BUILD WEB 2.0 APPLICATIONS WITHOUT HAND-CODING

USE APPLICATION GENERATION TO EXPEDITE WEB APPLICATION DEVELOPMENT

"Every minute spent on infrastructure programming is a wasted minute."

-Juval Lowy, .NET Software Legend

Increasingly, developers, CTOs, IT business analysts are turning to a new, breakthrough approach for rapidly developing robust Web applications: Application Generation.

Iron Speed Designer builds database and reporting applications for the Microsoft .NET and software-as-a-service cloud environments – without hand-coding. Simply point to an existing database and let Iron Speed Designer generate a visually stunning, feature-rich Web 2.0 application that is easy to customize and ready to deploy.

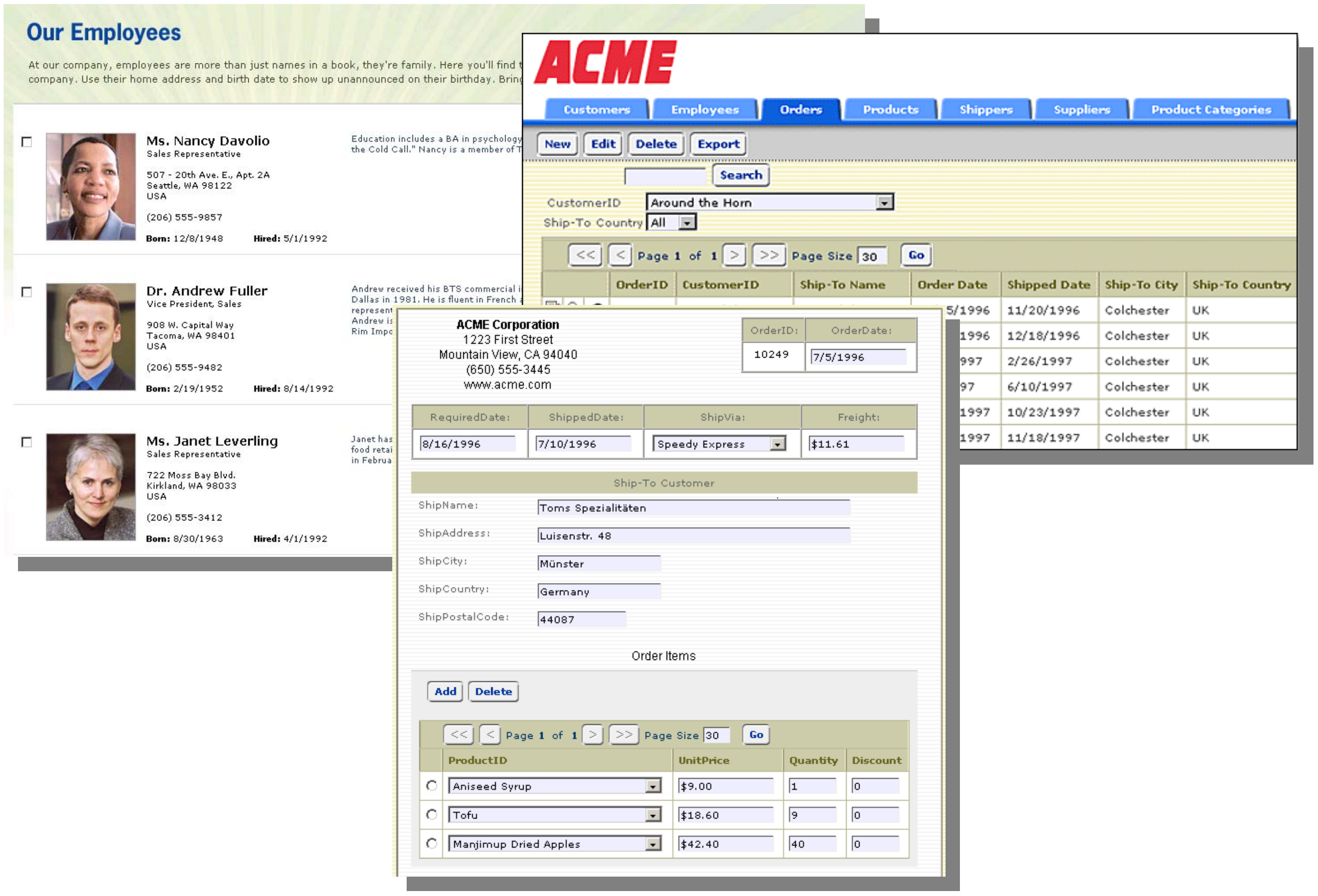
White Paper
February 2010

Iron Speed, Inc.
2870 Zanker Road, Suite 210
San Jose, CA 95134
http://www.ironspeed.com

Table of Contents

Build Web 2.0 Applications without Hand-Coding

Developing Web 2.0 applications provides an opportunity to bring desktop functionality to the Web by building visually attractive, user-friendly applications like these:



Our Employees

At our company, employees are more than just names in a book, they're family. Here you'll find t
company. Use their home address and birth date to show up unannounced on their birthday. Bring

Ms. Nancy Davolio
Sales Representative
507 - 20th Ave. E., Apt. 2A
Seattle, WA 98122
USA
(206) 555-9857
Born: 12/8/1948 **Hired:** 5/1/1992

Education includes a BA in psychology
the Cold Call." Nancy is a member of T

Dr. Andrew Fuller
Vice President, Sales
908 W. Capital Way
Tacoma, WA 98401
USA
(206) 555-9482
Born: 2/19/1952 **Hired:** 8/14/1992

Andrew received his BTS commercial i
Dallas in 1981. He is fluent in French a
represent
Andrew is
Rim Impo

Ms. Janet Leverling
Sales Representative
722 Moss Bay Blvd.
Kirkland, WA 98033
USA
(206) 555-3412
Born: 8/30/1963 **Hired:** 4/1/1992

Janet has
food retai
in Februa

CustomerID: Around the Horn
Ship-To Country: All

<< < Page 1 of 1 > >> Page Size 30

OrderID	CustomerID	Ship-To Name	Order Date	Shipped Date	Ship-To City	Ship-To Country
			5/1996	11/20/1996	Colchester	UK
			1996	12/18/1996	Colchester	UK
			997	2/26/1997	Colchester	UK
			97	6/10/1997	Colchester	UK
			1997	10/23/1997	Colchester	UK
			1997	11/18/1997	Colchester	UK

ACME Corporation
1223 First Street
Mountain View, CA 94040
(650) 555-3445
www.acme.com

OrderID:	OrderDate:
10249	7/5/1996

RequiredDate:	ShippedDate:	ShipVia:	Freight:
8/16/1996	7/10/1996	Speedy Express	$11.61

Ship-To Customer

ShipName:	Toms Spezialitäten
ShipAddress:	Luisenstr. 48
ShipCity:	Münster
ShipCountry:	Germany
ShipPostalCode:	44087

Order Items

<< < Page 1 of 1 > >> Page Size 30

ProductID	UnitPrice	Quantity	Discount
Aniseed Syrup	$9.00	1	0
Tofu	$18.60	9	0
Manjimup Dried Apples	$42.40	40	0

Web 2.0 applications look better than traditional desktop applications, and they're perceived as much easier to use. Web 2.0 applications let you engage with users like never before, and many organizations are enhancing Web applications to integrate customers and suppliers, in addition to internal users.

At the same time, building Web applications put you, the developer, in a completely different paradigm. Web 2.0 applications add additional development burdens of application security, stateless transaction management, and page-based navigation, as well as a variety of user interface features you wouldn't normally worry about in traditional desktop

An application generation tool should generate:

- Enterprise-class 3-tier architecture
- User interface web pages, including sophisticated data grids with advanced features like multi-table joins, reports, and filters.
- User interface code and database bindings
- SQL statements & stored procedures
- Transaction management code
- Workflow reporting
- End user account management and access control
- Multiple layers of application security
- Scalable applications
- Easily extensible class hierarchies

client-server applications. Even though platforms like the Microsoft .NET Framework deliver the basic OS-level services for these applications, you still must create hundreds of Web pages, code your application's business logic and write SQL statements.

IT departments are efficiently conquering many of these critical issues and freeing up resources using a new breed of software tools called 'application generators' to eliminate hand-coding. Quickly delivering applications positions them as 'heroes' in their organization.

Application generation takes a giant leap forward, offering a broader, more comprehensive solution, and some tools such as Iron Speed Designer, create your entire application for you without hand-coding.

Accelerating the development cycle without sacrificing performance, features and interoperability, are key benefits provided by application generation.

This white paper describes how one application generator, Iron Speed Designer, builds applications without hand-coding. It combines point-and-shoot application generation with Web 2.0 functionality, speeding development and reducing costs.

Business Benefits of Application Generation

Why is application generation so important and useful for modern application development?

- **Reduce software development costs**. Creating applications in a fraction of the time means you can develop and deploy applications faster and more efficiently with less cost in human resources. Your clients will be amazed at how quickly you can build proof-of-concept systems, gather feedback, and deploy applications.

  At Iron Speed, our customers report saving an average of $37,000 per application. The savings on just the first application more than pays for the cost of the tool.

- **Impress customers and clients**. Using application generation, you can quickly create prototypes and gather user feedback. This allows you to be very responsive to internal customers or impress external clients with your development capabilities.

  Moreover, generated applications have highly consistent and professionally designed user interfaces, giving your applications a finished look and feel, even though they may be prototypes or works-in-progress. Users can better visualize the finished application and will confer a degree of perceived quality typically not present in traditional hand-coding development until the end of the project.

- **Automate business processes**. Custom workflow processes that streamline your business can be quickly designed, built and deployed. Workflow design is often an iterative process, taking several cycles to ascertain the business process and then to reorganize and streamline it. Application generation lets you quickly refine screens and workflows, engaging application users in a more iterative development process.

- **Focus on higher priority tasks**. Eliminating large amounts of hand-coding lets you focus on other high-priority tasks and projects. While every organization has different priorities, why spend time hand-coding if you don't have to?

Development Team Benefits of Application Generation

- **Speed application development.** Application generation saves development time, letting you deliver applications more quickly. Iron Speed's customers report saving an average of 19 weeks per application.

  Most application development projects start with a handful of page mock-ups and a database schema. Iron Speed Designer generates a fully working application using your database schema as input, skipping much or all of the mock-up phase of the project.
- **Simplify application maintenance**. Generated applications follow a highly consistent architecture, allowing any developer maintain any application. There is little or no 'ramp up' time necessary for one developer to maintain another developer's application because the architectural knowledge transfers from one application to another.
- **Reduce testing time**. High-quality generated code is very stable, reducing your QA burden. You take advantage of the testing already applied to the application generation tool, plus the additional testing from thousands of other developers using the tool.
- **Maximize your IT department productivity**. Most developers dread building features such as paginated reports, hierarchical navigation, filtering, and full text search, because the amount of time required is disproportional to the value they create. Application users expect them, but they usually require considerable coding. Iron Speed Designer frees your team to focus on higher-level process design that supports your specific business needs.
- **Eliminate mind-numbing work.** A Web page may contain five or ten separate SQL queries, each of which has to be written, debugged, and vetted for performance. However, by using application generation tools, you don't have to worry about manually programming pages, controls, or SQL. They generate most, if not all, of this repetitive code for you.

The Iron Speed Designer Approach

Imagine creating a complete application just by pointing a wizard at the databases underlying your application-to-be? All the Web pages and components – tables, panels, controls and navigation – are automatically created and connected, and the underlying databases and associated code is generated for you. That's exactly what Iron Speed Designer does, illustrating the power of application generation.

Iron Speed Designer's approach is very straightforward:

1. Start with your existing database or other data asset.

2. Automatically create a set of fully functional Web pages for common database operations, such as adding, editing and viewing data.

3. Automatically create a set of fully functional reports and process workflows.

4. Automatically generate a working N-tier application with "pure" source code in native .NET languages, such as C# and Visual Basic, without any proprietary run-time servers or libraries.

5. Generate built-in application security using industry standard protocols.

6. Customize the generated application using means that enable non-traditional developers – DBAs and 'para-techs' – to build and customize Web applications.

What Types of Web 2.0 Applications can be Generated?

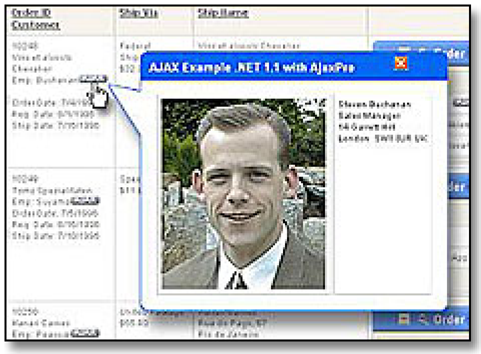
From a functional perspective, most Web 2.0 applications are database-driven, meaning they are used for:

- Data entry and management – collecting and editing data from users.
- Reporting and tracking – reporting, summarizing and visualizing data.
- Workflow and scheduling – basic step-wise business processes.
- Business process automation – orchestrating data flow between multiple systems.

If these application types sound like client server applications from the 1980's and 1990's, you're right. Modern Web 2.0 applications couple a rich internet user interface with a back-end server for centralized data storage and processing.

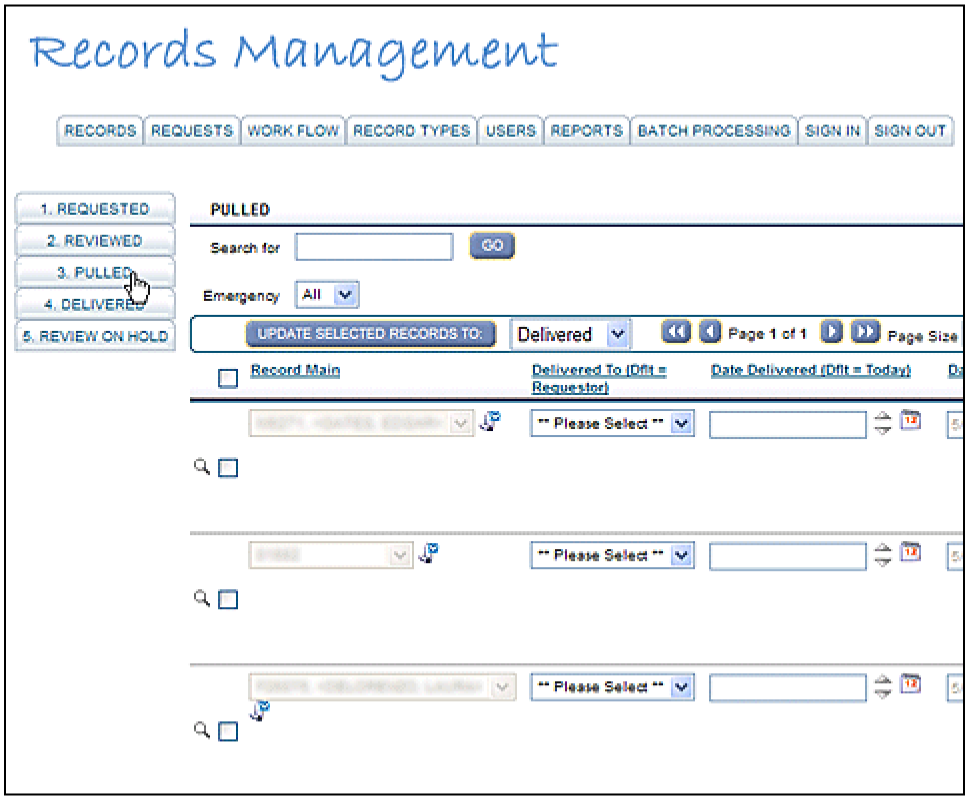
There is no precise definition of a Web 2.0 application. For our purposes, we'll describe them as applications that bring desktop application functionality to Web applications via rich transaction-centric user interfaces.

Ajax-enabled controls like pop-ups and rich text editors are examples of functionality frequently found in Web 2.0 applications.



How are these rich user interfaces implemented? Largely by using new technologies such as Ajax (Asynchronous Java and XML) and ASPX for the user interface, and web services and ADO.NET for data source access.

This Web 2.0 workflow application lets users process transactions from a single Web page.



What are some specific characteristics of Web 2.0 applications?

- Rich components are a hallmark, frequently 'mashed up' from a variety of sources to create a rich, comprehensive application. These components may be hosted on a variety of servers and sites, effecting a truly distributed application.

- Individual Web page controls interact intensively with server-side code. Interactive search, calendar and scheduling controls, and live data feeds, all interact with server-side code in real-time to provide users with dynamically updated information.

- They don't postback, or at least appear to postback. Postbacks refresh the page causing annoying flicker. Web 2.0 applications use Ajax and other techniquest to provide a 'smooth panel update' of the affected screen areas.

Fortunately Web 2.0 applications are ideally suited to an application generation approach.

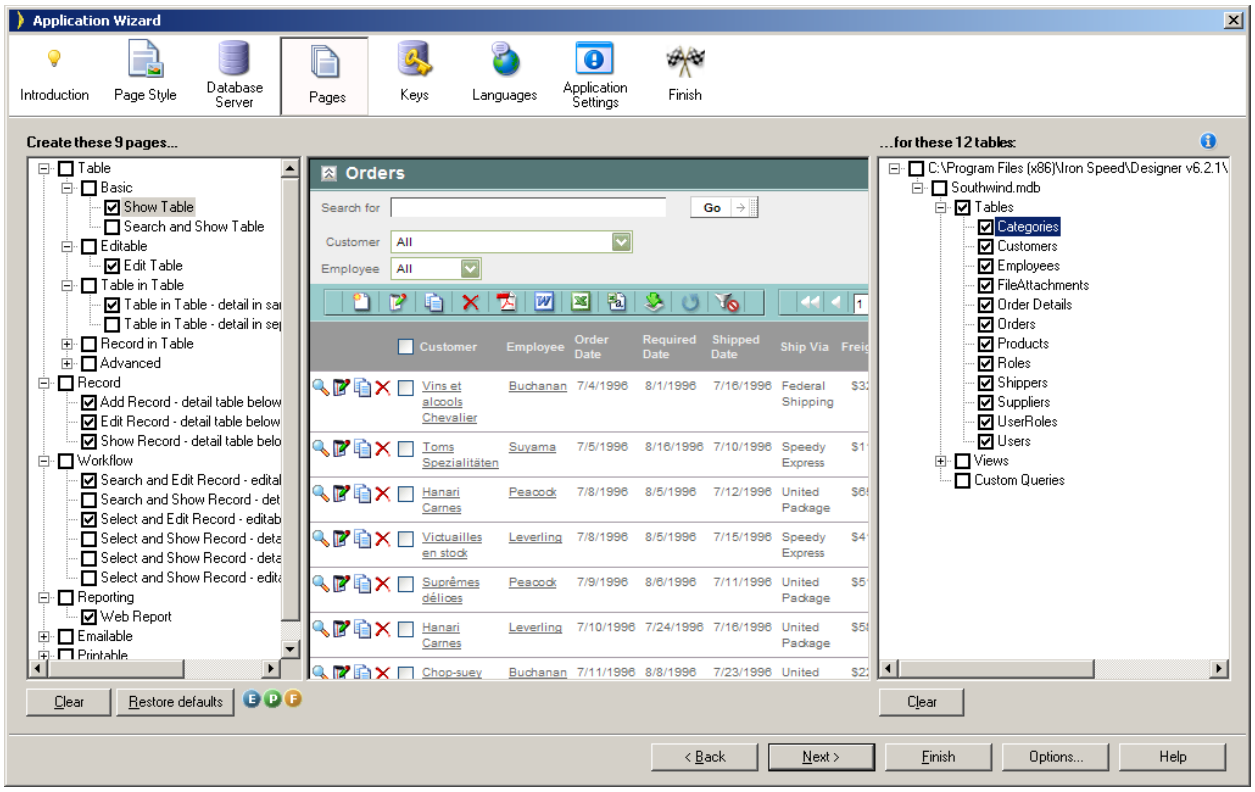
Generating Web 2.0 Applications

Translating an end-users' vision into a functional and intelligent Web application requires three main steps, each of them non-trivial.

- Design and create the Web pages to access and run your application, including data entry forms and reports;
- Build the application's basic Web 2.0 functionality — sophisticated features ranging from navigation and search to search and security;
- Connect Web pages and their components to the database, implementing the underlying SQL and transaction management code.

Application generation recognizes that the vast majority of most Web applications consist of fairly standard components – tables, panels, and controls – that lend themselves to being automatically generated and connected by an application generator.

A brief example illustrates this point. Suppose you need to build an order entry system that allows sales people to create, edit and retrieve sales orders. This application might have an input form for adding and editing orders and a table page for displaying orders for any given customer. Using an application generator, the developer selects various database tables and views necessary to build the application. When ready, the developer lets the application generator produce the necessary Web pages, source code and SQL, including code for all of the Web pages, controls, and database connectivity infrastructure.



Create these 9 pages...

- [ ] Table
  - [ ] Basic
    - [✓] Show Table
    - [ ] Search and Show Table
  - [ ] Editable
    - [✓] Edit Table
  - [ ] Table in Table
    - [✓] Table in Table - detail in sa
    - [ ] Table in Table - detail in se
  - [ ] Record in Table
  - [ ] Advanced
- [ ] Record
  - [✓] Add Record - detail table below
  - [✓] Edit Record - detail table below
  - [✓] Show Record - detail table belo
- [ ] Workflow
  - [✓] Search and Edit Record - edital
  - [ ] Search and Show Record - det
  - [✓] Select and Edit Record - editab
  - [ ] Select and Show Record - deta
  - [ ] Select and Show Record - deta
  - [ ] Select and Show Record - edit
- [ ] Reporting
  - [✓] Web Report
- [ ] Emailable
- [ ] Printable

Orders

Search for: Go

Customer: All

Employee: All

Customer	Employee	Order Date	Required Date	Shipped Date	Ship Via	Freig
Vins et alcools Chevalier	Buchanan	7/4/1996	8/1/1996	7/16/1996	Federal Shipping	$32
Toms Spezialitäten	Suyama	7/5/1996	8/16/1996	7/10/1996	Speedy Express	$1
Hanari Carnes	Peacock	7/8/1996	8/5/1996	7/12/1996	United Package	$6
Victuailles en stock	Leverling	7/8/1996	8/5/1996	7/15/1996	Speedy Express	$4
Suprêmes délices	Peacock	7/9/1996	8/6/1996	7/11/1996	United Package	$5
Hanari Carnes	Leverling	7/10/1996	7/24/1996	7/16/1996	United Package	$5
Chop-suey	Buchanan	7/11/1996	8/8/1996	7/23/1996	United	$22

...for these 12 tables:

- [ ] C:\Program Files (x86)\Iron Speed\Designer v6.2.1\
  - [ ] Southwind.mdb
    - [✓] Tables
      - [✓] Categories
      - [✓] Customers
      - [✓] Employees
      - [✓] FileAttachments
      - [✓] Order Details
      - [✓] Orders
      - [✓] Products
      - [✓] Roles
      - [✓] Shippers
      - [✓] Suppliers
      - [✓] UserRoles
      - [✓] Users
    - [ ] Views
    - [ ] Custom Queries

The Application Wizard in Iron Speed Designer generates sophisticated Web 2.0 pages from selected database tables and views.

Database and reporting applications like these lend themselves to application generation because they use typical database connectivity and application infrastructure. They all need data entry pages for the selected database tables and views, reporting pages to view and analyze the data in the application, application security, and other application infrastructure.

Data is entered into a web form and inserted into the database, and data is retrieved from the database and displayed on a Web page. These data operations are ideal for a template-based approach to application development, enabling an application generator to create the application and its infrastructure based on page and code templates using standardized and well-accepted ways to perform certain common operations.

Iron Speed Designer is different from other rapid application development tools because it generates complex user interfaces and all the supporting code and SQL, without you having to design pages or write any code. Using Iron Speed Designer, a custom, working application—not just a prototype or individual components—is quickly generated.

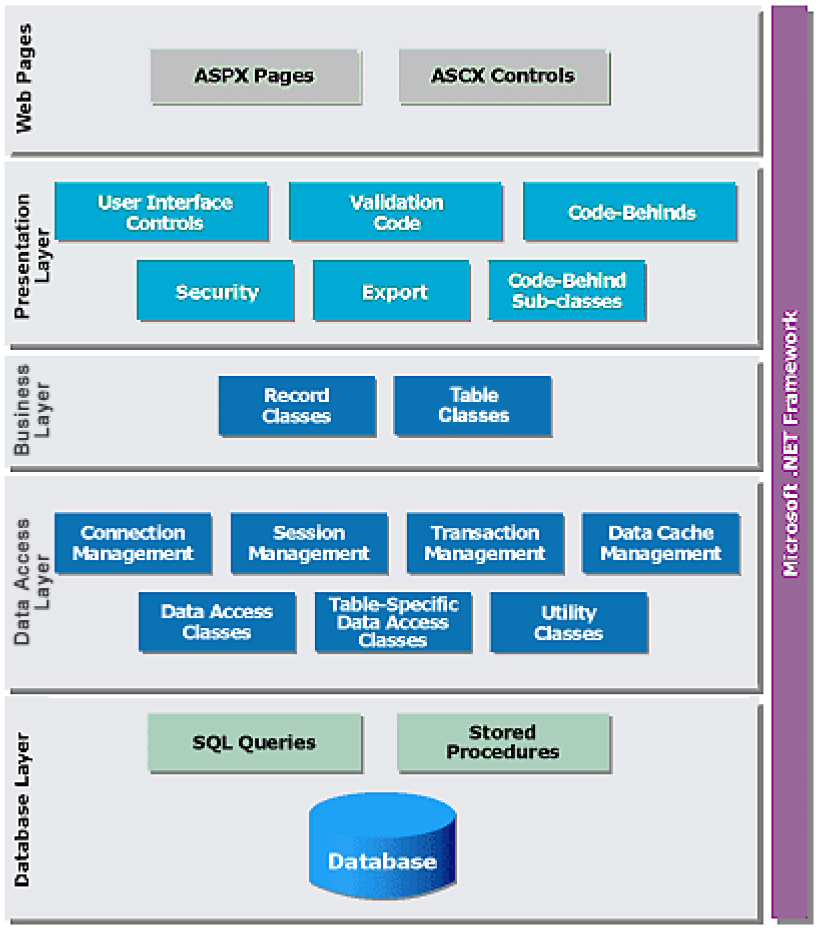
Generating an N-Tier Architecture

Most Web 2.0 applications are built on an N-tier architecture comprised of:

- Web pages — the Web pages comprising an application.
- Presentation Layer — the user interface code necessary to get and fetch data from the Web pages and operate the user interface controls.
- Application or Business Layer — where you typically add your code customizations, if any.
- Data Access Layer — SQL and transaction management code.
- Database Layer — database server-resident stored procedures.

Implementing transaction-based Web 2.0 applications can be difficult because of the stateless nature of the web. Stateless applications can require complex database management to implement complicated transactions, especially those spanning multiple Web pages such as multi-page data input wizards. Thin client browsers can display pages, but can't implement database transaction logic. As a result, the server-side application must manage every aspect of these database transactions.

Iron Speed Designer generates a complete N-tier application.



The next sections show how Iron Speed Designer generates these layers.

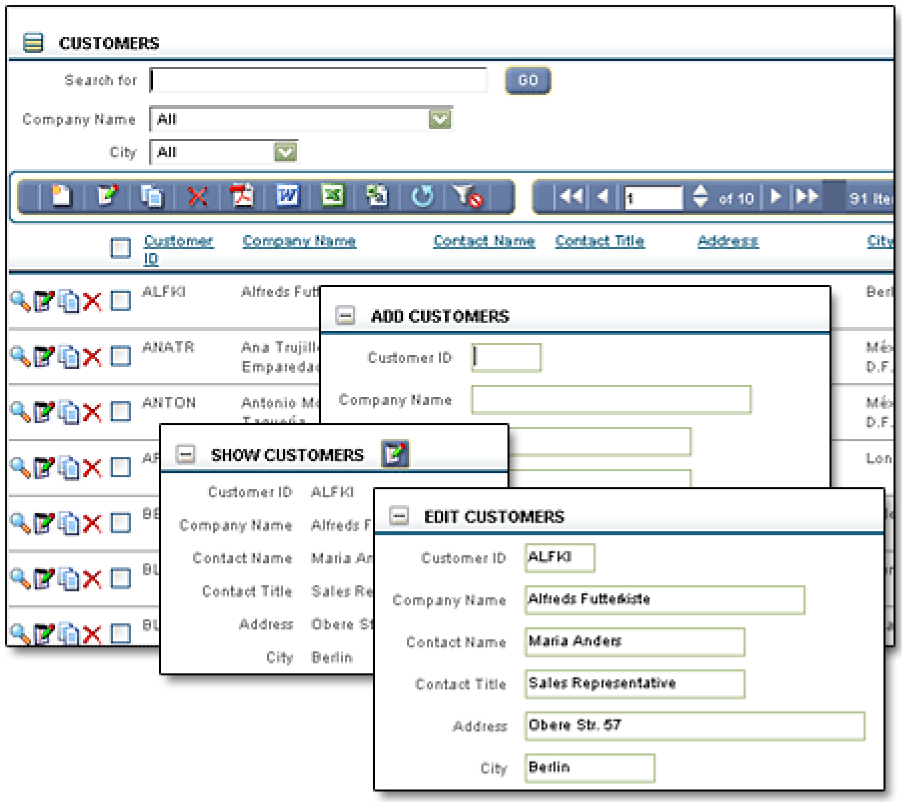
Generating Application Web Pages

While designing an application's look-and-feel is not trivial, repeating it over and over for each of the hundreds of pages in your Web application is a significant chore. Iron Speed Designer eliminates this struggle by automatically creating a suite of pages for each database table and view in your application. These pages reflect the basic operations performed with any database table: Create record, Retrieve record, Update record and Delete record –sometimes referred to as *C/R/U/D* pages.

- Add Record – a page for inputting data records.
- Edit Record – a page for editing data in an individual record.
- Show Record – a page for viewing data from one or more queries.
- Show Table – a paginated, interactive report view of a query's data.
- Edit Table – an editable data grid useful for editing a quantity of records.

All of the generated pages are automatically linked together with a menu navigation structure, providing an out-of-the-box application with no additional coding required.

These four standard database-connected Web pages can be automatically generated for each table in your application.



CUSTOMERS

Search for	
Company Name	All
City	All

	Customer ID	Company Name	Contact Name	Contact Title	Address	City
	ALFKI	Alfreds Fut				Berl
	ANATR	Ana Trujill Empareda				Méx D.F.
	ANTON	Antonio M Taqueria				Méx D.F.
	AR					Lon
	BE					
	BL					
	BL					

ADD CUSTOMERS

Customer ID	
Company Name	

SHOW CUSTOMERS

Customer ID	ALFKI
Company Name	Alfreds F
Contact Name	Maria An
Contact Title	Sales Re
Address	Obere St
City	Berlin

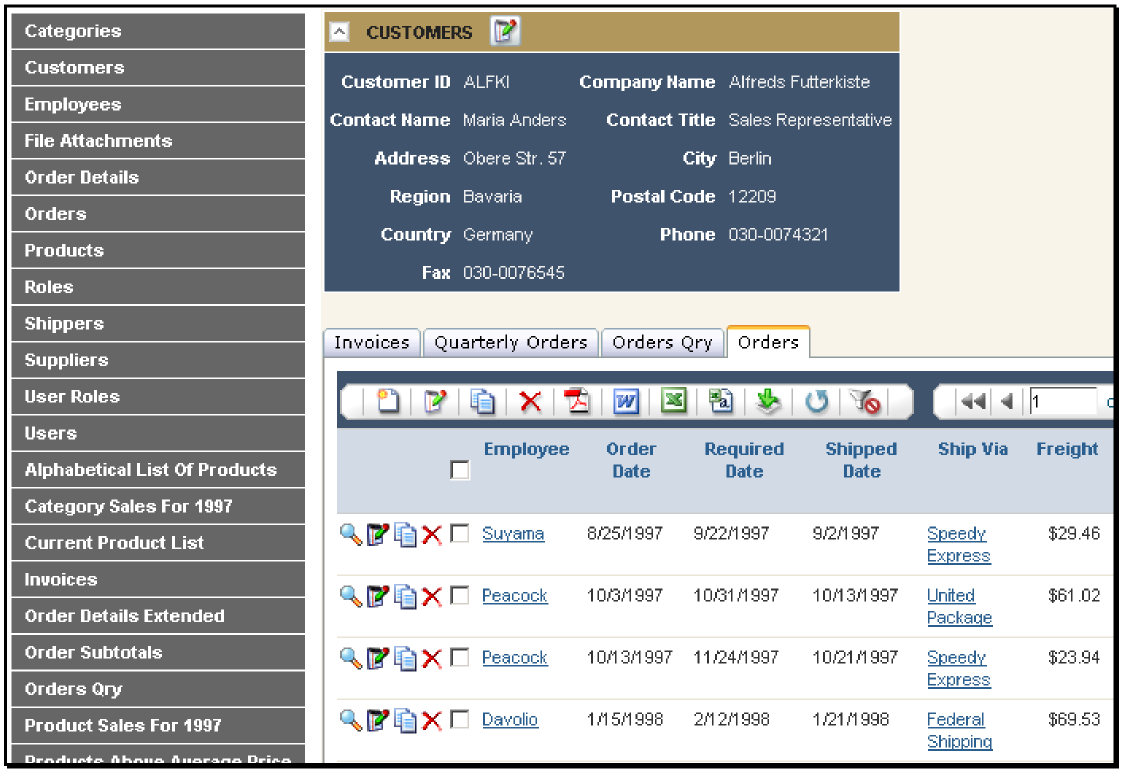
EDIT CUSTOMERS

Customer ID	ALFKI
Company Name	Alfreds Futterkiste
Contact Name	Maria Anders
Contact Title	Sales Representative
Address	Obere Str. 57
City	Berlin

Iron Speed Designer goes beyond basic C/R/U/D pages, however, by creating a variety of Web 2.0 pages and features you would normally expect to find only in custom desktop applications, such as:

- Master-detail pages – all generated pages can be master-detail (parent-child) pages referencing multiple database tables.
- Workflow pages – or displaying and processing workflow sequences that automate business processes.
- Reporting – pages that summarize data, including PDF and Microsoft Word reports that can be printed and emailed.
- Data integration – data import and export in a variety of formats for further analysis by application users.

Master-detail pages are among the advanced Web 2.0 features generated by Iron Speed Designer.



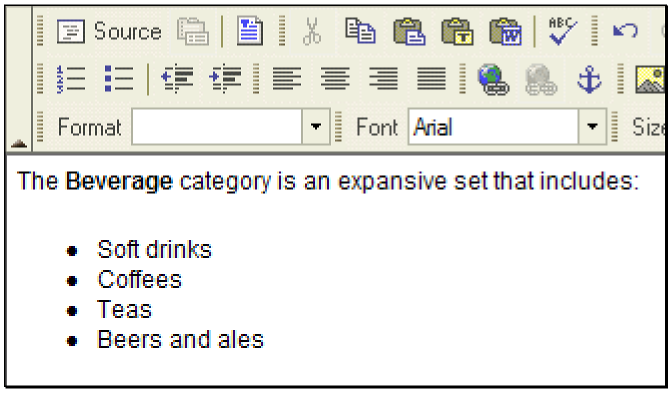
CUSTOMERS

Customer ID	ALFKI	Company Name	Alfreds Futterkiste
Contact Name	Maria Anders	Contact Title	Sales Representative
Address	Obere Str. 57	City	Berlin
Region	Bavaria	Postal Code	12209
Country	Germany	Phone	030-0074321
Fax	030-0076545		

Employee	Order Date	Required Date	Shipped Date	Ship Via	Freight
Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	$29.46
Peacock	10/3/1997	10/31/1997	10/13/1997	United Package	$61.02
Peacock	10/13/1997	11/24/1997	10/21/1997	Speedy Express	$23.94
Davolio	1/15/1998	2/12/1998	1/21/1998	Federal Shipping	$69.53

Web 2.0 features extend beyond page types to the components in those pages. These components might include rich text editors, tree controls, multi-level menus and tabbed panels for data input and display. Complex navigation defines Web 2.0 applications with features such as multi-step wizards for data collection and workflow processes governed by application user interaction.

Web 2.0 components like this rich text editor generated by Iron Speed Designer bring desktop functionality to Web applications.

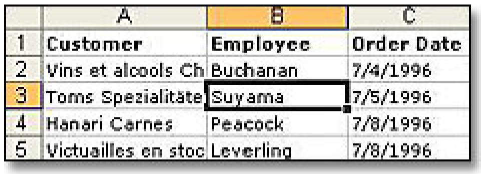


The Beverage category is an expansive set that includes:

- Soft drinks
- Coffees
- Teas
- Beers and ales

Advanced application features like integrated security, reporting, multi-lingual support, and data integration are also hallmarks of Web 2.0 applications.

Export formatted data directly to Microsoft Excel for further analysis and reporting.



	A	B	C
1	Customer	Employee	Order Date
2	Vins et alcools Ch	Buchanan	7/4/1996
3	Toms Spezialitäte	Suyama	7/5/1996
4	Hanari Carnes	Peacock	7/8/1996
5	Victuailles en stoc	Leverling	7/8/1996

Generating the Data Access Layer

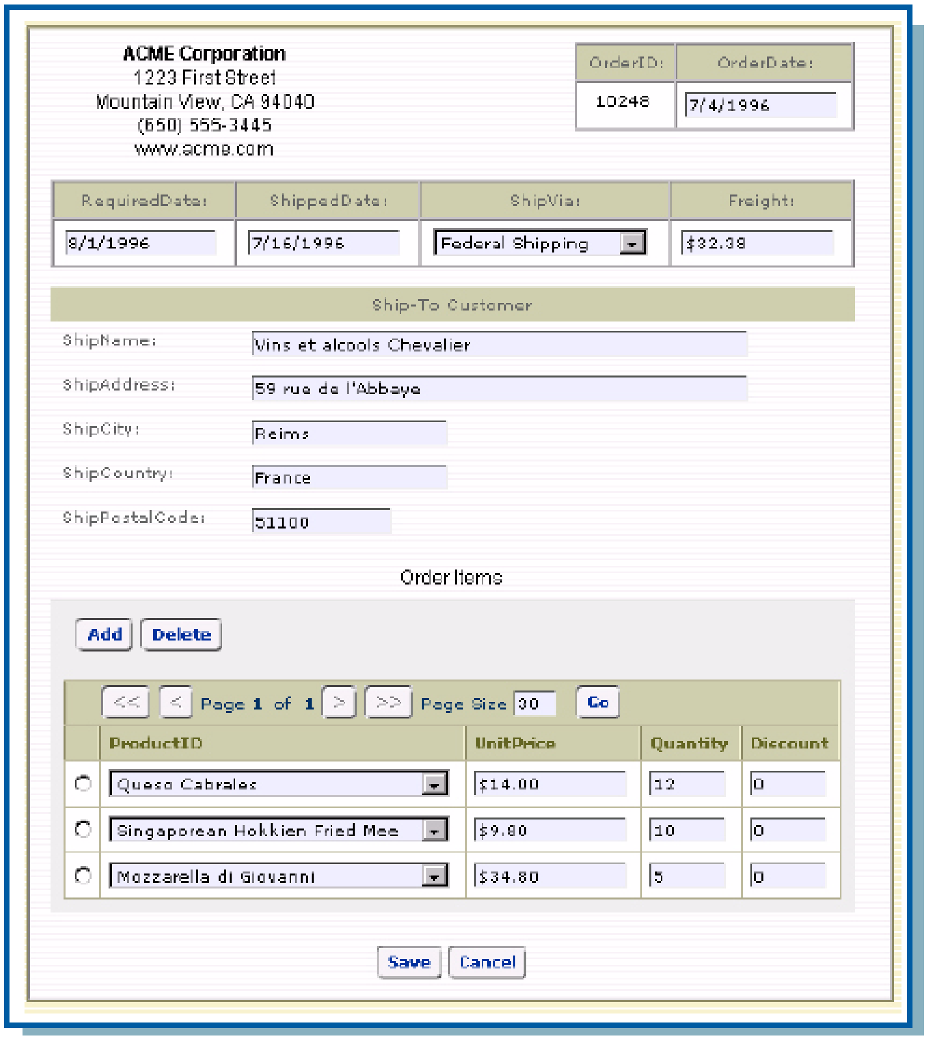
When generating the application, Iron Speed Designer generates data-bound controls, server-side code-behind logic and transaction management code that constitute the bulk of the application. Iron Speed Designer automatically generates every SQL query and the data access code for each web-based component that accesses the database. Thus, you do not have to write any code to create sophisticated database and reporting applications with Iron Speed Designer.

Even for experienced developers, application scalability is hard to implement. Since all of the business logic in Web applications is concentrated server-side in the application layer, scalability becomes a big issue for Web applications. Functionality isn't distributed to the client-side as it is with client-server applications, so you can't rely on the application user's machine to offload computationally intensive tasks. Hence, improperly designed applications can cause performance bottlenecks in the middle-tier.

Iron Speed Designer builds applications with a multi-tier architecture that scales easily, including optimized SQL, and advanced transaction, cache, and session management techniques.

Generating Application Security

Most applications serve a variety of constituents – customers, customer service, marketing, sales, and management, to name a few. It's almost universally common in contemporary Web applications to permit broad access to applications and their underlying data, and with broad use comes the need to partition data according to the user type – and sometimes down to the individual user as well.



ACME Corporation
1223 First Street
Mountain View, CA 94040
(650) 555-3445
www.acme.com

OrderID:	OrderDate:
10248	7/4/1996

RequiredDate:	ShippedDate:	ShipVia:	Freight:
8/1/1996	7/16/1996	Federal Shipping	$32.38

Ship-To Customer

ShipName:	Vins et alcools Chevalier
ShipAddress:	59 rue de l'Abbaye
ShipCity:	Reims
ShipCountry:	France
ShipPostalCode:	51100

Order Items

Page 1 of 1 Page Size 30

	ProductID	UnitPrice	Quantity	Discount
○	Queso Cabrales	$14.00	12	0
○	Singaporean Hokkien Fried Mee	$9.80	10	0
○	Mozzarella di Giovanni	$34.80	5	0

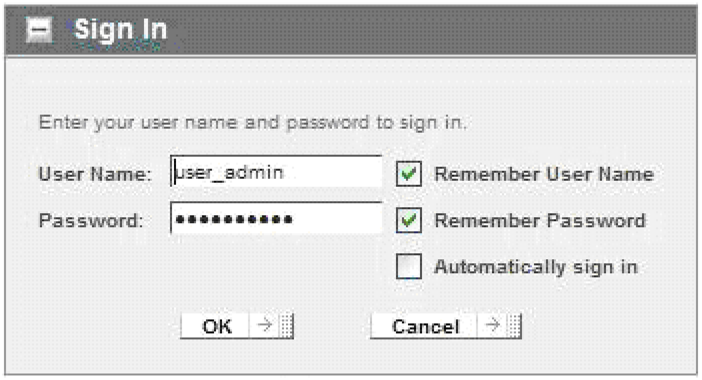
This seemingly straightforward "Edit Order" page can create code havoc when you have to add enterprise-class security, concurrency and transaction management code by hand.

Although most corporations take security precautions at the network level, the majority of application security must be built into the application itself. You can increase your application's security by taking advantage of the powerful benefits of application generation. For example, Iron Speed Designer generates applications with these built-in security features:

- Authentication – who can log into an application
- Authorization – what can someone do once they're logged in
- Component-level (field) security – who can see what field
- URL parameter encryption – prevents robotic attacks
- Automatic sign-out – guards against unattended machines
- Data transmission encryption – prevents unauthorized monitoring

Iron Speed Designer supports Active Directory, Windows Authentication and Database authentication and authorization.

Automatically generating features like role-based security can save many hours of programming.


Sign In

Enter your user name and password to sign in.

User Name: user_admin ✓ Remember User Name

Password: •••••••••• ✓ Remember Password

☐ Automatically sign in

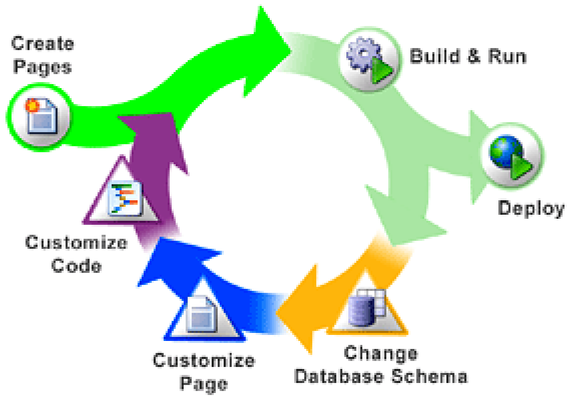
Broadly speaking, Iron Speed Designer generates role-based security that supports both authentication as well as authorization. Simple sign-in authentication governs who can sign into the application. A valid user name and password is required to sign in, giving you control over who can access your application.

With multiple-role authorization, individual Web pages are configured to accept users belong to one of several designated roles. For example, customer service representatives assigned to the “rep” role may be able to access most customer information but not customer credit card data. However, those belonging to the “manager” role can issue refunds or credits. Users can be assigned multiple roles, effectively giving them tailored access.

Customizing Generated Web Pages

It’s virtually impossible to generate applications that require no customization – extensions made by hand to the page layouts or to the generated code to add custom application logic, integrate with third-party components, and interface with external systems. While many administrative and data management applications can escape the need for customization, most generated production applications have some amount of customization.

Application generation encourages an iterative development style.



Customizing Page Layout

While an application generator can make many intelligent assumptions, it can't always create the optimal page layout or even decide which fields to place on a page and which to omit. So no matter what, you need to customize page layout as well as add and remove controls, panels, and navigation components. There are several approaches to page layout customization:

- Logical page editing using an editor that uses a logical representation of a page, generally based upon the controls and other user interface artifacts on the page. Iron Speed Designer uses this approach.

- Direct physical page editing using your favorite Web page editor, such as Dreamweaver, Microsoft Visual Studio, or Microsoft Front Page, to modify pages directly at the ASPX or HTML level.

There are pros and cons to both approaches. The chief benefit of logical page editing is it permits the application generator to regenerate your pages with your customizations whereas physical page editing does not. The application generator takes ownership of your page customizations and uses them to regenerate your page. This avoids the problem of 'page orphaning' whereby customizations made external to the application generator are lost when your page is regenerated.

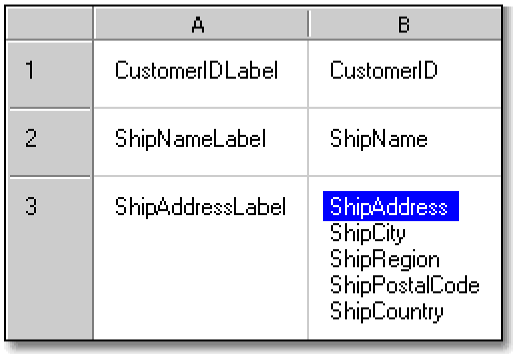
A second benefit of logical page editing is that no knowledge of HTML, ASPX or .NET concepts is required to customize page layout. This permits non-developers or para-technicals to build Web applications.

On the other hand, direct physical page editing lets you see what you're editing and how it will look. However, directly editing a page's HTML makes it almost impossible for the application generator to preserve your changes when regenerating the page. The application generator can't understand the external edits you've made and incorporate those into a regenerated page that may have a very different format than your modified page.

Moreover, WYSIWYG editing is nearly impossible to implement in the ASP.NET world. Most modern Web pages are dynamically rendered, making it almost impossible to create a truly WYSIWYG editor. The editor must be hooked to the actual database and render the data in situ as you edit, a very complex undertaking.

Iron Speed Designer uses a logical page editing approach featuring a spreadsheet layout editor as its editing paradigm.

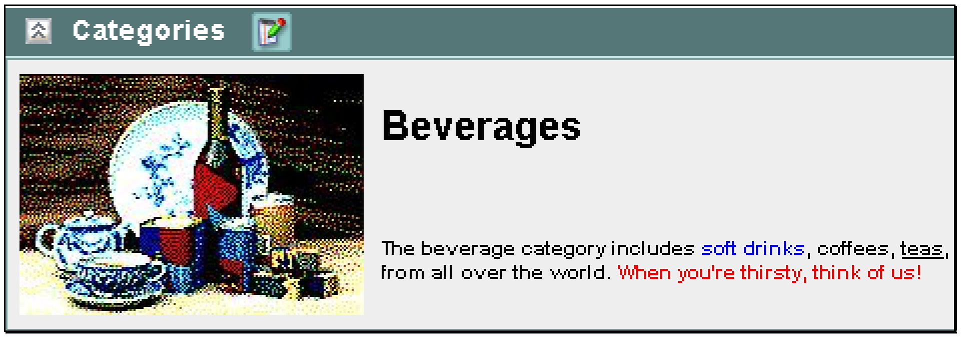
An intuitive layout spreadsheet makes page customization easy. There is no HTML or ASPX to learn!



	A	B
1	CustomerIDLabel	CustomerID
2	ShipNameLabel	ShipName
3	ShipAddressLabel	ShipAddress ShipCity ShipRegion ShipPostalCode ShipCountry

You can easily rearrange page layout by dragging data bound controls to their desired locations. When ready, the page is regenerated and can be viewed in an accompanying 'live preview' screen.

Pages are viewed on a live preview screen, displaying actual application data.



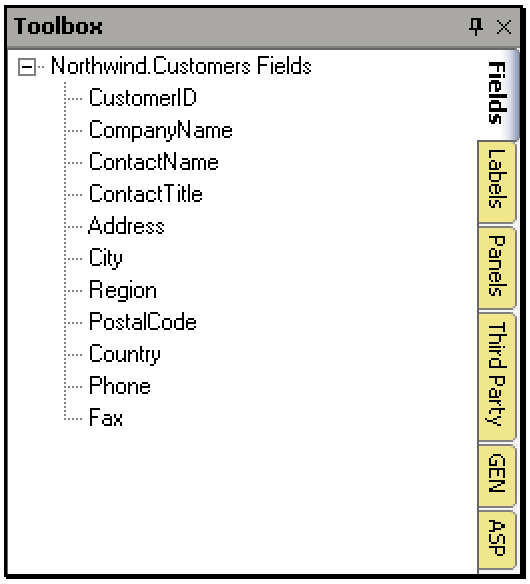
Categories

Beverages

The beverage category includes soft drinks, coffees, teas, from all over the world. When you're thirsty, think of us!

Additional data bound controls are easily added by dragging them from the Toolbox into the layout spreadsheet. You can, for example, drag complete panels – parent or child, record or table – onto your page, as well as third-party controls and other components. No hand-coding is required.

Drag data bound controls from the Toolbox onto your page. No coding is necessary to connect components.



Toolbox

Northwind.Customers Fields
- CustomerID
- CompanyName
- ContactName
- ContactTitle
- Address
- City
- Region
- PostalCode
- Country
- Phone
- Fax

Finally, a 'cell editor' is used for adding any custom HTML, ASP.NET tags, JavaScript or text the developer wishes to add to a cell. This custom content is preserved and regenerated as part of the page.

Customizing Look-and-Feel Using Style Sheets

How physically attractive are automatically created Web pages? You might be concerned that mechanically generated pages look ugly and, well, mechanical. Nothing could be further from the truth!

The difference between these two applications is strictly the style sheet.

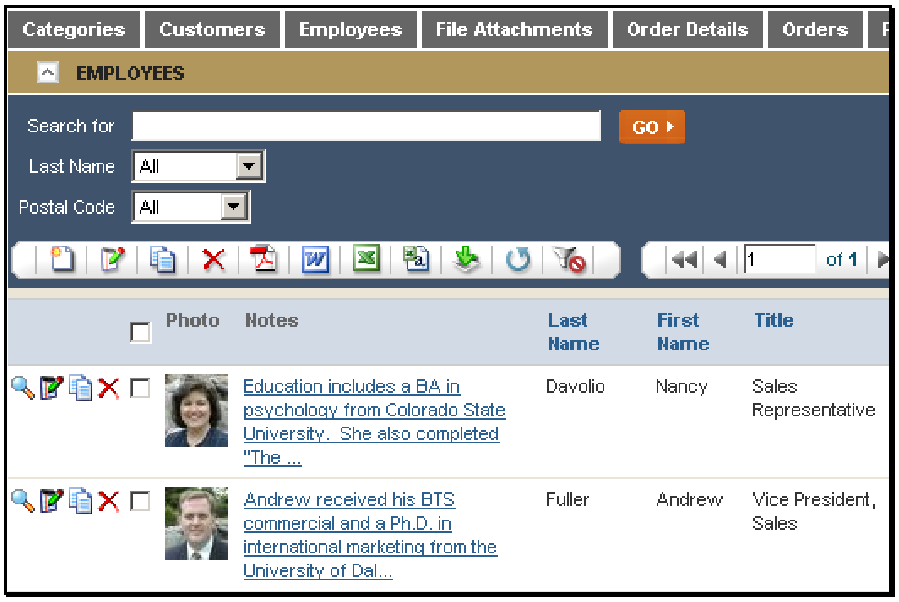


	Photo	Notes	Last Name	First Name	Title
		Education includes a BA in psychology from Colorado State University. She also completed "The ...	Davolio	Nancy	Sales Representative
		Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dal...	Fuller	Andrew	Vice President, Sales

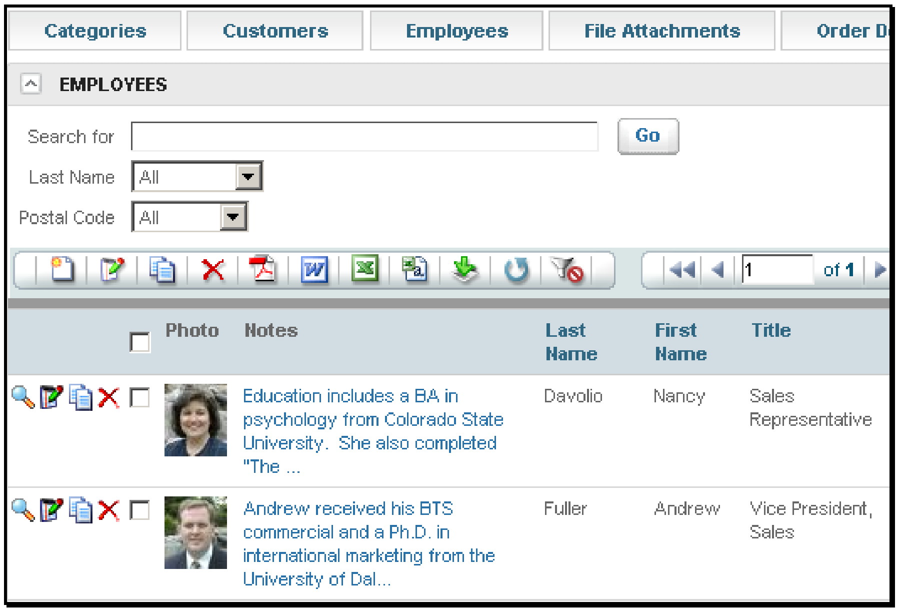
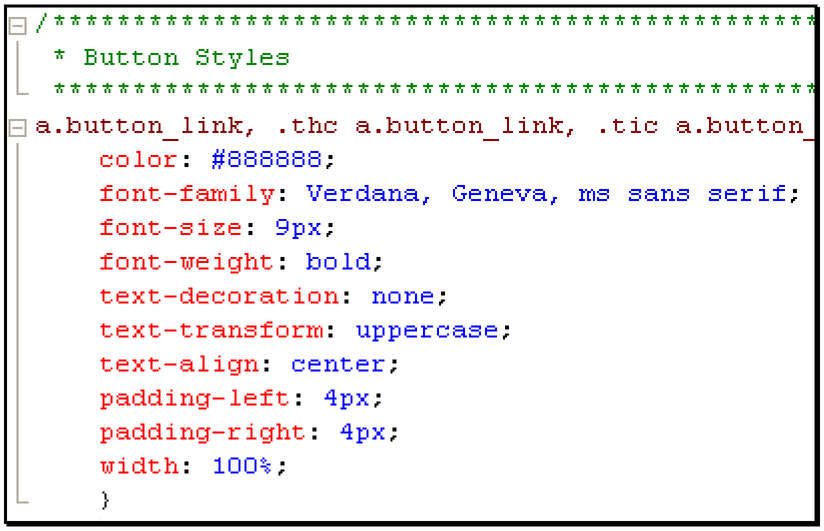


	Photo	Notes	Last Name	First Name	Title
		Education includes a BA in psychology from Colorado State University. She also completed "The ...	Davolio	Nancy	Sales Representative
		Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dal...	Fuller	Andrew	Vice President, Sales

Iron Speed Designer uses a template-driven approach to generate Web pages. Over 25 different professionally-designed style sheets are included with the product, providing a variety of layouts, color schemes, and other important visual attributes. In short, generated applications look great without any customization. More importantly, these style sheets can be customized, allowing developers to either modify one of the style sheets to their liking or create one of their own.

Style sheets control nearly every stylistic element in applications generated by Iron Speed Designer. This makes look-and-feel customization very straightforward.



```
/*************************************************
 * Button Styles
 *************************************************
a.button_link, .thc a.button_link, .tic a.button_
    color: #888888;
    font-family: Verdana, Geneva, ms sans serif;
    font-size: 9px;
    font-weight: bold;
    text-decoration: none;
    text-transform: uppercase;
    text-align: center;
    padding-left: 4px;
    padding-right: 4px;
    width: 100%;
    }
```

Nearly all look-and-feel aspects can be controlled by changing the appropriate CSS class, making it easy for non-developers and graphic designers, to modify the stylistic elements of the generated application.

Customizing Generated Code

You may want to customize the generated application in order to add custom business logic, add additional user interface components or integrate with external systems. Your application generator must produce code that is readable, easy-to-understand and easy-to-customize; otherwise the code won't be trusted and the time benefits achieved by application generation are lost. There are several concerns:

- Locating where to add a code customization. Application generators can produce voluminous amounts of code, so it's essential to be able to quickly locate the correct place to add a code customization.

- Understanding the code model. An application generator generates code in a certain code model. Customizing that generated code relies on your ability to understand that code model.

- Preserving code customizations when the application is regenerated. The best application generators can quickly regenerate an application using different developer preferences. That regeneration process must preserve any code customizations you've made so you don't lose any work.

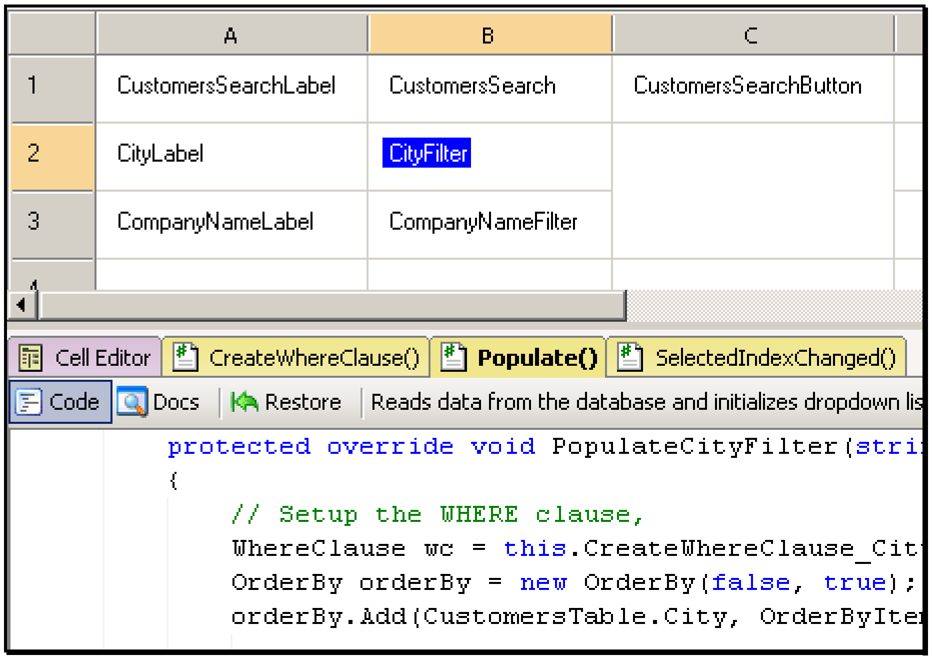
While there are different solutions, this section describes how Iron Speed Designer approaches these issues.

Locating the Right Code to Customize

Congratulations, you've generated 100,000 lines of code in 5 minutes! Now where do you place your code customization?

Iron Speed Designer takes an innovative approach to answering this question using a concept called 'code tabs'. Code tabs combine the page layout spreadsheet with a code editor. Selecting a specific control or user interface component in the layout spreadsheet displays relevant code methods in a set of code tabs displayed below the layout spreadsheet.

Selecting a page control displays relevant code methods in a set of code tabs.



	A	B	C
1	CustomersSearchLabel	CustomersSearch	CustomersSearchButton
2	CityLabel	CityFilter	
3	CompanyNameLabel	CompanyNameFilter	

```
protected override void PopulateCityFilter(stri
{
    // Setup the WHERE clause,
    WhereClause wc = this.CreateWhereClause_Cit
    OrderBy orderBy = new OrderBy(false, true);
    orderBy.Add(CustomersTable.City, OrderByIte
```

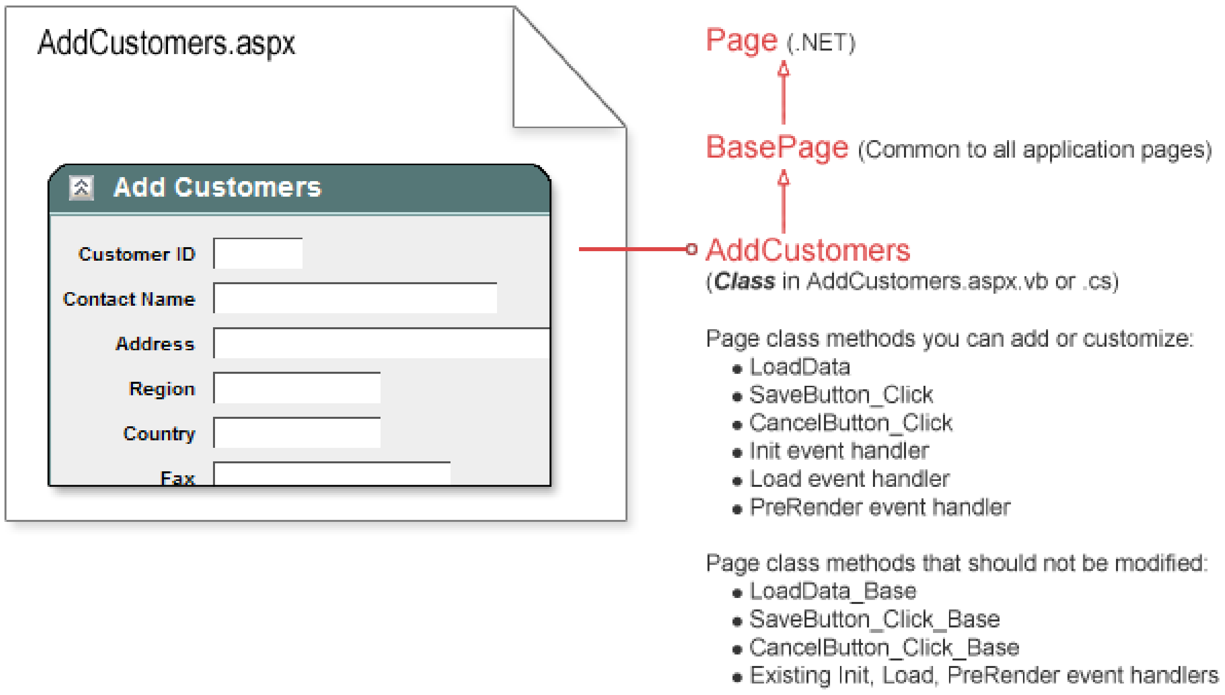
While this mechanism can't display all possible code methods – there would be way too many code tabs – it does highlight the most commonly customized methods. This has the benefit of focusing your attention on the important code methods, solving the 'forest from the trees' problem of trying to understand a very large body of code.

Companion code documentation is generated for each method and is easily accessed by clicking the 'Docs' button in the tool bar within the code tab.

Understanding the Code Model

The class hierarchies generated by Iron Speed Designer derive from the base classes in the .NET Framework. In most applications written for the Microsoft .NET Framework, the functionality and logic in the Application Layer are derived from a set of C# or Visual Basic .NET classes that implement page management, user interface controls, database access, and data validation.

Using your favorite code editor, such as Microsoft Visual Studio or the code editor in Iron Speed Designer itself, simply subclass the appropriate page, control or database class and add your code customization. The class hierarchy means your code customizations have access to the full functionality of the generated application, all the way down to the .NET framework.


AddCustomers.aspx
Add Customers
Customer ID
Contact Name
Address
Region
Country
Fax
Page (.NET)
BasePage (Common to all application pages)
AddCustomers
(Class in AddCustomers.aspx.vb or .cs)
Page class methods you can add or customize:
• LoadData
• SaveButton_Click
• CancelButton_Click
• Init event handler
• Load event handler
• PreRender event handler
Page class methods that should not be modified:
• LoadData_Base
• SaveButton_Click_Base
• CancelButton_Click_Base
• Existing Init, Load, PreRender event handlers

The inheritance hierarchy for a typical Web page generated by Iron Speed Designer.

You can override any of the functions or methods in the generated classes or the .NET Framework classes, or write your own methods in combination with the existing base class methods. Then, regenerate the application to reflect your changes.

Regenerating Without Losing Code Customizations

One of the biggest challenges of application generation is *code orphaning*. Code orphaning occurs when your application cannot be modified or extended without severing the application generator's ability to regenerate that application without losing your code customizations. Application generation encourages iterative development practices, but that's impossible if customizing your application removes your ability to regenerate it.

Iron Speed Designer tackles this by using the object inheritance model so that the inherited classes you create for your code customizations are never overwritten when the application is regenerated. Only the *generated classes* – the automatically generated portions of your application – are ever regenerated; your inherited classes and methods are never regenerated, preserving your code customizations.

As long as your modifications are segregated in the sub-classed methods and not in the generated classes, your application can be regenerated repeatedly without re-integrating your code extensions. Your code extensions are not overwritten when the application is regenerated, allowing you a high degree of flexibility in making code extensions to the application.

Contact Iron Speed

Building Web applications by hand is complicated and time-consuming. The ROI delivered by Iron Speed Designer means more efficient utilization of key personnel, reduced development and testing time, and more rapid deployment of robust applications.

Let us show you how Iron Speed Designer can accelerate your Web application development by eliminating hand-coding.

Telephone:	1-408-228-3420
Web:	http://www.ironspeed.com

Iron Speed, Inc.
2870 Zanker Road
Suite 210
San Jose, CA 95134

Please send comments, suggestions and ideas regarding this white paper to editor@ironspeed.com