

ComponentOne®
Studio™ Enterprise
Framework

 An Integrated Approach to .NET Development



Optimized for

 Microsoft
Visual Studio .net

ComponentOne® Studio™ Enterprise Framework

 An Integrated Approach to .NET Development

Abstract

ComponentOne Studio Enterprise is a comprehensive suite of data and user interface components for Microsoft Visual Studio .NET or any .NET-compatible development environment.

This paper describes the rationale behind the design of the ComponentOne Studio Enterprise framework. In addition to providing an overview of the four application layers (data, presentation, reporting, and transformation), it explains the benefits of the architecture as it relates to enterprise-wide development for .NET platforms.

Contents

Architectural Overview	1
Data Layer	3
Defining Reusable Schemas	4
Simple and Composite Tables	5
Datasets and Table Views	6
Business Logic and Validation Rules	7
Presentation Layer	8
Reporting Layer	11
Transformation Layer	14
Application Layer Interactions	15
Conclusion	16
ComponentOne Studio Enterprise Resources	17



Optimized for



Microsoft
Visual Studio .NET



ComponentOne Studio Enterprise Framework



www.componentone.com
www.componentone.co.uk

Architectural Overview

Among .NET component vendors, only ComponentOne offers a complete solution that targets all runtime environments (WinForms, WebForms, and mobile devices) and all application layers (data, presentation, reporting, and transformation). The cornerstone of ComponentOne Studio Enterprise is the data layer, which encapsulates business logic in reusable assemblies called **data libraries**.

APPLICATION LAYER	DESCRIPTION
DATA	The data layer provides a comprehensive data and business objects framework that serves .NET applications of any range, scalability, and architecture: <ul style="list-style-type: none">• Desktop applications• Client-server applications• 3-tier distributed applications• Enterprise-wide business object libraries
PRESENTATION	The presentation layer consists of a suite of best-of-breed user interface controls (such as grids, charts, and input editors) that interoperate seamlessly with the data layer, combining powerful data manipulation capabilities with unmatched flexibility in GUI design.
REPORTING	The ComponentOne Report Designer is a standalone application for creating XML-based report definition files at design time. At run time, reports are rendered in a customizable preview control (WinForms) or as HTML (WebForms).
TRANSFORMATION	The transformation layer extracts, converts, loads, and saves data from a variety of heterogeneous sources, including popular file formats such as XML, PDF, and ZIP. It is used internally by the presentation and reporting layers, and is surfaced in standalone components such as C1Zip and C1Pdf.

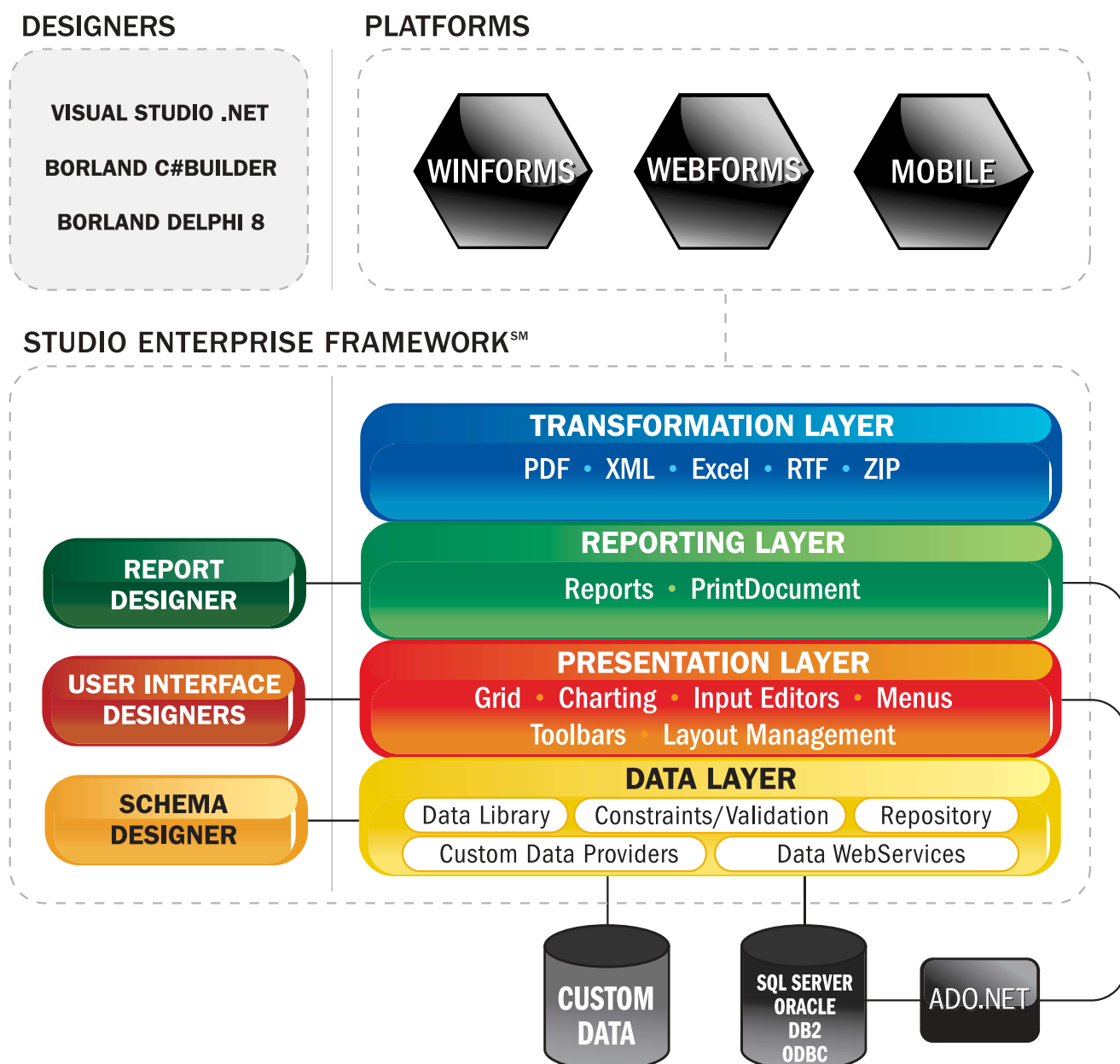
ComponentOne Studio Enterprise is not simply a collection of unrelated controls, but a unified suite of components that share a common framework. This architecture offers several significant advantages for enterprise-wide application development:

- ➔ **Completeness:** The framework spans all facets of an application, from low-level data access and business rules to end-user interaction and reporting.
- ➔ **Interoperability:** The layers work in concert to add unique functionality, making the whole greater than the sum of its parts. For example, multi-row presentation controls are optimized to work with the virtual mode feature of the data layer.
- ➔ **Granularity:** The choice of components is left to the developer. For example, presentation and reporting components need not talk to the data layer, but can interact with standard ADO.NET adapters. Likewise, the data layer can be used with intrinsic or third-party presentation controls and report writers.

Architectural Overview

- ➔ **Reusability:** The data library concept promotes the reuse of data schemas and business rules across applications and across platforms.
- ➔ **Programmability:** Rich object models complement easy-to-use designers, making it possible to create a complete report definition or data schema entirely in code.
- ➔ **Consistency:** Framework components use similar object models and documentation conventions, reducing the learning curve.

The following diagram shows how the application layers interact with each other and with external data sources.



ComponentOne Studio Enterprise Framework



Data Layer

Based on Microsoft ADO.NET technology, the data layer simplifies and enhances database application development in many significant ways:

- It promotes code-free data access through drag-and-drop designers and bindable components representing datasets, tables, and views.
- It is platform-neutral. Data libraries can be consumed by both WinForms and WebForms applications, without modification.
- It supports huge databases through a unique virtual mode technology that fetches data in segments while providing performance tuning options and control over background operations.
- It isolates the physical database structure from the logical structure needed for business objects and user interface design.
- It supports composite tables representing joins between two or more physical tables. Composite tables are exposed to data consumers as a simple table, and the data layer automatically handles updating and deleting of the physical rows in the underlying tables.
- It centralizes business logic by providing a repository for data validation rules, calculated fields, constraints, remote methods, and other domain-specific tasks.
- It automates server-side caching for WebForms applications at the global and session levels, and also supports memory-resident table views for optimized access to read-only data.
- It provides a rich object model and programmable business logic components for customizing the behavior of fetch, edit, update, insert, and delete operations.
- It generates type-safe classes that map to the tables and views exposed in a data library assembly.
- It saves development time and effort by making it possible (and easy) to maintain a single data library and deploy it in any application configuration.

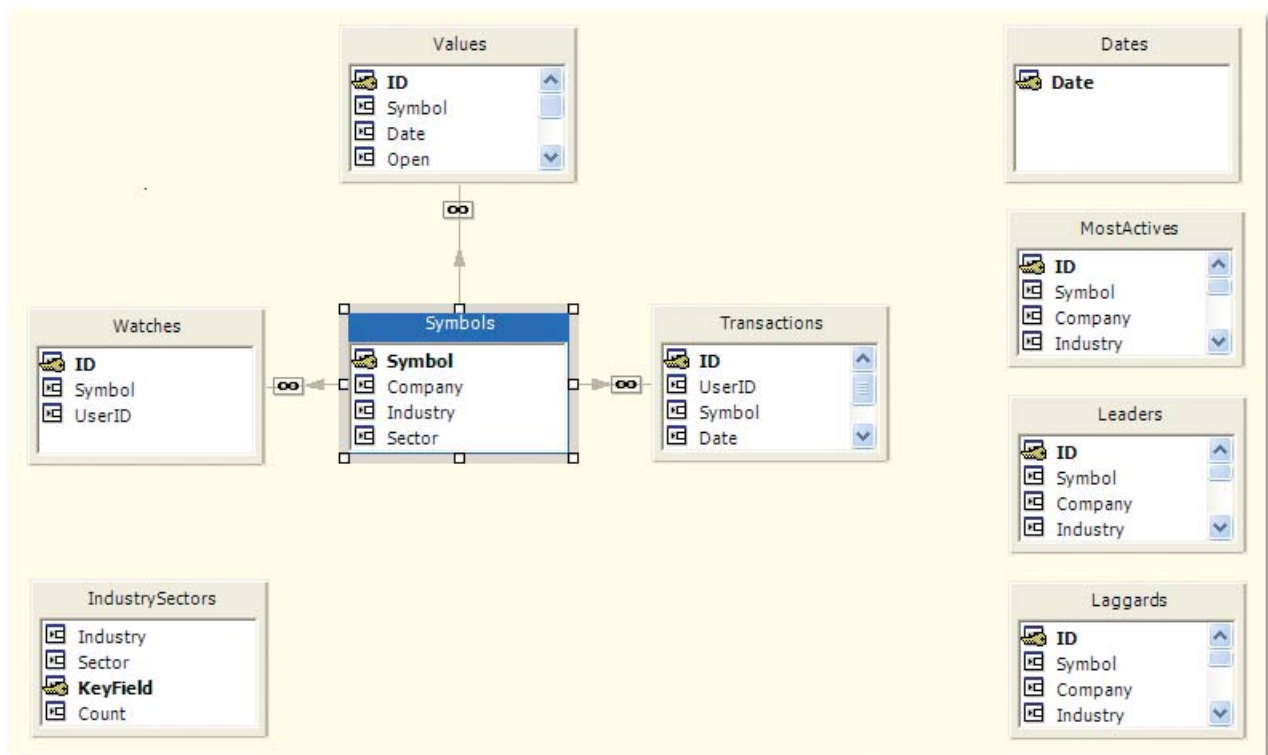
Data Layer

Defining Reusable Schemas

Defining Reusable Schemas

For enterprise development, the ComponentOne data layer makes it easy to develop a central repository of data schemas and business logic that can be reused in applications throughout the enterprise.

A schema is the root element of the data layer. It defines basic entities, such as tables and relations, along with their properties. A visual tool, the ComponentOne DataObjects Schema Designer, is used to create schema definitions. Typically, a developer creates a new schema by importing a database structure using the Schema Designer's Import Wizard. The Schema Designer provides a graphical representation of database tables and the relationships between them, as shown in the following diagram.



A data library project contains a single C1SchemaDef component that provides access to the Schema Designer. Client applications consume the data schema by adding a reference to the data library DLL.



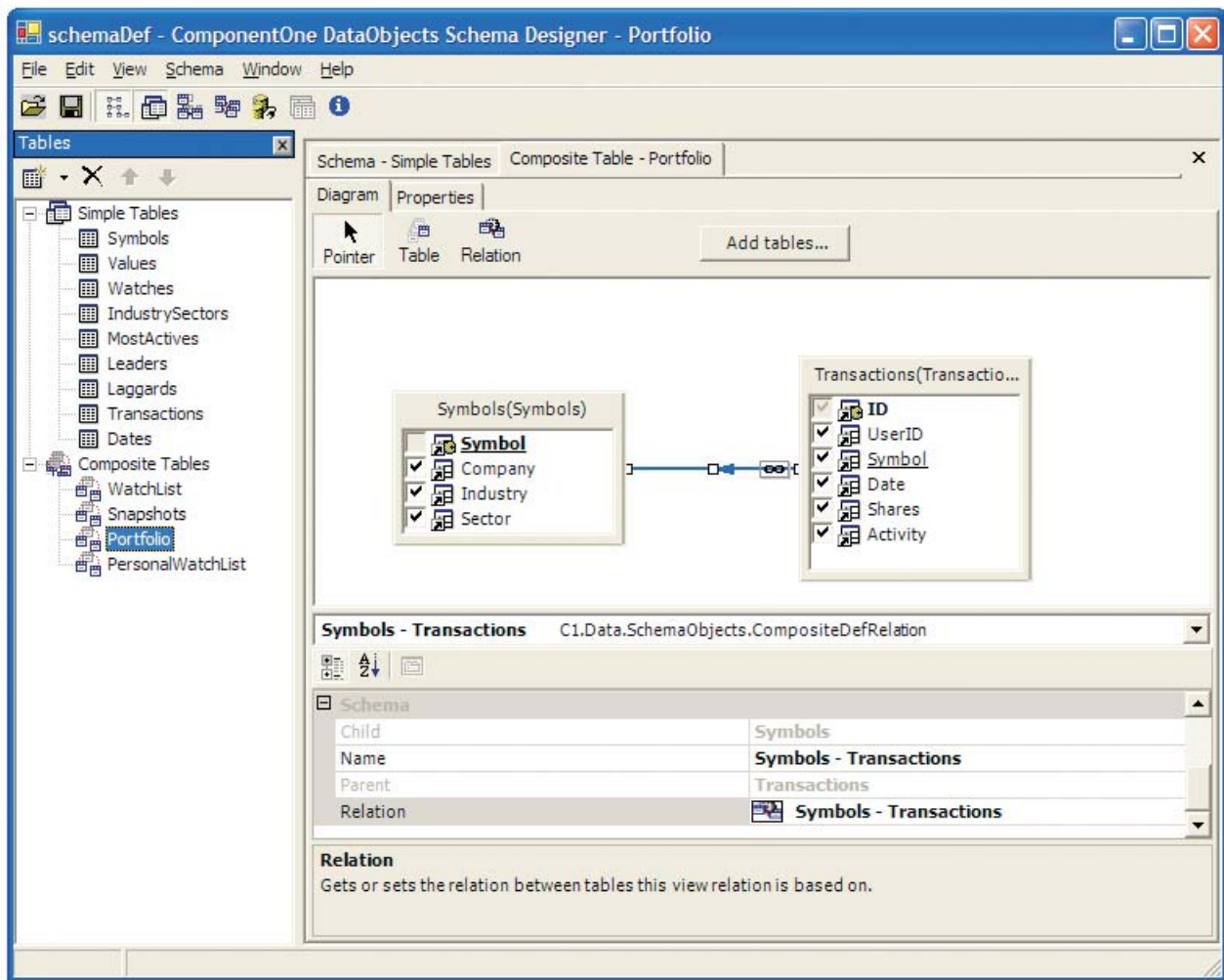
Simple and Composite Tables

Simple tables are the basic, elementary data objects in a schema. A simple table usually corresponds to a physical database table (or view). Simple tables need not mirror the structure of the underlying database table. Fields can be omitted, renamed, or reordered.

Simple tables can also be based on an SQL statement or stored procedure, either by setting the `SelectCommandText` property in the Schema Designer or by setting the SQL statement in code.

An unbound table is also considered to be a simple table. It is filled with data entirely in code, which makes it ideal for implementing custom data providers.

A **composite table** is an aggregate of two or more simple tables that is exposed to clients as a single table. A composite table row represents a row of each constituent simple table. In the Schema Designer, a composite table is defined by a diagram of simple table nodes connected by lines representing relations, as shown in the following figure.



Data Layer

📁 Datasets and Table Views

Composite table rows do not actually store simple table data; they store pointers to simple table rows instead.

Composite table support is one of the most powerful features of the data layer. By allowing developers to define business objects with data stored in multiple database tables, composite tables bridge the gap between the physical database structure and the logical structure needed for business logic and user interface design.

Composite tables represent business objects just as simple tables do. Developers can associate business logic code with composite tables, as with simple tables, using the `C1TableLogic` component events. Each composite table has a class associated with it, with properties corresponding to composite table fields.

Datasets and Table Views

A dataset is a collection of table views, optionally connected with view relations. Each table view represents a specific table, either simple or composite. There can be multiple table views representing a single table in a dataset. Typically, a table view represents a table as is, without modifying business logic and properties, and with table view fields corresponding exactly to the table fields. In this case, creating a dataset definition in the Schema Designer is a simple matter of selecting the desired tables. However, in more advanced cases developers can use the ability of a table view to override and modify the table's business logic and properties, and customize table view fields.

A dataset also serves as a transaction context. That is, modifications made in separate datasets belong to separate transactions; they are not visible across datasets. A schema can contain multiple dataset definitions.

A `C1DataSet` component represents a dataset defined in the Schema Designer. It contains the data as it is exposed to the user, and can be used by data bound controls and manipulated programmatically. By default, a `C1DataSet` component is filled with data at startup, when the data is requested. This feature can be disabled, and the dataset can be filled explicitly in code by calling its `Fill` method.

`C1DataSet` components can be assigned to the `DataSource` property of bound controls. Once this assignment is made, the `DataMember` property displays a list of the available table views.

When filling a `C1DataSet` component, developers can specify filter conditions in code, either as an argument to the `Fill` method or in the `BeforeFill` event handler. Filter conditions must be specified separately for each table view. Restricting one table view does not automatically restrict any other.



Business Logic and Validation Rules

Simple tables, composite tables, and table views can all be augmented with business logic. The simplest form of business logic is a calculated field, which evaluates an expression involving one or more database fields. For example, the expression for a calculated field called `ExtendedPrice` could be defined as:

$$\text{UnitPrice} * \text{Quantity} * (1 - \text{Discount})$$

Another common business logic construct is a constraint expression, which consists of one or more Boolean conditions that must be satisfied before an update can occur. For example, to prevent the end user from entering non-positive values for a particular field, its constraint expression can be specified as:

$$\text{UnitPrice} > 0$$

Constraints can also be assigned an error message string to be displayed by bound controls when the user attempts to enter a value that does not satisfy the expression.

For cases where expressions are inadequate to express business rules, developers can write event handlers for the `C1TableLogic` and `C1DataSetLogic` components. These events fire for a number of scenarios, including field changes, row updates, and dataset fetches, to name a few. The following example shows how the `BeforeFieldChange` event can be used to modify or disallow new field values:

```
private void table_Order_Details_BeforeFieldChange(
    object sender, C1.Data.FieldChangeEventArgs e)
{
    if (e.Field.Name == "Quantity")
    {
        if ((short)e.NewValue < 1)
            e.NewValue = (short)1;
    }
    else if (e.Field.Name == "Discount")
    {
        if ((float)e.NewValue < 0 || (float)e.NewValue > 1)
            throw new ApplicationException("Discount must be between 0 and 1");
    }
}
```

If the user enters a non-positive value for `Quantity`, this event handler changes it to 1, and the update proceeds. If the user enters an illegal value for `Discount`, an exception is thrown, and the change is disallowed.

Presentation Layer

The presentation layer consists of a set of award-winning visual controls that intermediate between the end user and the underlying data source. The following table lists the data bound controls in the presentation layer, by platform:

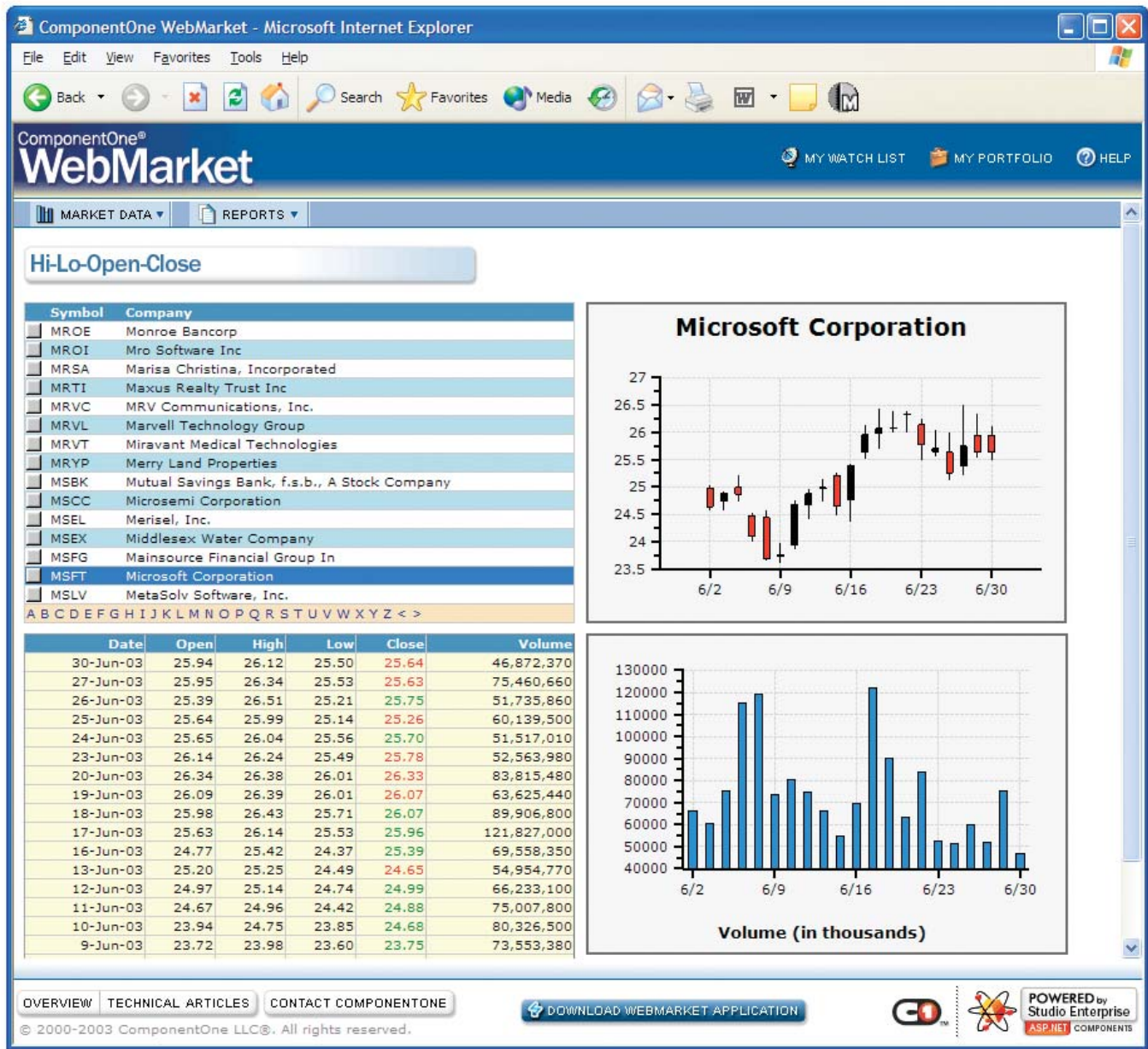
PRESENTATION LAYER COMPONENTS	
WINFORMS	<ul style="list-style-type: none"> • C1FlexGrid: A lightweight yet powerful grid control with support for outlining, subtotals, and hierarchical tree views. • C1TrueDBGrid: A heavyweight grid control with advanced data presentation and user interface features. • C1Chart: Two-dimensional chart control used to render bar, line, area, pie, radar, polar, and candle charts. • C1Chart3D: Three-dimensional chart control used to render surface, contour, scatter, and bar charts. • C1TextBox: Derived from System.Windows.Forms.TextBox, this control supports data formatting, edit masks, data validation, and other features. • C1NumericEdit: Derived from C1TextBox, this control specializes in numeric values and supports a dropdown calculator as well as a spin button. • C1DateEdit: Derived from C1TextBox, this control specializes in date values and supports a dropdown calendar as well as a spin button. • C1Label: Derived from System.Windows.Forms.Label, this control displays read-only formatted data. • C1PictureBox: Derived from System.Windows.Forms.PictureBox, this control displays picture images stored in a data source field. • C1DbNavigator: Toolbar control used for database navigation as well as adding, updating, and deleting rows. • C1List: List box control patterned after the object model of C1TrueDBGrid. • C1Combo: Combo box control patterned after the object model of C1TrueDBGrid.
WEBFORMS	<ul style="list-style-type: none"> • C1WebGrid: Similar to the Microsoft DataGrid, this control adds enhanced display, grouping, sorting, and user interaction functionality. • C1WebChart: An ASP.NET wrapper for the C1Chart control. • C1WebChart3D: An ASP.NET wrapper for the C1Chart3D control.



Presentation Layer

9

The following screenshot shows a WebForms application with two C1WebGrid and two C1WebChart controls. The master grid (top) is used to select a single stock symbol, which in turn populates the detail records in the bottom grid as well as the chart controls.

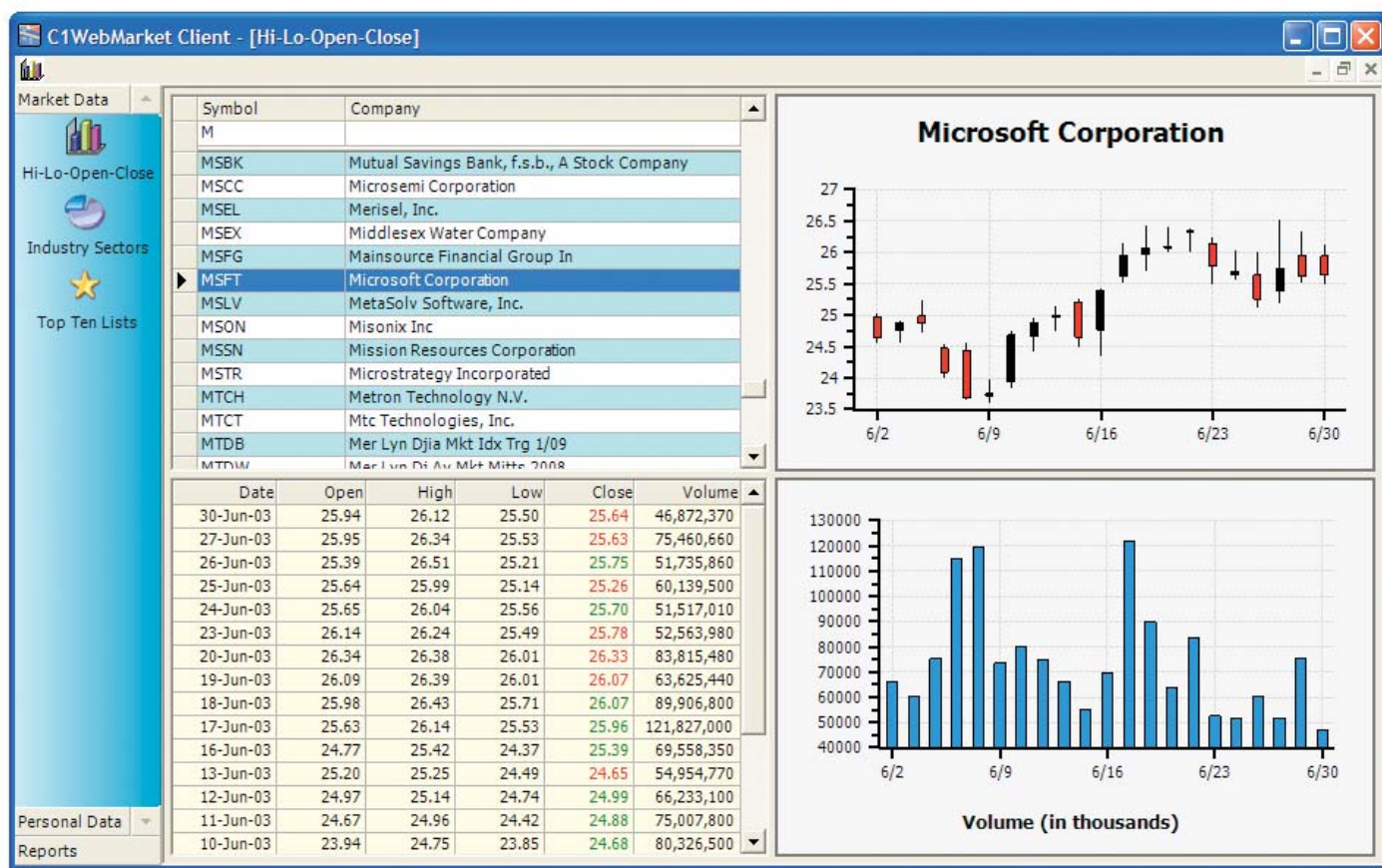


Note the alphabetic links at the bottom of the master grid. Since it would be impractical to display all master records at once, the application uses a custom pager to allow the user to select symbols beginning with a specific letter. When the user clicks one of these links, the appropriate filter condition is passed to the underlying dataset.

Presentation Layer

The next screenshot shows the same application written for the WinForms platform using C1TrueDBGrid instead of C1WebGrid, and C1Chart instead of C1WebChart. Note that the appearance is virtually identical except for the master grid. Since the WinForms platform does not have the same limitations as WebForms, this application uses the virtual mode feature of the data layer. When the form loads, the master grid is populated with a small subset of the available rows and is responsive to end user scrolling and navigation, even though only a portion of the data has been fetched.

Without the ComponentOne data layer, the grid would not load at all until the entire dataset was fetched.



Not only are the presentation layer components visually similar, but both WinForms and WebForms applications share the same data library assembly. This ensures that data validation rules are enforced consistently in all client applications, regardless of platform.



Reporting Layer

The reporting layer consists of a standalone application, the ComponentOne Report Designer, and the following components, listed by platform:

REPORTING LAYER COMPONENTS	
WINFORMS	<ul style="list-style-type: none">• C1Report: Hosts report definition files created with the Report Designer.• C1PrintPreview: Renders reports on the screen and provides commands for navigation, panning, zooming, and printing.• C1PrintDocument: Document creation engine that supports styles, tables, page setup, and rendering methods.
WEBFORMS	<ul style="list-style-type: none">• C1WebReport: An ASP.NET wrapper for the C1Report control.

The ComponentOne reporting layer offers a number of advantages:

- It imports existing report definitions from both Crystal Reports and Microsoft Access.
- It integrates natively with the ComponentOne data layer.
- It is platform-neutral. Report definitions can be consumed by both WinForms and WebForms applications.
- It makes deployment easier by allowing report definitions to be embedded within applications.
- It includes a powerful object model (C1PrintDocument) that can be used to programmatically render free-form reports.
- Source code for the Report Designer is also available.

Typically, a developer uses the Report Designer application to create report definitions, which are saved as XML files. Report definitions can be created from scratch or imported from Microsoft Access or Crystal Reports. Alternatively, report definitions can be created in code, either by using the object model to add groups and fields or by writing a custom XML file.

The C1Report component is used to host report definitions, which can be loaded at either design time or run time. Reports loaded at design time are embedded within the application, in which case the report definition file need not be distributed.

The C1Report component can render reports directly to a printer, into a C1PrintPreview control, or as HTML and PDF files that can be published on the Web. For ASP.NET applications, the C1WebReport control hosts the C1Report component and streams HTML or PDF reports to the client machine.

Reporting Layer

The following screenshot shows a WebForms application that uses C1WebReport to render a report as paged HTML with a basic navigation bar. A command button is also provided for viewing a PDF version of the same report.

ComponentOne WebMarket - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Mail Print W M

ComponentOne® WebMarket MY WATCH LIST MY PORTFOLIO HELP

MARKET DATA REPORTS

Top Ten Report

<< < Page 1 of 21 > >>

Monday, June 30, 2003

symbol	company	close	change	% change	volume
Most Actives					
INTC	Intel Corporation	20.81	-0.33	-1.56%	49,082,500
MSFT	Microsoft Corporation	25.64	-0.30	-1.16%	46,872,370
CSCO	Cisco Systems, Inc.	16.79	-0.29	-1.70%	46,851,170
SIRI	Sirius Satellite Radio, Inc.	1.69	-0.04	-2.31%	37,800,540
ORCL	Oracle Corporation	12.01	-0.48	-3.84%	33,799,350
WCOEQ	Worldcom GP	0.04	0.00	-5.00%	27,287,900
SUNW	Sun Microsystems, Inc.	4.65	-0.14	-2.92%	27,225,580
NXTL	Nextel Communications, Inc.	18.07	-0.81	-4.29%	26,240,060
SINA	Sina.com	20.25	2.46	13.83%	22,939,620
CMGI	CMGI, Inc.	1.51	0.13	9.42%	22,907,420
Leaders					
BREK	Brek Energy Corp	0.15	0.14	1400.00%	5,200
PMIC	Pacific Magtron Intl Corp	0.15	0.05	50.00%	2,200
CARD	PubliCARD, Inc.	0.08	0.02	41.82%	18,600
DCTHW	Delcath Systems Inc Wts 10-05	0.28	0.07	33.33%	34,700
BIGCE	Bridge Technology, Inc.	0.44	0.09	25.71%	3,600
DIRI	Direct Insite Corporation	1.85	0.37	25.00%	12,708
TRNS	Transmaton, Inc.	1.61	0.31	23.85%	3,400
BLSI	Boston Life Sciences	1.85	0.34	22.52%	403,938

OVERVIEW TECHNICAL ARTICLES CONTACT COMPONENTONE

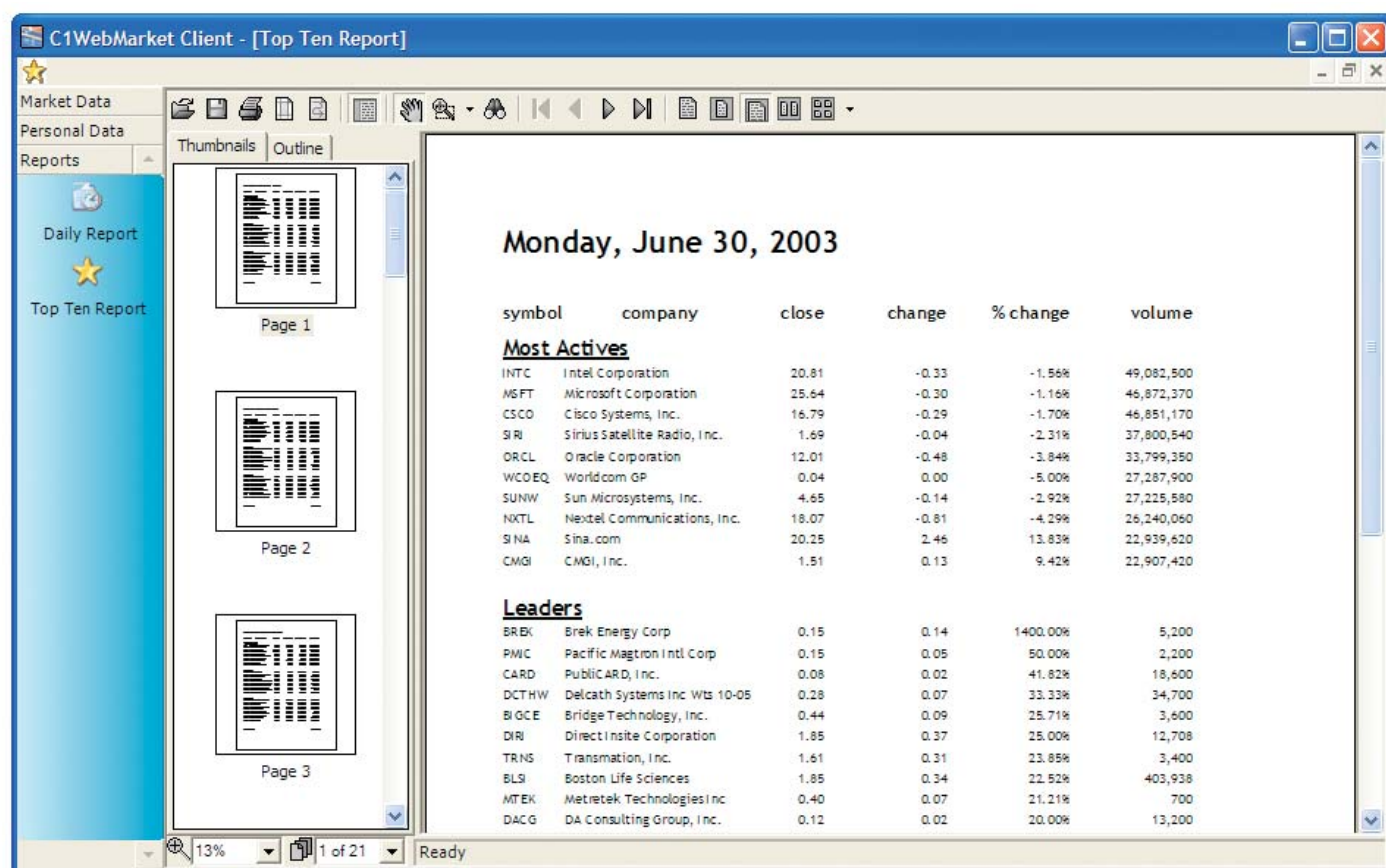
DOWNLOAD WEBMARKET APPLICATION

© 2000-2003 ComponentOne LLC®. All rights reserved.

POWERED by Studio Enterprise ASP.NET COMPONENTS



The next screenshot shows the same application written for the WinForms platform using C1Report and C1PrintPreview instead of C1WebReport. Note that the appearance of the report itself is identical, but C1PrintPreview provides a much richer navigation model, including thumbnail and outline views.



As an alternative to Microsoft Access-style banded reports, the C1PrintDocument component provides a rich object-oriented framework for advanced document formatting and processing. Its features include multi-column pages, rich text formatting options, tables (including nested tables), automatic column and page breaks, and PDF export. Instead of rendering a report definition created with the Report Designer, C1PrintDocument exposes an object model for rendering content in code, either by drawing objects directly at specific positions on a page, or in the block flow, where column and page breaks are handled automatically when there is not enough space to accommodate the drawing object.

C1PrintDocument integrates seamlessly with C1PrintPreview, and also recognizes C1-printable controls. For example, multi-page controls such as C1TrueDBGrid and C1FlexGrid use the C1PrintDocument component to print their contents.

Transformation Layer

Transformation Layer

The transformation layer extracts, converts, loads, and saves data from a variety of heterogeneous sources, including popular file formats such as XML, PDF, and ZIP. It is used internally by the presentation and reporting layers, and is surfaced in standalone components such as C1Zip and C1Pdf.

TRANSFORMATION LAYER COMPONENTS	
WINFORMS	<ul style="list-style-type: none"> • C1Pdf: Creates PDF documents, with support for security, compression, outlining, hyperlinks, and attachments. • C1Zip: Creates, opens, and manages ZIP files. Also supports data compression with .NET streams.
WEBFORMS	<ul style="list-style-type: none"> • Same as WinForms.

C1Pdf follows the style of the .NET Graphics class, which makes it instantly familiar. For example, the following code creates a document with a one-inch margin containing a single sentence:

```
// create the C1PdfDocument object
C1.C1Pdf.C1PdfDocument pdf = new C1.C1Pdf.C1PdfDocument();

// add content to the page
RectangleF rc = pdf.PageRectangle;
rc.Inflate(-72, -72);
Font font = new Font("Arial", 12);
pdf.DrawString("Hello World!", font, Brushes.Black, rc);

// save the document to a file
pdf.Save(@"c:\temp\hello world.pdf");
```

C1Zip provides three levels of data compression classes. At the highest level, developers can create and open ZIP files, test their integrity, and add, delete, and extract files. At the medium level, data can be compressed into or expanded from regular .NET streams, including memory, file, and network streams. At the lowest level, the ZStream class provides a 100% C# implementation of Zlib, the popular data compression library. ZStream is used by the higher level classes in C1Zip.



Application Layer Interaction

The following tables summarize how the four application layers interact with one another:

DATA LAYER INTERACTIONS	
PRESENTATION	<ul style="list-style-type: none">• Schema definitions and field properties are exposed to bound controls.• Calculated fields, filter conditions, and business logic events are evaluated.• Constraints and validation rules are evaluated during updates.• Virtual mode is available to bound controls.
REPORTING	<ul style="list-style-type: none">• Schema definitions and field properties are exposed to the Report Designer.• Calculated fields, filter conditions, and business logic events are evaluated.
TRANSFORMATION	<ul style="list-style-type: none">• Schema definitions can be loaded from or saved to XML files.

PRESENTATION LAYER INTERACTIONS	
DATA	<ul style="list-style-type: none">• Bound controls respond to data validation errors and constraint exceptions raised in the data library.• Multi-row controls are optimized to work with virtual mode datasets.
REPORTING	<ul style="list-style-type: none">• Grid controls provide print preview functionality.• C1-printable controls expose the entire dataset, not just the visible region.
TRANSFORMATION	<ul style="list-style-type: none">• Grid controls can export their contents to HTML and PDF.

REPORTING LAYER INTERACTIONS	
DATA	<ul style="list-style-type: none">• The Report Designer can import data schemas from C1DataSet and C1DataView components on the same form.
PRESENTATION	<ul style="list-style-type: none">• C1PrintDocument provides intelligent rendering of C1-printable controls.
TRANSFORMATION	<ul style="list-style-type: none">• Reports can export their contents to HTML, PDF, and Excel.

TRANSFORMATION LAYER INTERACTIONS	
DATA	<ul style="list-style-type: none">• Schema definitions can be loaded from or saved to XML files.
PRESENTATION	<ul style="list-style-type: none">• Grid controls can export their contents to HTML and PDF.
REPORTING	<ul style="list-style-type: none">• Reports can export their contents to HTML, PDF, and Excel.

Conclusion

Conclusion

ComponentOne Studio Enterprise is an indispensable set of tools for serious application development. It combines rich object models and unparalleled functionality with easy-to-use visual designers and components. Its data, presentation, reporting, and transformation layers are tightly integrated, yet flexible enough to be coupled with Microsoft data access components or third-party controls.

What makes ComponentOne Studio Enterprise unique is that it is not just a set of presentation layer components or reporting tools, but a complete package that spans all application layers. Data access is the foundation of any real-world application, and the ComponentOne data layer has no peer.

With ComponentOne Studio Enterprise, developers can add core functionality, improve performance, and build more extensible applications for any .NET platform.



Optimized for



Microsoft
Visual Studio .net



OVER 80 AWARDS



ComponentOne Studio Enterprise Framework



www.componentone.com
www.componentone.co.uk

C1WebMarket WinForms and WebForms Applications

The C1WebMarket applications demonstrate the versatility of ComponentOne Studio Enterprise and provide excellent examples of how these components can be used to produce functionally equivalent applications in both WinForms and WebForms environments. Visit the demonstration site at <http://c1webmarket.componentone.com> to view ComponentOne data, grid, chart, reporting, and user interface components in action, and discover how the ComponentOne Studio Enterprise Framework delivers a truly integrated approach to .NET development.

Key C1WebMarket Application concepts include:

- Using ComponentOne WebDataObjects for ASP.NET to implement a centralized data schema with business logic.
- Using ComponentOne WebGrid for ASP.NET to render tabular data in a variety of formats with client-side scripting for end-user interaction.
- Using ComponentOne WebChart for ASP.NET to present visually appealing charts and graphs of numeric data.
- Using ComponentOne WebReports for ASP.NET to display formatted reports in a customizable preview control or as PDF.

Key C1WebMarket Client Application concepts include:

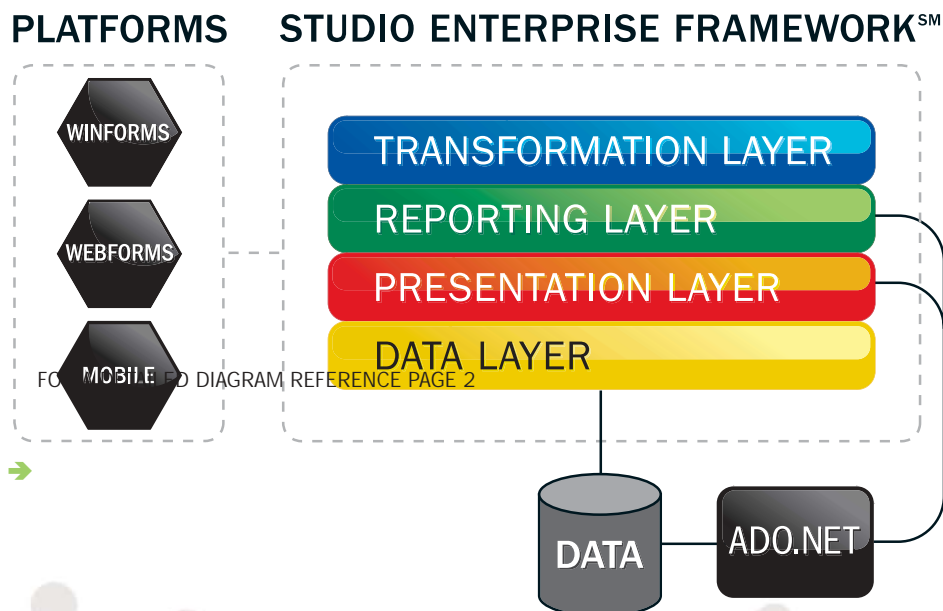
- Using ComponentOne DataObjects for .NET in a 3-tier distributed application.
- Using ComponentOne True DBGrid for .NET in conjunction with the virtual mode of DataObjects for .NET.
- Using ComponentOne Chart for .NET to present visually appealing charts and graphs of numeric data.
- Using ComponentOne Reports for .NET to display formatted reports with pan, zoom, outline, and thumbnail capabilities.

The WinForms client retrieves and updates data using a Web service that uses the same database as the ASP.NET application. Moreover, it uses the same data library assembly as the ASP.NET application. Data libraries are reusable assemblies created with ComponentOne DataObjects for .NET. They encapsulate all aspects of a data schema, including partitioning into logical datasets, composite tables, business rules, calculated fields, constraints, and custom methods.

In addition to the online demo, which contains context-sensitive help topics with code samples, developers can also download the full source code for both applications (requires the .NET Framework 1.1 to install). Visit <http://c1webmarket.componentone.com> and click the Download WebMarket Application button centered at the bottom of the page.

■ ComponentOne Studio Enterprise Framework

This unique framework, the backbone of the Studio Enterprise subscription service, targets all runtime environments and all application layers. The cornerstone of ComponentOne Studio Enterprise is the data layer, which encapsulates business logic in reusable assemblies called data libraries. Data access is the foundation of any real-world application, and the one-of-a-kind ComponentOne data layer has no peer.



ComponentOne
4516 Henry Street
Suite 500
Pittsburgh, PA 15213 USA

E-mail: info@componentone.com
Sales: 1.800.858.2739
Phone: 1.412.681.4343
Fax: 1.412.681.4384

ComponentOne Europe
Knapp House, Howley
Chard, Somerset
TA20 3DU UK

info@componentone.co.uk
0800 328 5271
E-mail: +44 (0)1460 234636
Sales: +44 (0)1460 234616
Phone:
Fax:

