Climb Inc.
Growing to Meet Your Needs

EspressReport® ES

EspressReport Enterprise Server のテクノロジー概要、
デプロイメント、インテグレーション、各種機能のホワイトペーパー

2026 年 2 月

株式会社クライム

目次

はじめに 4
レポートのデザイン 4
Organizer 4
Report Designer 5
レポートのレイアウト 6
Report Designer の各種機能 7
データ表示の視覚効果 8
あらゆるデータソースをエンタープライズ サポート 9
SQL による完全制御 9
XML サポート 11
Java オブジェクト／アレイデータ 11
SOAP データソース 12
データソースの更新 12
高度レポーティング機能 13
レポート パラメータの活用 13
スクリプティング機能 13
サブレポート 14
詳細情報の深掘り自動設定 14
アドホック レポーティング 14
レポートの公開 15
出力フォーマット 16
アプレットの活用 16
レポートビューア 16
ページビューア 16
スケジュール設定 16
レポートの自動公開 18
レポーティング ポータル 18
URL 19
API 19

メニューAPI 19
レポート API 20
スケジューラ API 21
サーバ拡張クラス 21
マップのデザイン 22
Google マップ 23
座標ファイル 23
Google マップの設定方法 23
ツールチップでレポート／チャート表示 24
ドリルダウン 26
SVG マップ 26
カラーコーディング 27
ドリルダウン 28
ダッシュボード 29
柔軟なレイアウト設定 29
共有パラメータ 30
ドリルダウンの追加 30
ダッシュボードの仕上げ 31
テクノロジーとアーキテクチャ 32
EspressReport ES のコンポーネント 32
EspressReport ES のアーキテクチャ 33
管理とセキュリティ 35
グローバル デプロイメント 36
まとめ 36

はじめに

EspressReport® Enterprise Server（ERES）は、EspressReport の Pure Java レポーティング エンジンをエンタープライズ規模に拡充して中央一元管理を可能にした、高性能レポーティング アーキテクチャを基盤としています。レポート生成機能はもちろん、エンドツーエンドのユーザ／データ セキュリティ、セルフサービス型ダッシュボード機能、高度アドホック クエリ、分析機能を完備しています。これらの豊かな機能性と Pure Java アーキテクチャの柔軟な拡張性がビジネスユーザの強力な武器となり、ウェブベースの高性能ビジネスインテリジェンス（BI）プラットフォームを確立しています。従来の各種 BI ツールと比べても突出したコストパフォーマンスを誇る、ツールキット形式で利便性に優れたレポーティング ソリューションです。

ERES を使用すると、さまざまなデータソースから簡単にデータを引き出して多種多様なチャートとレポートを作成できます。具体的には、ERES に組み込まれたチャート作成インターフェースのチャートデザイナー（Chart Designer）とレポート作成インターフェースのレポートデザイナー（Report Designer）を使用して、誰でも手軽にチャートとレポートを作成することができます。その場のニーズに応じた柔軟なレポート作成（アドホック レポーティング）にはシンクライアント型レポーティング ツール、クイックデザイナー（QuickDesigner）が便利です。さらに、作成したレポートやチャートのセキュリティ管理および仮想ファイル管理システムとしては、オーガナイザー（Organizer）インターフェースを活用できます。作成したレポートやチャートをオーガナイザーに取り込み、シンクライアント型のダッシュボードビルダー（Dashboard Builder）インターフェースで各要素を自由に組み合わせて、セルフサービス形式で実用性に富んだダッシュボードを構築できます。

デプロイメント機能も充実しており、作成したレポート、チャート、ダッシュボードはメニューを通じ、さらには URL リクエストを生成することで自動的に公開できます。ERES には、高度なスケジュール設定とアーカイブ機能が備わっており、レポートのバージョニング、メール送信スケジュール、バッチプリントジョブなどがサポートされています。

Java API インターフェースを通じた緻密なコンフィギュレーションとデプロイメントも利用可能で、作成したレポートやチャートを柔軟に取り込んで独自の構成環境を構築したり、他のアプリケーション環境に統合したりもできます。

レポートのデザイン

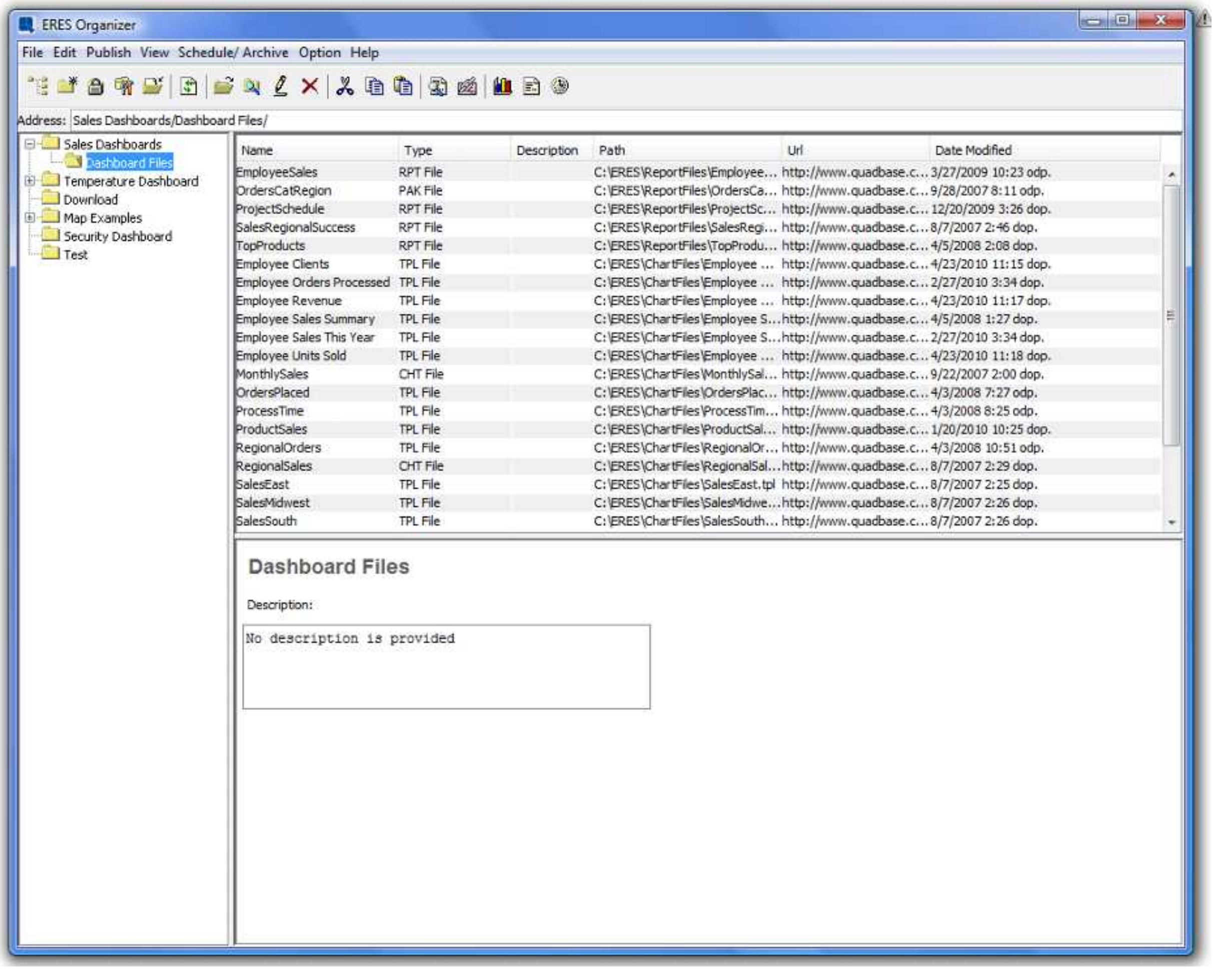
EspressReport ES（ERES）におけるレポートのデザインは、クライアント／サーバ アーキテクチャを基盤としています。ユーザは中心となるインターフェースからサーバにログインして、レポートやチャート、さらにはデータ／クエリツールへとアクセスできます。レポートを作成したらファイルシステムに保存でき、コーディングは一切不要でそのまま公開することができます。

Organizer

オーガナイザー（Organizer）は、EspressReport ES（ERES）の中枢を担うファイル管理インターフェースです。ユーザは、オーガナイザー内でチャートを作成し、チャートのデータソースやクエリを設定し、スケジュールやアーカイブ ジョブを作成できます。これらに加え、管理者ロールのユーザは一般ユーザの権限とアクセス権を設定できます。

オーガナイザーはチャート、レポート、マップ、ダッシュボードのリポジトリとしても機能します。オーガナイザーに追加されたダッシュボードはレポーティング ポータルから開くことも、URL や API を通じて開くことも可能です。オーガナイザーでは、エンドユー

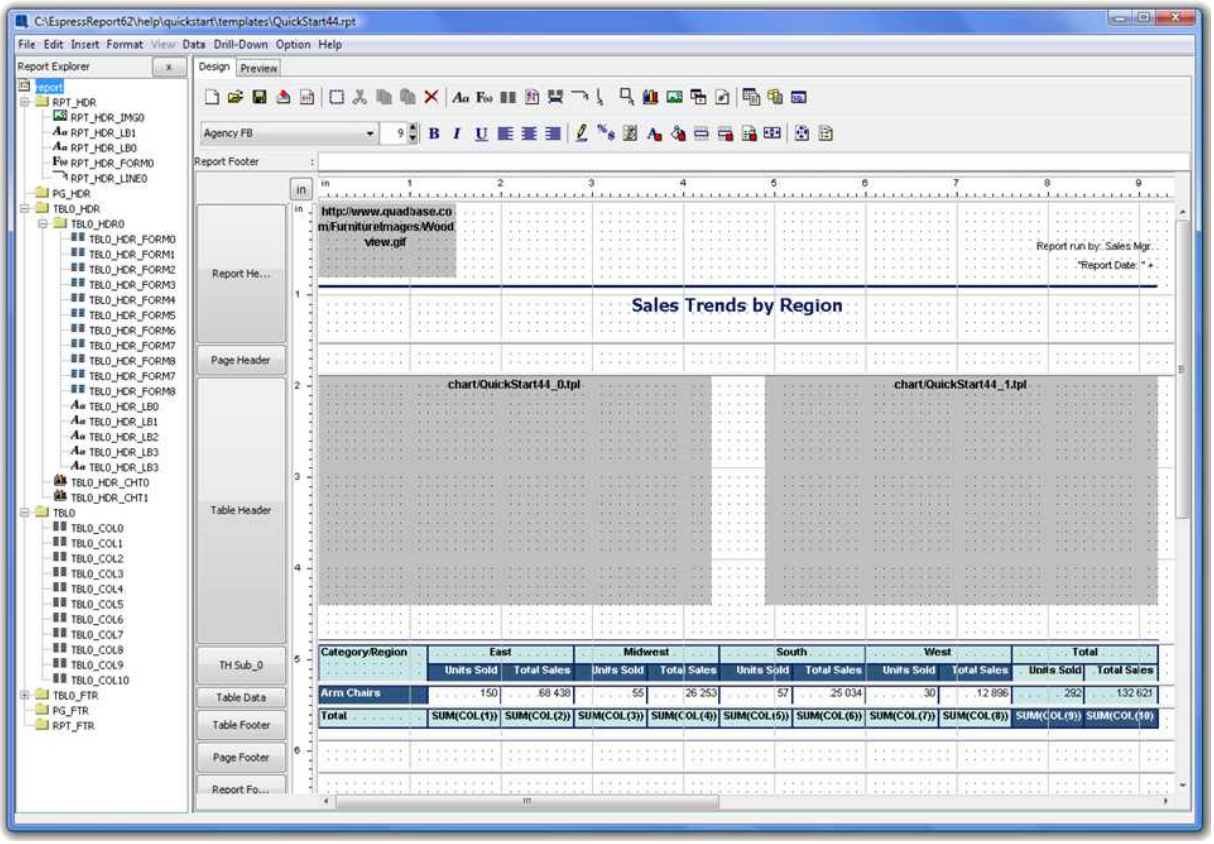
ザのローカル環境にとらわれず、いつでもレポートの作成や修正が可能で、デプロイメント環境の設定をカスタマイズする必要はありません。



Pure Java で設計されたインターフェースは実質的にいかなるプラットフォームでも実行可能なので、オーガナイザーは、クライアントサイドのアプリケーションまたはリモートのウェブブラウザのコンテンツとして機能します。

Report Designer

レポートデザイナー（Report Designer）は、オーガナイザー内で起動できる、レポートのデザインと開発を簡素化するツールです。使いやすいグラフィカル ユーザインターフェースにより、ドラッグ アンド ドロップの簡単な操作でレポートを自由にカスタマイズできます。従来のレポーティング ツールに慣れているユーザなら、レポートデザイナーのバンド型インターフェースに慣れるのにほとんど時間はかからないはずです。他の一般的なレポーティング ツールとの共通点も多く、新規ユーザも特別なトレーニングなしで、すぐに使いこなせるようになります。



レポートデザイナーは Pure Java コンポーネントであり、あらゆるプラットフォームで簡単に稼働させることができます。ウェブブラウザでアプレットとして実行できるので、クライアント環境でのインストールやセットアップの作業は必要ありません。柔軟性を最大限に高める API フックも用意されており、多種多様な構成環境に自動的に適用させることができます。

レポートのレイアウト

EspressReport ES（ERES）では、5 種類の基本的なレイアウトを使用してレポートを作成できます。いずれのレイアウトでも、表形式の入力データをグループ化されたフォーマットやサマリー フォーマットにマッピングすることで、レポートを構成することができます。マッピングはウィザードのわかりやすい手順に沿って実施でき、一度レポートを必要データで構成してしまえば、その後はレポート上のあらゆる要素を自由にカスタマイズすることができます。

基本的なレイアウトは以下の 5 種類です。

Simple Columnar Layout（シンプル カラム型レイアウト）— もっとも基本的なレイアウトです。単一の表形式でデータを表示するレポートで、グループ分けやセクションの区切りはありません。

Summary Break Layout（要約区切りレイアウト）— シンプル カラム型レポート同様、カラムデータを表形式で表示するレイアウトですが、データをセクションごとに区切って、要約フィールドを挿入することができます。

Crosstab Layout（クロスタブ レイアウト）― カラムデータを縦横の格子状のフォーマットで表したレイアウトで、スプレッドシートに近い形式です。横の行と縦の列を要約できて、複数次元のデータを二次元フォーマットで表示することができます。

Master & Details Layout（マスター詳細レイアウト）― マスターフィールドにもとづいてグループ化された表データのレポートレイアウトです。マスターフィールドが表データに対して 1 対多の関係を持つことで、より柔軟なレポート構成が可能になります。

Mailing Label Layout（メールラベルレイアウト）― クラスタでデータを事前にグループ分けしてメール／アドレスラベルのフォーマット化を容易にするためのレポートレイアウトです。

レポート作成ウィザードでは、データやレイアウトオプションの選択に加え、タイトルやページ番号など、レポートの補足コンテンツを追加したり、既定スタイルかカスタムスタイルを選択してレポートの形式を整えたりできます。これらの機能により、ユーザはレポートの各種要素の編成や見た目の調整に時間をかけることなく、完成形のレポートをすばやく生成することができます。

レポートをカスタマイズするパターンは数限りなく、レポートの各構成要素は実質上すべてカスタマイズ可能なので、すべてのデータはユーザが望む限り、ありとあらゆる形式でプレゼンテーションすることができます。

Report Designer の各種機能

EspressReport ES（ERES）はレポートデザインに画期的なアプローチを取り入れています。その特長は、デザインウィンドウに表示された複数のバンドが個々にレポートの異なるセクションを指し示すバンド型インターフェースに代表されます。この仕組みにより、各セクションにオブジェクトを自由形式で挿入し、調整することができます。オブジェクトはグループごとに一括で移動、フォーマット化できるので、さらに柔軟な調整が可能になっています。しかも、この機能はカラム形式のオブジェクトに対しても有効です。また、オブジェクトをグリッドに合わせて自動配置できるスナップ トゥ グリッドやガイドライン設定によって、レポートのさまざまな構成要素のポジショニングを整理する機能も充実しています。作成したレポートには、統合エクスプローラ インターフェースによる簡単ナビゲーションで、いつでもアクセスできます。

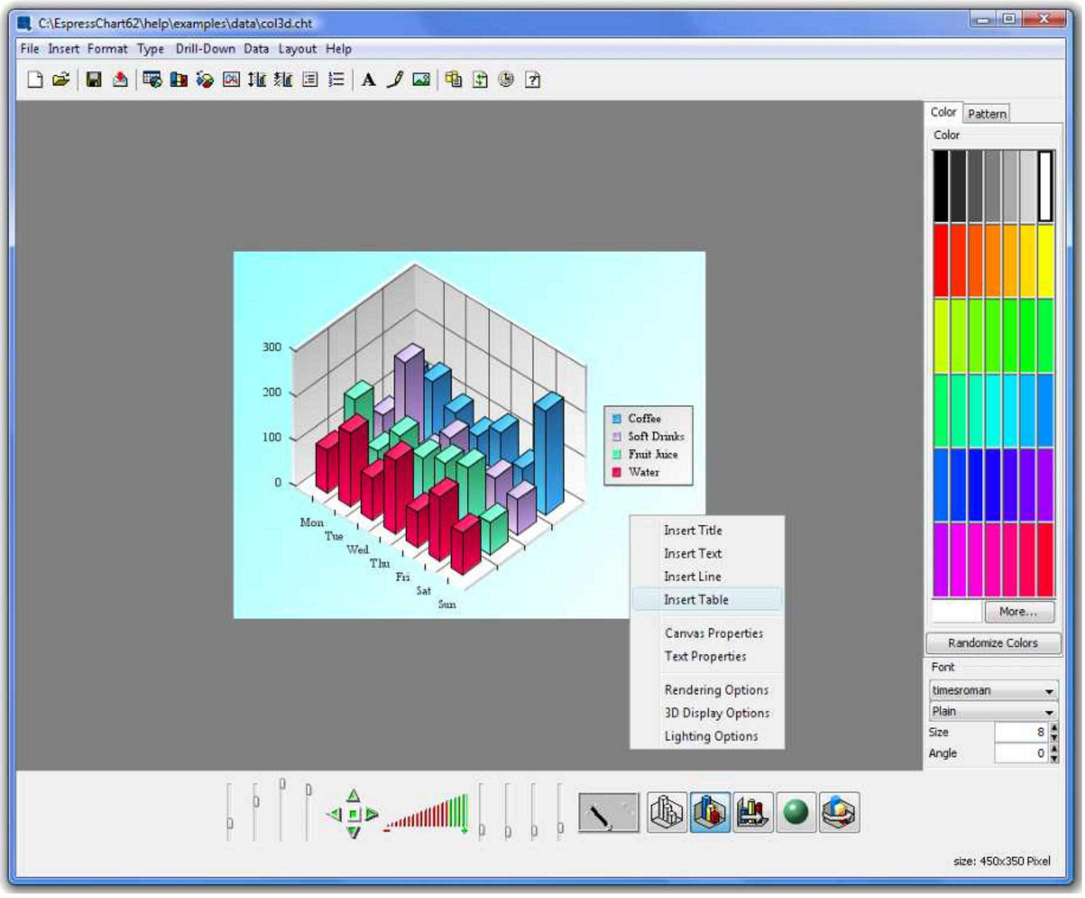
オブジェクトのフォーマットはグローバルに調整できます。つまり、ラベル、式、カラムフィールドなど、各種レポート オブジェクトにレポート全体にわたって適用されるオブジェクト プロパティを割り当てることができます。必要に応じて、オブジェクトの特定のプロパティを変更したり、デフォルトの属性を変更したりすることも可能です。グローバル フォーマットは XML ファイルに抽出できるので、手軽に他のレポートに同じフォーマットを適用することができます。

レポートデザイナーには、パラメータ化、スクリプティング、ドリルダウン、サブレポートなど、多くの高度なレポーティング機能も備わっており、これらすべてにシンプルなインターフェースからアクセスできます。ユーザはさまざまな機能をレポートに取り入れ、時間をかけずに、プロフェッショナルな仕上がりのレポートをデザインすることができます。

データ表示の視覚効果

EspressReport ES（ERES）は、レポートデザインだけでなく、チャートデザイン機能も充実しています。チャートデザイナー ユーティリティが内蔵されており、30 種類以上の二次元および三次元のチャートタイプから最適なタイプを選択して、データポイントをグラフ上に展開することができます。チャートは、色やフォントやサイズなど、あらゆる要素をカスタマイズでき、これをすべてチャートデザイナー ウィンドウで完成させてからレポートに挿入することができます。ERES のチャート作成機能は真の 3D チャートを生成できて、図表のパン、ズーム、ローテーションも可能なうえ、色の濃淡やアンビエント ライト効果などの微調整も自由自在です。

さらに高度なチャート機能としては、タイムシリーズ ズーミング、データ ドリルダウン、ランタイム テキスト置き換え、マウスオーバー ポップアップラベルなどがあります。統計や分析に役立つ機能としては、チャートに傾向線（トレンドライン）を引いたり、ノーマルカーブで正規分布を表したり、度数分布図（ヒストグラム）でデータの発生頻度や、パレート図でデータの優先順位を示すこともできます。箱ひげ図でデータの動向を示したり、チャートに制御線や制御領域を追加してデータの許容範囲を示したりもでき、さまざま手法を駆使して、チャートの視覚効果を高めることができます。



チャートのテンプレートはレポートとは別に保存されるので、チャートは単独で使用することも、レポートに挿入することも可能です。チャート作成機能の柔軟性により、レポートのみならず、ニーズに応じてウェブページなどにもチャートを追加でき、デー

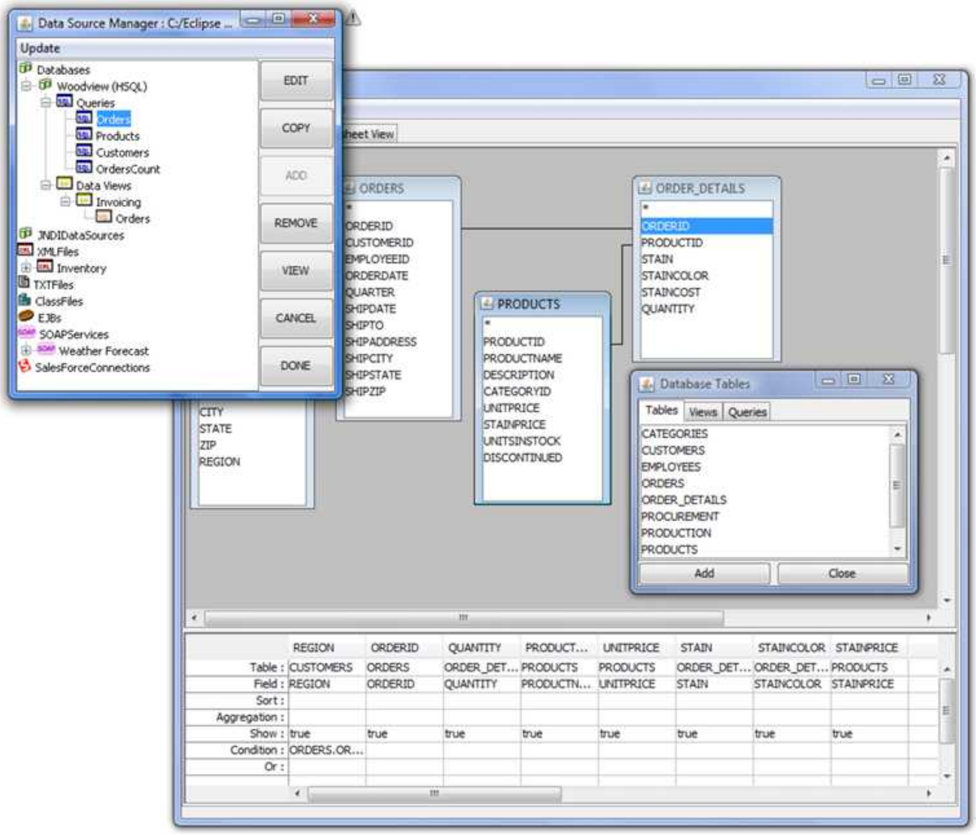
タを視覚的にわかりやすく提示することができます。ERES では、既存のウェブコンテンツに画像をダイナミックに自動ストリーミングすることもできます。

あらゆるデータソースをエンタープライズ サポート

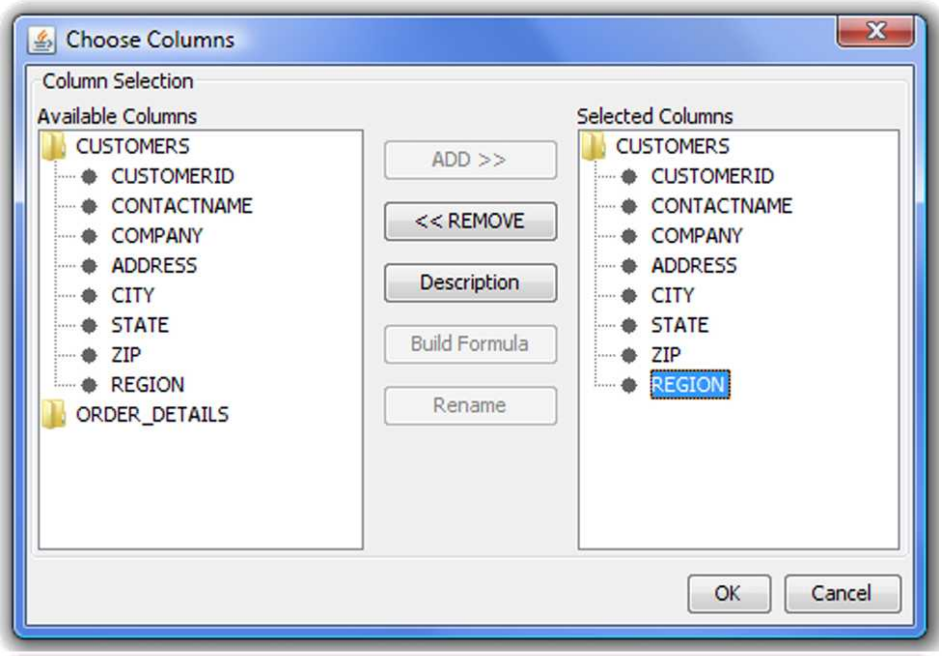
EspressReport ES（ERES）は、企業全体のあらゆるデータをレポート用に取得、管理できるインターフェースを備えています。JDBC による直接のデータベース呼び出しはもちろん、JNDI を通じて SQL を駆使することも、Java クラス、EJB からのデータ呼び出しや、コネクション プールによるアプリケーション モデル インフラストラクチャの活用も可能です。XML 形式のアプリケーション データや SOAP データソースにも対応しています。データソースとクエリは XML レジストリに保存され、複数のレポートで再利用できます。レジストリは、オーガナイザー（Organizer）から起動するデータソース マネージャー インターフェースを通じて作成、更新できます。

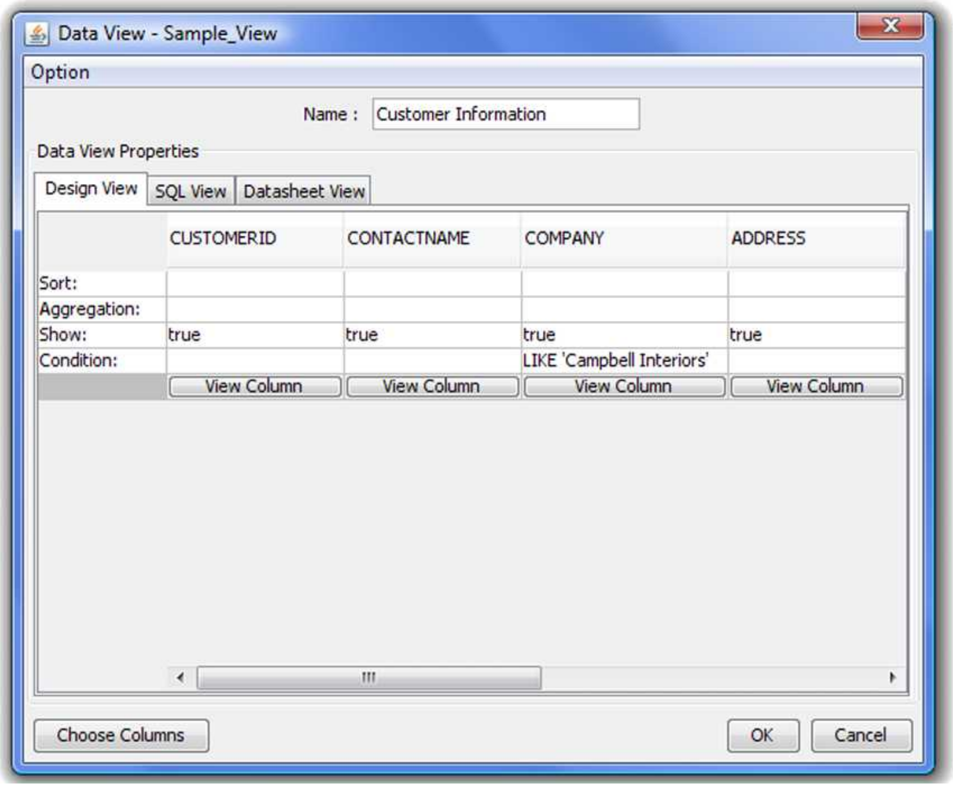
SQL による完全制御

他の多くのダッシュボード ツールとは異なり、EspressReport ES（ERES）では SQL 完全制御によるレポートデータのクエリが可能です。これに加え、3 種類のデータベース クエリ インターフェースが用意されています。上級者ユーザは、単に SQL ステートメントを入力したり、インポートしてクエリを実行し、レポートデータを生成できます。使用できる SQL ステートメントに制限は一切ありません。中級者ユーザはクエリビルダーの GUI を利用できます。クエリを QBE 形式で作成できるので、フィールド選択、条件、結合、集約の指定が簡単なうえ、カスタム データベース関数をインポートして複雑な式を編集できます。



データベースの構造やテーブル名などに精通していないユーザは、データビュー（Data View）インターフェースを利用できます。データビューは、管理者ユーザがテーブル、テーブル結合、フィールド、さらにはフィルタを事前に定義して、エンドユーザがローカルで使用できるように設定するものです。エンドユーザは基底となるデータベースの構造を理解していなくても、フィールド（通常、エイリアスで定義）やフィルタを選択して使用することができます。

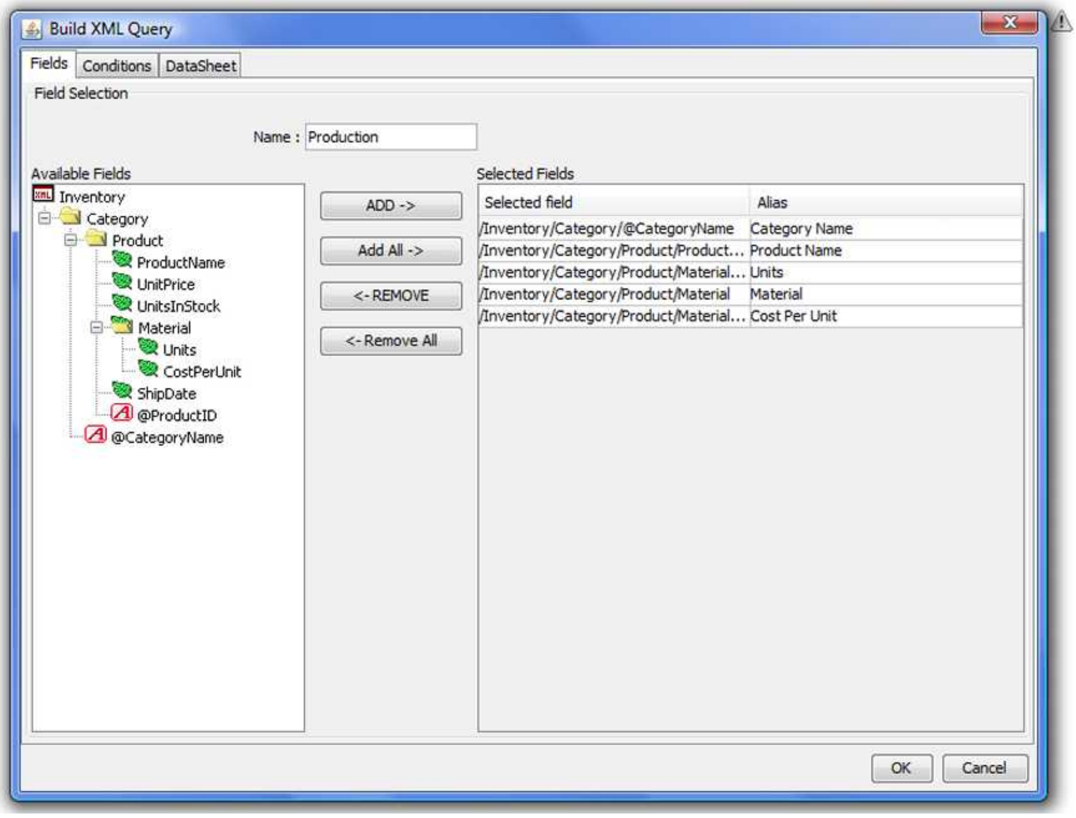




データビュー インターフェースは、QuickDesigner（クイックデザイナー）とも連携しているので、シンクライアントならぬゼロクライアントの HTML インターフェースでアドホック クエリを作成できます。

XML サポート

アプリケーション コンポーネント間のデータの転送やエンコーディングを XML に依存しているアプリケーション モデルは少なくありません。このようなデータをレポートに表示するために、EspressReport ES（ERES）は XML データを定義し、クエリを生成するための広範なインターフェースを備えています。クエリは、DTD または XML スキーマにもとづいて XML ファイルから生成できます。あるいは、サーブレット／JSP からの XML 出力ストリームからも生成可能です。



Java オブジェクト／アレイデータ

EspressReport ES（ERES）では、リレーショナル データベースや XML ソースからデータを取得するだけでなく、EJB への接続を通じて Java オブジェクトやアレイから直接データを取得することもできます。その場合、ランタイムには、データが直接レポートに受け渡され、デザインタイムではオブジェクト／アレイのデータが Java クラスファイルに抽出されます。アプリケーション データを ERES に取り込むためのインターフェースが数通り用意されています。

サンプルコード：レポート用に単純なデータアレイを取得

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

public class SampleData implements IDataSource {
     // データを引数として受け渡すための DbData をセット
     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);

      // からのコンストラクタを作成
      public SampleData() {};
      // getResultSet を実装
      public IResultSet getResultSet() { return data; }
}
```

SOAP データソース

EspressReport ES（ERES）では、SOAP（サービス指向アーキテクチャ プロトコル）を使用してデータを取得することもできます。SOAP データソースは、WSDL 準拠のタイプと Salesforce タイプの 2 通りがサポートされています。WSDL を使用して SOAP データに接続する場合は、サービスの URL、SOAP アクション、オペレーション名、パラメータなどの情報は必要ありません。必要情報を含む WSDL ファイルの場所のみ指定する必要があります。データソース マネージャー（Data Source Manager）と簡単なやり取りをするだけで、WSDL SOAP データソースの設定が完了します。

既存の Salesforce ユーザが Salesforce SOAP データソースを設定する場合も同じで、データソース マネージャーで簡単に設定することができます。ただし、ユーザは有効な Salesforce アカウントを持ち、信頼できるネットワークから Salesforce アカウントにアクセスできる必要があります。また、データソースの設定には SOQL（Salesforce オブジェクト クエリ言語）が欠かせません。クエリのパラメータは、データベース クエリと同じ要領で追加することができます。

データソースの更新

レポートデータはテンプレートとレジストリの両方に保存されます。これにより、レジストリ ファイルに依存しないレポート デプロイメントが可能になります。レポートレベルで行われたデータソースへの更新は、レジストリに保存されます。一方、レジストリに直接行われた更新（データベース接続情報の更新など）は、複数のテンプレートに簡単に反映させることができます。

この仕組みは、多くのテンプレートを使用するデプロイメントの移行（たとえば、開発環境から本番環境へのマイグレーション）を容易にします。変更が必要なのはレジストリに保存された接続情報に限られ、それもテンプレートに自動更新されます。

高度レポーティング機能

QuadBase の強力な第 4 世代レポーティング エンジンを搭載した EspressReport ES（ERES）には、レポートおよびチャート作成のあらゆるニーズに応える高度な機能が備わっています。テンプレートの汎用性を高める高度なスクリプティングとパラメータ化機能や、ほぼ無限のバリエーションでレポート作成を可能にする詳細コンフィギュレーションおよびレイアウト オプションが揃っています。

レポート パラメータの活用

EspressReport ES（ERES）には、レポートのコンテンツをパラメータ化して動的に置き換えるための、さまざまな機能が完備されています。XML クエリやデータベースへの動的なフィルタを何通りも追加することができます。レポートのパラメータを Java クラス ファイル データソースに関連付けたり、レポートデータに独自のフィルタ基準を設定したりすることも可能です。パラメータの値は、データベースフィールドや関数だけでなく、SELECT ステートメントにも割り当てることができます。計算式のパラメータでは、ランタイムでテキストや値を置き換えられ、レポート自体のラベルも置き換え可能です。

ランタイムのパラメータ値は自動設定することも手動入力することも可能です。ERES で特長的なのは、パラメータの選択肢を表示する HTML フォームを自動生成する機能や、サーブレットを使用して任意のフォーマットでレポートを出力できる点です。

「パラメータの連続指定」も、ERES に特長的な実用性の高い機能です。デフォルトでは、すべてのレポート パラメータを一度に入力することがプロンプト ダイアログで求められますが、パラメータの中には異なる大量の値を持つデータベースカラムに割り当てられているものもあり、一括指定に向いていない場合があります。大量の選択肢リストから選ぶのが困難であったり、パラメータの組み合わせが不適切で、何のデータも返さないパラメータを選んでしまう可能性もあります。

このような状況を回避するには、パラメータの入力順序を前もって設定しておくのが有効です。これを設定すると、ユーザはダイアログで既定の順序で一つずつパラメータを選択し、一つの選択が次の選択肢のフィルタとなって、無効な組み合わせの選択が防止されます。

スクリプティング機能

EspressReport ES（ERES）には便利なスクリプティング ライブラリが内蔵されており、レポートの要素を単純明快な構文のスクリプトを使用して、条件に応じてフォーマット化したり、ランタイムで調整したりできます。スクリプトでは条件を設定できるほか、セルのフォーマット化、特定のデータ行の処理、計算の繰り返し実行、レポート全体の修正まで行えます。

また、このスクリプティング機能は動的なフォーマット機能をテンプレートにまで適用でき、テンプレートの再利用を促進して、レポートをカスタマイズするためにコードを記述する開発者の手間と時間を省きます。

サブレポート

サブレポート機能を使用すると、複数テーブルを多様な構成で組み合わせたレポートを作成できます。1 つのレポートに複数の異なるデータソースからデータを取り込むことができ、パラメータ共有やリンク設定によって複数データソースを連携させて活用することができます。

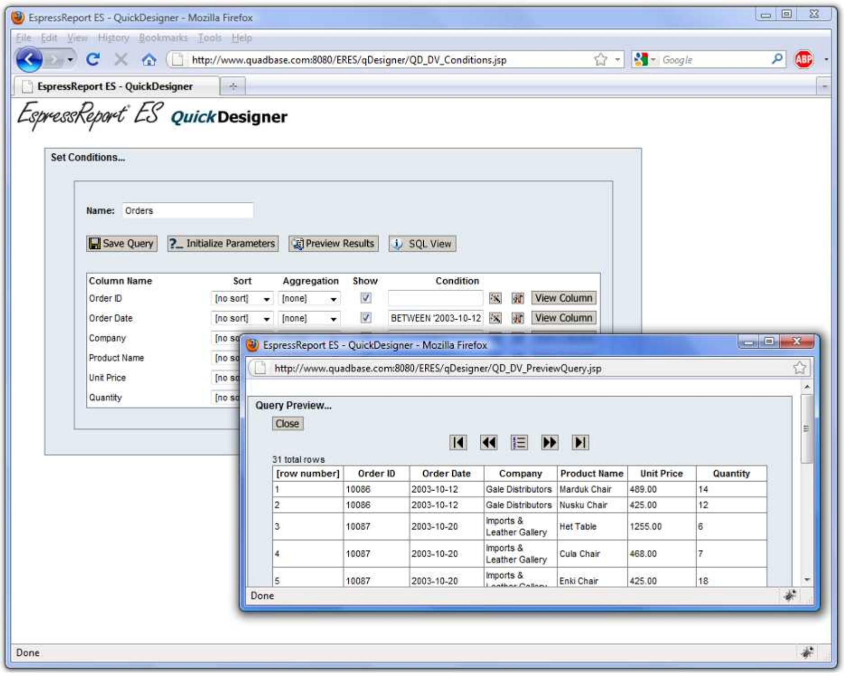
詳細情報の深掘り自動設定

レポートでもっとも必要とされる機能の 1 つに、最上位のレポートやチャートから詳細を関連情報に掘り下げていく「ドリルダウン」機能があります。EspressReport ES（ERES）では、レポートやチャートに簡単にドリルダウンを設定することができます。ERES では、ドリルダウンの各階層に 1 つのレポートまたはチャートテンプレートを設定するだけで階層間のリンクが自動的に処理されるので、コーディングなしでレポートにドリルダウンを実装できます。

アドホック レポーティング

上記の各種デザイン機能に加え、EspressReport ES（ERES）には高度なアドホック レポーティングとアドホック クエリ機能が備わっています。これは、誰でもすぐに使いこなせるシンプルなシンクライアント型のインターフェースで、エンタープライズ環境内のあらゆるユーザがケースバイケースのニーズに応じて柔軟にクエリを作成し、任意のデータソースからレポートを構築して公開できる仕組みです。

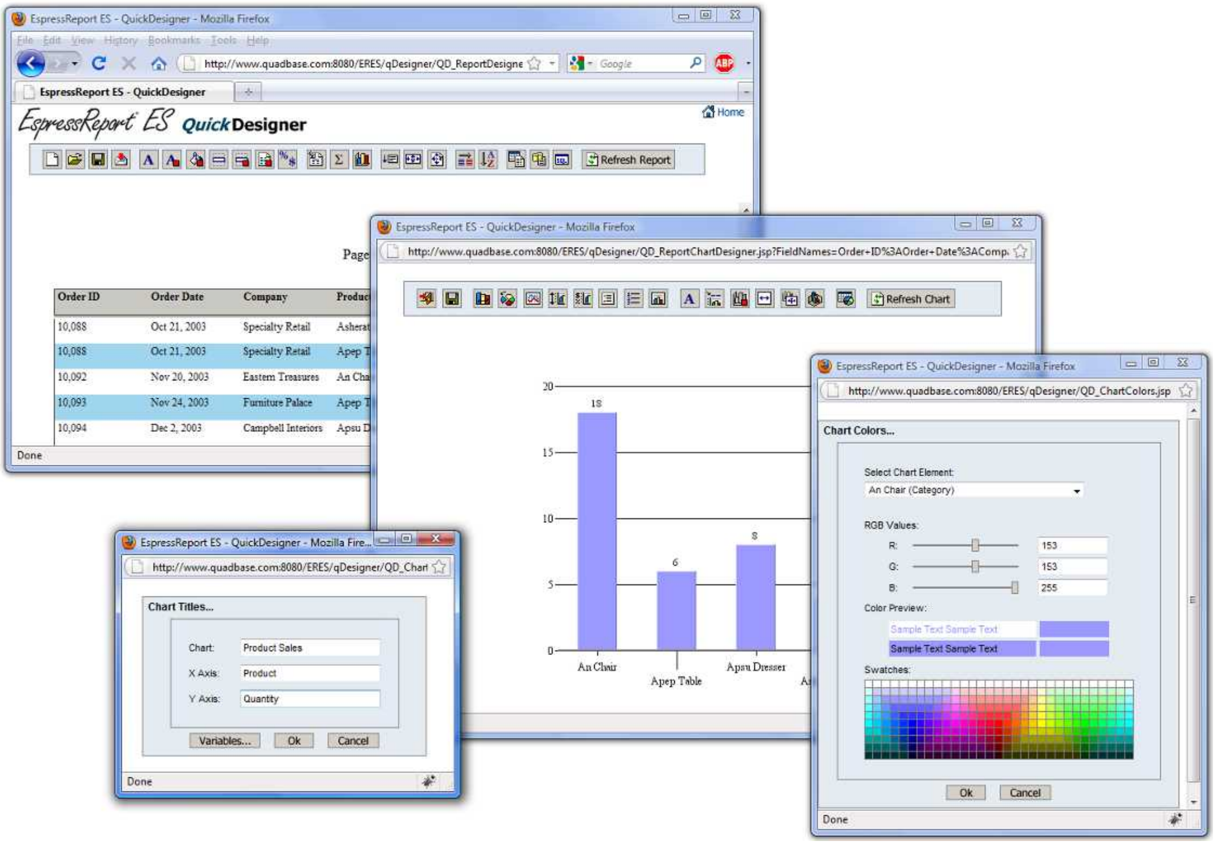
ERES のアドホック クエリおよびレポートは、クイックデザイナー（QuickDesigner）インターフェースで作成します。クイックデザイナーは、定義済みであれば、企業内のいかなるデータソースからでもレポートやチャートを作成できる、HTML ベースのシンクライアント型ツールで、これを使用してデータビュー（Data View）に対するクエリを作成することもできます。選択してクリックするだけの非常にわかりやすいユーザ インターフェースなので、誰でも基本的なレポーティング機能を、特別なトレーニングなしで使いこなせるようになります。



クイックデザイナーでクエリを作成したら、クイックデザイナー内のレポートデザイナー ページに移動して、レポートを作成／修正します。このユーザ インターフェースでは、レポートに各要素のグループ分けやサマリーを追加したり、ソーティングを変更したり、カラムの順番を並べ替えたりできます。ページ上部のツールバーを使用して、レポート フォーマットの基本的な編集、色やフォントの設定、データ形式の設定を行えます。レポートへのタイトル、ラベル、日付、ページ番号の追加も、このページで行えます。

チャートのフォーマット設定は別のインターフェースで行います。チャートは独立したオブジェクトとして作成することも、レポートの要素として追加することも可能です。チャートのサイズや色などの見た目の調整、タイトルの追加などを、チャート専用インターフェースで行えます。

レポートが完成したら、PDF や MS エクセルなど、さまざまな形式でクライアントサイドにストリーミングできます。サーバサイドのリポジトリに保存して、他のユーザと共有することも可能です。



レポートの公開

EspressReport ES（ERES）でレポートを公開する方法は多岐にわたります。ERES にはパブリッシング機能が統合されており、一切のコーディングなしでレポートを出力し、公開できます。また、高度なスケジュール設定が可能で、さまざまなフォーマットのレポートを指定の時間に公開することも可能です。さらに、Java API フックを利用すれば、各種レポート機能をウェブアプリケーションに統合できます。

出力フォーマット

EspressReport ES（ERES）では、レポートを多種多様なフォーマットで生成できます。基本的なフォーマットとしては、あらゆるブラウザと互換性のある標準的な HTML のレポート生成が可能です。より複雑なフォーマット要件を満たすには、DHTML スタイルシートを適用してレポートを生成することもできます。印刷用に、あるいは電子文書用にページ分けしたレポートが必要な場合は、高品質の PDF 文書を出力できます。MS Office ユーザ向けには、Excel スプレッドシートや MS Word で読み取り可能なリッチテキスト形式（RTF）でレポートを生成できます。そのほか、CSV、XML データファイル、区切りテキストファイルなどをサポートしています。

レポートだけでなく、チャートもさまざまなフォーマットで生成可能です。GIF、JPEG、PNG、PDF、WMF、Flash、SVG、BMP に加え、ハイパーリンク、ドリルダウン、ポップアップラベルなどのインタラクティブなイメージマップを生成できます。

アプレットの活用

EspressReport ES（ERES）は、アプレットやクライアントサイドのデプロイメントによるレポート公開にも対応しており、レポートやチャートの表示とインタラクションを可能にする各種ツールを提供しています。

レポートビューア

レポートビューア（Report Viewer）は、クライアントマシンにウェブブラウザを介してレポートを表示するアプレットです。AWT バージョンと Swing バージョンがあり、AWT バージョンではプラグインをダウンロードせずにレポートビューアを実行できます。レポートビューアはあらゆるレポート テンプレートにも対応でき、レポートをハイパーリンクで他のレポートに直接リンクすることも、ウェブページにリンクすることも可能です。レポートはページ分割された文書形式で表示されます。レポートデザイナーから内蔵アプレットで直接 HTML ページを生成することもできます。エンドユーザはページ間やリンクされたレポート間をポップアップメニューを使用して行き来でき、しかも、このポップアップメニューはカスタマイズ可能です。

ページビューア

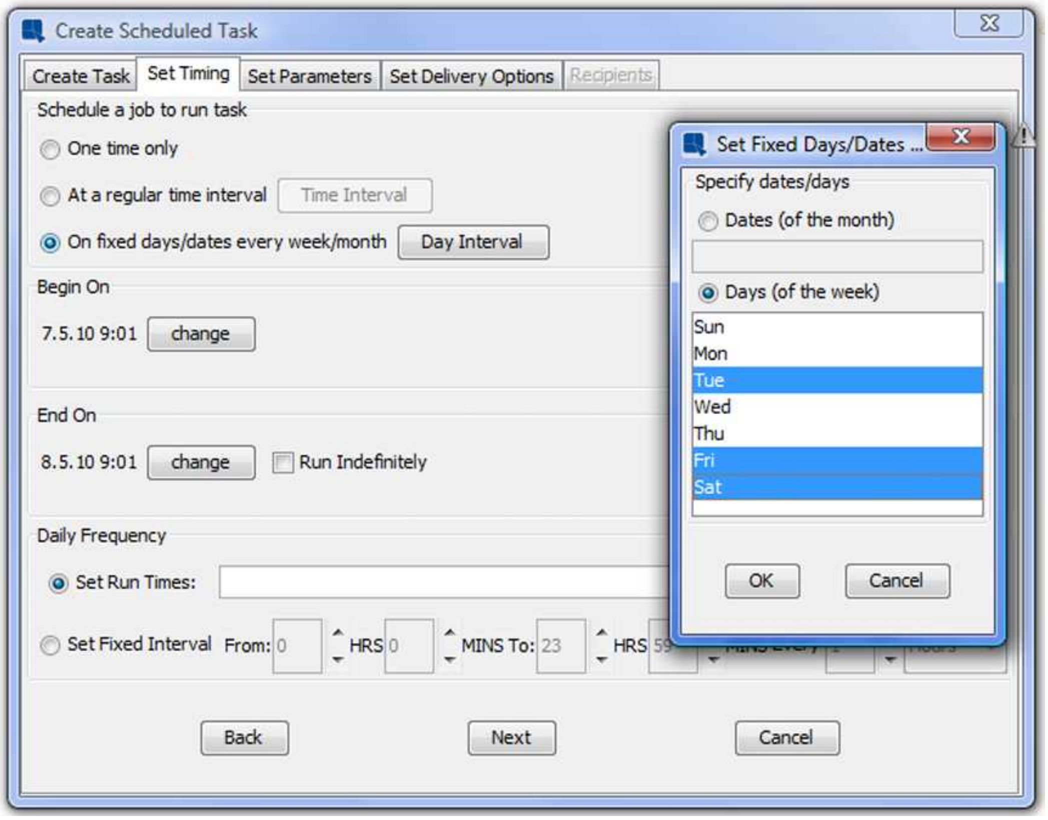
レポートの分量が膨大な場合、まるごとクライアントマシンにロードするのが実用的でない場合があります。その場合は、ERES のページビューア（Page Viewer）コンポーネントを使用すると便利です。ページビューアは、基本的にはレポートビューアと同じように機能しますが、クライアントサイドのリクエストに応じて必要なページだけを送るテクノロジーが組み込まれている点が異なります。ページビューアを使用すると、長大なレポートでもメモリ不足にならずに、クライアントマシンに必要に応じて表示したり、プレビュー表示したりできるようになります。

スケジュール設定

EspressReport ES（ERES）には、スケジュールにもとづいてレポートを発行できるオプションが充実しています。オーガナイザー（Organizer）インターフェースでスケジュールを設定でき、エクスポートされたレポートやチャートをさまざまなフォーマットで公開できます。また、レポートを履歴データとともに保存するレポート アーカイブも作成できます。

スケジュールを設定することで、レポートはサーバのファイルシステムに HTML／DHTML、PDF、Excel、CSV 形式で、チャートは PNG、GIF、JPEG、SVG 形式で保存できます。また、レポートおよびチャートを任意の受信者にメール送信することも可能です。さらに、ネットワーク プリンタに送信してバッチ印刷することも、直接 FTP サーバに送信して公開することもできます。

レポート バースティング機能を利用すると、長大なレポートはグループデータにもとづいて自動的にセクション分けされ、複数のユーザやグループに対して個別に公開することができます。バースティング機能によって、高ボリュームのスケジュール生成処理が過度なデータベース負荷やパフォーマンス低下をともなわずに実現できます。



レポート／チャートは、複数をまとめて一つのパッケージでスケジュールでき、一つのメールに複数のファイルを添付したり、リンクを送信したりできます。スケジュールには特定日数のインターバルを指定でき、たとえば、9:00AM から 5:00PM の間に 1 時間おきにレポートを発行するように設定したり、あるいは不定期で特定日時のエクスポートを指定したりできます。コンテンツをパラメータ化したレポートは、複数の受信者にパラメータ設定値にもとづいて送信することができ、たとえば、地域名をパラメータ化したレポートを異なるメール受信者グループに地域別に送信されるようにスケジュール設定できます。さらに、スケジュール設定では高度なエラーハンドリングや通知とログ生成もサポートされています。

ERES には、アーカイビング／バージョニング機能もあり、レポートの履歴を容易に保存／管理することができます。データのスナップショットが記載されたバージョン レポートも、普通のレポート同様に発行／公開できます。画面上で特定バージョンの日付を選択すれば、ERES でサポートされているどの形式でも自在にアーカイブ レポートを発行できます。

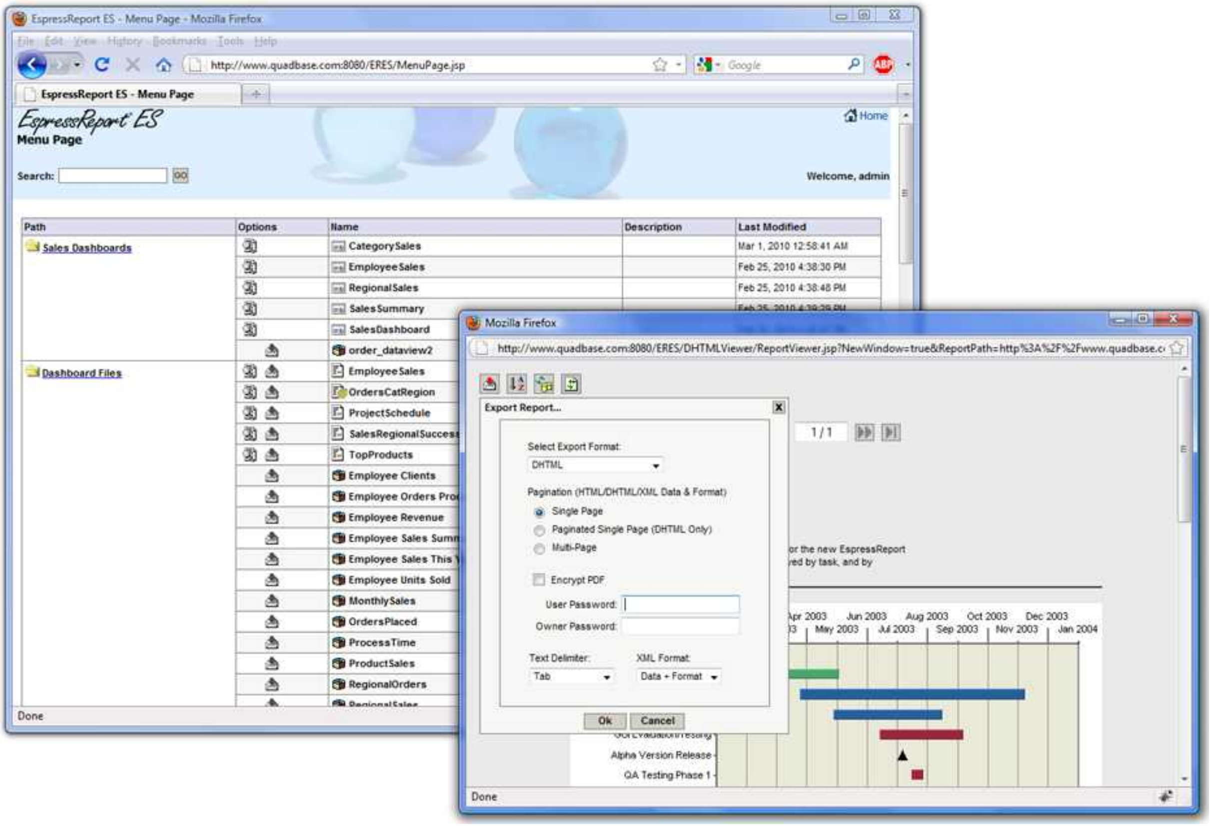
スケジューリング インターフェースには API フックも完備されており、ユーザはスケジュール ジョブをプログラミングによって自動的に作成／修正することもできます。

レポートの自動公開

EspressReport ES（ERES）で発行したレポートはリポジトリに保存され、オーガナイザー（Organizer）インターフェースからアクセスできます。これらのレポートは、一切コーディングせずに自動的にウェブに公開することができます。

レポーティング ポータル

ERES には JSP／JSTL 仕様のインタラクティブなレポーティング ポータルが備えられ、それを通じてリポジトリのレポートを閲覧／操作することができます。このポータルは、フルカスタマイズ可能なシンクライアント型インターフェースで、ユーザはオンデマンドでレポートを発行したり、スケジュールされた直近のレポートを表示したり、さらにはアーカイブから履歴レポートを取り出したりできます。



レポートの表示は DHTML ビューアで処理されます。単なるビューアにとどまらず、レポートのフィルタリング、マルチレベル ソート、グループ分けなどをこなすシンクライアント型インターフェースの多機能ユーティリティです。ユーザは、ページ上部のツールバーで各種ファンクションを制御できます。

URL

レポートやチャートは URL で呼び出して表示することもできます。URL はオーガナイザー インターフェースで生成することも、独自に作成することも可能で、エクスポート可能なすべてのフォーマットに対応しています。チャートは HTML ページとしてもエクスポートでき、既存のウェブコンテンツに動的な画像として追加することができます。

API

EspressReport ES（ERES）では、Java API の柔軟性をフルに活用できます。ほとんどすべてのインターフェースに API フックが用意されており、ユーザは独自にカスタマイズしたレポート デプロイメントを利用することも、ERES インターフェースを他のアプリケーションに統合することも可能です。

メニューAPI

レポート リポジトリへの API フックとして利用できるメニューAPI では、公開済みのレポートの一覧を表示できます。これには、担当ユーザ、権限、スケジュール、アーカイブなどの情報も含まれます。この API フックにより、ユーザは独自のレポーティング ポータルを構築したり、レポート リポジトリを他のアプリケーションに統合したりでき、そのようなインターフェースを通じてレポートの公開や更新が自在に行えるようになります。ユーザがレポート／チャートのテンプレートを修正した際には、デプロイメントやコーディングを調整することなく、更新されたレポートをそのまま発行することができます。

サンプルコード：リポジトリにアクセスして有効なレポートをユーザに提示

```
 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 {

       // オーガナイザーにアクセスして処理を可能に – QbMenu オブジェクトを作成
       qbMenu = new QbMenu(context.getAttribute(QbMenu.QBMENUDATA));

       // 指定のユーザ名とパスワードでログイン
       String userName = request.getParameter("USERNAME");
       String password = request.getParameter("PASS");
       boolean loginSuccess = qbMenu.login(userName, password);

       // ログインに成功したら、レポートとチャートのリストを作成
       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();
        }
     }
  }
```

レポート API

レポート API は、レポートやチャートに直接 API フックを適用できます。これを利用して、ユーザはレポートを発行するための独自のメカニズムとロジックを構築できます。レポート API により、レポートをユーザの指定したパラメータにもとづいて即座に生成でき、クライアントサイドのブラウザに直接ストリーミングできます。生成されたレポートはサーバにキャッシュすることも可能です。この API は、ほぼすべてのレポート オブジェクトに対応可能で、しかもレポートのデザインはプログラミングによって自動制御されます。また、レポートデザイナー（Report Designer）で作成されたテンプレートにも対応しているので、レポートのプロパティをランタイムで修正できます。レポート API のフックはレポートデザイナー インターフェースにも直接適用でき、プログラミングによってすべて自動的に実行可能です。

サンプルコード：レポートのデータソースを変更して、クライアントに HTML でストリーミング

```
 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 {

     // ERES サーバに接続
     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 {
```

```
        //データベースへの接続をプールから取得
        Connection conn = dbPool.getConnection();

        // QbReport オブジェクトを設定して接続時に受け渡し
        QbReport report = new QbReport((Applet)null, template, false, false,
false, true);
        report.getInputData().setAllDatabaseInfo(conn);

        // レポートを DHTML にエクスポートしてクライアントにストリーミング
        report.export(QbReport.DHTML, toClient);

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

      toClient.flush();
      toClient.close();
    }
  }
 }
```

スケジューラ API

スケジューラ API は、スケジュール設定プロセスに直接 API フックを適用できます。これを利用して、ユーザはスケジュールジョブをプログラミングで自動的に作成したり、既存のジョブを修正したりできます。アーカイブ ジョブの作成／修正も可能です。

サーバ拡張クラス

ERES のレポート公開機能には、独自のニーズに合わせたカスタム ビジネス ロジックを適用することもできます。これは、ポータルで表示するレポートはもちろん、URL を介したり、スケジュール設定で公開するレポートにも適用可能です。仕組みとしては、レポートやチャートが公開される際、そのプロセスに途中介入するカスタムサーバ拡張クラスを実装し、この拡張クラスがレポートやチャートの API を通じた修正を可能にして、修正後にまたレポートやチャートを公開プロセスに戻します。この仕組みにより、レポート／チャート生成へのカスタムロジックの適用が、別のシステムを導入せずに実現します。

サンプルコード：ERES で処理されるレポートにユーザ名と処理日を追加

```
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 {

        // レポート作成ユーザを表示するラベルを追加
        ReportCell userLabel = new ReportCell();
        userLabel.setText("Report Run By: " + username);
        userLabel.setFont(new Font("Arial", Font.PLAIN, 8));
        userLabel.setAlign(IAlignConstants.ALIGN_RIGHT);
        userLabel.setHeight(0.2);
        userLabel.setWidth(1.6);
        userLabel.setY(0);
        userLabel.setX(5.90);

        // レポートの処理日時を表示する式を追加
        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(IAlignConstants.ALIGN_RIGHT);
        runDate.setHeight(0.2);
        runDate.setWidth(2.5);
        runDate.setY(0.2);
        runDate.setX(5.0);

        // ページヘッダセクションにセルを追加
        report.getPageHeader().addData(userLabel);
        report.getPageHeader().addData(runDate);

        // 必要に応じてセクションの高さを指定
        if (report.getPageHeader().getHeight() < 0.4)
        report.getPageHeader().setHeight(0.4);

     } catch (Exception ex) {

        ex.printStackTrace();

     }

     return report;
   }
}
```

マップのデザイン

EspressReport ES（ERES）では、チャートとレポートに加え、マップも作成できます。具体的には、Google マップとSVG マップの 2 種類がサポートされています。Google マップは、対象となる地点の正確な位置を地図上に示すのに最適です。高品質の衛星画像で世界中を網羅し、柔軟なズームとクローズアップが可能なので、あらゆるニーズに対応できま

す。地理的な情報（都市名や郵便番号など）を含むレポートはすべて、Google マップを添えることで利便性が増すと言っても過言ではありません。一方、SVG マップは Google マップのように高度なズームには対応していませんが、地理的に、または距離や範囲でデータを簡単にわかりやすく比較するのに便利です。SVG マップ上でエリア別に色分けして、特定データ（人口、成長率、売り上げなど）の値の違いを視覚的に示すことができます。

Google マップ

Google が提供するオンライン地図です。そのため、ダッシュボードなどで表示するにはライセンスキーが必要になります。ERES にライセンスキーをインストールすれば、いつでも Google マップを表示できるようになります。マップで特定のデータポイントを示すには、経度と緯度の座標を指定しなければなりません。ERES で Google マップを使用する場合は、座標ファイルでこの情報を指定します。

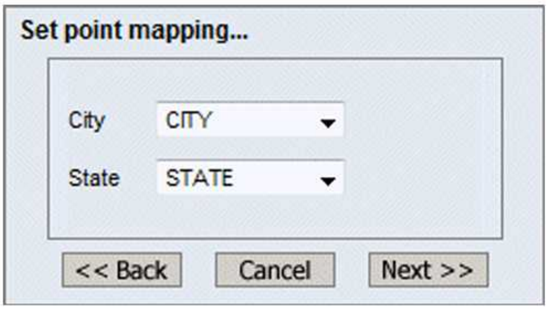
座標ファイル

座標ファイルを作成するには、まずデータソースを選択する必要があります。座標ファイルは、データソースから生成するか、ジオコーディングまたは手動で作成します。個々に経度と緯度の組み合わせを持つポイント ID のレコードでファイルが構成され、ポイント ID は単一フィールドと複数フィールドのどちらでも指定できます。ポイント ID は、たとえば以下のように指定します。

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

Google マップの設定方法

Google マップを作成するには、まずデータソースを選択する必要があります。ここで取得されるデータは、1 つ以上のフィールドが特定のポイント ID と一致するものでなければなりません。すなわち、データソースのフィールドを座標ファイルのポイント ID フィールドに割り当てるポイント マッピングをあらかじめ行っておく必要があります。



Google マップは、このフィールドにもとづいて、地図上にポイント ID の位置を表示します。たとえば、データソースのフィールドが座標ファイルの上記の例で示したフィールドに一致する場合、Google マップはイリノイ州シカゴの位置を経度 –87.624333、緯度 41.879535 だと認識して、それを地図上に表示します。

ツールチップでレポート／チャート表示

上記のとおり、Google マップはポイント マッピングにもとづいてデータソースの情報を地図上のどこに表示するかを判断しますが、その位置に関連するレポート情報を表示するには、ツールチップの設定が必要になります。

マップデザイナーのツールバーでツールチップ アイコンをクリックすると、クイックデザイナー（QuickDesigner）が起動します。

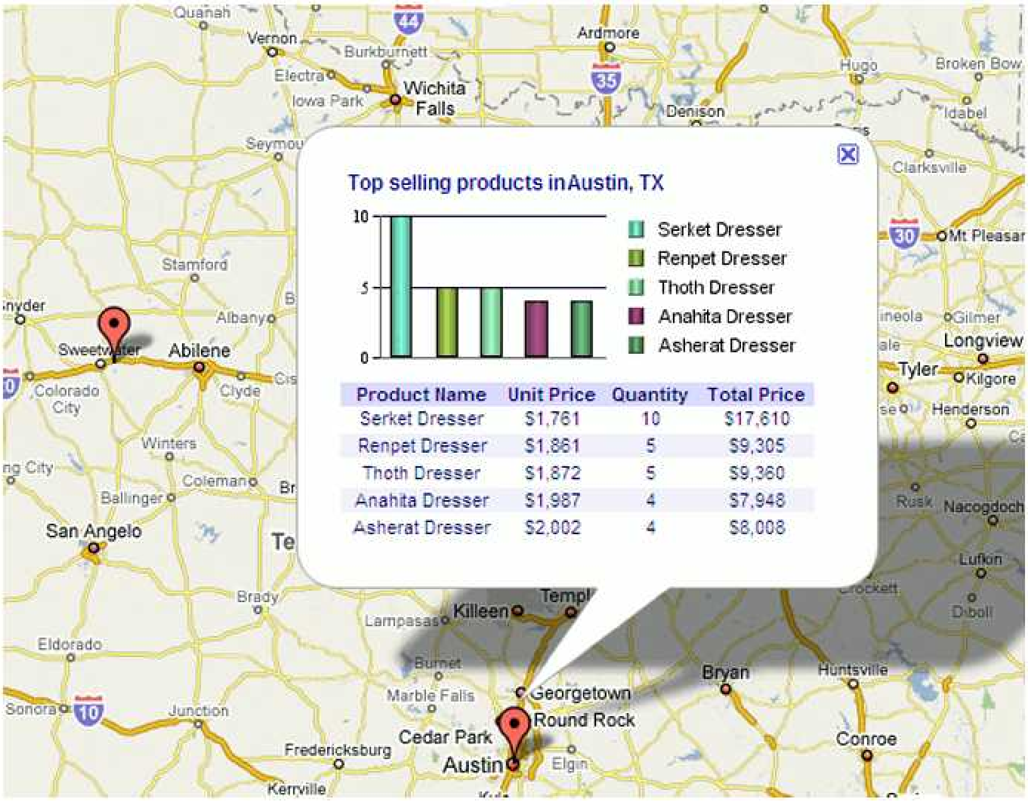
Data Source Results...

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 >>

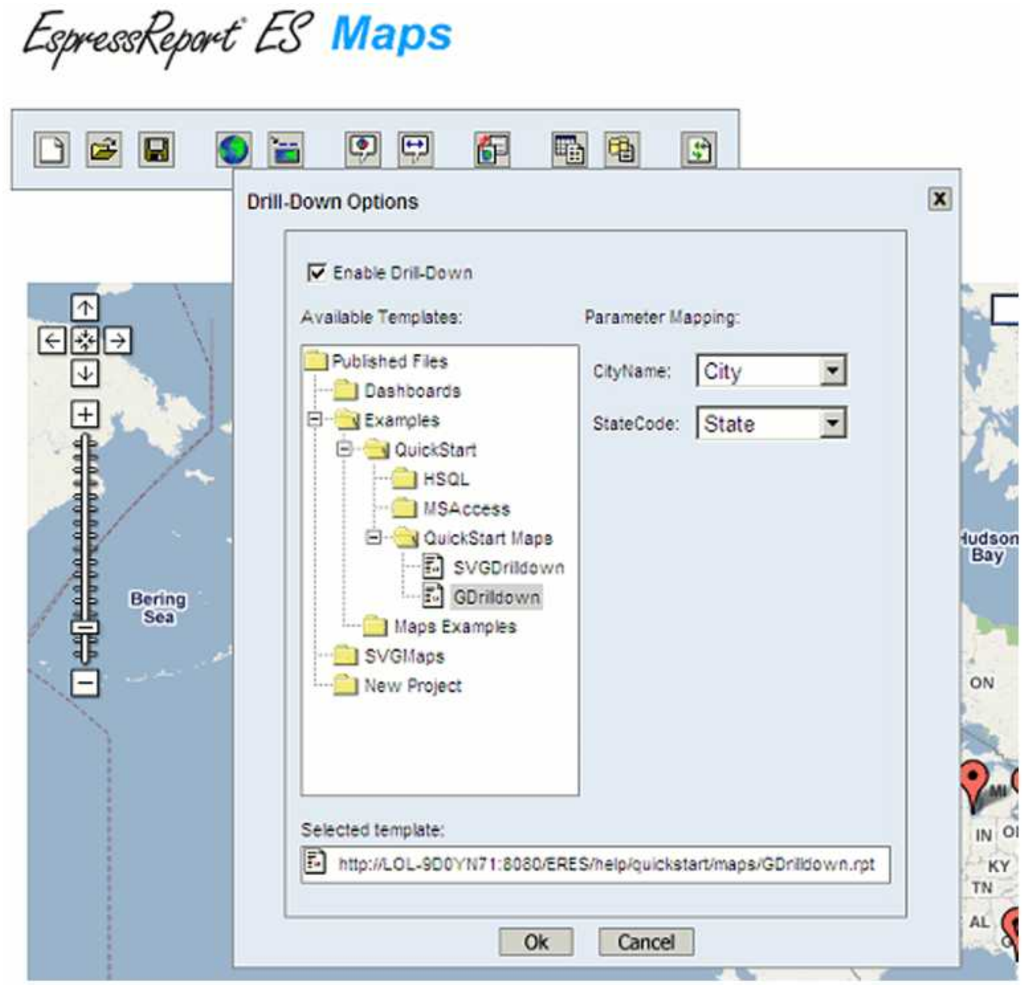
地図上でポイント ID の位置に表示されるバブルアイコンにマウス（カーソル）を合わせると、指定のレポートまたはチャート（チャートを含むレポートも可）が吹き出しで表示されます。



上記の例では、テキサス州オースティンのポイント ID に相当するデータソースに 5 件のレコードが含まれ、各レコードの 5 つのフィールド（Product Name、Unit Price、Quantity、Total Price）の値が吹き出しで表示されています。

ドリルダウン

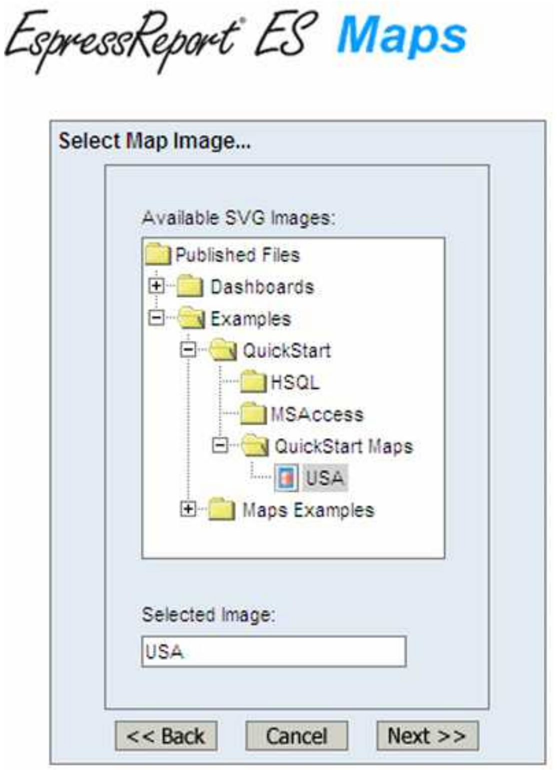
Google マップ上のツールチップで表示された情報は、さらに他の関連レポート、チャート、マップにドリルダウンする（詳細を掘り下げる）ことができます。



ツールバーにあるドリルダウン アイコンをクリックすると、ドリルダウン設定に利用可能なテンプレートがポップアップダイアログに表示され、必要なテンプレートを選択することができます。ここで選択するテンプレートには、マップのポイント ID に相当するパラメータが含まれていなければなりません。上記の例では、テンプレートで City と State のどちらか、または両方を指定する必要があります。

SVG マップ

Google マップの設定方法と同じく、ERES で SVG マップを使用するには、データソースを指定する必要があります。ただし、Google マップとは異なり、座標ファイルを指定する必要はなく、サードパーティのサポートに依存することもありません。最初にデータソースと SVG マップ イメージファイルをを選択して、SVG マップの設定を開始します。

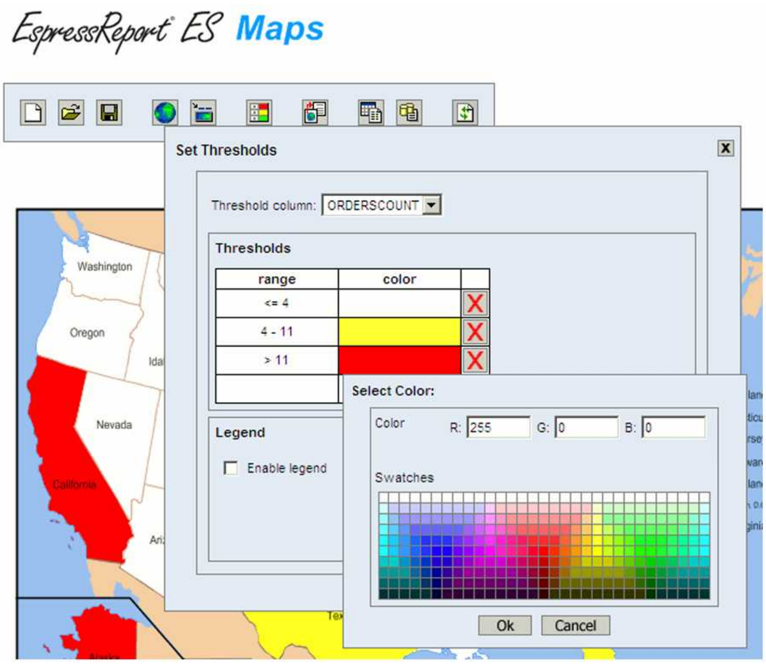


Map Images に指定する SVG イメージファイルは、地理的データを含むデータ構造になっています。これにより、SVG マップ上の特定地域を示すエリア ID が指定できます。SVG マップと同時に指定するデータソースにも、SVG マップのエリア ID に相当するデータフィールドが存在しなければなりません。データフィールドとエリア ID が一致することで、有効な地理情報の表示が可能となります。

基本的なコンセプトは、Google マップ設定時のポイント ID マッピングと同じです。

カラーコーディング

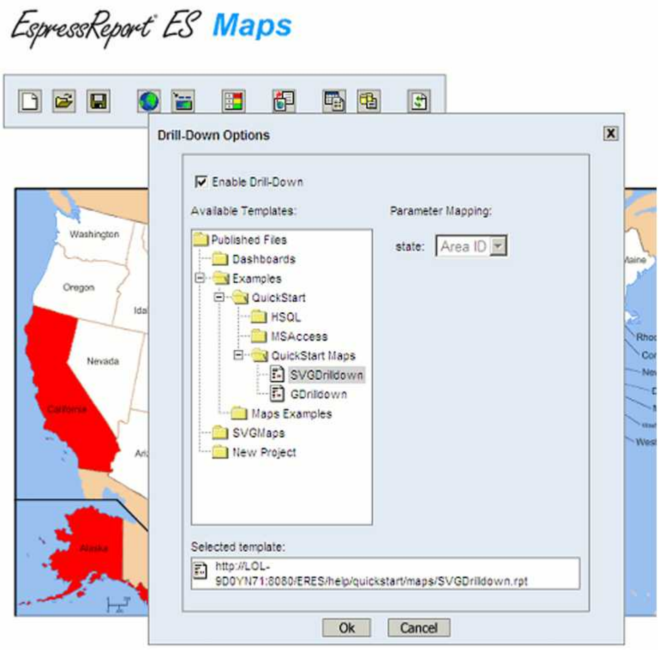
SVG マップでは、データソースの特定フィールドにおける値の範囲に色を指定することができ、データ値の地図上の分布状態を色別にわかりやすく示すことができます。



SVG マップには、カラーコーディングによる色分けの説明とマップタイトルを凡例として追加することもできます。

ドリルダウン

Google マップ同様、SVG マップでも表示された情報の掘り下げ（ドリルダウン）が可能で、パラメータ化されたチャート、レポート、マップのリストが表示されます。ただし、Google マップとは異なり、ドリルダウンは、エリア ID をチャート／レポート／マップのパラメータに割り当てることによってのみ設定可能です。



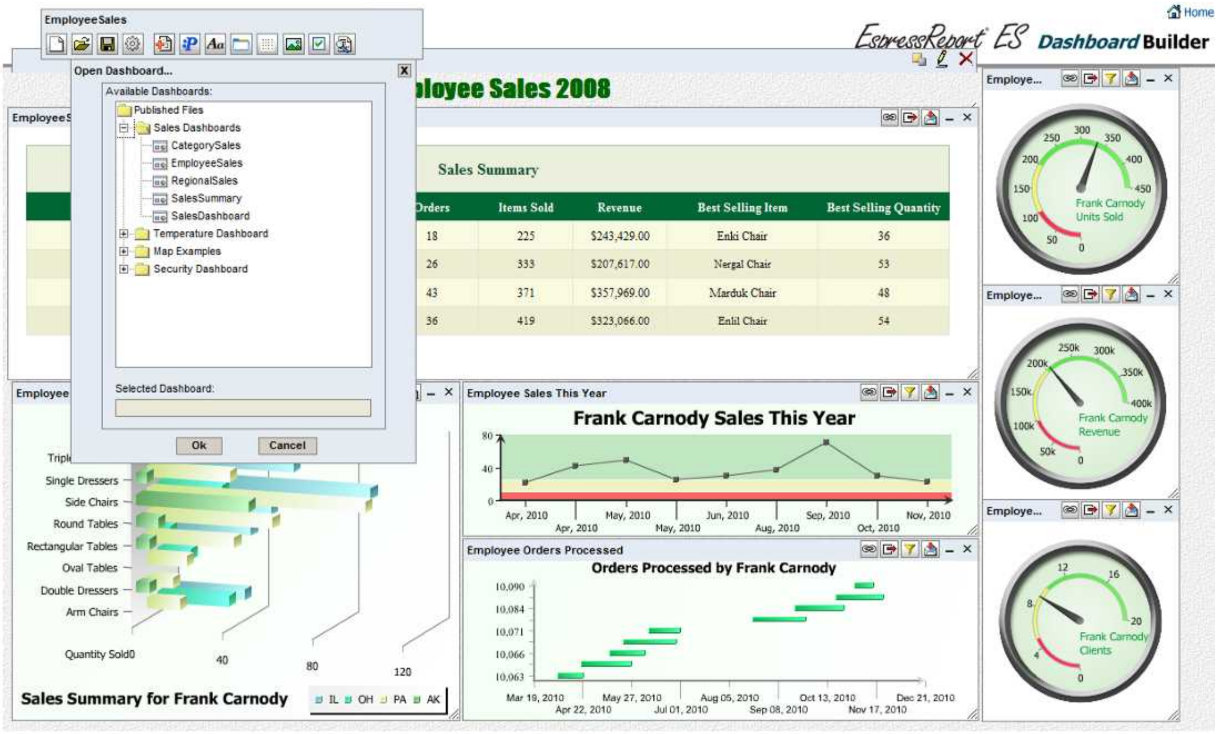
ダッシュボード

EspressReport ES（ERES）では、ダッシュボードを作成して公開することができます。ERES で作成したレポート、チャート、マップをダッシュボードの構成要素として自由に編集でき、シンクライアント型のダッシュボードビルダー（Dashboard Builder）インターフェースを通じてセルフサービスで高度なプレゼンテーションを構築できます。

ダッシュボード ビルダーは、多機能ながらもシンプルでユーザフレンドリーなシンクライアント型のデザインツールで、マウスで選択してクリックしたり、ドラッグ アンド ドロップしたりするだけで簡単に操作できます。他の多くのダッシュボード ツールとは異なり、スクリプティングやプログラミングは一切不要で、UI 操作のみでダッシュボード デザインが完結します。IT に疎いスタッフでも、あっと言う間に使いこなせるようになるので、トレーニングコストもかかりません。

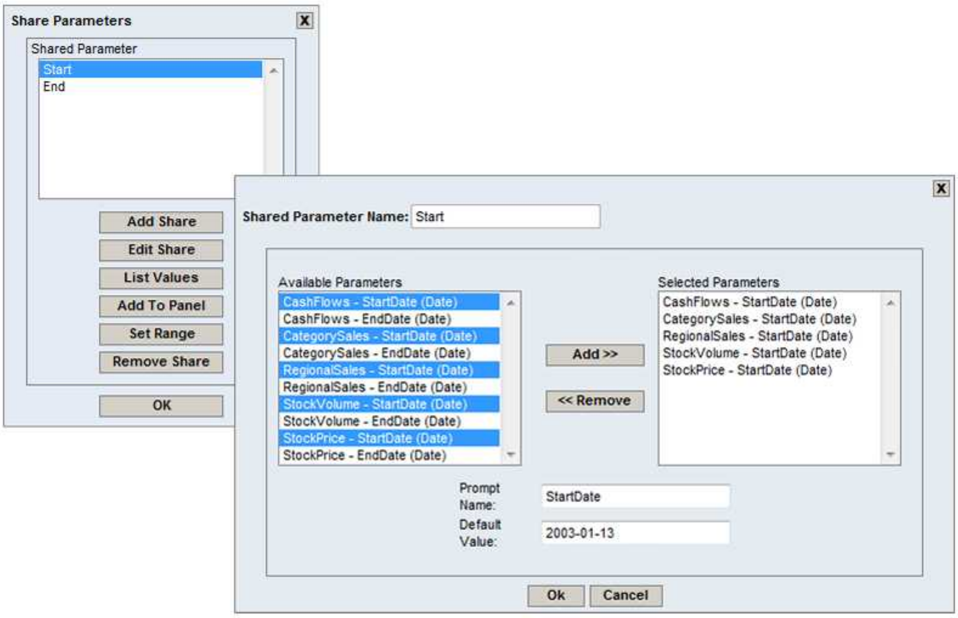
柔軟なレイアウト設定

ダッシュボードを構成する作業は、単にチャート、レポート、マップをオーガナイザー（Organizer）に追加していくだけです。それにより、各要素（レポート／チャート／マップ）をダッシュボード ビルダー（Dashboard Builder）で自由にレイアウトできるようになります。最初に、各要素が最適と想定される場所に自動的に配置されるので、ユーザは自由に位置を変えたり、必要に応じてサイズを調整したりできます。これには、ラベルや画像、パラメータも含まれ、ダッシュボードの品質を総合的に高め、プレゼンテーションとしてより説得力や訴求力のあるデザインを追求できます。



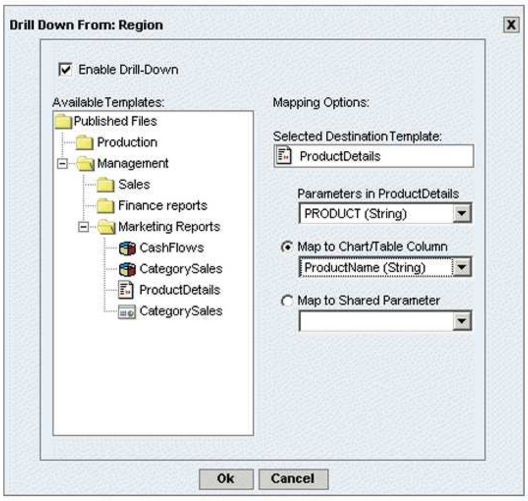
共有パラメータ

ダッシュボードに構成要素（レポート／チャート／マップ）を追加したら、各要素に共有パラメータを設定できます。これは、類似のパラメータを持つ要素間で値が同じパラメータがある場合に、それらを共有させて表示できる仕組みです。

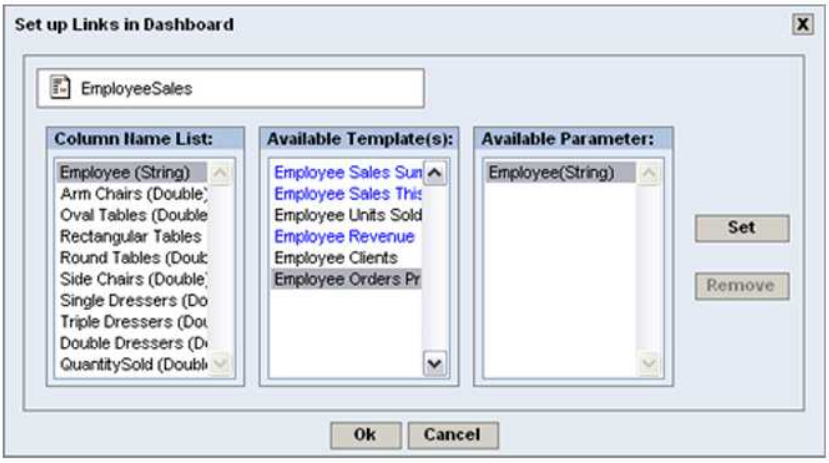


ドリルダウンの追加

ダッシュボード上の構成要素から関連レポート／チャートに詳細を掘り下げられるようにするには、単に Add/Modify Drill-Down（ドリルダウンの追加／修正）ボタンをクリックするだけです。ドリルダウン ダイアログが表示されるので、ドリルダウンするパラメータに合致するチャート、レポート、マップ、あるいは別のダッシュボードを選択します。パラメータの異なるチャート／レポート／マップを選択する場合は、それらに共有パラメータが割り当てられている必要があります。



Add/Modify Link（リンクの追加／修正）ボタンをクリックすると、ダッシュボード上の要素にリンクを付けることができます。ダイアログが開いて、リンクするカラム（フィールド）を選択でき、Available Templates（利用可能なテンプレート）パネルにパラメータ化されているレポート／チャートのリストが表示されるので、パラメータの一致する要素を選択してリンクさせます。



ダッシュボードの仕上げ

ダッシュボードのデザインは自由にカスタマイズできます。ラベルを追加したり、背景色を変えたり、背景イメージを追加したりなど、ダッシュボードの見た目を自由に調整できます。テキストのフォントサイズや色も選択でき、パラメータ パネルも同様にカスタマイズ可能です。

テクノロジーとアーキテクチャ

EspressReport ES（ERES）は Pure Java プロダクトであり、あらゆるプラットフォームで実行可能な柔軟性を備えています。このため、ユーザは既存のインフラストラクチャに（追加投資なしで）導入でき、J2EE アプリケーション環境への完全なデプロイメントが可能です。サーバサイド アプリケーション プロセスが追加で必要になることもありません。主要なアプリケーション サーバとデータベース エンジンは、ほぼすべてサポートされています。

EspressReport ES のコンポーネント

EspressReport ES（ERES）は、以下の主要コンポーネントから構成されています。

ERES Server ― ERES のバックエンド サーバです。アプリケーション サーバ内のサーブレット コレクションとして、あるいはサーブレット コンテナとしてデプロイされ、レポートやチャートの設定とデザインをつかさどります。また、ユーザ認証、スケジュール設定、アーカイブ作成を管理し、チャートデザイナー（Chart Designer）やレポートデザイナー（Report Designer）を実行するクライアントマシンへのファイル I/O とデータアクセスを提供します。

Organizer ―ERES の仮想ファイル管理システムとして機能する強力なグラフィカル ユーザ インターフェース（GUI）をオーガナイザー（Organizer）と呼びます。ユーザはこの GUI を通じてチャート、レポート、その他のファイルを仮想フォルダで管理でき、さらにチャート／レポートを表示するための URL 生成、チャート／レポートのスケジュールとアーカイブ、ファイルレベルの権限設定を行えます。データソースを管理してデータベース クエリを作成するためのインターフェースとしても機能します。

Report Designer ― オーガナイザー内で起動してレポートをデザイン／カスタマイズする GUI を、ERES ではレポートデザイナー（Chart Designer）と呼びます。ドラッグ アンド ドロップ式で誰でも簡単に操作できる GUI と広範な編集／フォーマット機能により、高度なレポートを迅速かつ簡単にデザインできます。

Chart Designer ― オーガナイザー内で起動してチャートをデザイン／カスタマイズする GUI を、ERES ではチャートデザイナー（Chart Designer）と呼びます。選択してクリックするだけのシンプルな GUI ですばやく簡単にチャートを作成でき、ほぼ無限の柔軟性でチャートをカスタマイズできます。

QuickDesigner ― ERES 内で起動するシンクライアント型のアドホック レポーティング ツールをクイックデザイナー（QuickDesigner）と呼びます。クエリを迅速かつ簡単に構築してシンプルなレポート／チャートを作成でき、簡易フォーマットオプションによってレポート／チャートをカスタマイズしてオーガナイザーに保存することができます。

Map Designer ― ERES 内で Google マップの作成／編集、座標ファイルの作成、SVG マップの設定を行えるシンクライアント型ツールをマップデザイナー（Map Designer）と呼びます。ERES Google マップに対しては、関連レポートを表示するツールチップやドリルダウンもカスタマイズできます。SVG マップもカラーコーディングによる地域の色分けやドリルダウンでカスタマイズできます。

Dashboard Builder — ERES 内のシンクライアント型ダッシュボード デザイン／作成ツールをダッシュボードビルダー（Dashboard Builder）と呼びます。テクノロジーに精通したユーザもそうでないユーザも、オーガナイザーに保存されたチャート、レポート、マップを使用して、簡単に会議資料やダッシュボードを作成／カスタマイズできます。

Report Viewer — レポートをウェブブラウザで動的に表示して操作可能にするアプレットを、レポートビューア（Report Viewer）と呼びます。レポートビューアによってレポートの各ページを自由にナビゲートして任意のページを表示したり、レポートをさまざまなフォーマットにエクスポートしたり、レポート上の特定の要素から詳細情報を掘り下げたり（ドリルダウン）できます。

Page Viewer — ページビューア（Page Viewer）は、レポートビューア同様、ERES に統合されたアプレットですが、リクエストに応じてレポートの任意のページだけをクライアントに送信できるページサービング テクノロジーを採用しています。膨大な分量のレポートを閲覧する際には欠かせないツールです。

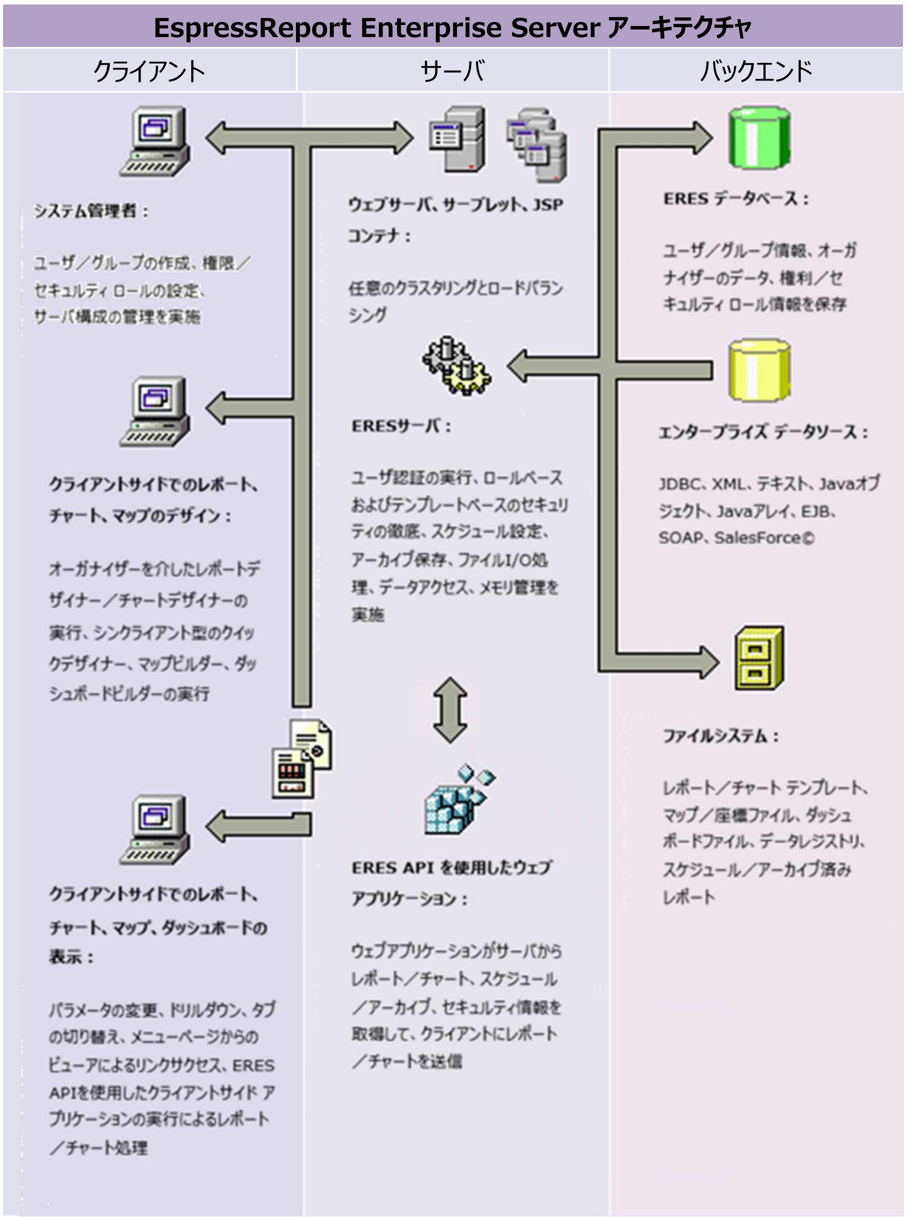
Chart Viewer — ERES のチャートをリモートで表示可能にするアプレットを、チャートビューア（Chart Viewer）と呼びます。画像を回転させたり、サイズを変更したり、拡大したり、パンしたりできます。特定のデータポイントをクリックすると、関連データを表示させたり、別のチャートにリンクしたりも可能です。ERES には、もう 2 つインタラクティブなチャート表示用に小型軽量アプレットがあり、それぞれ Mini Viewer と Micro Viewer と呼ばれています。

Report API —アプリケーション、アプレット、サーブレット、JSP でのレポートやチャートの作成、修正、実行を可能にする、ERES のシンプルで使いやすい Java アプリケーション プログラミング インターフェースです。ほとんどのチャートとレポートに対応しており、レポート／チャートをその場で簡単にカスタマイズできます。

ERES（Menu）API — ユーザが ERES サーバとオーガナイザーに組織的／体系的にアクセスできるようにするための Java アプリケーション プログラミング インターフェースです。この API によってユーザは、オーガナイザーでファイルや権限に関する情報を取得でき、レポートやチャートに対するラインタイム カスタマイゼーションを実行できます。スケジュールまたはアーカイブ済みのレポートにも対応可能です。

EspressReport ES のアーキテクチャ

EspressReport ES（ERES）は、ウェブアプリケーション環境へのデプロイメントを基盤とする設計になっています。以下の図は ERES デプロイメントの基本的なアーキテクチャを表したものです。



ERES サーバは、アプリケーション サーバ／サーブレットおよび JSP コンテナ内のサーブレット コレクションとしてデプロイされます。さらに、この ERES サーバに接続する別のサーブレット／JSP がサーバにデプロイされ、ERES サーバから情報（ファイル名、スケジュール情報、スケジュール済みレポート／チャートなど）を取得します。

ERES は、単一のサーバにも、クラスタ環境にもデプロイできます。クラスタで実行させる場合、1 つのサーバがロードバランサーの役割を担い、共通のバックエンドファイル構造を維持しながら、受信されたすべてのクライアント リクエストをクラスタの各メンバーにルーティングします。この構成は、数千単位のユーザに対応できる高可用性デプロイメントへのスケーリングを可能にします。

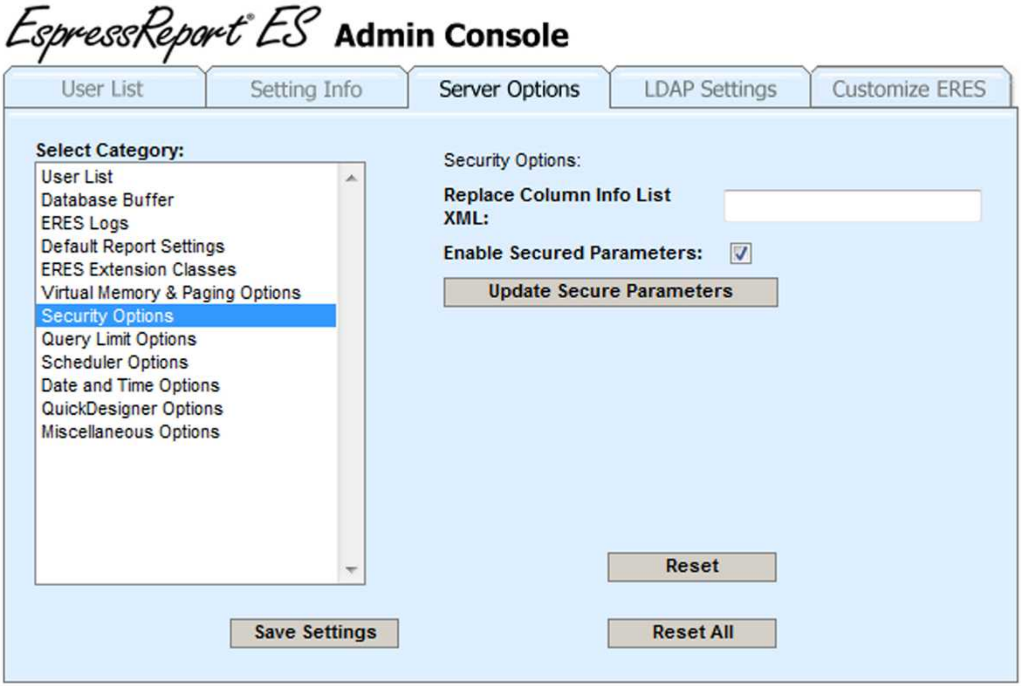
ERES バックエンドでは、ユーザ／グループ情報、オーガナイザーのファイルおよびその構造、セキュリティと権限に関する情報は、すべてデータベースに保存されます。データアクセスやレポート／チャートのバッファリング、サーバサイドのファイル I/O もサーバで実行されます。

ERES には、ユーザが独自のユーザ／グループやカスタム セキュリティ モデルを統合することもできます。LDAP システム内のユーザ／グループに直接統合でき、そのレコードがリレーショナル データベースに保存されます。ユーザが任意のカスタム セキュリティ モデルを ERES アーキテクチャにプログラムによって自動的に統合できるよう、オープンサーバ エクステンションも用意されています。

クライアントサイドでは、管理者がユーザ、権限、サーバ設定をリモートの管理コンソールまたはオーガナイザー（Organizer）から更新することができます。クライアントでは、オーガナイザーを介してレポートデザイナー（Report Designer）やチャートデザイナー（Chart Designer）を起動してレポートやチャートを作成でき、シンクライアント型のクイックデザイナー（QuickDesigner）を起動してアドホック レポーティングや分析も行えます。メニューまたはチャートを表示するクライアント システムでは、メニューへのアクセスはもちろん、ERES サーバの URL 呼び出し、他のアプリケーションに接続してのチャート、レポート、マップの呼び出しも実行できます。

管理とセキュリティ

ERES サーバおよびユーザベースの管理ファンクションは、ほぼすべてリモートで実行できます。管理者はシンクライアント型の管理コンソールを通じてユーザやグループを作成でき、サーバ機能を設定することができます。セキュリティと権限に関する各種設定はオーガナイザー インターフェースから適用できます。



EspressReport ES は、ロールベースで安全性の高いパラメータと要素単位のセキュリティ体制に加え、ユーザ／グループ権限を設定／管理できる強固なセキュリティ フレームワークを採用しています。レポートの開発者にはオーナーシップ モデルによる権限が適用され、レポートの処理が増加／変更するに連れてセキュリティが自動的に施行されます。これに加え、管理者は、従来的なファイル／フォルダ単位のセキュリティ フレームワークも施行できます。

セキュリティ機能は、権限の管理にとどまらず、レポートのテンプレートを通じても適用できます。レポートの構成要素をテンプレート設定で限定したり、レポートを閲覧するユーザやグループに応じてデータカラムが変更されるように設定したりすることができます。さらに、レポートへのフィルタ適用も自動設定でき、ロールベースのセキュリティが徹底されます。これにより、同じレポートやチャートが表示されれても、閲覧するユーザのセキュリティ クレデンシャルが異なれば、表示されるデータも異なるようになります。ユーザのロールにもとづいて、閲覧が承認されたものだけを閲覧できる仕組みです。これらに加え、LDAP や任意の外部セキュリティモデルとの統合もサポートされています。

グローバル デプロイメント

EspressReport ES は完全グローバル サポートで、世界中の国／言語で使用することができます。ロケール ベースのフォーマットにより、ユーザの地域／タイムゾーンに合わせてデータを正しく表示できます。レポート、チャート、マップ単位でも国際的な文字コードがサポートされており、UI やメッセージのテキストを翻訳するためのユーティリティ プログラムも用意されています。エクスポート形式は、自動的に UTF-8 文字コード（フォントマッピングを要する PDF を除く） で生成され、すべての言語に対応しています。

まとめ

EspressReport Enterprise Server®（ERES）は、エンタープライズ規模のレポート機能をシンプルで使いやすいユーザ インターフェースで利用可能にしたレポーティング ソリューションです。ERES を使用すれば、定型レポートの作成はもちろん、レポートの公開スケジュール設定やアーカイブ保存、アドホック レポーティングやクエリの作成など、すべてコーディングなしで実行できます。しかも、これらの操作は簡単で覚えやすく、特別なトレーニングも必要ありません。

ERES のアーキテクチャは完全な Pure Java 設計のため、あらゆるプラットフォームで実行可能なうえ、アプリケーション サーバ環境にデプロイできるので、確立されたシステム基盤のうえでスケーラビリティを実現できます。一般的に、BI アプリケーションは複雑なものが多く、レポートツールは特定アプリケーションに縛られがちですが、ERES の高度な機能、多様なパブリッシング オプションと圧倒的な使いやすさは、他の BI ツールでは決して味わえない利便性を提供します。

EspressReport ES の詳細については、**www.climb.co.jp/soft/** をご参照ください。

Copyright © 2024 Quadbase Systems, Inc. All rights reserved. EspressReport ES, Quadbase Systems, Inc., and the Quadbase logo are trademarks of Quadbase Systems, Inc. Java, JDK, AWT, Swing, and JDBC are trademarks of Oracle Corp. in the United States and other countries. All other trademarks belong to their respective owners. 翻訳©株式会社クライム