

EspressReport® ES

Technical Overview

**Technological Overview,
Deployment/Integration Options,
and Detailed Feature Specifications
for EspressReport Enterprise Server**

Contents:

I. Introduction	4
II. Designing Reports	4
A. The Organizer Interface.....	4
B. Report Designer.....	5
1. Report Layouts.....	6
2. Using Report Designer.....	7
C. Visual Data Representation	7
D. Enterprise Data Source Integration	8
1. Complete SQL Control	9
2. Extensive XML Support	11
3. Java Object/Array Data	11
4. SOAP Data Source.....	12
5. Updating Data Sources	12
E. Advanced Reporting Features.....	13
1. Extensive Parameterization.....	13
2. Powerful Scripting Features.....	13
3. Sub-Reports	13
4. Automatic Drill Down	14
F. Ad-Hoc Reporting.....	14
III. Running Reports	16
A. Output Formats.....	16
B. Using Applets	16
1. Report Viewer.....	16
2. Page Viewer	16
C. Scheduling	16

D. Automatic Publishing	18
1. Reporting Portal	18
2. URLs	18
E. APIs	19
1. Menu API	19
3. Scheduler API	21
4. Server Extensions	21
IV. Designing Maps	23
A. Google Maps	23
1. Coordinate File	23
2. Creating Google Map	23
3. Tool Tip Report/Chart	24
4. Drill-down	26
B. SVG Maps	27
1. Color Coding	28
2. Drill-down	29
V. Dashboards	30
A. Flexible Layouts	30
B. Shared Parameters	30
C. Adding Drill-Down	31
D. Complete Customization	32
VI. Technology & Architecture	33
A. EspressReport ES Components	33
B. EspressReport ES Architecture	34
C. Administration & Security	35
VII. Global Deployment Ready	36
VIII. Conclusion	36

I. Introduction

EspressReport® Enterprise Server (ERES) provides a powerful, centralized reporting architecture that leverages the functionality of EspressReport's pure Java reporting engine and extends it to the enterprise. With built-in report delivery capabilities, end-to-end user/data security, self-service dashboards, advanced ad-hoc query and analysis features and scalable pure Java architecture, ERES empowers business users and provides a robust web based business intelligence (BI) reporting platform that offers superior price/performance ratio against legacy BI implementations as well as tool-kit based reporting solutions.

With ERES, users can easily draw data from a variety of sources and create a wide range of charts and reports using the integrated Chart Designer and Report Designer interfaces. Ad-hoc reporting is supported through the thin-client QuickDesigner. The Organizer interface provides a virtual file management and security system for reports and charts. Reports and charts in the Organizer can be used to build actionable, self-service dashboard through the Dashboard Builder which is also a thin client interface.

Deployment features allow users to automatically publish reports, charts and dashboards through the integrated menu features, and by generating URL requests. ERES supports advanced scheduling and archiving features, allowing users to version reports, and schedule email, and batch printing jobs.

Advanced configuration and deployment is available through the Java API interfaces. Programmatically, users can get a handle to created reports and charts, and develop their own deployment vehicles for reports and charts, or integrate them within other application environments.

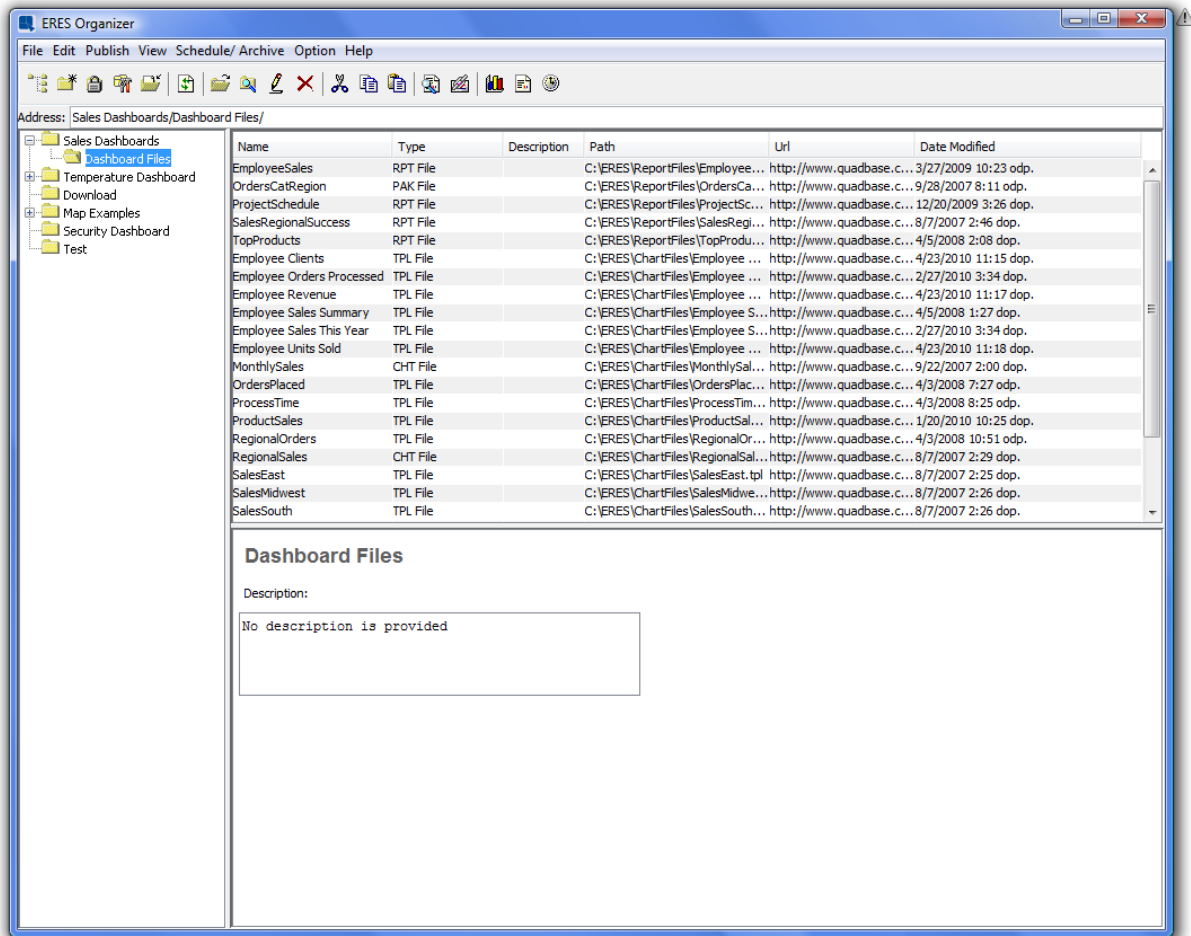
II. Designing Reports

In EspressReport ES, report design is handled in a client / server architecture. A central interface allows users to login to the server and access the report, chart, and data/query tools. Completed reports can be saved to the file system and published directly without any coding.

A. The Organizer Interface

The Organizer is the central report management interface for EspressReport ES. Within the Organizer users can create reports and charts, setup report rights data sources and queries, create schedule and archive jobs, and set user privileges and access rights (administrator only).

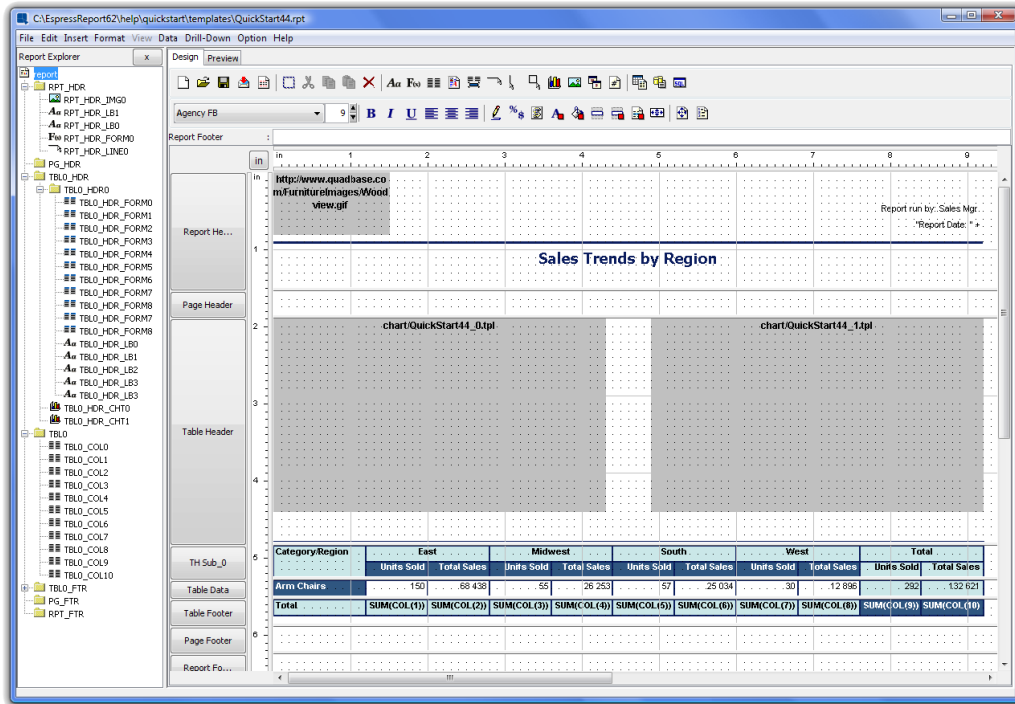
The Organizer also serves as a repository of published reports. All of the files added to the Organizer can be run from the reporting portal, from URL calls, and via the API. The Organizer allows users to create and modify reports without requiring any changes to the deployment environment.



The pure Java interface can run on virtually any platform, and can be run as a client-side application, or loaded remotely through a Web browser.

B. Report Designer

Launched from the Organizer, the Report Designer makes it easy to develop and design reports. It is an easy-to-use graphical user interface that allows reports to be customized in a drag-and-drop style environment. Users, who have experience with traditional reporting tools, will instantly recognize the banded-style interface of Report Designer. With a similar interface to many traditional reporting tools, it offers a short learning curve for new users.



As a pure Java component, the Report Designer can easily run on any platform. It can also be loaded as an applet on a Web browser with no client installation. For maximum flexibility API hooks are provided for the Designer, allowing it to be launched programmatically in a number of different configurations.

1. Report Layouts

For creating reports, ERES provides five basic layouts for the report data. Each of the layouts allows users to map the tabular input data into different grouped and/or summarized formats. Mapping options are set using an intuitive wizard. Once the initial report is populated with data, users are free to customize nearly every element in the report.

Simple columnar layout: This is the most basic of all the report types. It presents data in a single table without any grouping or breaks.

Summary break layout: Like the simple columnar report, the summary break report takes columnar data and presents it in a tabular form. However, it allows users to break the data into sections, and insert summary fields.

Crosstab layout: A crosstab report is a report format that shows and summarizes columnar data in a matrix-like form. Crosstab reports often resemble spreadsheets. Both rows and columns are summarized, allowing multi-dimensional data to be displayed in a 2 dimensional format.

Master & details layout: A master & details report is a set of tabular data that is grouped according to a master field. The master field usually has a one-to-many relationship with data in the data table.

Mailing label layout: A mailing label report allows data to be pre-grouped in clusters, making it easy to format data to generate mailing/address labels.

In addition to selecting the data and layout options, the report wizard also allows users to add titles, page numbers, and other elements to the report, as well as select a pre-defined or custom style to format the report. These features allow users to generate finished reports quickly without spending a lot of time formatting report elements.

Users who wish to customize reports have a myriad of options available to them. Virtually every report element can be customized allowing users to create just about any type of data presentation they can imagine.

2. Using Report Designer

EspressReport ES takes an innovative approach to report design. It uses a banded-style interface, where each band in the design window represents a section of the report. Objects can be inserted and manipulated into each section in free form. ERES also adds a unique manipulation feature that allows users to move and format groups of objects at once. This feature allows users to manipulate objects in a columnar format as well. Powerful positioning features like snap to grid, and guidelines make it easy to align and position the various elements in a report. The integrated report explorer interface makes it easy to navigate through reports.

Users can also manipulate object formats globally. This feature allows users to assign object properties for each type of report object (label, formula, column field, etc.), at a report-wide level. This allows the specific properties of those objects to be changed, and also changes the default attributes of objects. The global formats can also be exported to an XML file, and passed between reports.

The Designer also allows users to easily implement many advanced reporting features including parameterized reports, scripting, drill down, and sub-reports. ERES provides simple interfaces for all these features, allowing end-users to easily integrate powerful features into reports. These unique design features, make it easy for users to design professional-looking reports without a large investment of time.

C. Visual Data Representation

EspressReport ES also provides a large charting feature set. The incorporated chart designer utility allows users to select from over 30 different two and three-dimensional chart types with which to plot report data. Nearly every aspect of the chart can be customized within the chart designer window before a chart is entered into the report. ERES can draw charts with true 3D rendering, allowing pan/zoom, rotation, and light source modification.

Advanced charting 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.

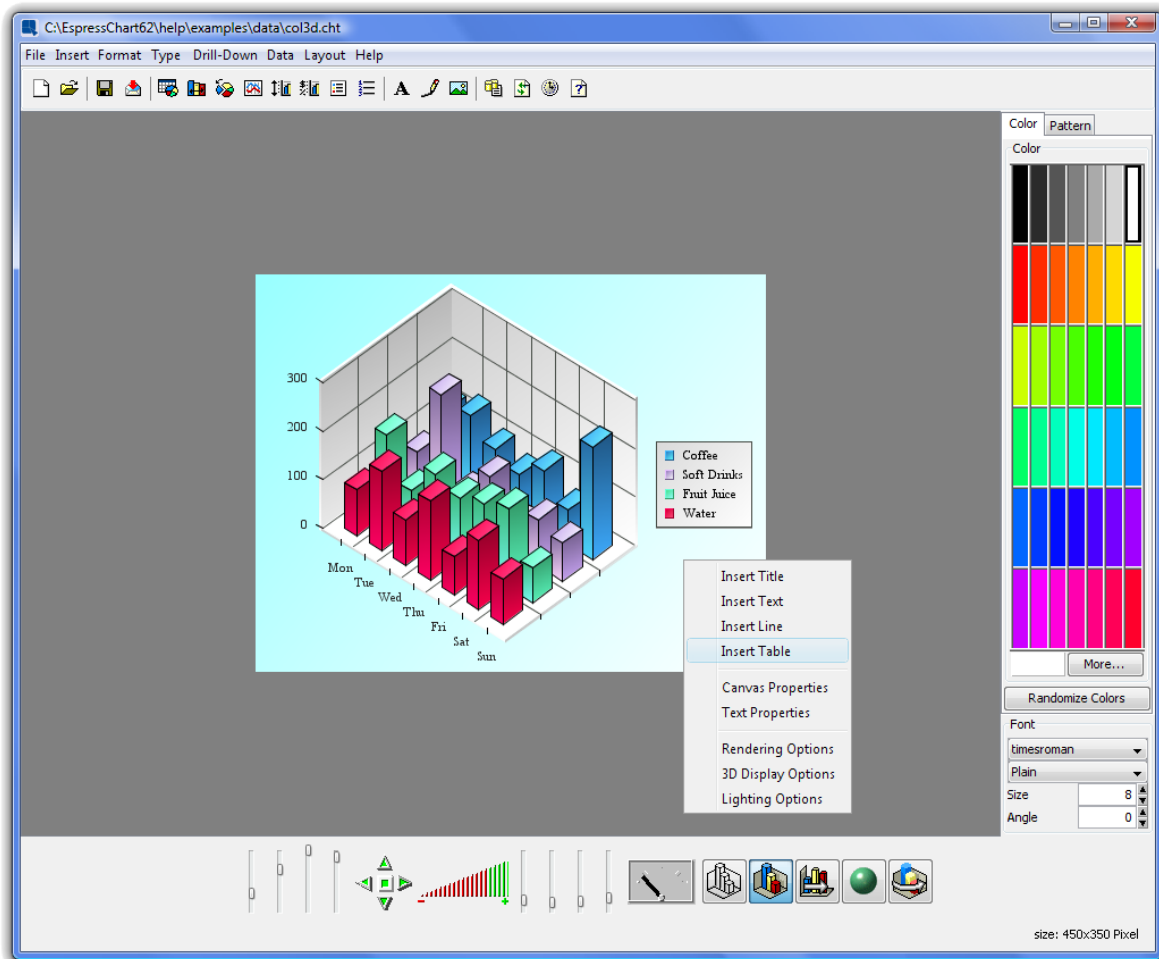


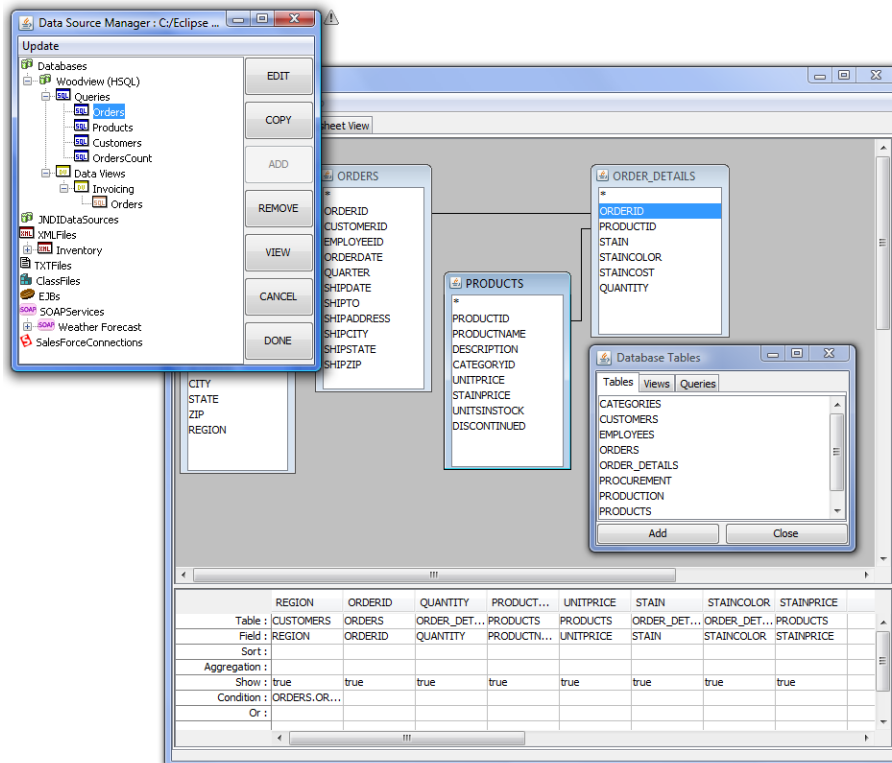
Chart templates are saved separately from reports, allowing charts to be embedded in reports or deployed on their own. This flexibility allows users to add graphical data representation to any Web page, not just reports. ERES can automatically stream dynamic images to any existing Web content.

D. Enterprise Data Source Integration

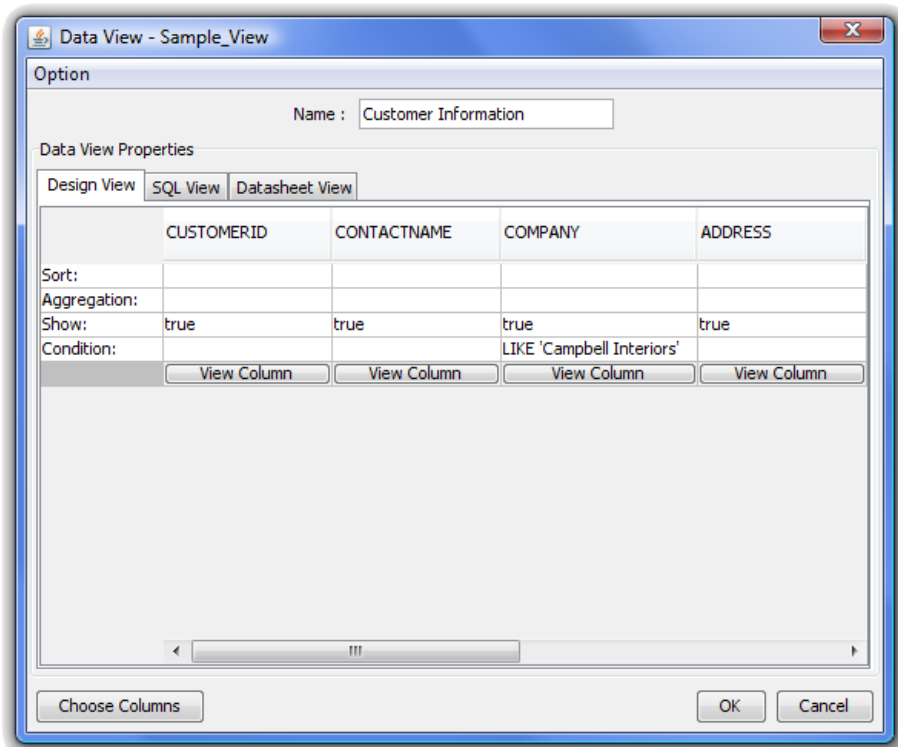
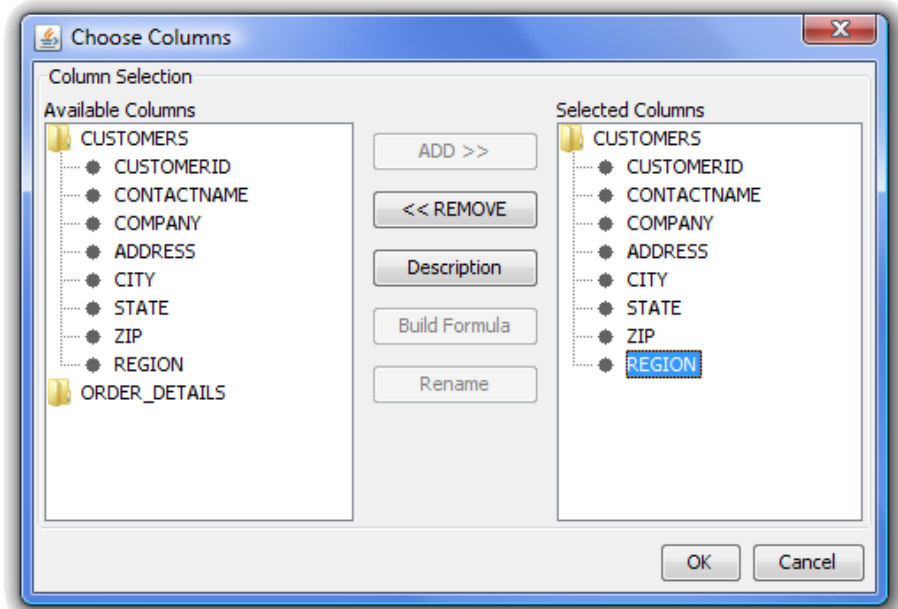
EspressReport ES provides interfaces to manage and retrieve report data from across the enterprise. Users can directly query databases via JDBC or using JNDI data sources with full SQL control. In addition data can be retrieved from Java classes, EJBs, or using connection pools to leverage application model infrastructure. SOAP data sources or application data returned in XML format can also be queried. Data sources and queries are stored in XML registries, allowing them to be re-used across multiple reports. Registries are created and edited through the data source manager interface, which is launched from the Organizer.

1. Complete SQL Control

Unlike many report writers, ERES gives users full control over the SQL that is generated to retrieve the report data. In addition, three different interfaces are provided for users to query databases. For advanced users, ERES allows you to simply enter or import a SQL statement. The full SQL statement can be imported and run intact to generate report data. For intermediate users, ERES provides a graphical query builder. 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.



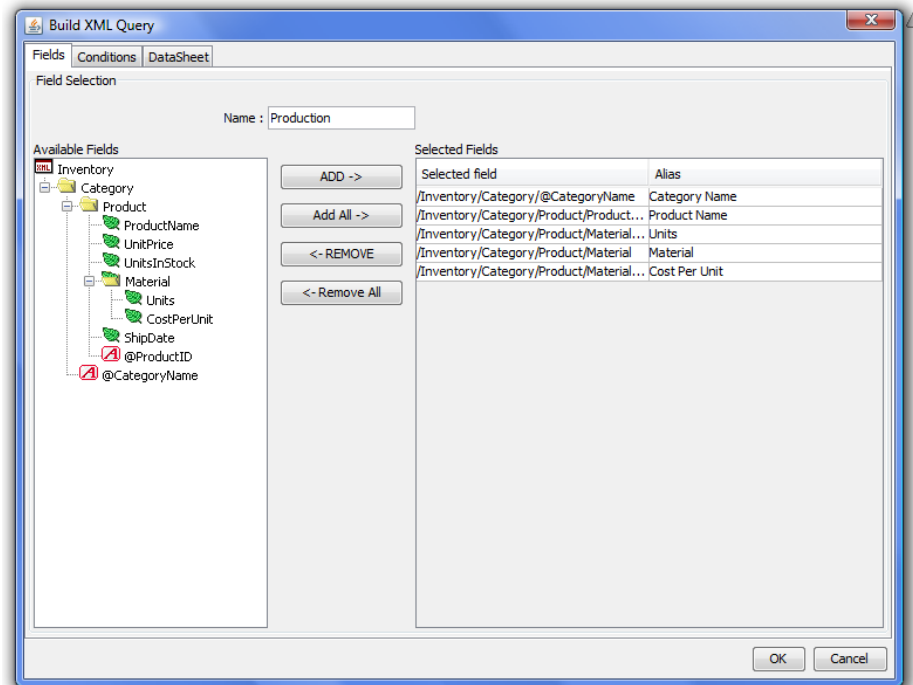
For users with limited knowledge of the underlying database names or structure, ERES provides the data view interface. In building a data view an administrator can pre-define tables, joins, fields, and even filters creating in effect a local schema for the user to select from. End users can then select (typically aliased) fields, and filter results without any knowledge of the underlying database structures.



The data view interface also works in conjunction with the QuickDesigner, allowing users to create ad-hoc queries against data views in a zero client HTML interface.

2. Extensive XML Support

Many application models rely on XML to encode and transfer data between application components. To report on this type of data ERES includes extensive interfaces to set up XML data definitions and query them. Based on a DTD or XML schema, users can then generate queries from an XML file, or an XML output stream from a servlet/JSP.



3. Java Object/Array Data

In addition to directly retrieving data from relational databases or XML sources, EspressReport ES lets you pass data directly into a report from Java objects or arrays, or by connecting to EJBs. At run-time data can be passed directly into the report, and at design-time object/array data can be pulled from a Java class file. Several different interfaces are provided for users to bring their application data into ERES.

Code Sample: Class to return a simple data array for a report

```
import java.awt.*;
import quadbase.reportdesigner.ReportAPI.*;
import quadbase.reportdesigner.util.*;

public class SampleData implements IDataSource {

    // Setting DbData for passing data as arguments
    String dataType[] = {"string", "string", "double", "double"};
    String fieldName[] = {"OrderID", "Product", "Price", "Quantity"};
    String records[][] = {{ "1001", "Chair", "325.00", "4"},
                          { "1001", "Table", "1211.00", "1"},
                          { "1020", "Dresser", "2214.00", "1"},
                          { "1020", "Table", "1211.00", "1"},
                          { "1020", "Cabinet", "4021.00", "2"},
                          { "1031", "Chair", "325.00", "2"},
                          { "1044", "Table", "1211.00", "1"},
                          { "1044", "Dresser", "2214.00", "1"} };

    DbData data = new DbData(dataType, fieldName, records);

    // create a empty constructor
    public SampleData() {}

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

}
```

4. SOAP Data Source

ERES 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.

5. Updating Data Sources

Report data is stored in both the template and in the data registry. This unique approach allows reports to be deployed independently of the registry files. At the same time, modifications to the data source performed at the report level can be saved back to the registry, and modifications to the registry (like database connection information) can be easily propagated to groups of templates.

This feature makes it easy to migrate installations with many templates (say from a development to a production environment). Only the connection information in the registry needs to be changed, and templates can be updated automatically.

E. Advanced Reporting Features

EspressReport ES uses Quadbase's powerful fourth generation reporting engine that contains all the advanced features that users demand in reports and charts. Powerful scripting and parameterization features facilitate template re-use, and advanced configuration and layout options allow users to create reports for nearly any application.

1. Extensive Parameterization

EspressReport ES provides a number of features making it easy to parameterize reports. Users can easily add a number of dynamic filters to database and XML queries. Also, users can associate report parameters with Java class file data sources, and implement their own filtering criteria on the report data. ERES also allows users to map parameter value to select statement in addition to database field and function. Formula parameters allow users to do run-time text/value substitution in calculations or labels in the report itself.

At run-time parameter values can be supplied programmatically, or submitted by the user. ERES includes a unique feature that generates an HTML form with parameter options automatically, and returns the report in the desired format using a servlet.

A very powerful feature that ERES supports is "cascading parameter". By default, the user is prompted to enter all of the report parameters at once in the prompt dialog. This configuration, however, may not be the best approach if some parameters are mapped to database columns with a significant number of distinct values. It can be difficult to select from a very large list, and depending on the parameter combination, users may be able to select sets that don't return any data.

To assist with these problems, ERES provides a feature that allows the user to configure the order in which the parameters should be entered. With this feature enabled, the user enters parameters in the dialog, in a pre-defined order. As such each selection will be applied as a filter to the next parameter prompt(s). Using cascading parameters can limit the number of distinct values presented to the user, and can prevent the user from selecting invalid parameter combinations.

2. Powerful Scripting Features

For run-time and conditional formatting, EspressReport ES includes a powerful internal scripting library that allows users to get a handle to, and modify most report elements using simple, intuitive syntax. Using scripts, users can set conditions, format cells, get a handle to specific rows of data, write loops for running calculations, and even modify whole report sections.

These powerful scripting features allow a great deal of dynamic formatting to be pushed to the template level. This allows for greater template re-use, and cuts down on the amount of code developers need to write to customize reports.

3. Sub-Reports

Extensive sub-report features allow users to create reports with multiple tables in many configurations. Using the sub-report features users can incorporate data from many different sources into a single report. Powerful parameter sharing and linking features allow users to create relationships between disparate data sources.

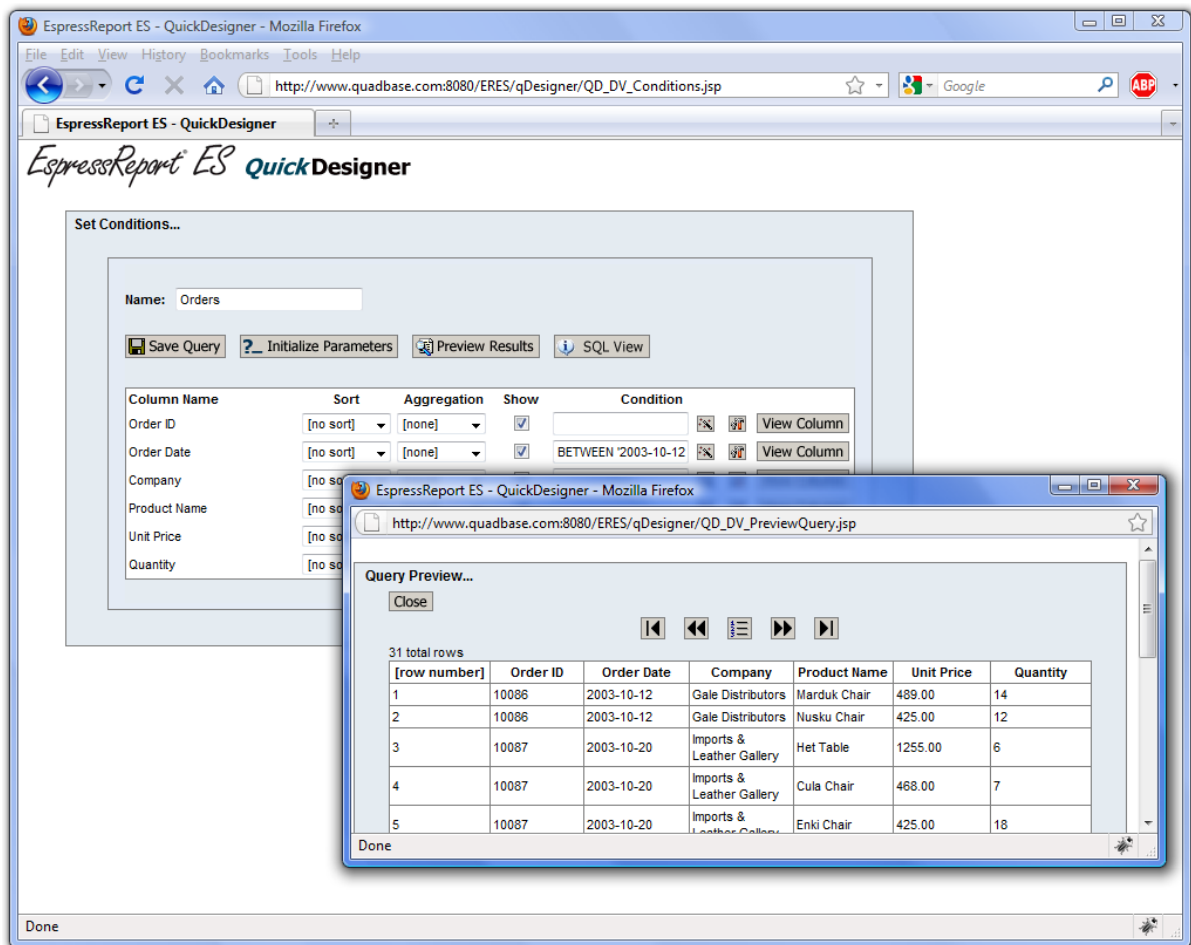
4. Automatic Drill Down

One of the most common reporting needs is the ability to drill into top-level reports and charts to pull underlying information. ERES supports this functionality, and provides tools to make it easy to deploy reports and charts in this configuration. Using the drill down features in ERES, users only have to design one report or chart template for each level of drill down. Linking between layers is handled automatically, and can be deployed without any additional coding.

F. Ad-Hoc Reporting

In addition to all of these design features, EspressoReport ES also includes powerful ad-hoc query and reporting capabilities. These simple thin client interfaces allow users across the enterprise to create queries, build reports from any data source, and publish the results without any training.

Ad-hoc queries and reports in EspressoReport ES are created through the QuickDesigner interface. This HTML-based thin-client tool allows users to create reports and charts from any defined enterprise data source, or build queries against Data Views. The simple point-and-click interface provides an extremely short learning curve, and allows users to quickly master basic reporting functions.



The screenshot displays the EspressoReport ES QuickDesigner interface in a Mozilla Firefox browser. The main window shows the 'Set Conditions...' dialog with a 'Name' field set to 'Orders'. Below this, there are buttons for 'Save Query', 'Initialize Parameters', 'Preview Results', and 'SQL View'. A table lists columns with their respective sort, aggregation, and condition settings. An overlaid 'Query Preview...' window shows a table with 31 total rows, displaying columns for Order ID, Order Date, Company, Product Name, Unit Price, and Quantity.

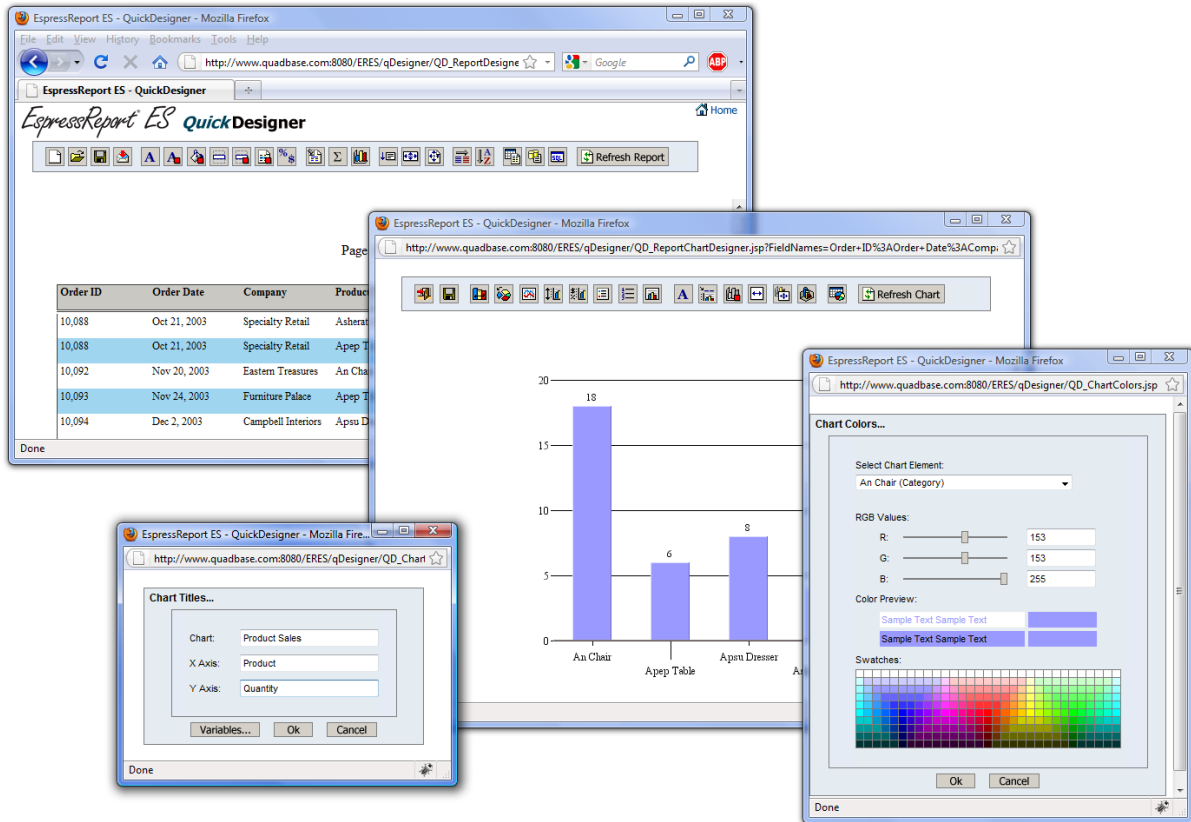
[row number]	Order ID	Order Date	Company	Product Name	Unit Price	Quantity
1	10086	2003-10-12	Gale Distributors	Marduk Chair	489.00	14
2	10086	2003-10-12	Gale Distributors	Nusku Chair	425.00	12
3	10087	2003-10-20	Imports & Leather Gallery	Het Table	1255.00	6
4	10087	2003-10-20	Imports & Leather Gallery	Cula Chair	468.00	7
5	10087	2003-10-20	Imports & Leather Gallery	Enki Chair	425.00	18

After creating a query, users can create and modify a report in the report designer portion of the QuickDesigner. In this interface, users can add grouping/summaries to the report, modify sorting, and

change column order. The intuitive toolbar at the top of the window allows users to perform basic report formatting, setting colors, fonts, and data formatting. Users can also add titles, labels, and dates/page numbers to the report.

A separate interface allows users to add and format charts. Charts can be added to reports, or created independently. The chart interface allows users to set chart size, colors, and other options. Users can also add and modify chart titles.

Once a report is finished, users can stream the report back to the client in a number of formats like PDF or MS Excel. Finished reports can also be saved back to the server-side repository to be published and shared with other users.



III. Running Reports

EspressReport ES has many different options for publishing reports. Integrated publishing features allow users to run and deploy reports without any code. Advanced scheduling features allow timed report delivery in a number of formats, and powerful Java API hooks allow users to integrate reporting functions into Web applications.

A. Output Formats

EspressReport ES can generate reports in several different formats. For basic reports, users can generate reports in standard HTML, compatible with any browser. For more sophisticated designs, users can generate reports in DHTML style sheets to handle the more complicated formatting requirements. To print reports, or for paginated output, ERES can generate high-quality PDF output. For MS Office users, ERES can generate reports as Excel spreadsheets, or Rich Text files (readable by MS Word). Other outputs include CSV, and XML data files, and delimited text.

Charts can be generated as GIF, JPEG, PNG, PDF, WMF, Flash, SVG, and BMP with interactive image maps for hyperlinks, drill down, and pop-up labels.

B. Using Applets

For applet or client-side deployment EspressReport ES provides several viewer components to display and interact with reports and charts.

1. Report Viewer

The Report Viewer is an applet that loads the report on the client machine via a Web browser. There are two versions of Report Viewer, an AWT and a Swing version. The AWT version allows users to run Report Viewer without downloading any plug-ins. The applet can view any report template, and allows reports to be hyperlinked to other reports directly, as well as Web pages. Reports are displayed in a paginated format. Users can generate an HTML page with the applet imbedded directly from Report Designer. The applet displays reports in a paginated format. Users navigate between pages and linked reports, using a customizable pop-up menu.

2. Page Viewer

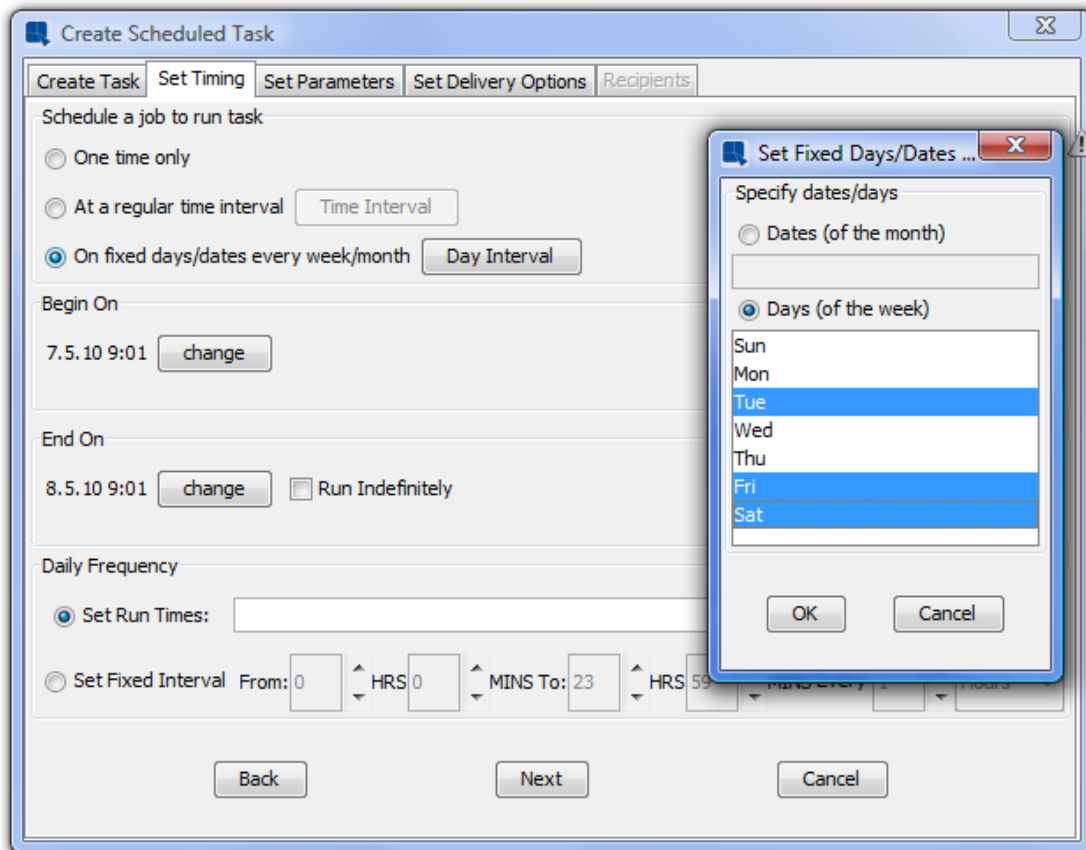
For large reports, loading the entire report on the client isn't generally feasible. For these scenarios, ERES provides the Page Viewer component. Page Viewer works much in the same manner as Report Viewer, but incorporates a page serving technology that sends report pages to the client only as they are requested. With Page Viewer, users can load/preview large reports on the client, without running out of memory.

C. Scheduling

A number of schedule based delivery options are available in EspressReport ES. Created through the Organizer interface, schedules can deliver exported reports and charts in a number of formats. They can also create report archives that store reports with historical data.

Schedules can save reports to the file system on the server in HTML/DHTML, PDF, Excel and CSV format and charts in PNG, GIF, JPEG and SVG format. Schedules can also email reports and charts to selected recipients. Reports and charts can be sent directly to network printers for batch printing, or published directly to FTP servers.

Report bursting features allow large reports to be automatically sectioned based on grouped data and delivered separately to different users/groups. Bursting allows high-volume scheduled generation and delivery without extensive database/performance overhead.



Users can schedule multiple reports and charts into a single package. Send a single email with multiple attachments or links to the recipients. Users can specify time intervals for fixed day schedules. For example, run a report every hour between 9AM - 5PM, or specify the exact export time for irregular schedules. A single parameterized report can be sent to multiple recipients based on the parameter set values. For example, a report with a region parameter can be scheduled to deliver each region's report to a different set of email recipients. The scheduler also supports robust error handling/notification, and schedule log.

Archiving/versioning features make it easy for users to save historical reports. Versioned reports contain a snapshot of data, and can be run/published in the same manner as standard reports. Interfaces allow users to select a version date, and run the archived report in any supported format.

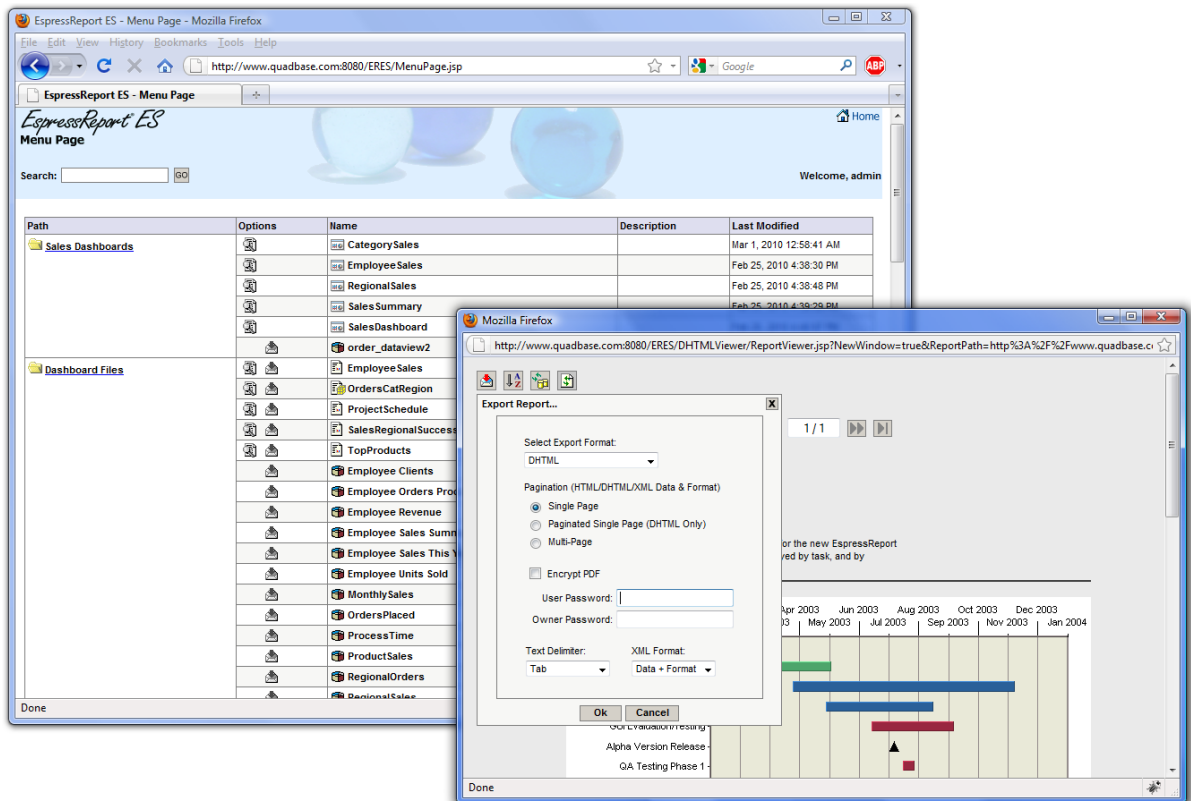
ERES also provides API hooks into the scheduling interfaces, allowing users to create and modify schedule jobs programmatically.

D. Automatic Publishing

EspressReport ES contains a repository of published reports in the Organizer interface. These reports can be run via the Web automatically without implementing any deployment code.

1. Reporting Portal

EspressReport ES includes an interactive JSP/JSTL reporting portal. This fully customizable thin-client interface allows users to view and run reports in the repository. Within this interface, users can run reports on-demand, and view the latest scheduled reports. Users can also pull up historical reports defined within archives.



Reports can be run using the DHTML Viewer component. This powerful utility allows users to perform filtering, multi-level sorting, and grouping in a thin-client interface. Functions are controlled using a small toolbar at the top of the page.

2. URLs

EspressReport ES also allows reports and charts to be run via URL calls. URLs can be generated in the Organizer interface, or users can create their own. They can be used to run and configure reports and charts in all export formats. Charts can be exported as complete HTML pages, or streamed as dynamic images to existing Web content.

E. APIs

The Java APIs extend the most flexibility to ERES users. API hooks are provided to most of the interfaces allowing users to create their own customized report deployments, or integrate the ERES interfaces within their own applications.

1. Menu API

The menu API provides hooks into the report repository. Using these methods users can access a list of published reports, along with users, privileges, and schedule and archive information. These hooks allow users to create their own reporting portals, or integrate the report repository into their own applications. These interfaces allow reports to be published and updated as users create and modify report and chart templates without any adjustments to deployment code.

Code Sample: Get a handle to the repository and return available reports for a user

```
import quadbase.reportorganizer.menuAPI.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class menuCode {

    Vector menuVector = new Vector();

    public void processRequest(HttpServletRequest request, ServletContext context)
throws Exception {

        try {

            // Get a handle to the Organizer - create a QbMenu object
            qbMenu = new QbMenu(context.getAttribute(QbMenu.QBMENUDATA));

            // Use supplied username and password to login
            String userName = request.getParameter("USERNAME");
            String password = request.getParameter("PASS");
            boolean loginSuccess = qbMenu.login(userName, password);

            // If login is good build a list of reports and charts for the user
            if (loginSuccess) {
                MenuNode[] menuNode = qbMenu.getMenuNodeList(null);
                for (int i = 0; i < menuNode.length; i++) {
                    for (int j = 0; j < menuNode[i].getRowItemCount();
j++) {
                        MenuTableRow menuTableRow =
menuNode[i].getMenuTableRowAt(j);
                        menuVector.add(menuTableRow.getName());
                    }
                }
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

2. Report API

The report API provides hooks directly into reports and charts. Using these methods users can develop their own mechanisms and logic to run reports. Report API can generate reports on the fly based on user parameters, and stream them directly to the client browser, or cache the generated reports on the server. Nearly every report object can be controlled using the API, and reports can be designed purely programmatically. The API can also run templates built in Report Designer, while modifying report properties at run-time. The Report API also provides hooks directly into the Report Designer interface, allowing it to be loaded and run programmatically.

Code Sample: Change a report's data source and stream to the client as HTML

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

public class runReport extends HttpServlet implements SingleThreadModel {

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

        // Connect to the ERES Server
        QbReport.useServlet(true);
        QbReport.setServletRunner("http://" + req.getServerName() + ":" +
req.getServerPort());
        QbReport.setServletContext(quadbase.common.client.ServerMessage.
getServletContext());

        res.setContentType("text/html");
        OutputStream toClient = res.getOutputStream();
        Vector reportList = menuCode.getMenuVector();
        String template = reportList.get(3).toString();

        try {

            // Get a database connection from the pool
            Connection conn = dbPool.getConnection();

            // Create a QbReport object and pass in the connection
            QbReport report = new QbReport((Applet)null, template, false,
false, false, true);
            report.getInputData().setAllDatabaseInfo(conn);

            // Export the report to DHTML and stream to the client
            report.export(QbReport.DHTML, toClient);

        } catch (Exception ex) {
            ex.printStackTrace();
        }

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

    }
}
```

3. Scheduler API

The scheduler API provides hooks directly into the scheduling process. Using these methods users can create schedule jobs programmatically and modify existing jobs. Users can also create and modify archive jobs.

4. Server Extensions

Users can also apply custom business logic to EspressoReport ES's publishing features. This includes reports run in the portal, via URLs, and scheduled reports. Users can implement custom server extensions that will intercept reports and charts before they are run. The extension classes can make any API-supported modifications to the report or chart, and then return it to the publishing process. These features allow users to apply custom logic to report/chart generation without having to implement their own deployment vehicles.

Code Sample: Add username and run date to reports as they are executed in ERES

```
package extensionClasses;

import java.lang.*;
import java.awt.*;
import quadbase.reportorganizer.ext.*;
import quadbase.reportdesigner.ReportAPI.*;
import quadbase.reportdesigner.ReportElements.*;
import quadbase.reportdesigner.util.*;
import quadbase.reportdesigner.report.Formula;

public class MyMenuPageListener implements MenuPageListener {

    public QbReport modifyBeforeRun(QbReport report, String username) {

        try {

            // Create a new label to show user that is running the report

            ReportCell userLabel = new ReportCell();
            userLabel.setText("Report Run By: " + username);
            userLabel.setFont(new Font("Arial", Font.PLAIN, 8));
            userLabel.setAlign(IAAlignConstants.ALIGN_RIGHT);
            userLabel.setHeight(0.2);
            userLabel.setWidth(1.6);
            userLabel.setY(0);
            userLabel.setX(5.90);

            // Create a new formula to show the report run date/time
            ReportCell runDate = new ReportCell();
            Formula runDateFormula = new Formula("ReportRunDate", "\"Report Run
Date: \" + printDateTime(getCurrentDateTime(), \"MMM dd, yyyy h:mm a\")");
            report.addFormula(runDateFormula);
            runDate.setFormulaObj(runDateFormula);
            runDate.setFont(new Font("Arial", Font.PLAIN, 8));
            runDate.setAlign(IAAlignConstants.ALIGN_RIGHT);
            runDate.setHeight(0.2);
            runDate.setWidth(2.5);
            runDate.setY(0.2);
            runDate.setX(5.0);

            // Add new cells to page header section
            report.getPageHeader().addData(userLabel);
            report.getPageHeader().addData(runDate);

            // Adjust section height if necessary
```

```
        if (report.getPageHeader().getHeight() < 0.4)
            report.getPageHeader().setHeight(0.4);

    } catch (Exception ex) {

        ex.printStackTrace();

    }

    return report;

}
```

IV. Designing Maps

In addition to charts and reports, the user can create maps. Two types of maps are supported, namely Google maps and SVG maps. ERES Google Maps are useful for displaying exact spots on the maps. Google maps use high quality satellite imagery, a very wide zoom range is possible, making it possible to display every imaginable area of the world. Numerous applications are possible. Basically any report that requires geographical information (e.g., city, state postal code) as part of the report can use a map. SVG maps on the other hand do not give wide zoom range. As such, it is mainly designed for a quick overview for comparing data points on a geographical or spatial basis. Areas on the SVG maps can be color coded to show the different ranges of values of certain data of interest (such as revenue, population, growth rate etc.).

A. Google Maps

As the name implies, Google maps are powered by Google. As such, a license key is required to run the maps. Once the license key is installed in ERES, it's ready to display Google maps. To display desired data points on the map, coordinates of longitudes and latitudes are required. In ERES Google maps, this information is supplied by the coordinate file.

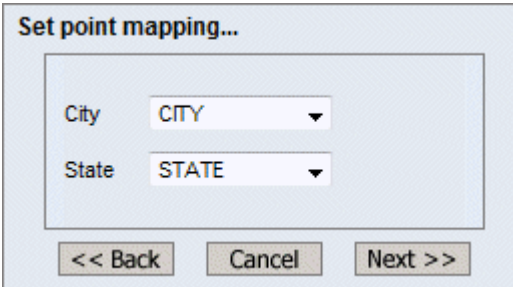
1. Coordinate File

To create a coordinate file, the user needs to first choose a data source. The coordinate file can be generated by one of three methods (1) from the data source, (2) by geocoding or (3) manually. The coordinate file consists of records that give a point ID for each pair of longitude and latitude coordinates. A point ID can be a single field or multiple fields. An example of a point ID is (city, state). So, the record would look like the following:

"Chicago", "IL", "-87.624333", "41.879535", i.e. (city, state, longitude, latitude)

2. Creating Google Map

To create a Google map, a data source has to be selected as in making any report/chart. The data retrieved must have a field or fields that match the point ID. The user must set point mapping, i.e. map the fields in the data source to the point ID fields in the coordinate file.



The image shows a dialog box titled "Set point mapping...". Inside the dialog, there are two dropdown menus. The first is labeled "City" and has "CITY" selected. The second is labeled "State" and has "STATE" selected. Below these dropdowns, there are three buttons: "<< Back", "Cancel", and "Next >>".

The matching field or fields is what allows Google map to display data for a location (point ID) on the map. For example, the fields (city, state) in the map's data source match the (city, state) fields in the coordinate file such that Google map knows Chicago, IL is at: (longitude, latitude) = (-87.624333, 41.879535).

3. Tool Tip Report/Chart

With point mapping set, Google map knows where to display information for each record in the data source on the map. But to display pertinent reporting information on the map, the user needs to add a tool tip report/chart.



Clicking on the tool tip icon on the tool bar in map designer will bring up the QuickDesigner.

Data Source Results...

CUSTOMERID	COMPANY	ADDRESS	CITY	STATE	ZIP
1	Allied Furniture Emporium	384 Broad St.	Littletown	NY	18322

First 10 records...

<< Back Cancel Create Chart >> Create Report >>

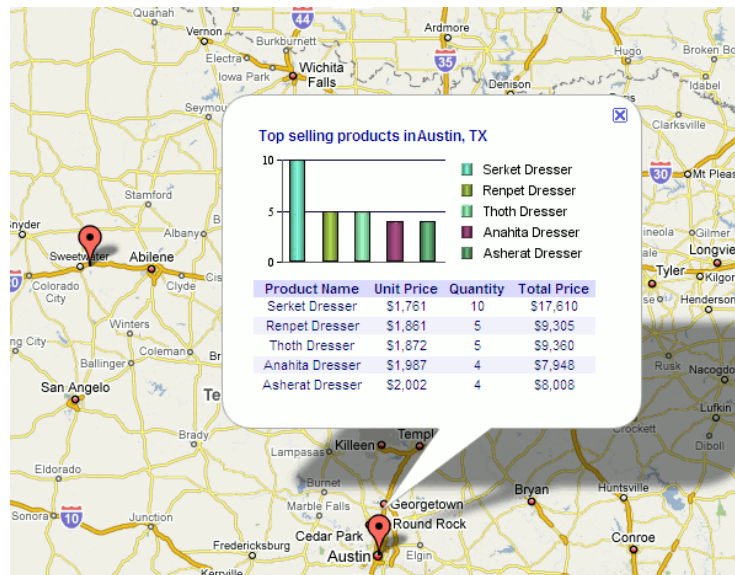
EspressReport ES QuickDesigner



Page 1 of 1

CUSTOMERID	COMPANY	ADDRESS	CITY	STATE	ZIP
1	Allied Furniture	384 Broad St.	Littletown	NY	18322

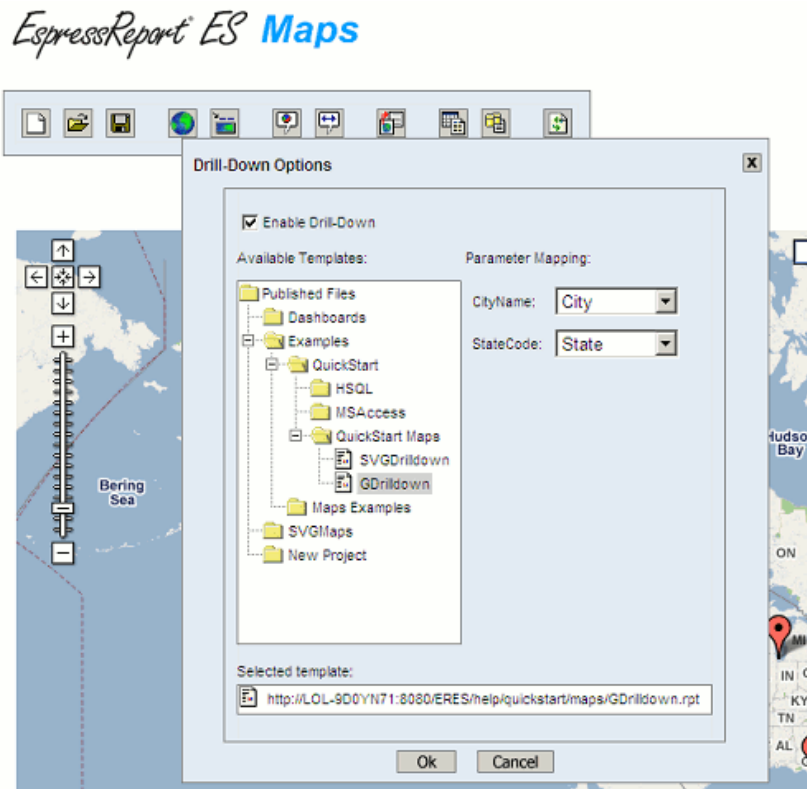
The created report or chart (a report can also contain charts) will show up when the mouse is over a bubble which indicates a point ID exists on that location on the map.



In the above example, there are five records in the map data source that match the point ID (Austin, TX). Five fields for the record are displayed to show (Product Name, Unit Price, Quantity, Total Price).

4. Drill-down

ERES Google maps allow drill-down from a tool tip bubble to another report, chart or map.

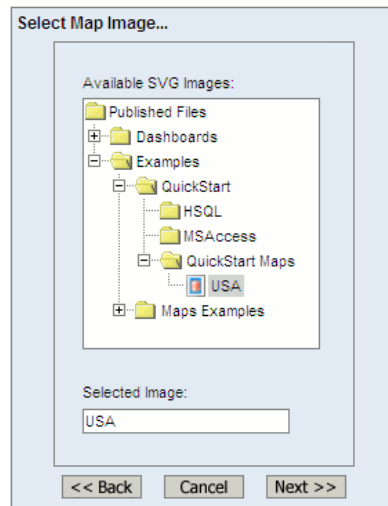


By clicking on the drill-down icon on the tool bar, a dialog will pop up showing the available templates for setting up drill-down. The template selected for drill-down should have parameters that match the point ID in the map. In the above example, the template should have either "city" or "state", or both as parameters.

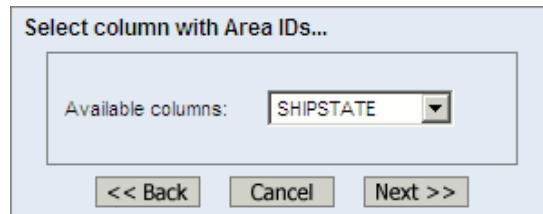
B. SVG Maps

Similar to ERES Google maps, an ERES SVG map requires a data source. But unlike Google maps, it does not require a coordinate file and it runs without external third party support. To create an SVG map, the user has to select a data source and an SVG map image file.

EspressReport ES Maps



Map Images are SVG image files with data structures containing geographic data. As such there are area IDs which can be used to identify areas on the SVG map. The data source used has to have a data field that match the area ID's in order to make the map display useful information.

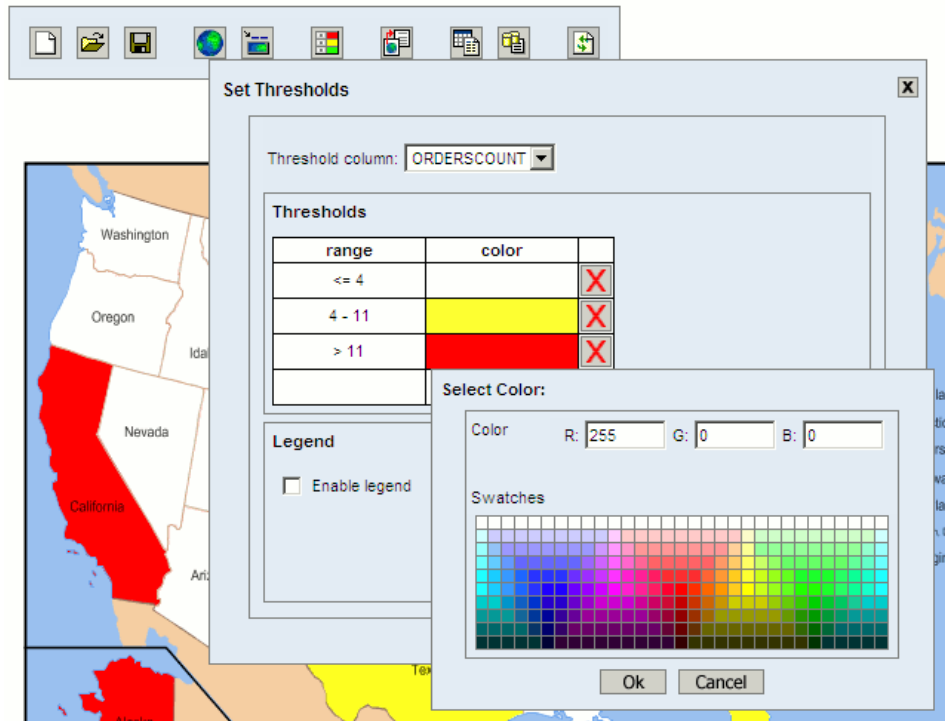


The concept is similar to setting point ID mapping in ERES Google maps.

1. Color Coding

The user can map colors to specific ranges of values of a given field in the data source so that a quick overview of distribution of data values geographically can be recognized easily.

EspressReport ES Maps

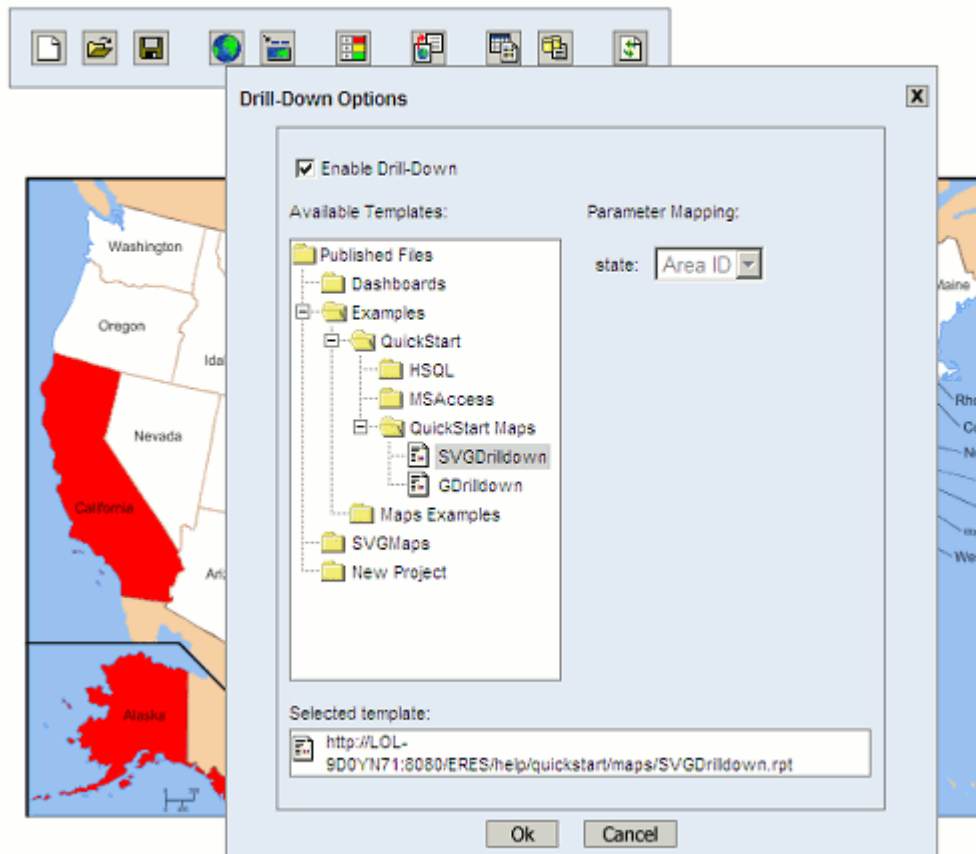


A legend for the color coding and map title can be added to the SVG map to complete the presentation.

2. Drill-down

As in Google maps, SVG maps support drill-down. A list of parameterized charts/reports/maps will be presented to the user. But unlike Google maps, there is only one way to set up the drill-down link in SVG maps, i.e. mapping the Area ID to the parameter in the chart/report/map.

EspressReport ES Maps



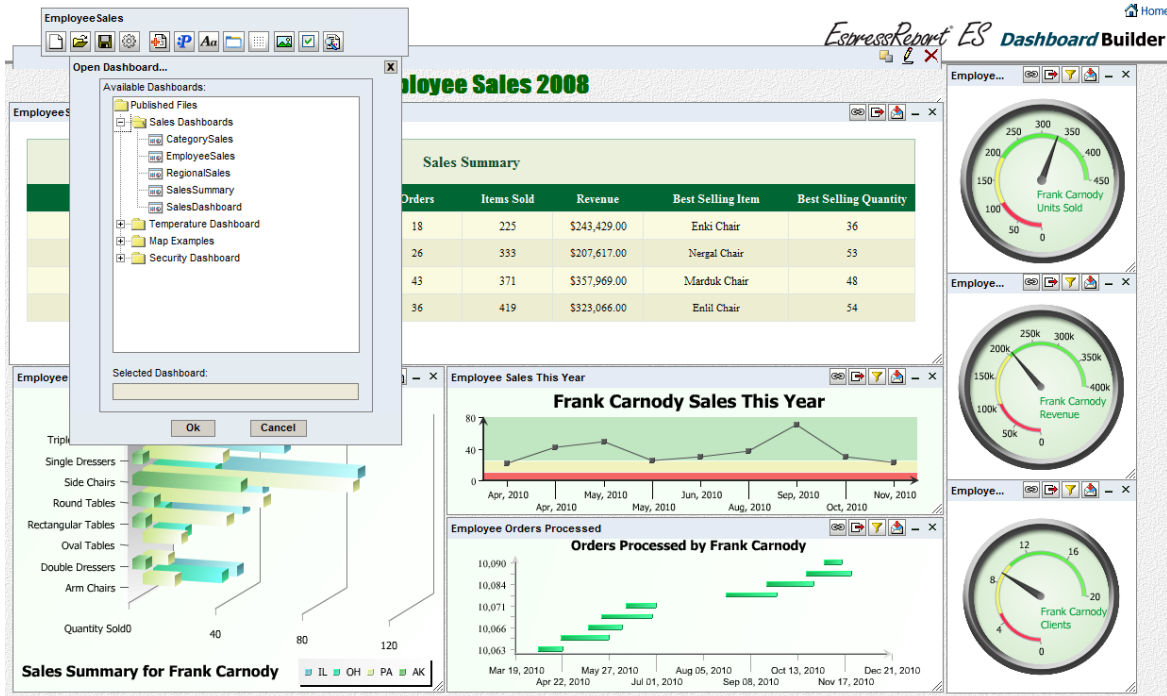
V. Dashboards

EspressReport ES allows users to construct and deploy dashboards. These self-service briefings can be built using any deployed report, chart or map using the thin-client dashboard builder interface.

The Dashboard Builder is a thin client design tool that offers a powerful, yet user friendly, drag-and-drop, and point-and-click environment. Unlike other competitive products, the user can make the entire dashboard with the UI without any scripting or programming. As such, it empowers a wide range of non-technical users without having to expense IT resources.

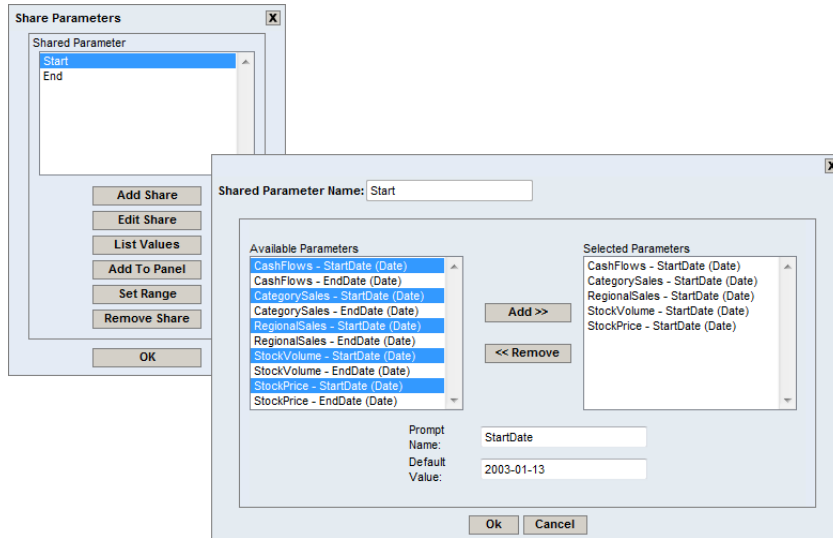
A. Flexible Layouts

To create a dashboard, simply add the available charts, reports and maps in the Organizer. The dashboard builder allows the user to layout the elements (reports, charts, maps) in any way desired. It will automatically place the elements in an approximate layout. The user can then re-position and re-size (if necessarily) the items (including labels, images and parameters) to make a presentation quality dashboard.



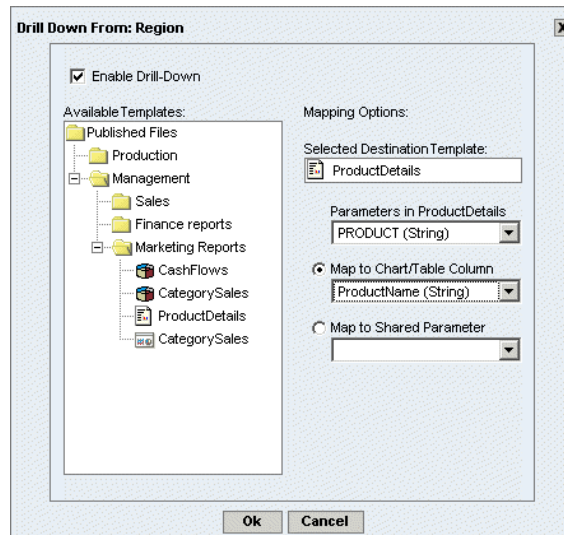
B. Shared Parameters

Once the users have the reports/charts/maps inserted into the dashboard, they can set up shared parameters provided that some of the elements have similar parameters that can share the same values.

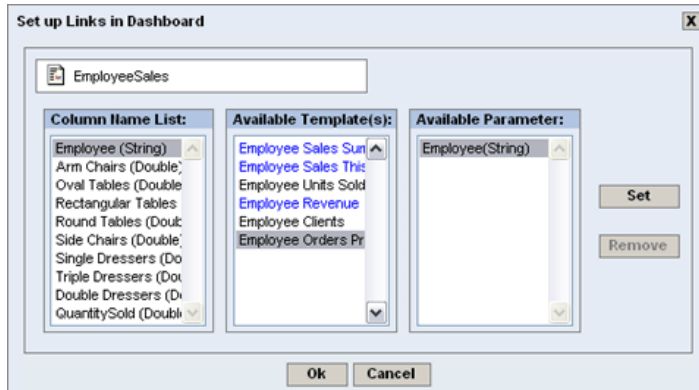


C. Adding Drill-Down

To add a drill-down chart or report to an element, simply click on the “add/modify Drill-Down” button. The drill-down dialog will pop up. Select a chart, report, map or another dashboard that has a matching parameter for drill-down. If the chosen chart/report/map has other parameters, they must be mapped to shared parameters.



To add a link to an element, click on the “add/modify link” button. A dialog will pop up for the user to choose the column for linking. If there are parameterized reports/charts in the dashboard, they will be shown in the “Available Templates” panel. The user can then select the ones with matching parameters for linking.



D. Complete Customization

The look and feel of the dashboard is highly customizable. The user can add labels, background color, background images to the dashboard. Text font, size, color can all be customized. The parameter panel can be customized in the same way.



VI. Technology & Architecture

EspressReport ES is a pure Java product, giving it the flexibility to run on nearly any platform. Designed to leverage users' existing infrastructure investments. It deploys completely within J2EE application environments, with no separate server-side application processes. It runs with most major application servers and database engines.

A. EspressReport ES Components

EspressReport ES has eleven major components.

ERES Server: The server is the back-end to ERES. It deploys as a servlet collection within an application server or servlet container, and manages design and deployment for Reports and Charts. The server handles user authentication, scheduling, archiving, and provides the data access and file I/O for clients running Chart Designer and Report Designer.

Organizer: The Organizer is a powerful graphical user interface that acts as a virtual file management system. It allows users to organize charts, reports, and other files in virtual folders, generate URLs deploying charts and reports, schedule/archive charts and reports, and set file-level privileges. It also provides users with an interface for managing data sources, and designing database queries.

Report Designer: The Report Designer is a graphical user interface, launched within the Organizer that allows users to create and customize reports. The simple drag and drop style interface and extensive editing/formatting capabilities, makes report design quick and easy.

Chart Designer: The Chart Designer is a graphical user interface, launched within the Organizer that allows users to design and customize charts. Its point and click interface enables the swift and easy creation of charts with an almost infinite degree of flexibility.

QuickDesigner: The QuickDesigner is a thin-client ad-hoc reporting tool. It allows users to quickly design queries and build reports. Limited formatting options allow users to customize reports, and save them back to the Organizer.

Map Designer: The Map Designer is a thin-client tool for creating/editing Google maps, creating coordinate files, and SVG maps. EDAB Google Maps can be further customized with tool-tip reports and drill-down. SVG maps can be customized with color coded areas and drill-down as well.

Dashboard Builder: The Dashboard Builder is a thin-client dashboard design/creation tool. It allows developers as well as non-technical users to easily build custom briefings/dashboards using charts, reports and maps deployed in the Organizer.

Report Viewer: Report Viewer is an applet that enables you to view and manipulate a report dynamically through a web browser. Within the Report Viewer, users can navigate from page to page in the report, export the report to a variety of static formats, and drill-down on report elements.

Page Viewer: Page Viewer is an integrated applet like Report Viewer that uses page serving technology. With Page Viewer, pages in the report are only sent to the client when requested. This allows users to view/preview large reports.

Chart Viewer: The Chart Viewer is an applet that allows charts to be viewed remotely. It enables viewers to rotate, resize, zoom and pan a chart. Viewers can click on individual data

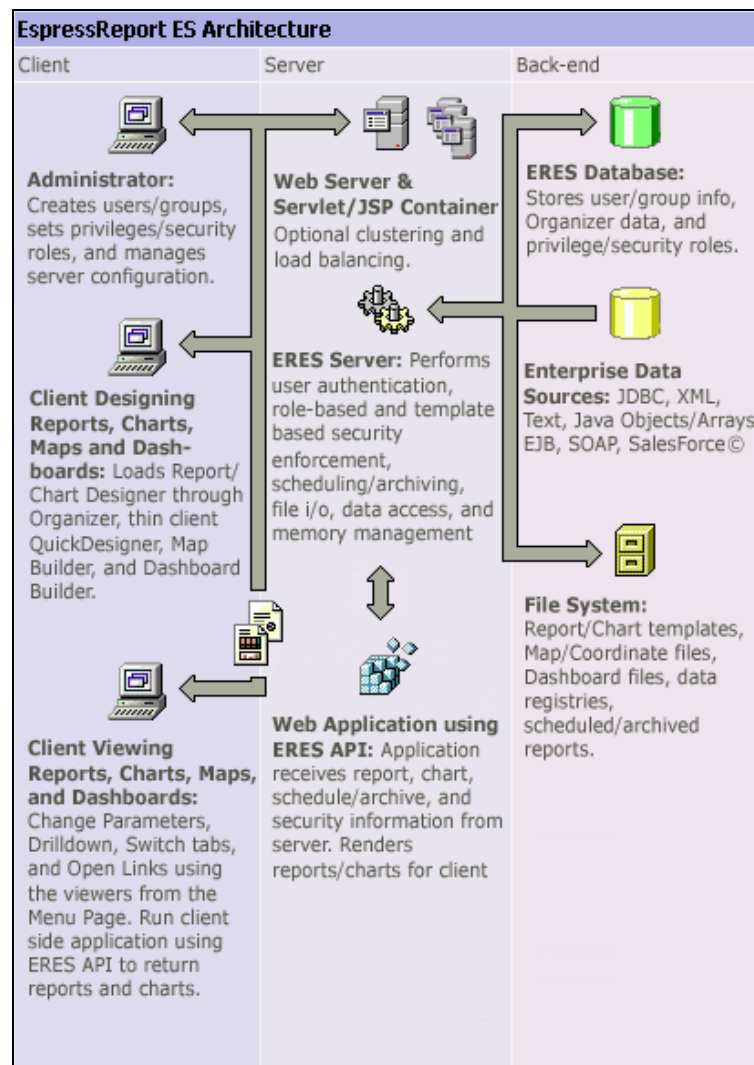
points to either recover the data associated with the data point, or link to another chart. ERES includes two other applets for interactive chart viewing, namely, Mini Viewer and Micro Viewer. Each version provides a successively smaller applet size with diminished functionality.

Report API: The Report API is an easy-to-use Java application programming interface that allows users to create, modify, and run reports and charts within applications, applets, servlets and JSPs. Users can easily get a handle to most chart or report elements making it easy to customize reports and charts on the fly.

ERES (Menu) API: The ERES API is a Java application programming interface that allows users to interface programmatically with the ERES Server and Organizer. Users can get a handle to the Organizer to retrieve information about files and permissions. Users can also get a handle to scheduled and archived reports.

B. EspressoReport ES Architecture

EspressoReport ES is designed to deploy in a Web application environment. The following diagram illustrates the basic architecture of an ERES deployment.



The ERES server is deployed as a servlet collection within an application server/servlet & JSP container. Other servlets/JSPs can be deployed in the server to connect to, and retrieve information from the ERES server (like file names, schedule information, and schedule reports/charts).

ERES can be deployed in a single server or in a clustered environment. When running in a cluster, one server acts as a load balancer, and routes all incoming client requests to cluster members, while maintaining a common back-end file structure. This configuration option allows ERES to scale to high-availability deployments with thousands of users.

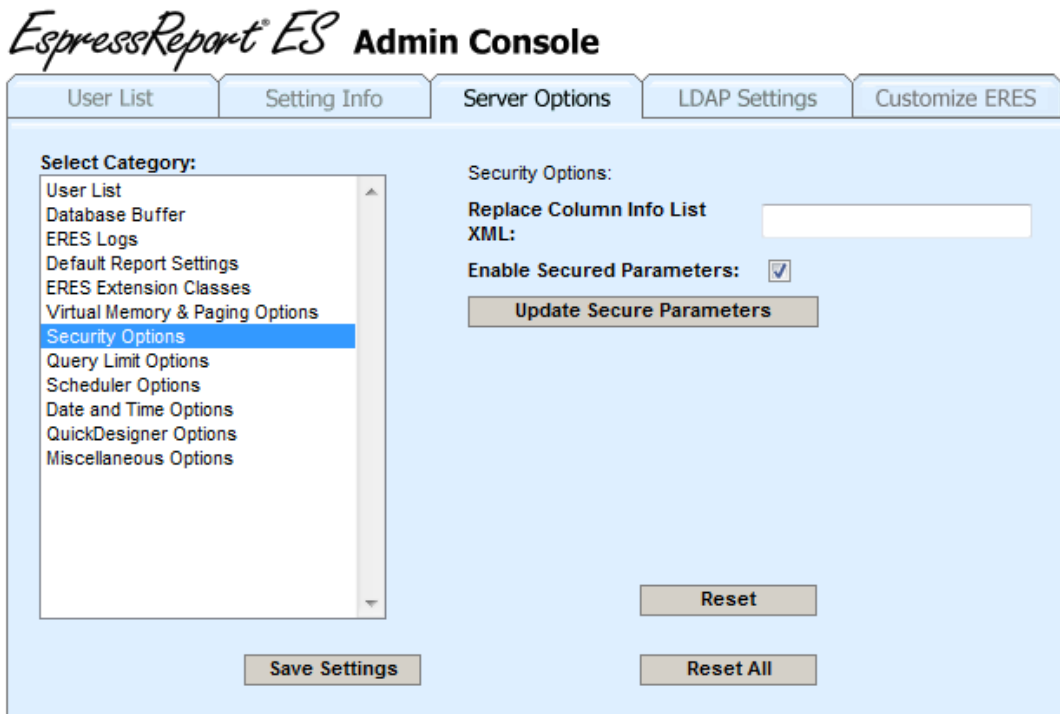
On the back end, information for ERES including user/group information, the organizer structure and files, and security and privilege information are saved in a database. The server also performs data access and buffering for reports and charts, and file I/O on the server-side.

Users can also integrate custom security models and users/groups with ERES. ERES allows direct integration with users/groups in LDAP systems, and stored in relation databases. In addition ERES provides open server extensions that allow users programmatically integrate any custom security model with the ERES architecture.

Clients can be administrators modifying users, privileges, and server settings from the remote Admin Console, or the Organizer. Clients can load the Report Designer and Chart Designer through the Organizer to create reports and charts, or load the thin-client QuickDesigner for ad-hoc reporting and analysis. Clients viewing reports and charts can access the menu, make URL calls to the ERES server, or connect to other applications to retrieve charts and reports.

C. Administration & Security

Almost all of the administration functions for the ERES server and users' base can be performed remotely. The thin-client admin console allows administrators to create users and groups, and configure server features. Security and privileges can be applied in the Organizer interface.



EspressReport ES provides a powerful security framework to handle user/group privileges as well as role based secured parameter and element level security. ERES binds privileges to report developers using an ownership model that allows security to be enforced automatically as reporting implementations grow and change. Administrators can also apply a more traditional file/folder security framework.

Beyond privileges, security features can be extended to report templates. Templates can be configured to have specific report elements or data columns change depending on which user/group is viewing the report. In addition filters can also be automatically applied to reports to facilitate role-based security. As such, users with different security credentials viewing the same reports/charts see different data, i.e. only what they are authorized based on their role. LDAP and custom external security model integration are also supported.

VII. Global Deployment Ready

EspressReport ES fully supports Internationalization. Locale specific formatting allows data elements to be displayed correctly for specific locale/time-zone. International characters are supported in reports, charts and maps. A utility program is included for the user to translate any text in the UI and messages to any language. Export formats are automatically generated with UTF-8 character encoding (except for PDF where font mapping is required) fully supporting all languages.

VIII. Conclusion

EspressReport ES is a powerful reporting solution that couples enterprise reporting features with simple interfaces that are easy to learn and use. ERES allows users to run canned reports, set schedules and archives, and create ad-hoc reports and queries without any coding.

Its pure Java architecture allows it to run on almost any platform, and it deploys completely within application server environments, allowing users to leverage the scalability of established systems. ERES's powerful feature set, publishing options, and ease of use make it a strong alternative to heavy business-intelligence applications, and limited, application-specific reporting tools.