

# User's Guide



# CodeCharge Studio User's Guide

---

© 2002 YesSoftware, Inc.  
2226 A. Westborough Blvd. #405  
South San Francisco, CA 94080  
Phone: (650) 754-9810  
Fax: (866) 312-8049

**[www.codecharge.com](http://www.codecharge.com)**

# Table of Contents

<b>I.</b>	<b>Introduction.....</b>	<b>6</b>
	Welcome.....	6
	Features .....	6
	IDE (Integrated Development Environment).....	6
	Application Design.....	6
	Code Generation .....	7
	System and Server Requirements.....	7
	System Requirements .....	7
	Server Requirements.....	7
<b>II.</b>	<b>Getting Started .....</b>	<b>8</b>
	Downloading CodeCharge Studio.....	8
	Installing, Upgrading and Repairing.....	8
	Installing CodeCharge Studio .....	8
	Upgrading, Repairing or Uninstalling CodeCharge Studio .....	12
	Setting up the work environment .....	14
	ASP Setup.....	15
	C# Setup .....	15
	Cold Fusion Setup .....	16
	Java Servlets.....	16
	PHP Setup .....	17
	Perl .....	18
<b>III.</b>	<b>Using CodeCharge Studio .....</b>	<b>19</b>
	The CodeCharge Studio Workspace .....	19
	The Command Area .....	20
	The Project Explorer.....	25
	The Toolbox .....	26
	The Properties Window .....	27
	Customizing the Workspace .....	29
	Keyboard Accessibility.....	34
	Translation Files.....	36
	Project Settings.....	37
	Project.....	38
	Publishing.....	39
	Connections .....	40
	Security .....	41
	Security Groups.....	42
<b>IV.</b>	<b>Databases and Data Sources .....</b>	<b>43</b>
	Connections.....	44
	Creating Connections .....	44
	Design Connection Settings.....	45
	Server Connection Settings .....	46
	Creating a Connection String (DNS-Less Connection).....	47
	Creating an ODBC Data Source Name (DSN) .....	49
	Form Data Sources.....	53
	Using the Visual Query Builder Window .....	60

	Using the Table Parameter Window .....	64
	Using the Stored Procedure Parameter Window .....	66
	Using the SQL Parameter Window .....	67
<b>V.</b>	<b>Controls and HTML Elements.....</b>	<b>69</b>
	HTML Elements .....	69
	Form .....	69
	Text Box .....	69
	Password .....	70
	Text Area .....	70
	Checkbox .....	70
	Radio Button .....	70
	Dropdown .....	71
	Listbox.....	71
	Button.....	71
	Submit Button.....	71
	Reset Button .....	72
	Image Button .....	72
	Label .....	72
	Hyperlink .....	73
	Image .....	73
	ImageLink.....	74
	Space.....	75
	Line Break.....	75
	Horizontal Rule.....	76
	Paragraph Break .....	76
	Bookmark.....	76
	Controls .....	76
	Include Page .....	77
	Label .....	77
	Sorter .....	78
	Navigator.....	79
	Link .....	80
	Text Box .....	81
	Hidden.....	81
	Text Area .....	81
	Checkbox .....	81
	Image .....	81
	RadioButton .....	82
	List Box .....	82
	ImageLink.....	82
	Button.....	82
<b>VI.</b>	<b>Forms .....</b>	<b>83</b>
	Grid Form .....	83
	Record Form.....	84
	Using the Builders to create forms.....	84
	Application Builder.....	85
	Grid Builder .....	100
	Record Builder.....	109
	Search Builder.....	112
	Authentication Builder.....	116

	Grid and Record Builder .....	119
<b>VII.</b>	<b>Object Properties .....</b>	<b>128</b>
	Project Properties .....	128
	Form Properties .....	129
	Grid Properties .....	129
	Record, Search and Login Form Properties .....	130
	Control Properties .....	130
	HTML Properties .....	134
	Connection Properties .....	135
<b>VIII.</b>	<b>Working with Documents .....</b>	<b>137</b>
	Design Mode .....	137
	HTML Mode .....	138
	Code Mode .....	139
	Preview Mode .....	140
	Live Page .....	140
	Using the Microsoft FrontPage Add-In .....	141
	Using External HTML Editors .....	142
<b>IX.</b>	<b>Client Side Event Actions .....</b>	<b>144</b>
	Set Focus .....	144
	Validate Entry .....	145
	Validate Form .....	145
	Validate Maximum Length .....	145
	Validate Minimum Length .....	146
	Validate Maximum Value .....	146
	Validate Minimum Value .....	146
	Validate Required Value .....	147
	Regular Expression Validation .....	147
	Custom Code .....	147
<b>X.</b>	<b>Appendix A .....</b>	<b>148</b>
	Java and JSP Implementation Architecture .....	148
	Java Servlet Control Flow .....	148
	JSP and XML Model .....	150
	Approaches .....	151
	Features .....	151
	Components .....	152
	Requirements .....	153
	Tomcat 4.0 scenarios .....	153
	Resin 2.0 scenarios .....	154
	WebLogic 7.0 scenarios .....	155
	End User License Agreement .....	157
	Copyright .....	160

# I. Introduction

## **Welcome**

CodeCharge Studio is a powerful Rapid Application Development (RAD) tool used to design, develop and deploy database driven web applications in six different languages (ASP/VBScript, .Net/C#, Cold Fusion 4, JSP, PHP and Perl) and running off a score of different databases. Time is money and this is particularly true in the fast paced WWW environment. CodeCharge Studio is squarely aimed at saving the developer a considerable amount of time and money by generating robust, extendable and highly configurable code that would otherwise require a significant investment in time and effort if it were to be written from scratch.

With its ability to generate web applications in six different languages, CodeCharge Studio doubles up as a learning tool that can be used to learn about web application development and make quick inroads into the field by learning from code that actually works, and is used in real life applications.

## **Features**

### **IDE (Integrated Development Environment)**

- Powerful, professional IDE
- Built-in web editor
- Full integration with Microsoft FrontPage
- Built-in powerful code editor with color code highlighting
- Fast, on-the-fly, two-way code generation
- Extreme extensibility: User interface and components created in an open XML format making them easy to customize and expand

### **Application Design**

- Design web applications using a simple point-and-click process
- Use Builders to create Data Entry Forms and Grids/Tables
- Design and apply custom Themes to forms on the page
- Define Login access to each page by specifying a minimum Security Level or Group
- Define multiple database connections per application
- Include pages as components within other pages
- Create custom Builders in JavaScript and XML
- “Hook” into generated code by creating Events and Actions
- Create and share custom Actions, which automatically place programming code within Events
- For more flexibility, manually modify generated code, which is fully preserved during subsequent code generations!

## Code Generation

- Generates object-oriented applications in ASP, .NET/C#, Java, PHP, Perl and Cold Fusion
- Generated code works with almost all databases via JET, ODBC, ADO, DBI and PHPLib
- Generated code is Windows and Linux compatible
- Helps beginners quickly get familiar with .NET and other web technologies, while rapidly building web applications
- Helps intermediate and advanced developers focus on the system architecture and design, while eliminating tedious coding

## *System and Server Requirements*

### System Requirements

The following are the minimum requirements needed for a system to run the CodeCharge Studio application

- Intel Pentium Processor or equivalent
- 64 MB available RAM
- 20 MB available disk space
- 256-color monitor capable of 800 x 600 resolution
- Windows 95, 98, ME, NT4, 2000 or XP
- Database application (MS Access, MS SQL, MySQL, Oracle, etc.)
- Microsoft Internet Explorer 5.5 or higher with JavaScript enabled.

### Server Requirements

In order to be able to run the generated pages, you need a Web server on the Windows or Unix/Linux platform that supports one of the following languages:

- ASP 3.0/VBScript 5.0+
- ASP.Net/C#
- Cold Fusion 4.0.1 or higher
- Java JDK 1.2 and Servlet API 2.2
- PHP 4.0

The server should also be capable of accessing the database required by the generated pages. If using a host server in a remote location, the server should have FTP support to enable uploading of the generated pages.

## II. Getting Started

### *Downloading CodeCharge Studio*

You can acquire a copy of the CodeCharge Studio installation executable from the following address: (<http://www.CodeCharge.com/Studio>).

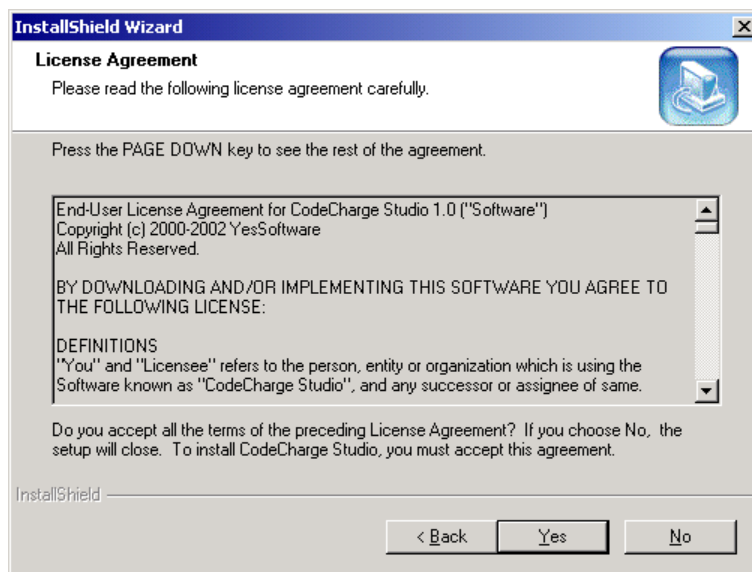
The installation file features a 30-day trial period that allows prospective users to test run the system before deciding on whether to make a purchase. Note that both licensed users and trial users download the same file that is fully functional but time-limited unless activated using a valid serial number.

### *Installing, Upgrading and Repairing*

#### **Installing CodeCharge Studio**

Before beginning the installation process, you should ensure that your system meets the requirements stipulated above in the [System requirements](#) section.

1. To begin the installation process, locate the CodeCharge Studio setup file and double-click on it. Before you can proceed with the installation process, you should read and agree to the License Agreement. Click on the **Yes** button to indicate your acceptance of the License Agreement and proceed to the next step.

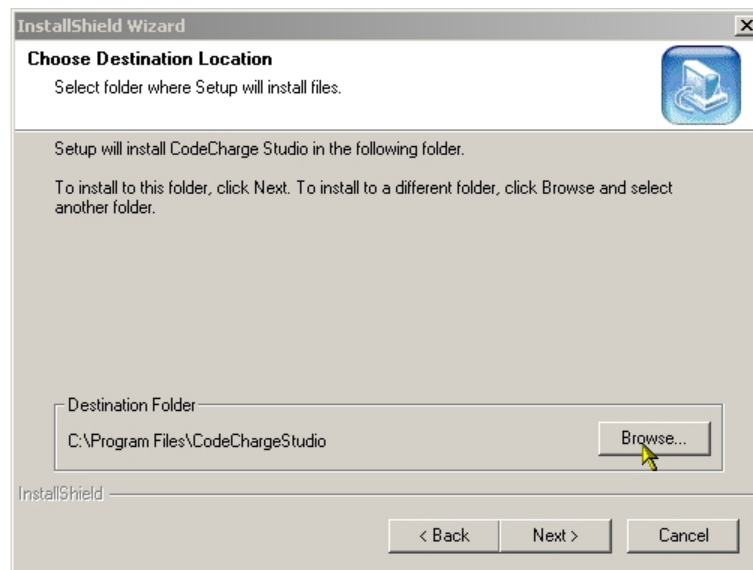


2. In the next window, you are presented with fields to enter your name, company name and Serial Number. If you don't have a serial number and want to take advantage of the 30 day trial period, simple enter "Trial" in the Serial Number field. Click **Next** to proceed.



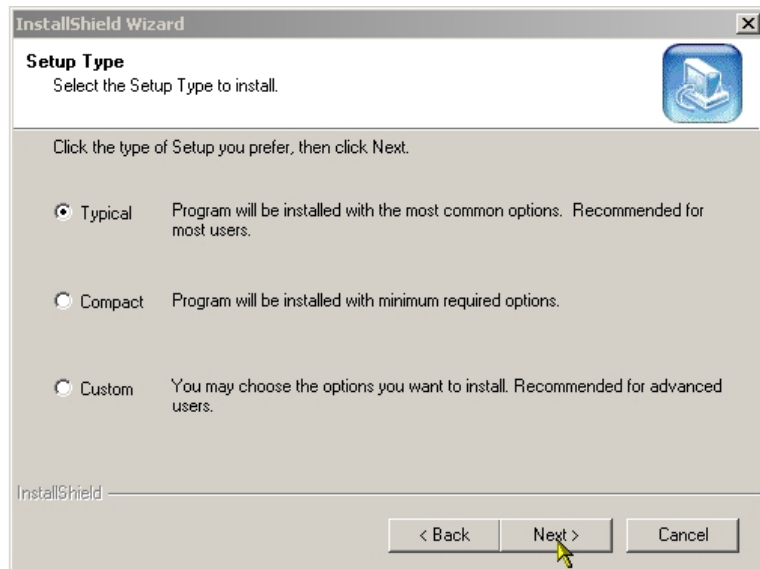


3. In the ensuing window, confirm or specify the destination folder where you want CodeCharge Studio to be installed. The default recommended path for installing CodeCharge Studio is C:\Program Files\CodeCharge Studio. However, if you wish to change this path, click **Browse** and specify an alternate path. Once done, click **Next** to proceed to the next step.

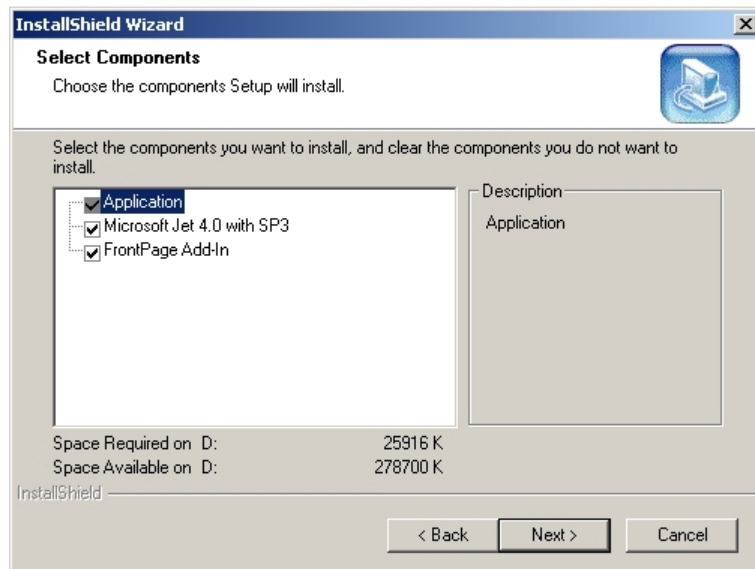


4. In the next window, you are presented with three setup options from which to choose. The Typical option installs all the CodeCharge options as shown in step 5 below. The Compact option installs the CodeCharge application files but does not install the examples or Microsoft

Jet 4.0. Finally, the Custom option allows the user to select the components to be installed.



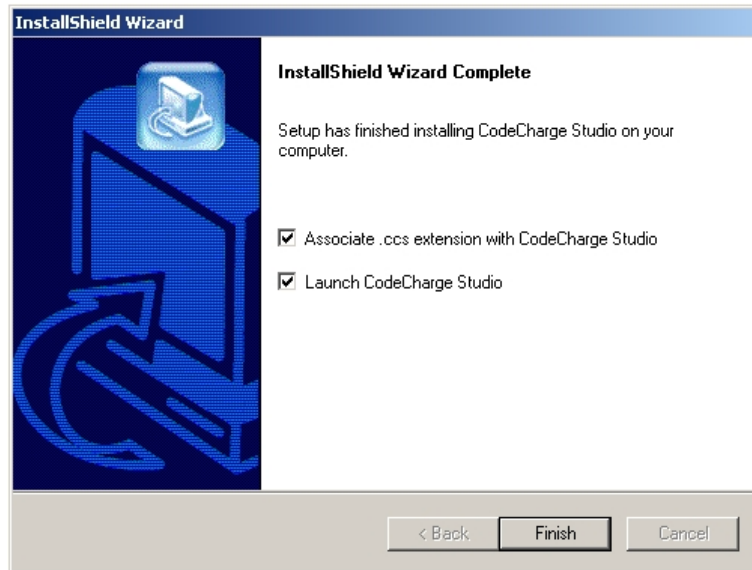
5. If you selected the Typical or Custom options in step 4 above, you are then presented with the following window that shows the components to be installed. You can then specify the components that you want installed.



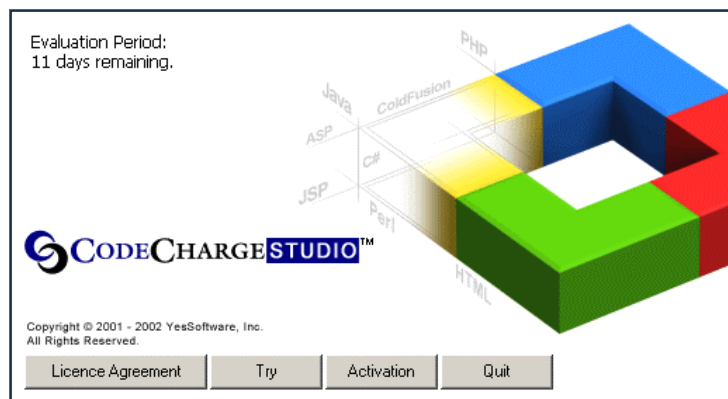
6. After all the necessary files have been installed and configured, you will then be presented with the following window to confirm the success of the installation. If you have both CodeCharge and CodeCharge Studio installed on the same machine, you have a choice of deciding the

default application to be used to open files with the (.ccs) extension. If only CodeCharge Studio is installed on the machine, you should associate the .ccs extension with CodeCharge Studio.

At this point, if you want to launch CodeCharge Studio, you can select the “Launch CodeCharge Studio” checkbox then click on the **Finish** button to complete the installation process.



If you installed CodeCharge Studio in trial mode, you have 30 days during which to evaluate the fully functional software. When you launch the software, a flash screen will show the number of days remaining for the evaluation. The screen will also have four buttons, one of which will be the **Try** button. Click on this button to start using the software in trial mode.



On the same flash screen, there will also be an **Activation** button that can be used to activate the software using a valid serial number. This can be done by entering the serial number directly into the activation window. Alternatively, you can use a browser to navigate to the web address indicated in the

activation window. Use your Serial Number and Installation Code to acquire an activation key then enter it into the Activation Key field.

CodeCharge Studio Activation

Information

Name

Joe Public

Company

Your Company, Inc.

Address

1000 Developers Way

City, State, Zip

San Francisco, CA, 00000

Country

USA

Email

joe@public.com

Where purchased (Store or Reseller + City/Country):

CodeCharge Website

Activation

Product: CodeCharge Studio    Version: 1. 0. 4. 0

Serial Number

000-000-000-0000

Installation Code

1F70-5E6F-EA83-3B79

☒ Auto-activation through the Internet

Obtain Activation key at  
<http://www.codecharge.com/activation>  
(have Your Serial Number and Installation Code ready)

Activation Key

OK

Cancel

### Upgrading, Repairing or Uninstalling CodeCharge Studio

In the event that a current CodeCharge Studio installation is corrupted or otherwise damaged, you can run the same setup file to modify or repair the installation or remove it entirely. To do so:

1. Double-click on the setup file to begin the process.
2. In the ensuing window, select whether you want to **Modify**, **Repair** or **Remove** the current CodeCharge Studio installation then click the **Next** button. If you selected the Repair option, please jump to step 4. For the Remove option, please jump to Step 5.


InstallShield Wizard

Welcome

Modify, repair, or remove the program.


Welcome to the CodeCharge Studio Setup Maintenance program. This program lets you modify the current installation. Click one of the options below.

☒ **M**odify




Select new program components to add or select currently installed components to remove.

☐ **R**epair



Reinstall all program components installed by the previous setup.

☐ **R**emove



Remove all installed components.

InstallShield

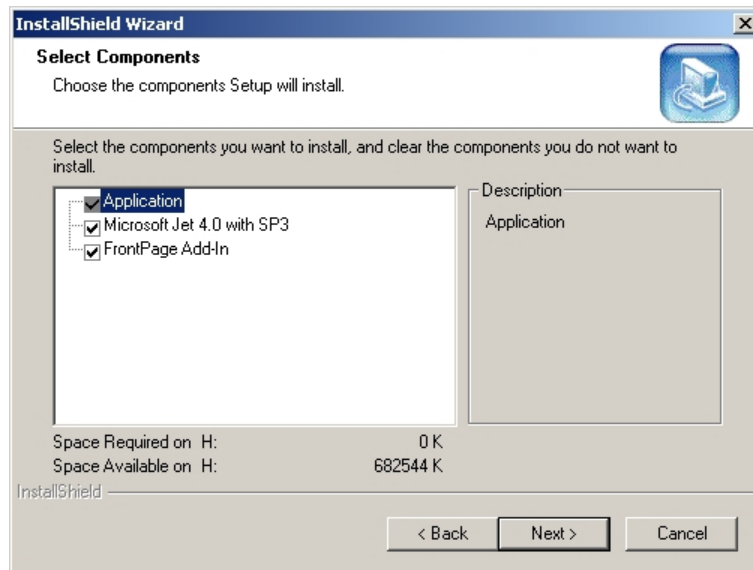
< Back

Next >

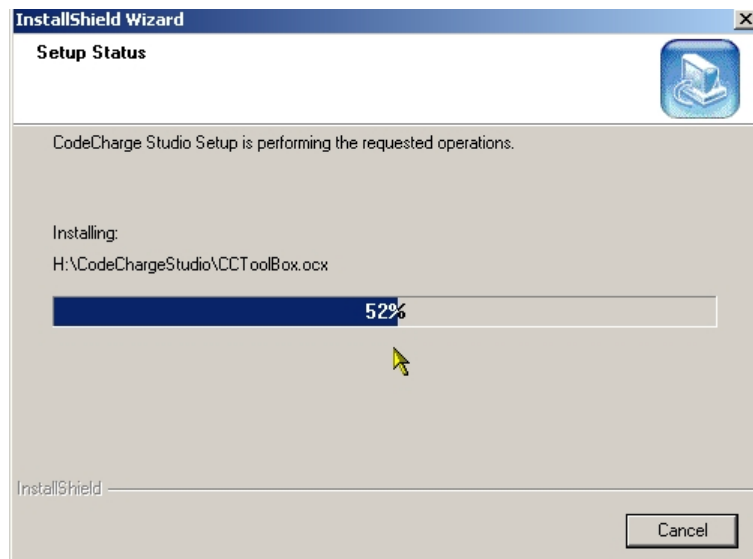
Cancel

12

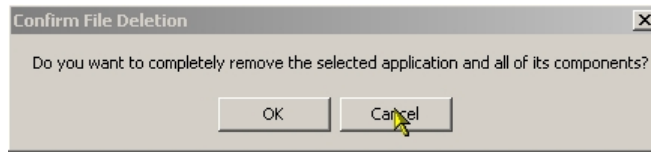
- For the Modify option, you will be presented with a list of components that you can install or remove. To install a currently unavailable component, activate its respective checkbox. Similarly, to uninstall a component that is already installed, uncheck its checkbox. After making your selections, click the Next button to proceed.



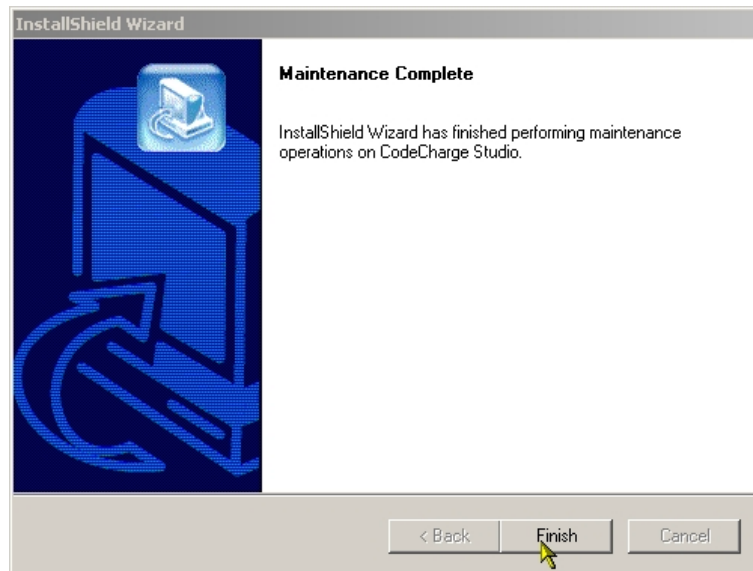
- If you selected the Repair option, the wizard will now proceed to check the current installation and repair any problems that may be diagnosed while a dialog window shows the progress of the repair operation.



5. If you selected the Remove option, you will be presented with a dialog window asking you to confirm whether you want to proceed with the removal. At this point, you can either cancel and return to the main installation window or proceed with the un-installation.



6. Finally, when all the operations have been completed successfully, a window appears to confirm the completion of the maintenance process. Click on the Finish button to close the dialog box.



## ***Setting up the work environment***

Having setup CodeCharge Studio, you are almost set to start generating and viewing dynamic database driven web pages. However, as you may be well aware by now, CodeCharge Studio works in concert with database and server software so these should be put in place before proceeding to use CodeCharge. While it is possible to generate web pages that aren't based on database content, such an approach is more the exception than the norm and even then, a server would be required to view the pages.

Considering that there are six different languages for generating pages and upwards of a dozen possible

databases, there are obviously many configurations that could be used. Each combination of language and database has its own unique merits. In the table below, we examine some of the more typical setups per language. Note that this is not to say that you cannot use setups beyond those stipulated here.

## ASP Setup

Operating System	<ul style="list-style-type: none"><li>• Windows 98</li><li>• Windows NT</li><li>• Windows 2000</li><li>• Windows XP</li></ul>
Web Server	The server must support ASP 3.0 [VBScript 5.0+] <ul style="list-style-type: none"><li>• Internet Information Server (IIS) 4.0 or 5.0</li><li>• Personal Web Server</li></ul>
Database Server	<ul style="list-style-type: none"><li>• Microsoft Access</li><li>• MS SQL Server</li><li>• Oracle</li><li>• MySQL</li><li>• PostgreSQL</li><li>• DB2</li><li>• FoxPro</li><li>• Excel</li><li>• Flatfile, CSV, textfile</li></ul>

## C# Setup

Operating System	<ul style="list-style-type: none"><li>• Windows 2000</li><li>• Windows XP</li><li>• Windows .Net Server</li></ul>
Web Server	Requires installation of the Microsoft .Net SDK 1.0.3705 or later. <ul style="list-style-type: none"><li>• Internet Information Server (IIS) 5.0</li></ul>

Database Server	<ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• MS SQL Server</li> <li>• Oracle</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• DB2</li> <li>• FoxPro</li> <li>• Excel</li> <li>• Flatfile, CSV, textfile</li> </ul>
-----------------	---

## Cold Fusion Setup

Operating System	<ul style="list-style-type: none"> <li>• Windows 98</li> <li>• Windows NT</li> <li>• Windows 2000</li> <li>• Windows XP</li> <li>• Unix/Linux</li> </ul>
Web Server	<p>All servers must have the Cold Fusion 4.0.1+ server installed.</p> <ul style="list-style-type: none"> <li>• Internet Information Server (IIS) 4.0 or 5.0 with Cold Fusion</li> <li>• Personal Web Server with Cold Fusion</li> <li>• Apache server with Cold Fusion</li> </ul>
Database Server	<p>When using Cold Fusion, the server connection string has to be made using ODBC.</p> <ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• MS SQL Server</li> <li>• Oracle</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• DB2</li> <li>• FoxPro</li> <li>• Excel</li> <li>• Flatfile, CSV, textfile</li> </ul>

## Java Servlets



Operating System	<ul style="list-style-type: none"> <li>• Windows 98</li> <li>• Windows NT</li> <li>• Windows 2000</li> <li>• Windows XP</li> <li>• Unix/Linux</li> </ul>
Web Server	<ul style="list-style-type: none"> <li>• Internet Information Server (IIS) 4.0 or 5.0</li> <li>• Personal Web Server</li> <li>• Apache</li> </ul> <p>In addition to the web server itself, you need to have a Java application server that supports Servlet API 2.2 and JDK 1.2 e.g.</p> <ul style="list-style-type: none"> <li>• Tomcat</li> <li>• IBM WebSphere</li> <li>• Macromedia JRun</li> </ul>
Database Server	<ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• MS SQL Server</li> <li>• Oracle</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• DB2</li> <li>• FoxPro</li> <li>• Excel</li> <li>• Flatfile, CSV, textfile</li> </ul>

## PHP Setup

Operating System	<ul style="list-style-type: none"> <li>• Windows 98</li> <li>• Windows NT</li> <li>• Windows 2000</li> <li>• Windows XP</li> <li>• Unix/Linux</li> </ul>
Web Server	<p>All the web servers have to have the PHP 4.0+ engine installed.</p> <ul style="list-style-type: none"> <li>• Internet Information Server (IIS) 4.0 or 5.0</li> <li>• Personal Web Server</li> <li>• Apache</li> </ul>

Database Server	<ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• MS SQL Server</li> <li>• Oracle</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• DB2</li> <li>• FoxPro</li> <li>• Excel</li> <li>• Flatfile, CSV, textfile</li> </ul>
-----------------	---

## Perl

Operating System	<ul style="list-style-type: none"> <li>• Windows 98</li> <li>• Windows NT</li> <li>• Windows 2000</li> <li>• Windows XP</li> <li>• Unix/Linux</li> </ul>
Web Server	<p>All servers must be Perl enabled with a version of Perl greater than or equal to 5.0. Some popular distributions of Perl include ActiveState Perl, Mod_Perl for Apache and Indigo Perl. Experienced users can also download and compile the Perl code.</p> <ul style="list-style-type: none"> <li>• Internet Information Server (IIS) 4.0 or 5.0</li> <li>• Personal Web Server</li> <li>• Apache server</li> </ul>
Database Server	<p>Perl requires installation of a suitable database interface (DBI) in order to connect to a database. This is a separate package that can be obtained from <a href="http://www.cpan.org">www.cpan.org</a>. e.g. For a MySQL database, the DBD:mysql is required.</p> <ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• MS SQL Server</li> <li>• Oracle</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• DB2</li> <li>• FoxPro</li> <li>• Excel</li> <li>• Flatfile, CSV, textfile</li> </ul>

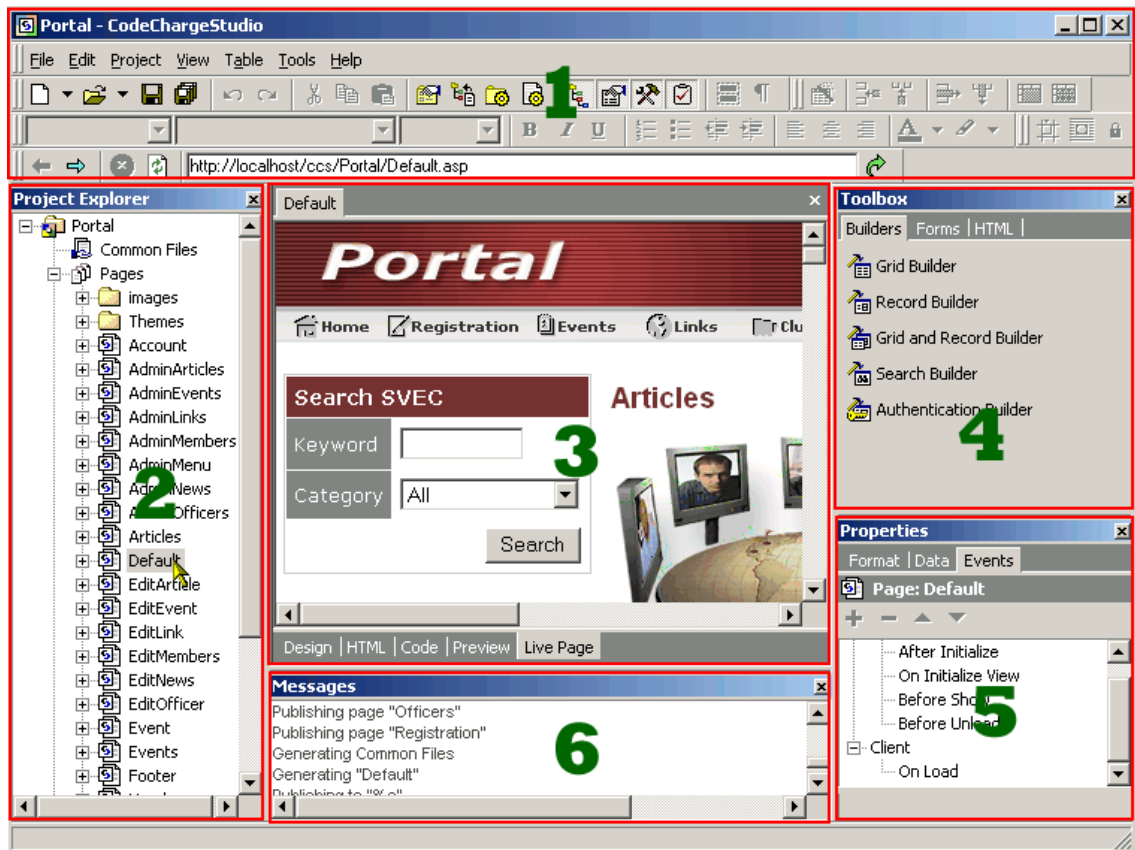
### III. Using CodeCharge Studio

#### *The CodeCharge Studio Workspace*

This section introduces the CodeCharge Studio workspace, illustrating the main areas of the interface and shows how you can customize the workspace to maximize productivity.

The workspace is the graphic user interface (GUI) with which you interact during the process of using CodeCharge Studio. The following are the main portions of the CodeCharge Studio workspace:

The illustration below shows the main parts of the CodeCharge Studio workspace.



1. **Command Area** – This is the top portion of the workspace that consists of the Title bar, Menu bar and various Toolbars used to provide quick access to different commands.
2. **Project Explorer** – This panel contains a list of the various elements that make up a project.
3. **Document window** – This is where most of the action goes on. It is here that project pages are opened, edited and viewed. The document window can either be in Design, HTML, Code, Preview or Live Page mode.

4. **Toolbox window** – This panel provides one-click access to various builders, forms and controls that can be used to add features to project documents.
5. **Properties window** – As the name suggests, the properties window displays relevant information pertaining to the currently selected document element. It can be used to adjust various properties depending on the currently selected element.
6. **Messages window** – This window provides feedback concerning processes that are executed such as publishing of a project.

Apart from the above major components, the workspace also has other minor components such as the Status bar and Tooltips.

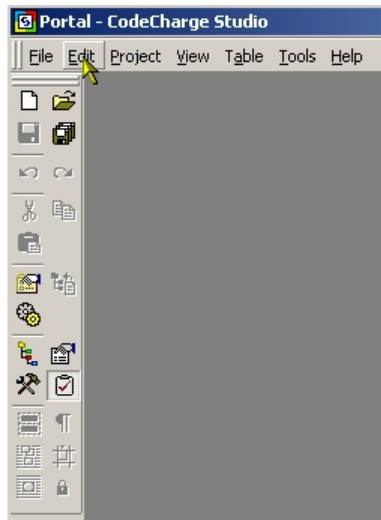
## The Command Area



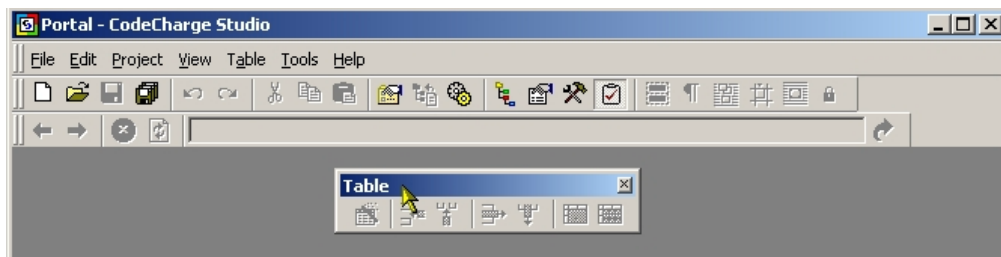
**Title Bar:** The title bar contains the name of the application (CodeCharge Studio) as well as the name of the currently open project. The right end of the title bar also contains buttons that can be used to minimize, maximize or close the application window.

**Menu Bar:** The menu bar features seven drop-down menus (File, Edit, Project, View, Table, Tools and Help). Each of these menus contains a series of options that carry out various functions when clicked. Notice that the main menu names as well as some of the sub-menus have an underlined letter. This is indication that that menu can be executed using a short-cut key combination by pressing the Ctrl key together with the underlined letter.












**Toolbar:** The toolbar provides quick access to many of the commonly used commands that are available in the menu bar. If you are concerned about screen space, you can use the View menu in the menu bar to hide some or all of the toolbars. Each of the toolbars can be dragged and docked on the top, left or right sides of the application window.











You can also drag the toolbars away from the sides of the application window and in so doing create a floating toolbar.


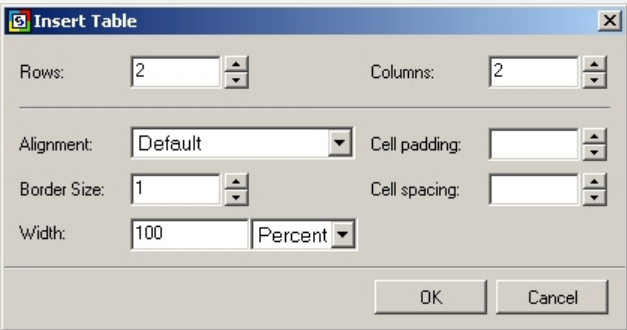








The following is a description of the commands available in the different toolbars.

Command	Description
<b>Standard</b>	
	Create a new Page.
	Open an existing Page.
	Save the current page.
	Save all changes made in multiple documents.
	Undo the last action.
	Redo the last action that was undone.
	Cut the current selection and place it in the clipboard.
	Copy the current selection to the clipboard.
	Paste the current contents of the clipboard.
	Open the Project Settings window.
	Synchronize code to reflect changes made.



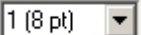
	Publish the project.
	Publish the current page.
	View the Project Explorer window.
	View the Properties window.
	View the Toolbox.
	View the Message window.
	Show borders.
	Reveal paragraph and new-line tags.












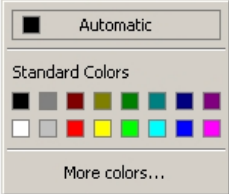

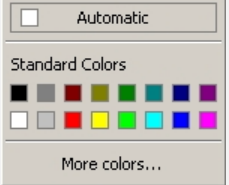







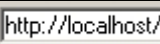

### Table

	<p>Open the Insert Table window</p>  <ul style="list-style-type: none"> <li>• Rows - Specify the number of rows for the new table.</li> <li>• Columns - Specify the number of columns for the new table.</li> <li>• Alignment - Select an alignment for the table [Default, Left, Center, Right]</li> <li>• Border Size - Specify the width of the table borders.</li> <li>• Cell Padding – Specify the amount of space between the border of the cell and its contents.</li> <li>• Cell Spacing – Specify the amount of space between adjacent cells.</li> <li>• Width – Specify the width of the table in pixels or as a percentage of the available space.</li> </ul>
---	--

	Insert a row above the current row.
	Insert a column before the current column.
	Delete the current row.
	Delete the current column
	Merge the currently selected cells.
	Slip the current cell.

### Formatting

	Select a format for the currently selected text.
	Select a font.
	Select a font size.

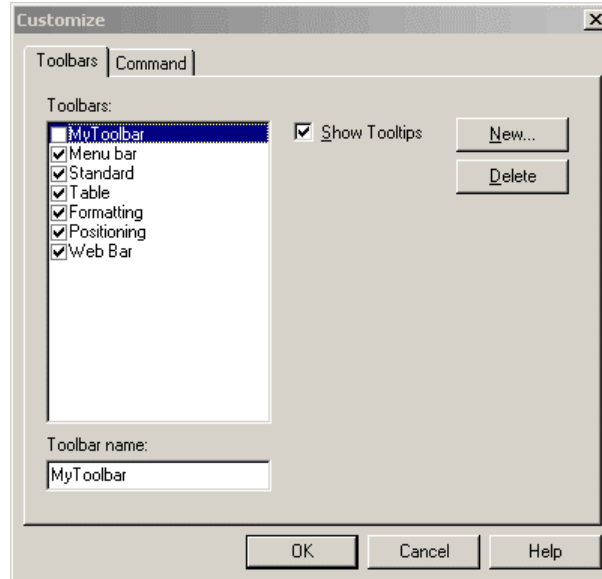
	Convert the selected text to bold typeface.
	Convert the selected text to italics typeface.
	Underline the selected text.
	Insert line numbers.
	Insert bullets.
	Decrease indentation.
	Increase indentation.
	Left align.
	Center align.
	Right align.
	<p>Select a font color for the current text.</p> 
	<p>Select a background color for the current font.</p> 
<b>Positioning</b>	
	Align the position of the control to the closest grid line.
	Assign the current control an absolute rather than relative position.
	Fix the position of an absolutely positioned control.
<b>Web Bar</b>	
	Navigate backwards.
	Navigate forwards.
	Stop the current operation.
	Refresh the current page.
	The URL of the current page.
	Go to the URL specified in the URL field.

Using the **Tools** → **Customize Toolbar...** menu option, you can configure the appearance of the various toolbars as well as add new toolbars.

- **Show Tooltips:** When this option is activated, informative Tooltips appear when the mouse

hovers over an option in the toolbar.

- **New...:** Clicking on this button adds a new toolbar to which you can later on add buttons using the Command tab.
- **Delete:** Use this button to delete a toolbar you created yourself. The six default toolbars cannot be deleted.
- **Reset:** The Reset button appears in place of the Delete button mentioned above when one of the six default toolbars is selected. The reset button has the effect of returning the toolbars to their factory appearance after they have been customized to look differently.
- **Toolbar name:** If you add a new toolbar, you can use this field to alter its name.

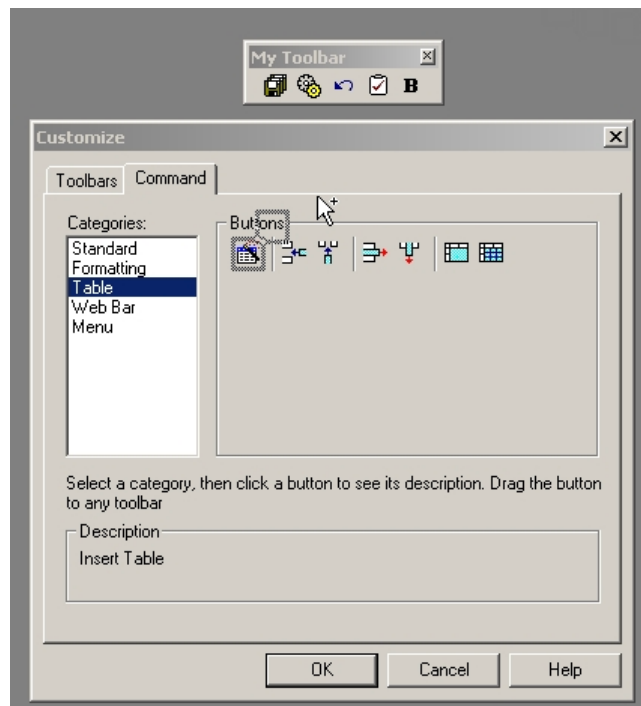


Under the Command tab, you can:

- View the various options available in a particular toolbar as well as see the tooltip text associated with the option.
- Add commands to a new or existing toolbar. To add a command to a toolbar, click and hold down the mouse button as you drag the command icon to the location where you want it to appear on the toolbar. Note that you can also remove command icons from a toolbar by clicking and dragging the icon away from the toolbar.

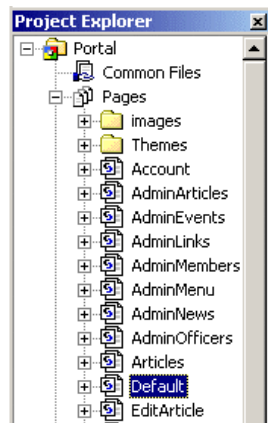
The illustration below shows the process of adding a command icon to a custom toolbar called My Toolbar.





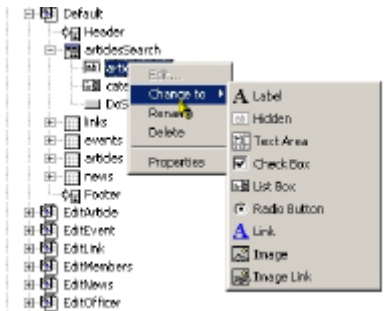
## The Project Explorer

The project explorer is essentially a hierarchical drop-down menu that provides access to the main components that make up a project. If the Project Explorer is not visible, you can use the **V**iew menu to make it visible. Like the toolbars and other windows, the Project Explorer window can be docked to the side of the application or against other windows such as the Toolbox. Alternatively, it can be a floating window.



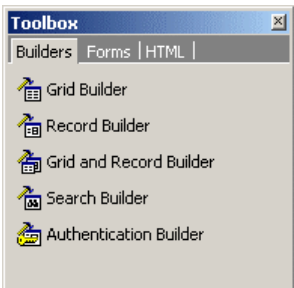
The Project Explorer window works in conjunction with the Document window and the Properties window. Double clicking on a page within the Project Explorer window will cause it to be opened in the Document window. At the same time, the Properties window displays the various properties for the currently selected

component. Using the Project Explorer, you can drill down to the basic elements of a project and execute various commands by right clicking on an element to bring up a popup menu.

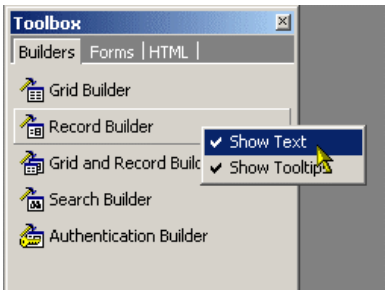


## The Toolbox

The Toolbox consists of three tabbed windows featuring a series of commands for adding various components to a page.



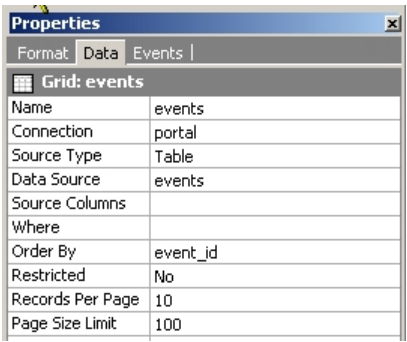
Depending on how familiar you are with the various options on the Toolbox you can opt whether or not to have it display text along with the Icons. You can also enable tooltips to be displayed when the mouse pointer hovers over a particular option. Right clicking within the Toolbox brings up a popup menu where you can toggle these features.



# The Properties Window

As mentioned earlier, the Properties window acts in conjunction with the Project Explorer as well as the Document window. The Properties window consists of three tabs (Format, Data, Events). You will notice that the contents of each of the tabs in the Properties window changes according to the currently selected element in the Project Explorer or Document window.

When using the Properties window, a caption that appears between the tabs and the list of properties displays the type and name of the currently selected element. For example, in the illustration below, a Grid form called events is the currently selected element.



It is advisable to make a habit of checking the caption before setting any property or event values so as not to end up making changes to the wrong element.

All the items in the Format tab and the Data tab display properties and events in a Name-Value format. The name of the property or event appears to the left of the window while its attendant value, if any, appears to the right. In some of the properties, drop-down lists are used to specify a series of options from which to pick a value while in others you can simply type in a value. Additionally some properties use popup dialogs of varying complexity and functionality to specify the property's value.

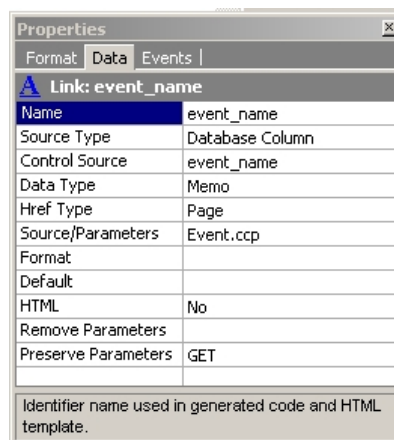
Unlike the other windows that we have seen so far, the Properties window features its own Status bar that is used to display helpful information about the currently selected property. For instance, in the illustration below, the status bar at the bottom of the Properties windows displays information about the height property of the <TR> tag.



**Format Tab:** The Format tab contains properties and events for various HTML/CSS elements. For the most part, the Format tab is mainly used when a page is opened in the Document window in HTML mode. In this case, selecting a HTML tag within the document will cause the Format tab to display the Properties and events applicable to that particular tag.

**Data Tab:** Whereas the Format tab displays HTML/CSS properties and events, the properties displayed under the Data Tab are implemented at the programming language level and can usually be controlled programmatically. As with the Format tab, the properties displayed under the Data tab vary depending on the currently selected control in the Project Explorer or Document window.

The following illustration shows the Data Tab of the Properties window displaying the properties for a Link control called event\_name.



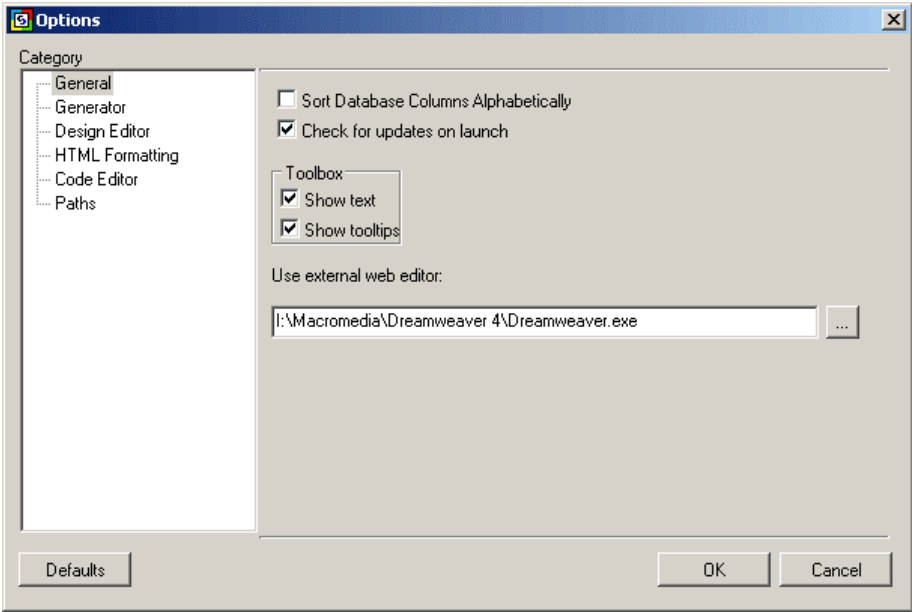
**Events Tab:** Events are designated points that occur during the process of code execution. Using the Events tab, it is possible to add Actions or custom code that would then be executed at the point in the code where the event occurs. Actions are predefined snippets of code that perform common functions such as validating a value such as a phone number or zip code or sending an email message. However, since there cannot be actions for all the possible operations that you may want to perform in an event, you can also enter your own

custom code into an event. Note that any code added into a Server event is language specific. If you change the language in which the project pages are generated, you have to write new event code in the new language. However, Client events are implemented using browser scripting languages (JavaScript, JScript) and as such are universally applicable regardless of programming language choice.

## Customizing the Workspace

By using the **Tools** → **Options...** menu, you can access additional configuration options that further customize the CodeCharge Studio workspace. The Options window has a list of six categories each with a set of configurable options.

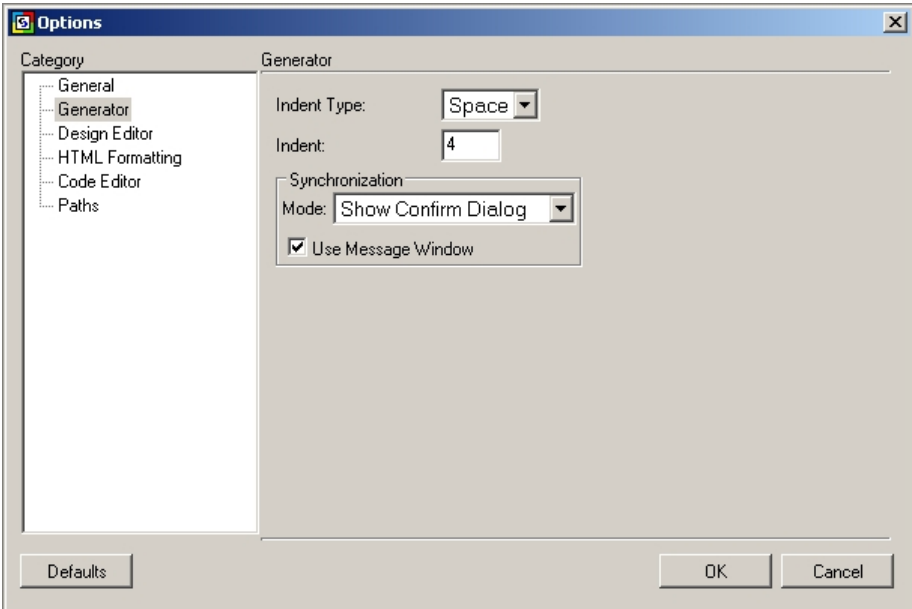
### GENERAL



Property	Description
Sort Database Columns Alphabetically	If selected, database fields listed in the CodeCharge Studio IDE during design time will be ordered alphabetically rather than in the order in which they appear in the database. For instance, when using the builders to create components, the fields will be shown alphabetically if this option is selected.
Check for updates on launch	If selected, CodeCharge Studio will connect to the internet and check for new software updates. If there are new updates, a window will open with the details of the update including where to obtain it.
Show text	If selected, caption text will appear next to each of the buttons in the Toolbox
Show tooltips	If selected, caption text will appear as a tooltip when the mouse pointer hovers over each of the buttons in the Toolbox.

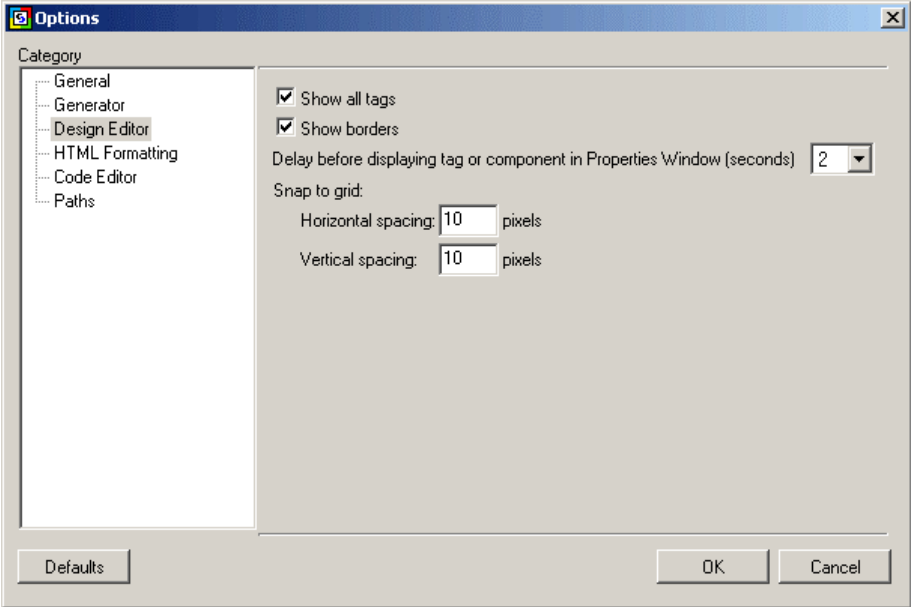
Use external web editor	In this field, you can specify the path to an external editor that you would like to use to edit the HTML pages.
-------------------------	--

GENERATOR



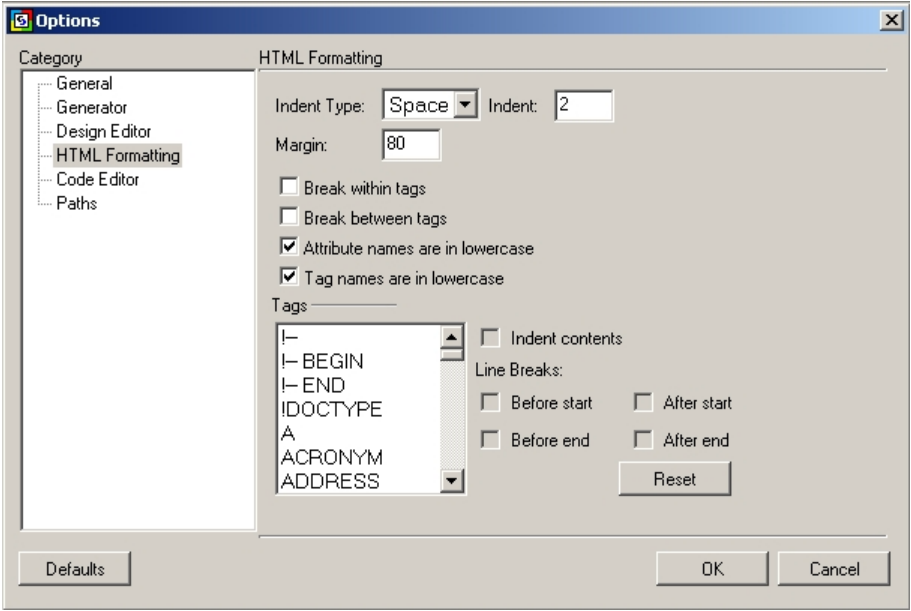
Property	Description
Indent Type	Select whether to uses spaces or tabs to indent the generated code
Indent	Specify the number of spaces to be used for indenting
Synchronization Mode	<p>During the process of creating a page, the user often deletes some components from the page and this is subsequently detected by CodeCharge Studio during the process of synchronization. This option allows you to specify the action to be taken when synchronizing page content:</p> <ul style="list-style-type: none"> <li>• Show Confirm Dialog – A dialog appears asking the user to confirm the deletion of components that were not found on the page.</li> <li>• Delete by default – Any components not found on the page are automatically deleted.</li> <li>• Don't Delete by Default – Components are not deleted from the page even if they are not found in the page code.</li> </ul>
Use Message Window	If selected, the Message window will also be used to display synchronization warnings.

# DESIGN EDITOR



Property	Description
Show All Tags:	If selected, Line Break and Paragraphs Breaks will be shown using icons.
Show Borders:	If selected, a border will be shown around all HTML tables in Design mode.
Delay before displaying tag or component in Properties Window (seconds):	Set the amount of time to wait before switching the Properties window to the currently selected control.
Horizontal spacing	While in design mode, the document window has an imaginary “snap to grid” that can be used to position elements on the page using the Positioning toolbar. Use this option to specify how far apart the grid lines should be horizontally.
Vertical spacing	Specify how far apart the lines of the snap to grid should be vertically.

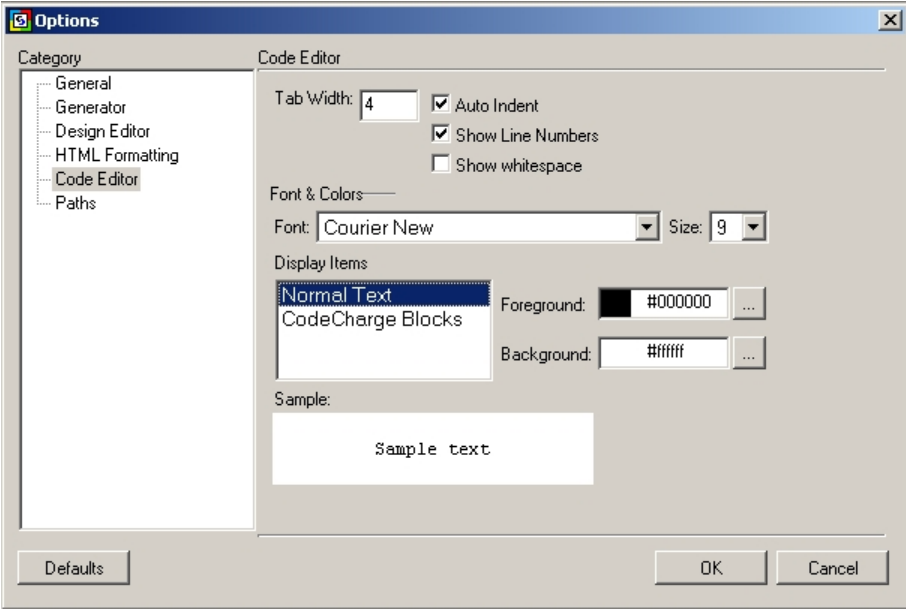
# HTML FORMATTING



Property	Description
Indent Type	Select whether to use space or tab indenting.
Indent	Specify the number of spaces to be used for indenting.
Margin:	The maximum number of characters that a string in a page can span.
Break within tags:	If a tag is longer than the Margin length, a line break will be inserted even if the tag is not closed.
Break between tags:	Line breaks for strings longer than the Margin length will only be inserted in between tags.
Attribute names are in lowercase:	Attribute names for tags will be in lowercase.
Tag names are in lowercase:	Tag names will be in lowercase.
<b>Tags</b>	
Indent Contents	Specify whether a specific tag will be indented.
<b>Line Breaks</b>	
Before Start:	Insert a Line break before the start of the tag.
After Start:	Insert a Line break after the start of the tag.
Before End:	Insert a Line break before the end of the tag.
After End:	Insert a Line break after the end of the tag.
Reset:	Revert to the factory settings.

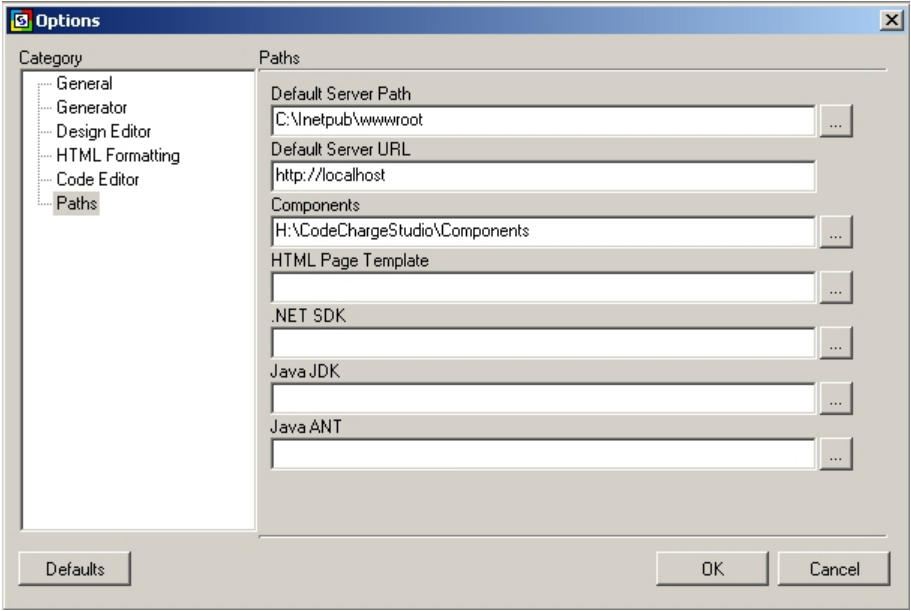


# CODE EDITOR



Property	Description
Tab Width	Specify the number of spaces to be used to represent a tab.
Auto Indent	If selected, code will be indented automatically.
Show Line Numbers	If selected, line numbers are shown on the left margin of the code editor
Show whitespace	If selected, white spaces are visually represented by periods (.) and tabs are represented using the symbol >>.
Font & Colors	
Font	Select the font in which the code will be displayed
Size	Select the size of the display font
Display Items	Select either Normal Text or CodeCharge Blocks to configure their display options
Foreground	Select the foreground color for the selected display item
Background	Select the background color for the selected display item.
Sample	This shows a preview of the display item according to the selections made

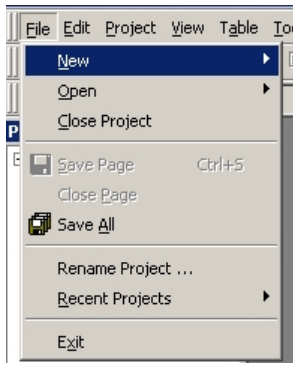
PATHS



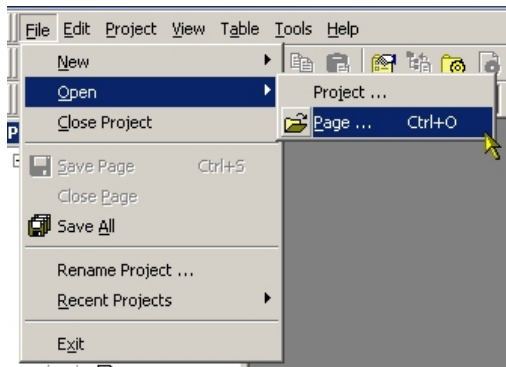
Property	Description
Default Server Path:	Specifies the default path to the web server where pages will be published.
Default Server URL:	Specifies the HTTP URL that maps to the default server path mentioned above.
Components:	Specifies the path where CodeCharge Studio components are located.
HTML Page Template:	Specifies the path to a HTML template page that will be used as the base for all newly created pages.
.Net SDK:	Specifies the location of the .Net framework SDK if available.
Java JDK:	Specifies the location of the Java JDK if available.
Java ANT:	Specifies the location of the Java ANT if available.

Keyboard Accessibility

Many of the commands available on the **Menu bar** can be accessed using keyboard keys rather than using the mouse. This is done by using combination keys that map to particular menu options. Looking at the **Menu bar** options, notice that each of them has a single letter underlined. Holding down the *Alt* keyboard key then typing the underlined letter has the same effect as clicking on the menu option concerned. For instance, typing the *Alt-F* keys will open the File menu shown below.



Notice that within the menu option, some of the sub-options also have underlined letters too. You can navigate further to these sub-options by simply pressing the underlined letter. For instance, within the File menu, you can open a page by pressing the O key followed by P.



Some of the sub-options under the Edit menu are unique in that they can be accessed directly without first opening the Edit menu option. These are commands that are very frequently used when working with documents and as such, they have dedicated access keys that are not only used within CodeCharge Studio but would also be found in most windows applications. The commands are as follows:

Command	Combination Keys
Cut	<i>Ctrl – X</i>
Copy	<i>Ctrl – C</i>
Paste	<i>Ctrl – P</i>
Select All	<i>Ctrl – A</i>
Find ...	<i>Ctrl – F</i>
Replace ...	<i>Ctrl – H</i>
Go to...	<i>Ctrl – G</i>

Committing these keys to memory could significantly improve the efficiency with which you edit

documents.

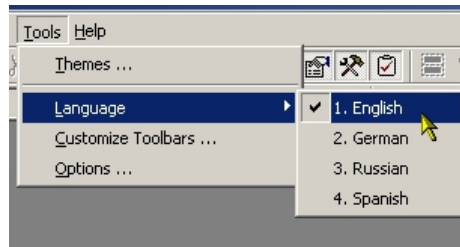
## Translation Files

It is possible to significantly localize CodeCharge Studio as well as the pages generated by using translation files. Within the CodeCharge Studio installation folder, the Components folder has a folder called Translations that contains two folders: IDE and Site

### IDE

The contents of the IDE folder are plain text files that contain the text that appears within the CodeCharge Studio IDE. Each line within the text files is a name–value pair representing an IDE component and its corresponding text. You can create files for different languages by duplicating an existing file and altering the value that appears to the right of the equals sign.

Once you create additional files, you can select the language to be used by using the Tools → Language menu option.

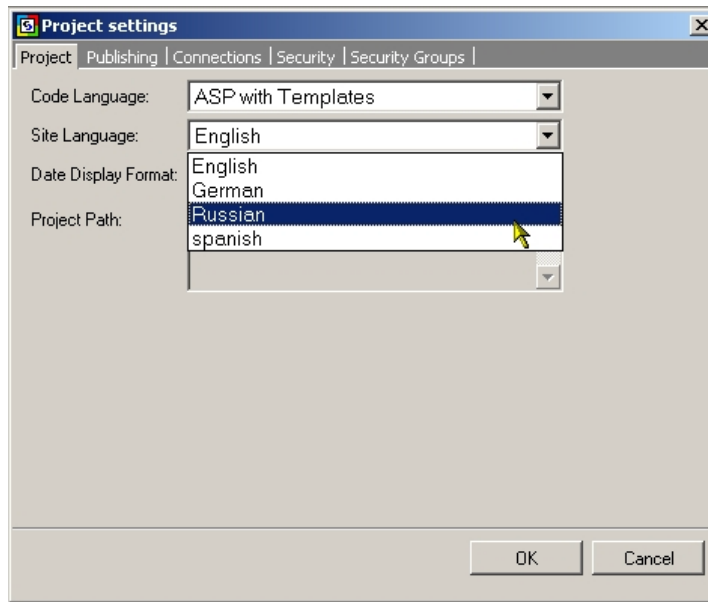


### SITE

In a similar manner to the IDE translation files, you can create translation files for the generated page. In this case, the translation files are in XML format but can similarly be edited using any text editor. The files contain a series of tags of the form:

```
<Msg name="SortBy" value="Sort by"/>
```

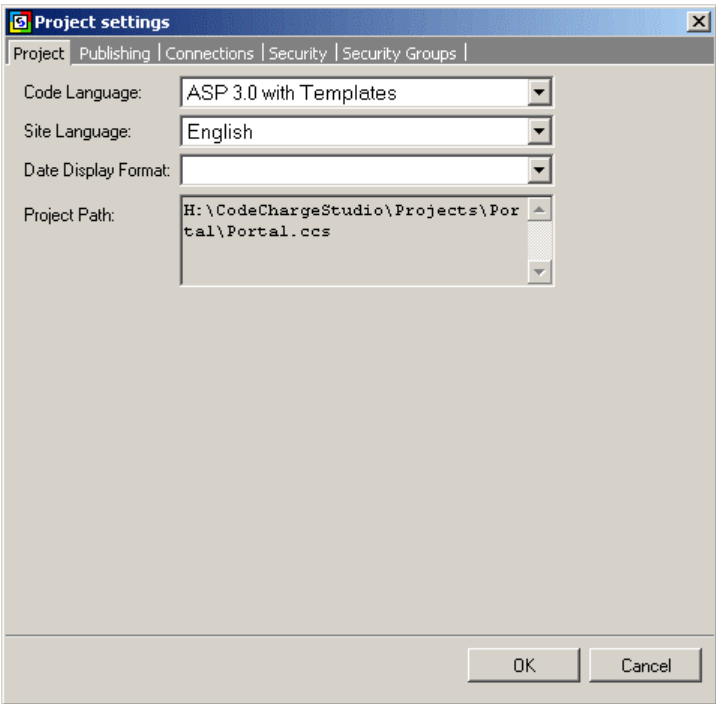
To accommodate a new language, simply duplicate an existing file and alter the *value* attribute of the *Msg* tag. Do not change the *name* attribute of the tag otherwise the tag will not serve its intended purpose. Save your changes to the file, giving it an appropriate name then re-start CodeCharge Studio. You will then be able to select the new site language from within the Project Settings window.



## ***Project Settings***

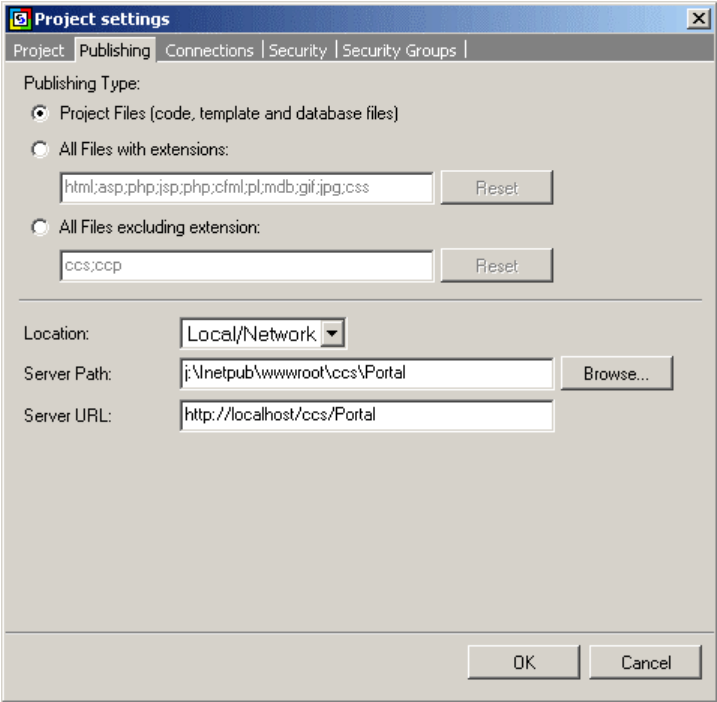
When working with a project, there are certain settings that apply globally to the project and these can be accessed using the **Project → Settings...** menu. The **Project Settings** window contains a series of five tabs under which the various settings are categorized.

# Project



Property	Description
Code Language	Select the programming language in which the code will be generated
Site Language	Select the spoken language to be used.
Date Display Format	Select the format in which you want date values to be displayed in the project. If no selection is made, date values will be displayed in the default database format.
Project Path	This is the path to the CodeCharge Studio file for the project.

# Publishing



Property	Description
Project Files	Select this option if you want all relevant project files to the published.
All Files with extensions:	Use this option to specify specific extensions to be used to determine which files to publish.
All Files excluding extensions:	Use this option to specify extensions whose corresponding files will not be published.
Location:	Specify whether the generated pages will be published to a local computer or network or whether they will be published to a remote computer via FTP.
Server Path:	The filing system path to where the pages will be published.
Server URL:	The HTTP URL that maps to the path specified above.

FTP	Location:	FTP
	Server:	
	Path/Folder:	
	Login:	
	Password:	
	Live site URL:	
	<input type="checkbox"/> Passive mode	
	<div>OKCancel</div>	

Server	The address (IP address of fully qualified domain name) of the FTP host machine.
Path/Folder	The location within the server where the pages will be uploaded.
Login	The login username required to access the FTP server.
Password	The password required to access the FTP server.
Live site URL	The HTTP URL that maps to the location where the files are uploaded.
Passive mode	Specify whether to use a passive connection. This can be helpful when the connection is made from behind a firewall. Also, use this option if you encounter difficulties maintaining a connection to the host.

Connections

Project settings

Project | Publishing | Connections | Security | Security Groups |

Connection name

portal

Add...

Modify...

Remove

Test

OK

Cancel

Under this tab is a list of all the database connections configured for the project. A project can have multiple database connections each connecting to a different type of database.



Property	Description
Connection name	The unique name used to identify a particular connection.
Add...	Click this button to initiate the process of adding a new connection.
Modify...	Select an existing connection then click this button to modify it.
Remove	Click this button to remove an existing connection.
Test	Click this button to test whether a connection is working properly.

## Security

Project settings

Project | Publishing | Connections | Security | Security Groups |

Security Method:

None

Table

Login Page:

Login.ccp

...

Connection:

internet

User Table:

users

User ID Field:

user\_id

Login Field:

user\_login

Password Field:

user\_password

Level/Group Field:

group\_id

Advanced...

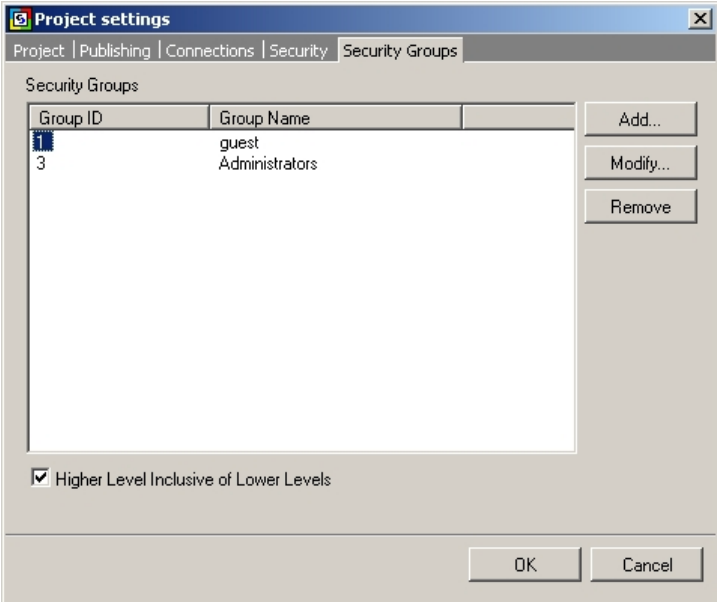
OK

Cancel

Property	Description
Security Method:	<ul style="list-style-type: none"> <li>None: No site security is implemented</li> <li>Table: Security will be based on information contained in a database table</li> </ul>
Login Page:	Select the project page that will be used to enter authentication details by the users.
Connection:	Select the name of the connection to be used for security information.
User Table:	Select the database table containing user security information.
User ID Field:	Select the database table field containing the unique ID that identifies the users.
Login Field:	Select the database table field containing the login usernames of the users.
Password Field:	Select the database table field containing the login passwords of the users.

Level/Groups Field:	If applicable, select the database table field containing the security levels of the users.
---------------------	---

Security Groups



Property	Description
Security Groups:	<ul style="list-style-type: none"> <li>Group ID: The unique value used to identify the group.</li> <li>Group Name: A descriptive name for the group.</li> </ul>
Add...	Click this button to add a new security group.
Modify...	Click this button to modify an existing security group.
Remove	Click this button to remove an existing security group.
Higher Level Inclusive of Lower Levels	If this option is selected, users with a higher security level will be able to access pages designated with a lower security level. Otherwise, an exact security level match will be needed in order to access any page.

## IV. Databases and Data Sources

One of the main features that makes CodeCharge Studio stand out from other web development tools is its ability to put database content on the web. More often than not, some interaction with databases will inevitably occur during the process of working with a CodeCharge Studio project.

As far as databases go, there is a wide range of applicable options each with its own pro's and con's. It would be inconceivable to accommodate all or the vast majority of the different options if each were unique. Thankfully, this is not the case since the Standardized Query Language (SQL) provides a uniform way of interacting with databases and the content contained therein. Broadly speaking, four main components make up the database experience:

1. The database software itself, otherwise known as archive or repository (e.g. Access, Oracle, MySQL, etc)
2. The driver/provider used to connect to the database. Essentially, this acts as a bridge between the database and the language being used. (e.g. ODBC, OLE-DB, Perl DBI)
3. The programming language being used. The programming language includes programming constructs that enable it to establish connections to the database and execute SQL commands against it. (e.g. ASP, PHP, Perl, etc)
4. The SQL language that is used to command the database. It is important to make the distinction that it is SQL and not the programming language that actually commands the database. The programming language is used to establish a connection to the database and then it sends SQL commands to be executed over the opened connection. The SQL transmitted is largely standard but different products do have proprietary SQL commands above and beyond the standard commands.

Various factors influence the choice of database used. These include but are not limited to:

- The amount of information being stored
- The speed and versatility required of the resultant application
- Security, backup and recovery concerns
- Price as well as operating cost of the database
- Compatibility with the intended programming language
- Availability of database drivers to connect to the database using the chosen programming language
- Availability of enhanced functionality such as stored procedures, procedural programming, replication, etc.
- Availability of a web host who supports the database

- Ease of use

Having selected the database to use and ensured that all the necessary components are in place, the next step is to make the connection within CodeCharge Studio. Note that you are not limited to a single database but can use multiple databases within the same project.

## Connections

Within CodeCharge Studio, a Connection is a link to a database for the purpose of creating pages based on the structure and contents of the database. The majority of the forms and field controls used within CodeCharge Studio draw their content from database fields.

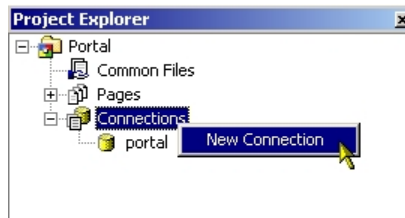
Any connection used within CodeCharge Studio has two sides to it. For designing pages within CodeCharge Studio, a **Design** Connection is used. This is the connection that CodeCharge Studio uses to communicate with the database. On the other side of this is the **Server-Side** connection that is used by the generated pages to communicate with the database.

Often times, the Design connection is the same as the Server-side connection. However, there arises occasions that warrant the use a Server connection that is different from the Design connection. This is usually the case when the generated pages have to be uploaded to a third party host machine where remote database access is not allowed or would be too slow.

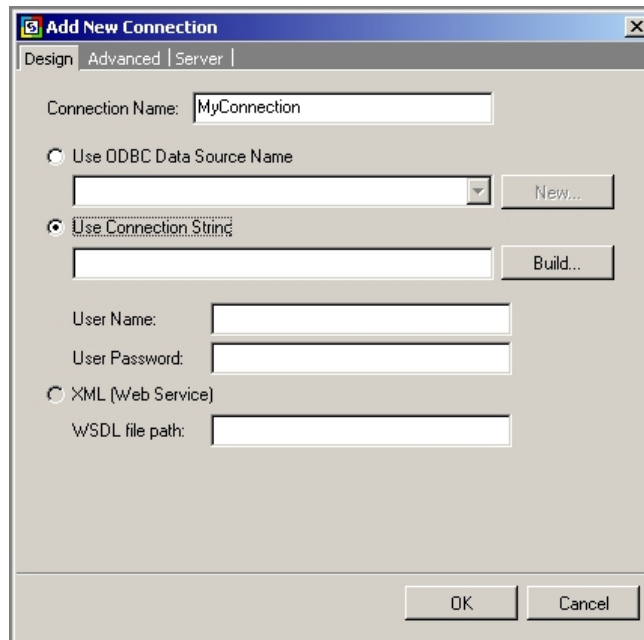
We shall now examine the process of creating connections within CodeCharge Studio.

## Creating Connections

There are two ways of creating a Connection to be used in a particular project. The first method involves using the **Connections** option within the Project Explorer window. Right-click on the **Connections** option and select **New Connection** as shown below.



You are then presented with the **Add New Connection** window where you specify the various properties of the Connection.



Notice that the **Add New Connection** window has three tabs whose functions are as follows:

- **Design:** This is used to configure the Design part of the connection as will be used within CodeCharge Studio.
- **Advanced:** This tab has advanced options that are supported by some databases.
- **Server:** This is used to configure the Server-side Connection if it is different from the Design connection.

## Design Connection Settings

The following are the configuration options available for the Design Connection.

**Connection Name:** This unique user selected name identifies the connection. No two connections within the same project can have identical names.

**Use ODBC Data Source Name:** This option should be selected if the connection to the database will be done via ODBC. Using the listbox field, you can then select one of the ODBC DSN's that is currently registered on the machine. Alternatively, you can [create a new ODBC DSN](#) by clicking on the **New...** button.

**Use Connection String:** This option is used to specify a connection string that contains the details necessary to make a connection to the database. Some of the details included in the connection string are:

- Database driver to be used
- Location of the database file

- Username and Password needed to connect to the database

If you are an experienced user, you can simply type in the proper connection string into the provided field. However, most people would find it necessary to click the **Build** button and use the **Data Link Properties** windows to [create the connection string](#).

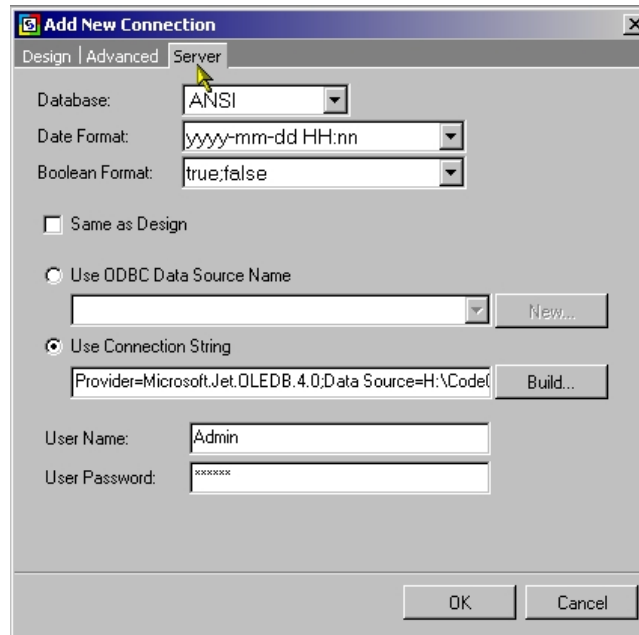
**User Name:** This is an optional field used to specify the username required to login to the database.

**User Password:** This is another optional field used to specify the password required to login to the database.

## Server Connection Settings

The **Server** tab of the **Add New Connection** window is used to specify additional properties for the database connection that is used in the generated pages.

In situations where the generated pages are deployed to a web server located on the development machine, it usually is the case that the Design connection is the same as the Server connection. This means that the connection used within CodeCharge Studio is the same as that used in by the generated pages. However, if the generated pages are to be deployed to a remote web server, it is usually necessary to configure a Server connection that is different from the Design connection.



The screenshot shows the 'Add New Connection' dialog box with the 'Server' tab selected. The 'Database' dropdown is set to 'ANSI'. The 'Date Format' dropdown is set to 'yyyy-mm-dd HH:nn'. The 'Boolean Format' dropdown is set to 'true:false'. The 'Same as Design' checkbox is unchecked. The 'Use ODBC Data Source Name' radio button is selected, and the 'New...' button is visible. The 'Use Connection String' radio button is also selected, and the 'Build...' button is visible. The 'User Name' field is set to 'Admin' and the 'User Password' field is masked with 'xxxxxx'. The 'OK' and 'Cancel' buttons are at the bottom.

The first three fields under the Server tab (**Database**, **Date Format**, and **Boolean Format**) apply regardless of whether the Server Connection is the same as the Design connection or not.

- **Database:** The selection made in this field influences the nature of the SQL used in the generated pages. The default option for this value is ANSI, which refers to the ANSI SQL92 standard. ANSI SQL is standard SQL that does not implement any database specific extensions or syntax. You should use this option if your database is not listed in the drop-down field. Otherwise, you can select your particular database from the drop-down list so that the SQL generated is specifically geared for that database.

Bear in mind that if a particular database is specified other than ANSI, the SQL generated would most likely not work if a different database was used. As such, if you would like the generated pages to be able to function if used in conjunction with different databases, you should select the ANSI SQL 92 option.

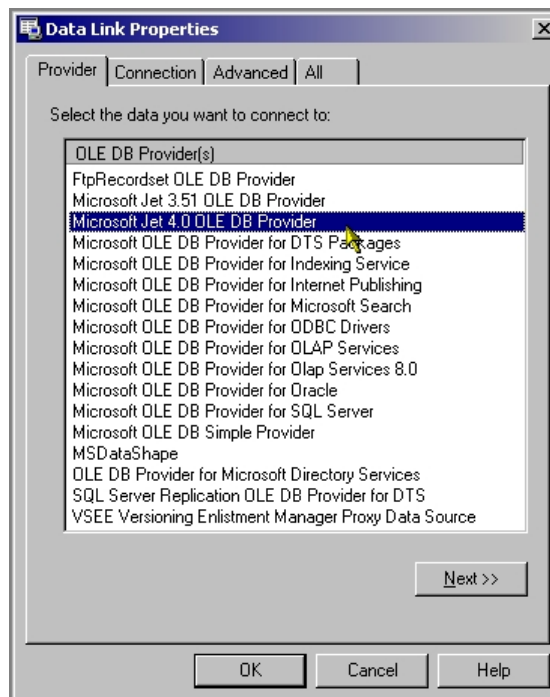
- **Date Format:** Specify the format to be used for date values that are entered into the database. For instance, if you specify yyyy-mm-dd: HH:nn a date such as 11-29-2001 would be entered into the database as 2001-11-29: 00:00.
- **Boolean Format:** Specify the format in which Boolean values will be entered into the database.

If the **Same as Design** checkbox is not activated, you have to specify an ODBC DSN or Connection string to be used for the Server Connection. The process of doing so is essentially similar to that used for the Design connection except that the database referenced should be the one that will be used on the deployment server by the generated pages.

## ***Creating a Connection String (DNS-Less Connection)***

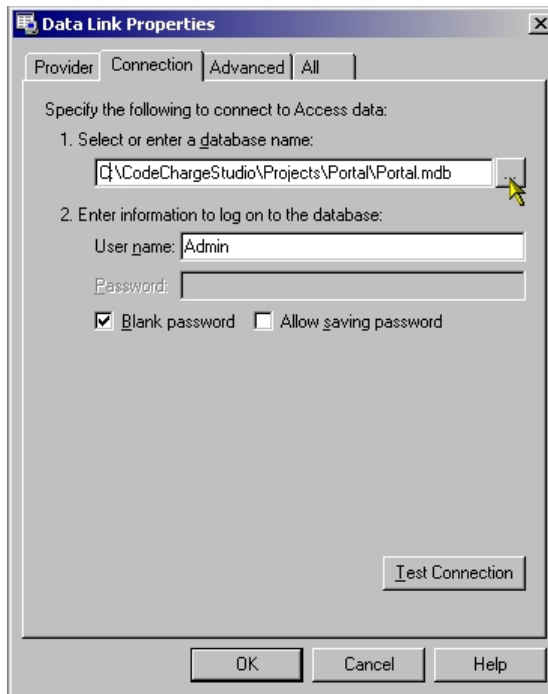
An alternative to using a DSN connection is using a DNS-Less connection that is in essence a connection string containing the details necessary to establish the connection to the database. The makeup of a connection string varies depending on the database being used as well as the provider/driver. While it is possible to simply type out a proper connection string, the better alternative would be to use the **Data Link Properties** window to construct one for you.

The **Data Link Properties** window can be accessed from the **Add New Connection** or **Modify Connection** windows by clicking on the **Build** button.

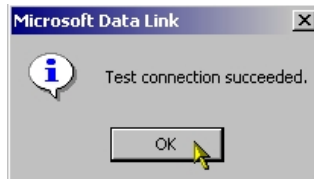


1. The first tab in the Data Link Properties window is the **Provider** tab that is used to specify the database provider/driver to be used. Under this tab will be listed all the database providers currently installed on the machine. Select the appropriate provider for the database you want to connect to then click on the **Next** button to move to the **Connection** tab. In the illustration above, the Microsoft Jet 4.0 OLE DB Provider has been selected in order to create a connection string to an Access database.
2. The fields displayed in the Connection tab will vary depending on the Provider selected in the Provider tab. In the case where the Microsoft Jet 4.0 OLE DB Provider was selected, the Connection tab will look like the illustration below. In this case, it is necessary to enter a path to the Access database file in the **Select or enter a database name** field. If the Access database file is password secured, you would also need to enter the necessary authentication details in the **User name** and **Password** fields. Otherwise, as is most frequently the case, the **Blank password** checkbox is activated to indicate that no authentication is required to access the database.





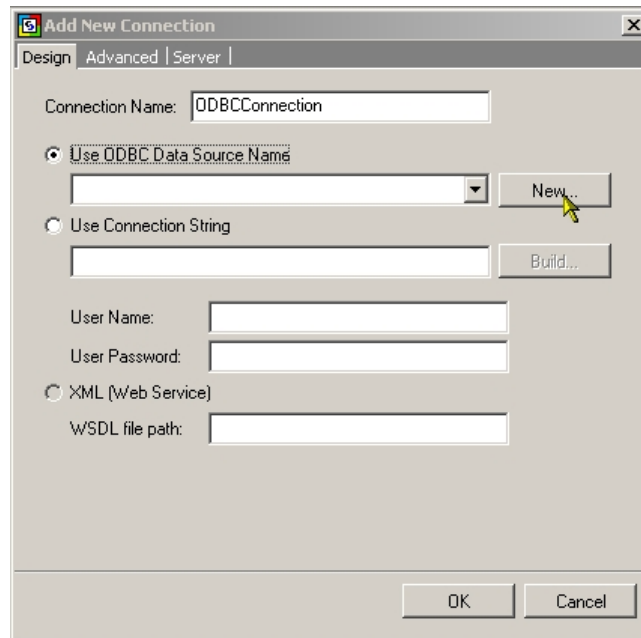
3. After entering the necessary details, it is a good idea to click on the Test Connection button to make sure that the configuration works correctly. If all is well, a dialog box appears to confirm the soundness of the connection otherwise, an error message is displayed depending on what the problem is.



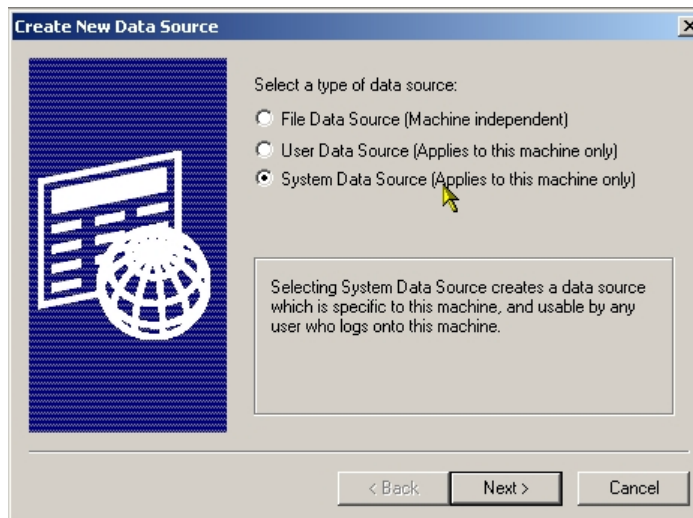
## ***Creating an ODBC Data Source Name (DSN)***

If you need to create a New ODBC DSN, you can do so from either within CodeCharge Studio or by using the **Data Sources (ODBC)** option available in the system Control Panel. In either case, the steps involved are essentially identical so we shall demonstrate how the process is accomplished from within CodeCharge Studio.

The first step is to open the **Add New Connection** window. This can be done by right clicking on the **Connections** option in the Project Explorer window and selecting **New Connection**.



Select the **Use ODBC Data Source Name** radio button to indicate your intention to use an ODBC DSN then click on the **New...** button to begin the process of creating a new DSN. You then have to select the type of DSN to create. There are three choices to select from as described below:



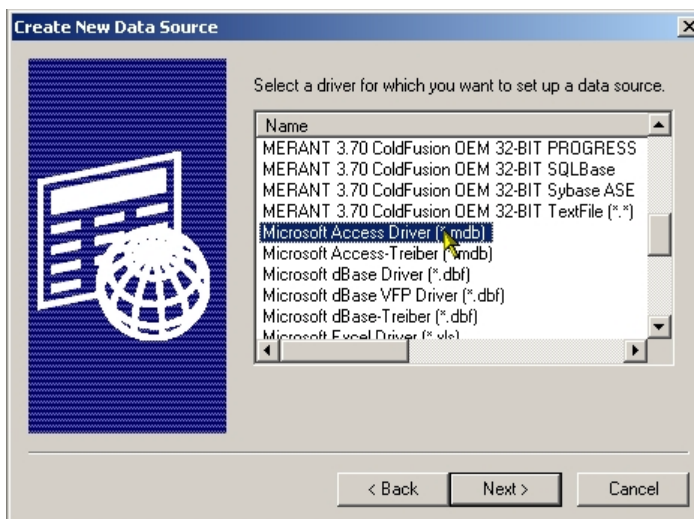
- **File DSN:** This is a data source that can be shared among all users who have the same drivers installed. A file DSN need not be user-dedicated or local to a computer
- **User DSN:** A User DSN is local to a computer and accessible only by the current user.

- **System DSN:** This is a DSN that is local to a computer but not user-dedicated; any user with privileges can access a System DSN.

Unless you have a specific reason for doing otherwise, you should opt to create a System DSN that will be accessible to all users including the account under which the web server process runs.

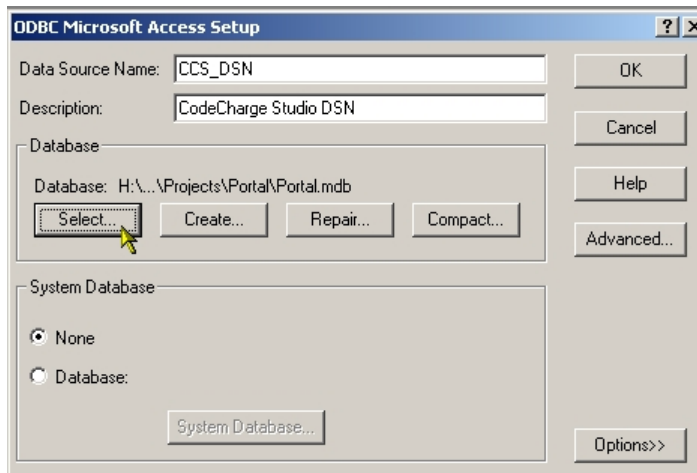
Once you have selected the DSN type, click on the **Next** button to proceed to the next step.

You are then presented with a list of ODBC drivers that are currently installed on the Local machine. Scroll through the list to locate the driver for the database you intend to use then select it.

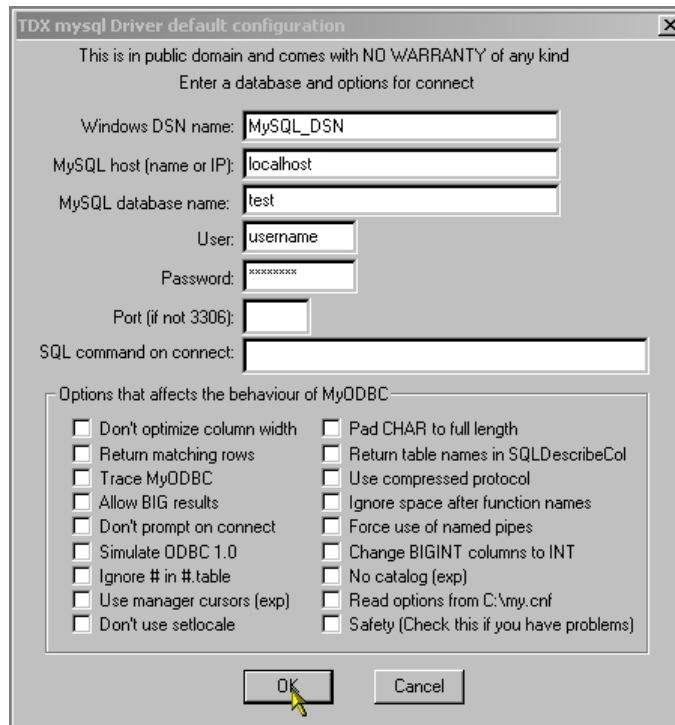


Click **Next** to advance to the next step, then in the ensuing window, click on the **Finish** button to create the data source.

Depending on the driver you selected, you may or may not be required to enter additional information. For an Access database driver, you are presented with the window below to specify the name of the DSN, an optional description as well as locate the Access file to be associated with the DSN.



Alternatively, if you opted to create a DSN for a MySQL database, you would be presented with a window similar to that shown below. Notice that in this case, instead of specifying a file for the database, you are required to identify the host machine on which the MySQL server resides. You also have to enter a username and password to be used to connect to the database since this is mandatory in the case of a MySQL database.



Creating an ODBC DSN for other databases such as MSSQL, Oracle, Excel, etc is inherently similar to the above steps except for the last step that varies depending on the database driver. However, once you have

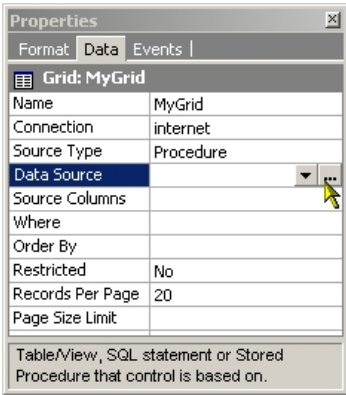
created a DSN, you can use and re-use it in multiple projects by simply selecting it from among the list of available DSN's.

## Form Data Sources

In CodeCharge Studio, dynamic data from databases is displayed using fields that are in turn contained within different types of forms. The data that could be displayed in a form field depends on the data source configured for the form itself. Put in another way, a forms data sources determines the selection of database fields that a field in that form could possibly display.

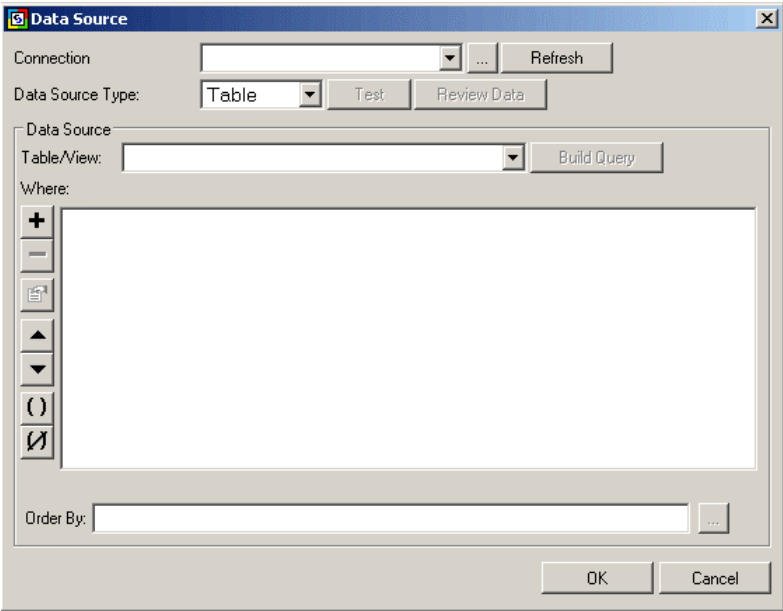
In this section, we shall examine how form data sources are configured. The main emphasis here is on how to configure a form data source from scratch as opposed to using a [form builder](#) to automate the process of building the form as well as its data source.

A form data source is configured using the **Data Source** window. The **Data Source** window can be accessed using the Data Source property of a form in the **Data** tab of the **Properties** window.

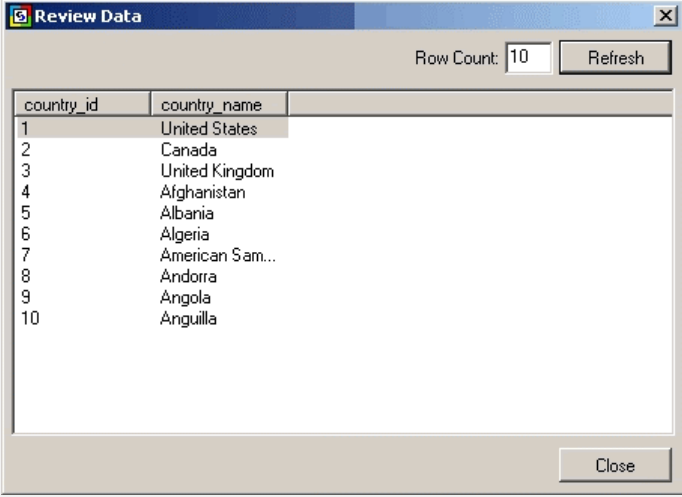
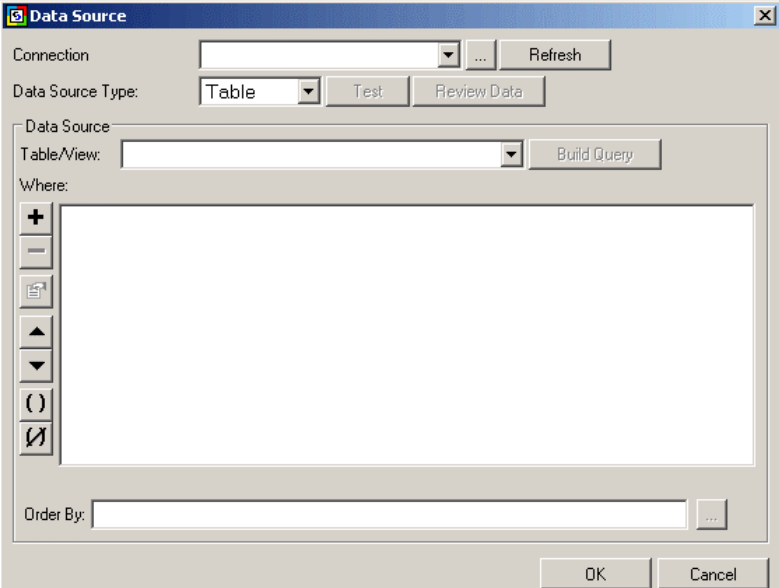


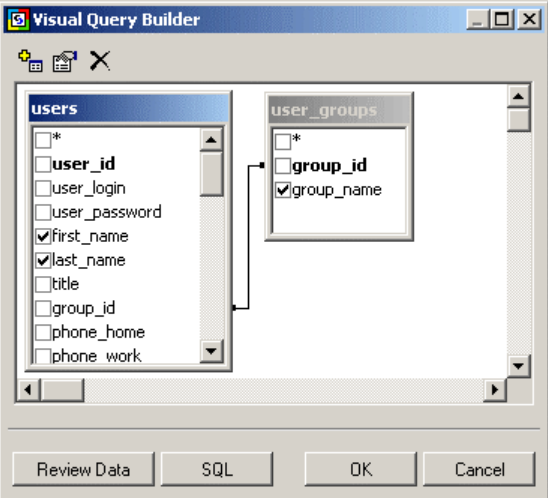
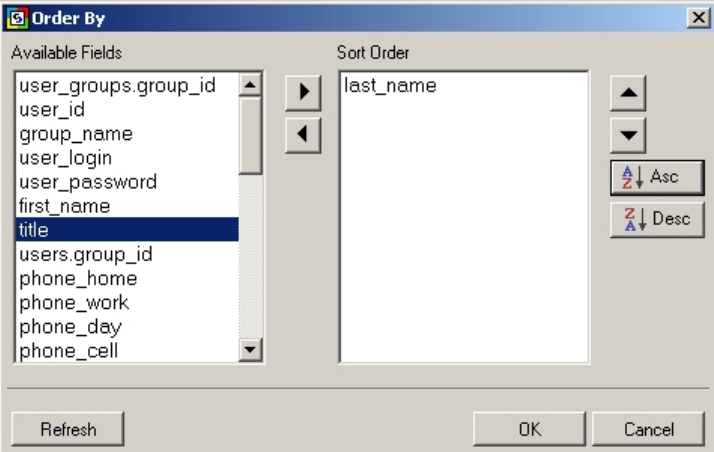
In addition to facilitating the configuration of a data source for a form, the **DataSource** window is also used to setup input parameters. Input parameters are values that are received by the form and are used in the process of filtering the data that is displayed in the form. Input parameters can be received from other forms, via the URL, cookies or session variables. For instance, a search form working in conjunction with a grid form makes uses of input parameters. The search form submits the input parameters and the grid form receives then and uses them to filter the records that are retrieved and displayed on the form.

The options available in the data source window are as follows:


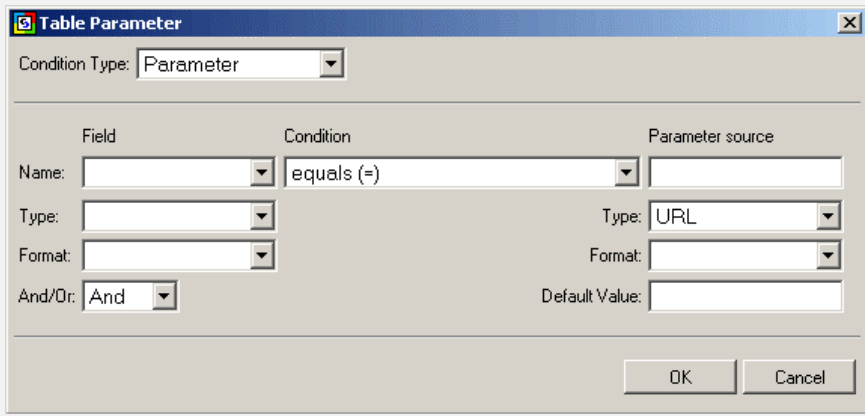








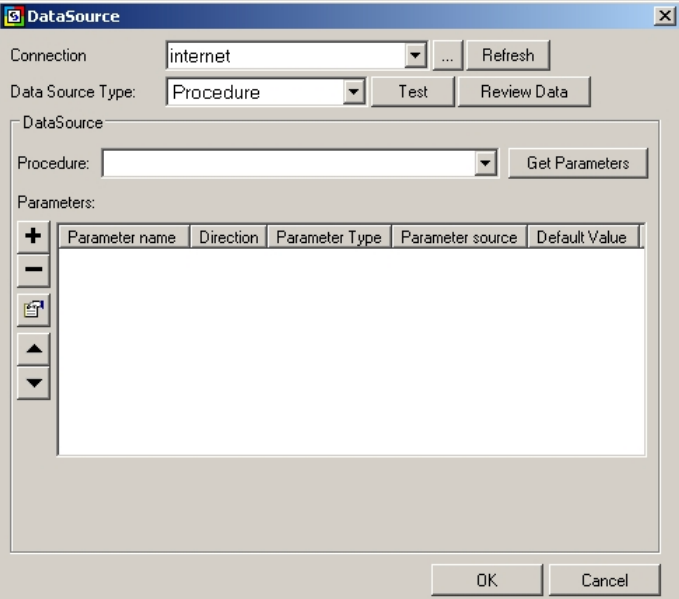

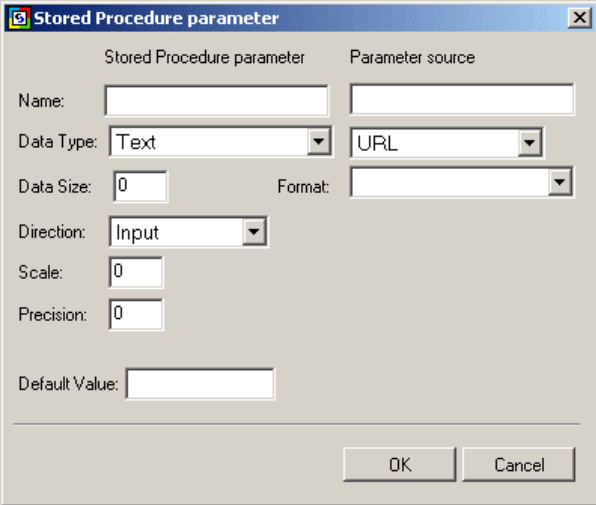


Property	Description
Connection	The name of the connection where the data will be retrieved.
Refresh	If you make changes to the database during the process of using the DataSource window, click on this button to update the window and reflect the current database status.
Data Source Type	Select the type of data source to be used to obtain the data [Table, Procedure, SQL].
Test	This button is only functional when the data source type is a procedure or SQL query. Click this button to execute the procedure or SQL query and confirm its validity.



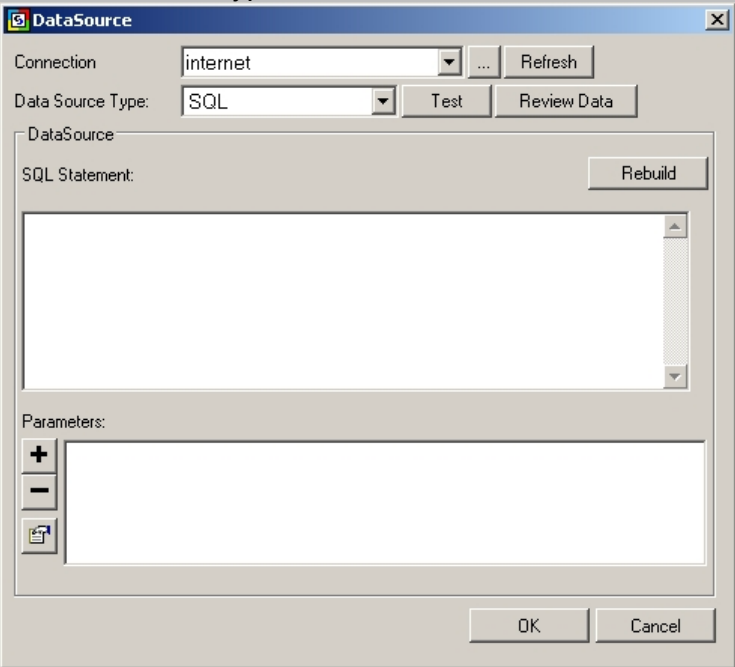

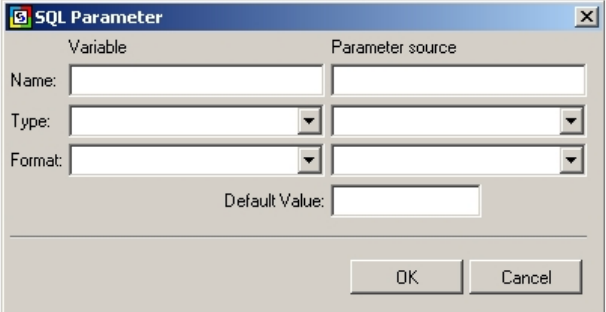


Review Data	<p>Depending on the data source type selected, clicking this button will retrieve and display a sample of the database content.</p>  <p>You can specify the number of records returned by entering a value in the Row Count field then clicking on the Refresh button to retrieve the records.</p>
Data Source	<p>The layout and options in the Data Source portion of the window vary depending on the value of the Data Source Type field. We shall first examine the case where a Table is being used as the data source.</p> 
Table/View	Select the database table or view from where the data will be retrieved.

Build Query	<p>If the form requires data that resides in two or more tables, it is necessary to construct a SQL statement that picks the required fields from the concerned tables. Clicking on this button will open the <a href="#">Visual Query Builder</a> window shown below where you can construct the SQL query with joins to different tables.</p>  <p>For more information on how to use the <b>Visual Query Builder</b> window, please refer to the section entitled <a href="#">Using the Visual Query Builder</a> window.</p>
Order By	<p>Use the adjacent button to select fields by which the data should be ordered. You can specify whether the field should be sorted in ascending or descending order.</p> 
Where	<p>The options in this portion of the window are used to construct the where clause of the SQL statement.</p>



	<p>Click this button to add a parameter to the where clause. Please refer to the section on <a href="#">Using the Table Parameter window</a> for more information.</p> <div data-bbox="399 164 1256 581">  </div>
	<p>Click this button to remove a parameter from the where clause.</p>
	<p>Click this button to view the configuration of an existing where clause parameter.</p>
	<p>Click this button to move an existing parameter upwards in the order or appearance.</p>
	<p>Click this button to move an existing parameter downwards in the order or appearance.</p>
	<p>Click this button to enclose a parameter definition within brackets. You can group together multiple parameters by holding down the <i>ctrl</i> key while clicking on them then click on the button to place a bracket around all the selected parameters.</p>
	<p>Click this button to remove existing brackets around a parameter definition or multiple parameters.</p>

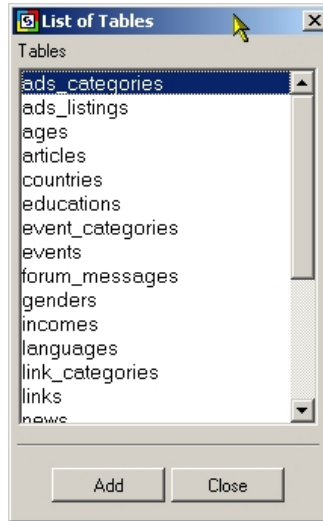
Data Source	<p>We now turn to the options available in the <b>Data Source</b> window when <i>Procedure</i> is selected in the Data Source Type Field.</p> 
Procedure	Select the stored procedure to be used as the data source.
Get Parameters	Click this button to retrieve the stored procedure parameters from the database.
	Click this button to add a stored procedure parameter. Please refer to the section on <a href="#">Using the Stored Procedure Parameter window</a> for more information.
	
	Click this button to remove an existing stored procedure parameter.
	Click this button to view the configuration of an existing stored procedure parameter.

	Click this button to move a stored procedure parameter upwards in the order of appearance.
	Click this button to move a stored procedure parameter downwards in the order of appearance.
Data Source	<p>Finally, the following are the options available in the <b>Data Source</b> window when SQL is selected in the Data Source Type Field.</p> 
SQL Statement	The SQL statement to be used as the data source for the form.
Rebuild	Use this button to reset an SQL query to a previous state. For instance, if you have a query them make changes to it, you can use this button to return to the original query.
	Click this button to add a SQL parameter. Please refer to the section on <a href="#">Using the SQL Parameter Window</a> for more information.
	
	Click this button to remove an existing SQL parameter.
	Click this button to view the configuration of an existing SQL parameter.

## Using the Visual Query Builder Window

When using the builders to create a grid form or during the process of configuring the Data Source property for a grid form, you might need to use the **Visual Query Builder** window to create a SQL query based on multiple tables. The options available in this window are designed to be relatively intuitive and easy to use compared to the task of writing a join SQL query from scratch. This section explains how to use the **Visual Query Builder** window to build SQL statements without the hassle.

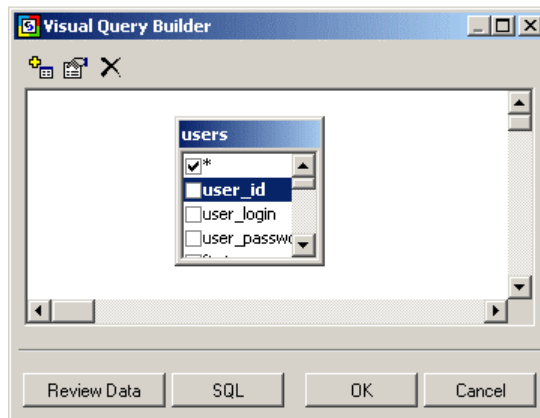
When the **Visual Query Builder** window first appears, another smaller window also appears that contains a list of the database tables available in the current connection.



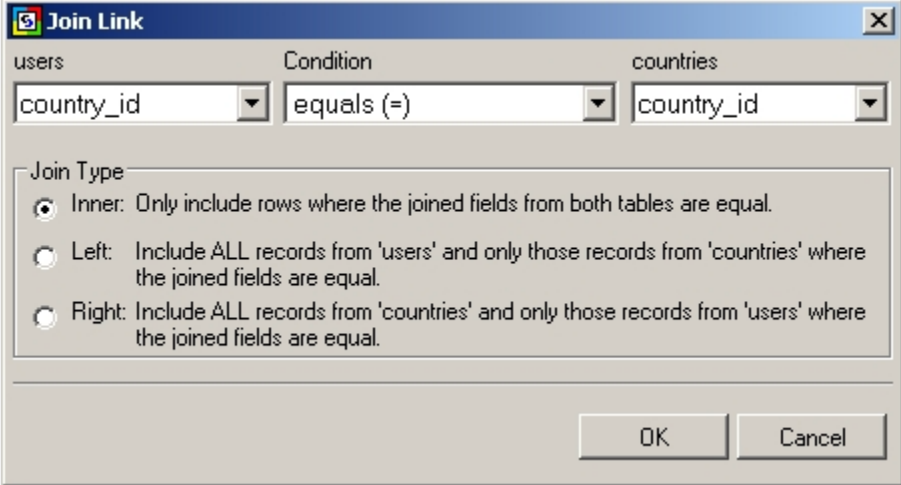
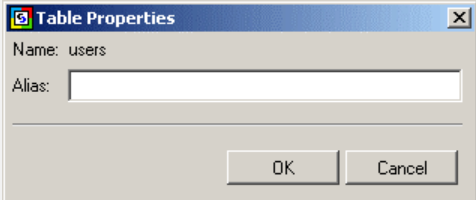



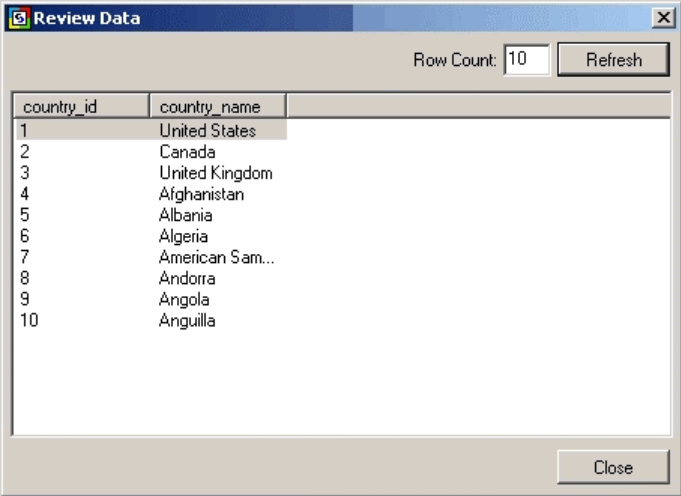
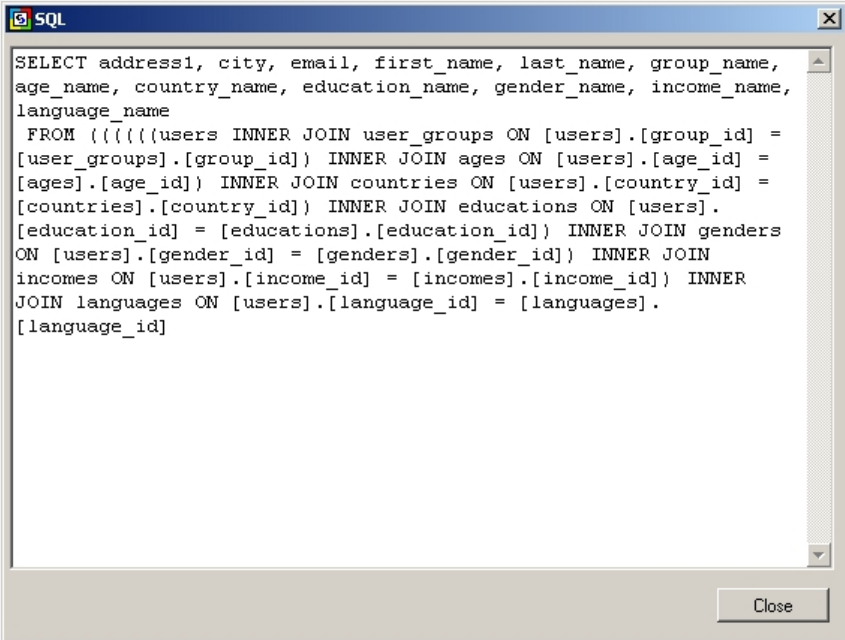
From this list of tables, you should select the tables you want to use in building the SQL query and click on the **Add** button. You can select multiple tables by holding down the *ctrl* key while clicking on the table names. Once done, click on the **Close** button to close the window.

The tables you selected appear within the **Visual Query Builder** window, each in a small window containing a list of the fields in the table. Depending on how the fields in the different tables are named, you might notice that some tables are joined together by lines. These lines represent relationships between the tables. The **Visual Query Builder** window automatically creates these lines if it encounters two tables with fields that are potential primary and foreign key fields based on their name and data type.

There are also a number of other buttons in the **Visual Query Builder** window whose functions are described below:

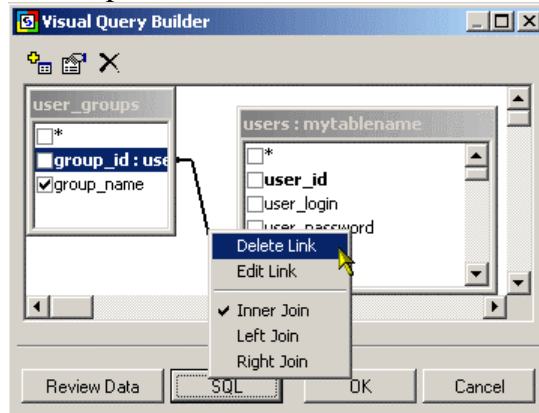


Property	Description
	<p>This buttons displays the list of tables available in the connection in case you need to add a table.</p>
	<p>Click this button to display the <b>Join Link</b> or <b>Table Properties</b> windows. The Join Link window contains the properties of a join. Joins are the lines that appear between two tables to represent a relationship between the tables. Click on a join line then click on this button to display the Join Link window.</p> <div data-bbox="415 812 1308 1298">  </div> <p>If you click on the properties button when a table is selected, the <b>Table Properties</b> window appears and you can enter an alias name for the table.</p> <div data-bbox="625 1411 1096 1611">  </div>

	With a table or join selected, click this button to delete the table or join.
Review Data	<p>Clicking this button displays the <b>Review Data</b> window that shows sample data based on the SQL query configured in the <b>Visual Query Builder</b> window. You can determine the amount of data displayed by entering a value in the Row Count field then clicking on the Refresh button.</p> 
SQL	<p>This button displays the SQL statement corresponding to the table and join setup in the <b>Visual Query Builder</b> window.</p> 
Ok	Click this button to save the current configuration and exist the window.
Cancel	Click this button to close the window and discard any changes.

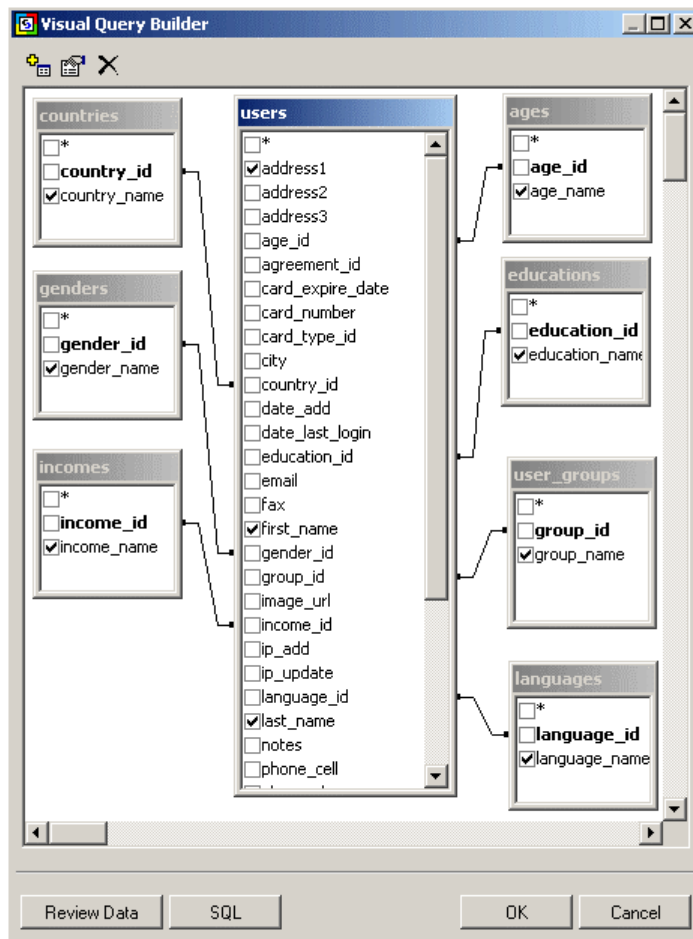
After adding tables to the **Visual Query Builder** window, you then proceed to create join links between the different tables.

- Some join links are automatically created based on matching field names and data type. If these relationships are wrong or you don't need them, you can delete them by right clicking on them and selecting the Delete Link option.



- You can create new join links by dragging a field from one table to another field in a different table. Once the join link appears between the tables, you can use the properties button to confirm the joined fields as well as set the join type and the operator used for the join criteria [e.g. equals (=) , Less than (<), Is Null, etc].
- Notice that top field in all the tables has an asterisk instead of a field name. The asterisk is a SQL wildcard that represents all the fields in the table. As such, if you want to make use of all the fields in the table, make sure the checkbox adjacent to the asterisk is activated. Otherwise, you can select individual fields by activating their corresponding checkboxes. You can use the **Review Data** window to confirm that you have selected all the required fields.

The illustration below shows the **Visual Query Builder** window with a number of tables with join links between them. You can maximize the **Visual Query Builder** window as well as move around the individual tables to achieve a better visualization of the relationships.



## Using the Table Parameter Window

The **Table Parameter** window is used to define input parameters when a table is being used as the data source. Input parameters are used in the *where* clause of the SQL statement to determine the records to be retrieved from the table. For instance, the values submitted by a search form are received as input parameters by the corresponding grid or record form and subsequently used to retrieve records based on the submitted search criteria.



Condition Type: Parameter

Field	Condition	Parameter source
Name: <span></span>	<span>equals (=)</span>	<span></span>
Type: <span></span>		Type: <span>URL</span>
Format: <span></span>		Format: <span></span>
And/Or: <span>And</span>		Default Value: <span></span>

OK Cancel

The options available in the **Table Parameter** window vary depending on the selection made in the Condition Type field that appears at the top of the window. We shall first examine the available options when the Condition Type field is set to *Parameter*.

Property	Description
Field	
Name	Select the name of the database table field that will be matched against the input parameter.
Type	Select the data type of the field: <ul style="list-style-type: none"> <li>• Text</li> <li>• Integer</li> <li>• Float</li> <li>• Date</li> <li>• Boolean</li> <li>• Memo</li> </ul>
Format	Specify the format of the field if applicable e.g. a Boolean field could have the format 0;1 or Y;N
And/Or	Specify whether the field should be included in the where clause using the ‘AND’ operator or the ‘OR’ operator.

Condition	Specify the condition to be used to match the field value and the parameter value: <ul style="list-style-type: none"> <li>• equals (=)</li> <li>• does not equal (&lt;&gt;)</li> <li>• less than (&lt;)</li> <li>• less than or equal (&lt;=)</li> <li>• greater than (&gt;)</li> <li>• greater than or equal (&gt;=)</li> <li>• begin with (like '...')</li> <li>• does not begin with (not like '...%')</li> <li>• end with ('%...')</li> <li>• does not end with (not like '%...')</li> <li>• contains (like '%;..%')</li> <li>• does not contain (not like '%...%')</li> <li>• is null (is null)</li> <li>• not null (is not null)</li> </ul>
Parameter Source	Specify the name of the input parameter.
Type	Select the source of the parameter: <ul style="list-style-type: none"> <li>• Expression</li> <li>• URL</li> <li>• Form</li> <li>• Session</li> <li>• Application</li> <li>• Cookie</li> </ul>
Format	Specify the format of the parameter value if applicable e.g. a Boolean field could have the format True/False or Yes/No.
Default Value	Specify a default value to be used incase no value is received for the parameter.

## ***Using the Stored Procedure Parameter Window***

The **Stored Procedure Parameter** window is used to define input parameters when a stored procedure is being used as the data source.

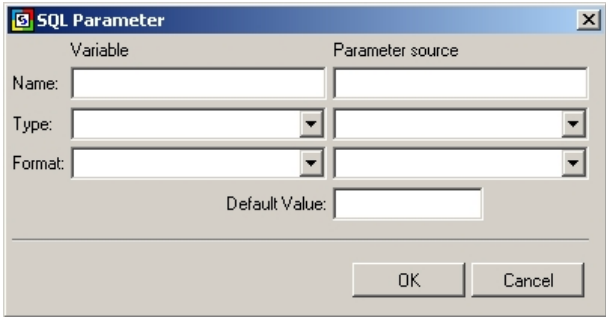
The screenshot shows a Windows-style dialog box titled "Stored Procedure parameter". It has two tabs: "Stored Procedure parameter" and "Parameter source". The "Stored Procedure parameter" tab is selected and contains the following fields: "Name:" (text box), "Data Type:" (dropdown menu showing "Text"), "Data Size:" (text box showing "0"), "Direction:" (dropdown menu showing "Input"), "Scale:" (text box showing "0"), "Precision:" (text box showing "0"), and "Default Value:" (text box). The "Parameter source" tab is also visible and contains a "URL" dropdown menu and a "Format:" dropdown menu. At the bottom right are "OK" and "Cancel" buttons.

Property	Description
Stored Procedure parameter	
Name	Specify the name of the stored procedure parameter
Data Type	Select the data type of the stored procedure parameter
Data Size	If applicable, specify the data size of the stored procedure parameter
Direction	Stored procedure parameters can be of different types. Select the type of the current parameter being referred to [Input, Input/Output, Output, Return Value].
Scale	Specify the number of decimal points for a number
Precision	Specify the number of digits for a number.
Parameter Source	
Name	The name of the input parameter.
Parameter Type	Select the type of the input parameter. <ul style="list-style-type: none"> <li>• Expression</li> <li>• URL</li> <li>• Form</li> <li>• Session</li> <li>• Application</li> <li>• Cookie</li> </ul>
Format	If applicable, specify the format of the input parameter value e.g. a Date could have the format mm/dd/yyyy.
Default Value	Specify a default value to be used incase no value is received from the input parameter source.

## Using the SQL Parameter Window

The **SQL Parameter** window is used to define input parameters when a SQL query is being used as

the data source.



Property	Description
Variable	
Name	Specify the name of the SQL variable/field.
Type	Select the data type of the variable/field: <ul style="list-style-type: none"><li>• Text</li><li>• Integer</li><li>• Float</li><li>• Date</li><li>• Boolean</li><li>• Memo</li></ul>
Format	If applicable specify the format of the variable/field e.g. a Boolean parameter could have the format true;false.
Parameter Source	
Name	Specify the name of the input parameter source.
Type	Select the type of the input parameter. <ul style="list-style-type: none"><li>• Expression</li><li>• URL</li><li>• Form</li><li>• Session</li><li>• Application</li><li>• Cookie</li></ul>
Format	If applicable, specify the format of the input parameter value e.g. a Boolean field could have the format True/False or Yes/No.
Default Value	Specify a default value to be used incase no value is received from the input parameter source.

# V. Controls and HTML Elements

In the process of creating a page, various controls and HTML elements are added to the page. Some of the controls are used to expose dynamic data that is retrieved from the database while others are purely functional such as buttons.

The CodeCharge Studio Toolbox provides a convenient way to add controls and HTML elements to a page. Before delving into the details of these components let us see how the two differ.

When viewed using a browser, Controls are no different from HTML elements. The distinction between the two exists only within CodeCharge Studio. Of the two, HTML elements are simpler and are made up entirely of HTML markup. When you add a HTML element to a page, the necessary HTML code is added to the page but you don't see the element listed in the Project Explorer window.


Controls are more advanced than HTML elements and some such as the Navigator and Sorter controls are composed of different types of HTML elements. Controls can be programmed dynamically using the programming language and they have properties that can be changed programmatically to alter their behavior. When you add a control to a page, the name of the control appears in the Project Explorer window.

## HTML Elements


The HTML elements are listed under the HTML tab of the CodeCharge Studio Toolbox. During the process of designing your pages, you can add HTML elements to the page as needed. After adding a HTML element, you can further customize it using the options available under the **Format** tab of the **Properties window**. You can configure a number of HTML properties or specify actions to be performed when certain events occur.

The following are the available HTML elements.


### Form

HTML	<form></form>
Image	 <i>(only visible in Design Mode)</i>
Description	A form is a container object within which you can place other form controls that submit values. In of itself, a form is not useful but it is indispensable when you need to use controls that display and submit values for processing. A Form element has an action property that is used to specify the page or resource to which the form values should be submitted.

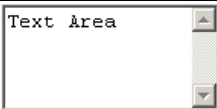
### Text Box

<b>HTML</b>	<code>&lt;INPUT TYPE="text" ID="TextBox1" NAME="TextBox1"&gt;</code>
<b>Image</b>	
<b>Description</b>	A text box is used to provide single line input. It is usually used when a small amount of data is to be entered.


## Password

<b>HTML</b>	<code>&lt;INPUT TYPE="password" ID="Password1" NAME="Password1"&gt;</code>
<b>Image</b>	
<b>Description</b>	This is very similar to the Text Box except that all characters entered are masked using asterisks. Most login forms will contain a password field.


## Text Area

<b>HTML</b>	<code>&lt;TEXTAREA ID="TextArea1" NAME="TextArea1" ROWS="4" COLS="15"&gt;Text Area&lt;/TEXTAREA&gt;</code>
<b>Image</b>	
<b>Description</b>	A Text Area field is used when a large amount of data needs to be entered. It is possible to set the number of rows as well as columns to be displayed. The control also has scrollbars to allow adding text beyond the specified rows and columns.


## Checkbox

<b>HTML</b>	<code>&lt;INPUT type="checkbox" id="Checkbox1" name="Checkbox1"&gt;</code>
<b>Image</b>	
<b>Description</b>	A checkbox is used to toggle the selection of an option. It can otherwise be viewed as a Boolean or On/Off control.


## Radio Button

<b>HTML</b>	<code>&lt;INPUT TYPE="radio" ID="Radio1" NAME="Radio1"&gt;</code>
<b>Image</b>	
<b>Description</b>	Radio button controls are used for the selection of mutually exclusive options. A number of radio buttons with the same name are presented and out of these, the user can select only one.

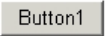
# Dropdown

HTML	<pre>&lt;SELECT ID="Select1" NAME="Select1"&gt;&lt;OPTION&gt;Select1&lt;/OPTION&gt;&lt;/SELECT&gt;</pre>
Image	
Description	A Dropdown is used to present a list of options from which the user can select one. The user clicks on the dropdown arrow to view all the options available.

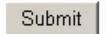
# Listbox

HTML	<pre>&lt;SELECT ID="Select1" NAME="Select1" SIZE="3"&gt;&lt;OPTION&gt;Select1&lt;/OPTION&gt;&lt;/SELECT&gt;</pre>
Image	
Description	A Listbox is essentially the same as a dropdown with the distinction of being able to display multiple options. With a listbox, it is also possible for the user to select more than one option by specifying the Multiple property.

# Button

HTML	<pre>&lt;INPUT TYPE="button" ID="Button1" NAME="Button1" VALUE="Button1"&gt;</pre>
Image	
Description	A Button is a control that performs a specified action when clicked. Usually, the OnClick event of the button is assigned some script code that executes the desired function.

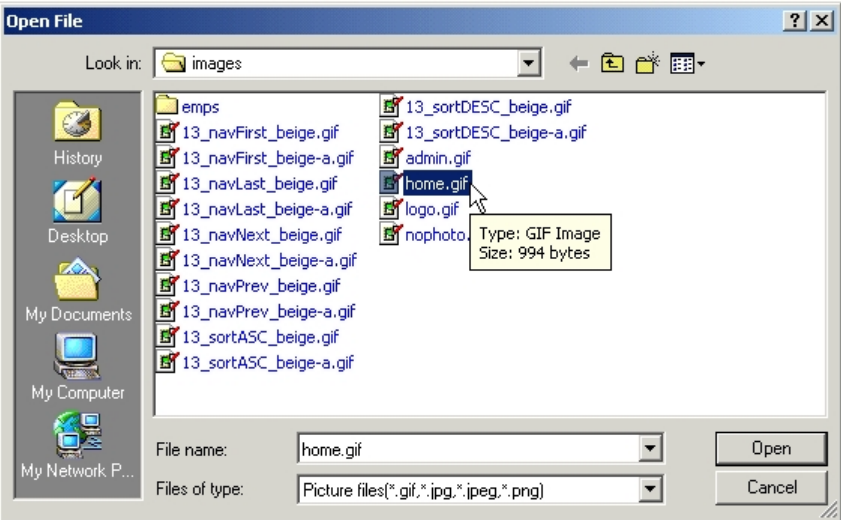
# Submit Button

HTML	<pre>&lt;INPUT TYPE="submit" ID="Submit1" NAME="Submit1" VALUE="Submit"&gt;</pre>
Image	
Description	When placed within a HTML form, a submit button serves the purpose of submitting all the fields within the form to the designated action page. The action page is specified in the action property of the Form element in which the submit button is placed.

## Reset Button

HTML	<code>&lt;INPUT TYPE="reset" ID="Reset1" NAME="Reset1" VALUE="Reset"&gt;</code>
Image	
Description	A Reset button is used to return the fields in a form to the state they originally were in when the page loaded. For instance, if a field was empty then a value was entered, clicking on the Reset button would clear the value.

## Image Button

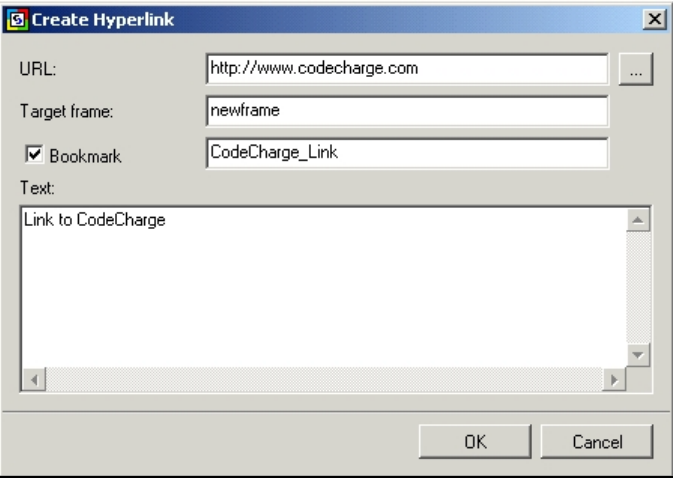
HTML	<code>&lt;input type="image" id="ImageButton1" name="ImageButton1" src="images\home.gif"&gt;</code>
Image	
Description	<p>This is the graphical equivalent of the Submit Button. When you click on the Image Button option in the <b>Toolbox</b>, a dialog box appears where you can select the image to be used to represent the button.</p> 

## Label

HTML	<code>&lt;label&gt;&lt;/label&gt;</code>
Description	As the name suggests, Labels are used to provide text that identifies other elements within the page. They can otherwise be termed as captions.

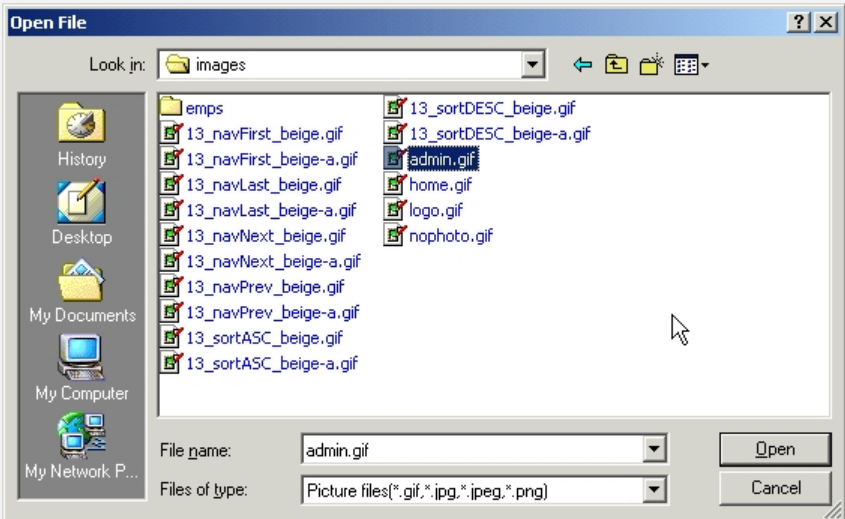


# Hyperlink

HTML	<code>&lt;a name="CodeCharge_Link" href="http://www.codecharge.com" target="newframe"&gt;Link to CodeCharge&lt;/a&gt;</code>
Image	<a href="#">Link to CodeCharge</a>
Description	<p>This is a link to another resource:</p> <ul style="list-style-type: none"><li>• <b>URL:</b> Specify the location of the resource to be linked to</li><li>• <b>Target Frame:</b> If using frames, enter the name of the frame where you want the linked resource to be opened.</li><li>• <b>Bookmark:</b> If want the link to also serve as a named anchor, enter a unique name to identify it.</li><li>• <b>Text:</b> This is the text the user will see and click on.</li></ul> <div></div>

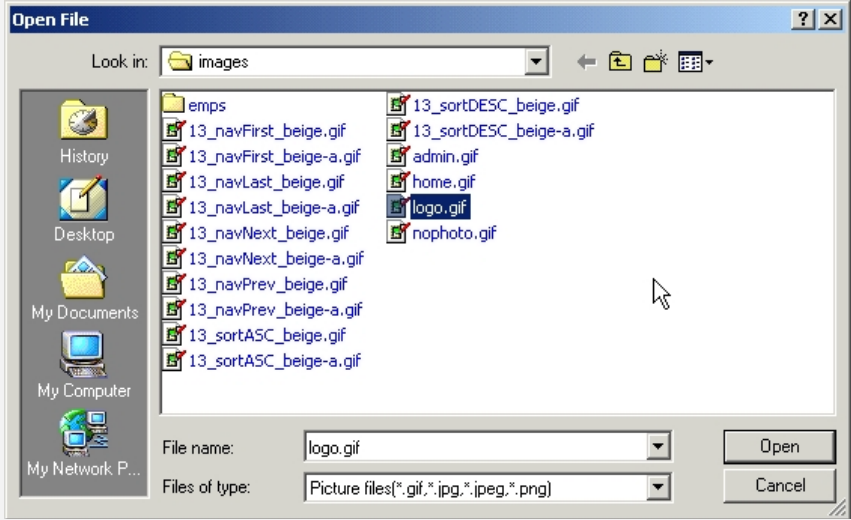
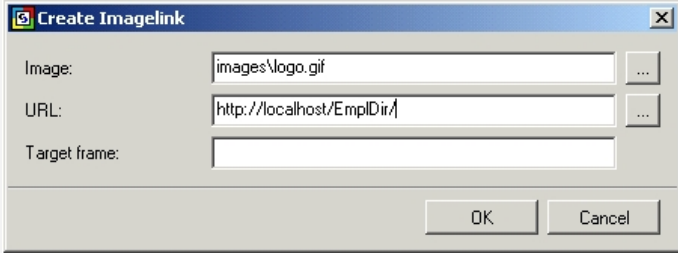
# Image

HTML	<code>&lt;IMG ID="Image1" NAME="Image1" SRC="images\admin.gif"&gt;</code>
Image	

<b>Description</b>	<p>This is used to embed an image in the current page. When you click on the image option in the <b>Toolbox</b>, a dialog box appears where you can navigate and select the image file to be used.</p> 
--------------------	---

### ImageLink

<b>HTML</b>	<pre>&lt;a href="http://www.codecharge.com"&gt;&lt;img src="images/logo.gif" border="0"&gt;&lt;/a&gt;</pre>
<b>Image</b>	


<b>Description</b>	<p>This is very similar to creating a HyperLink except that instead of having text to represent the link, an image is used.</p> <p>When you click on the ImageLink option in the Toolbox, a dialog box appears where you can select the image to be used.</p>  <p>After selecting an image, you then have to specify the URL of the resource to be linked to. If applicable, you can also enter the name of a frame where the link will be opened.</p> 
--------------------	--

Space

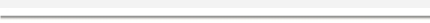
<b>HTML</b>	&nbsp;
<b>Description</b>	This is a non-breaking space meaning that a line break cannot occur in its place. It is used for instance when you don't want a line break to occur between two words.

Line Break


<b>HTML</b>	 
-------------	------

Image	 (only visible in Design Mode)
Description	A Line Break forcibly ends the current line of text and continues the flow of content on the next line. A Line Break is defined by a carriage return or a line feed. When in <b>Design</b> mode, you can enter a Line Break by typing the <i>shift+Enter</i> keys together.

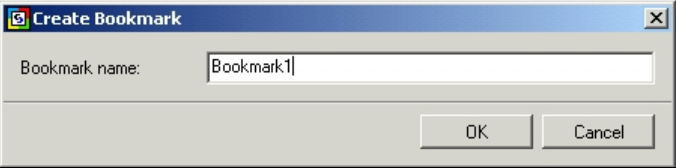
Horizontal Rule

HTML	<HR>
Image	
Description	This is simply a customizable line often used to demarcate sections of a page to make their difference more discernable.

Paragraph Break

HTML	<P></P>
Image	 (only visible in Design Mode)
Description	A Paragraph Break represents a paragraph and has the effect of placing an empty space between two adjacent paragraphs.

Bookmark

HTML	<a name="Bookmark1"></a>
Description	<p>A Bookmark is essentially a named anchor that can be the destination of another link. Anchors are often used in long pages so that the user can conveniently jump to designated positions within the document by using links to the anchors.</p> 

Controls

Controls are dynamic elements placed on a page and are often used to display database content. Controls play a central role in any given form that deals with database content. Controls and their values can also be controlled programmatically.

In their stripped down form, controls are pretty much HTML elements albeit with more functionality due to the server side code that interacts with them. In this section, the various controls available in the Toolbox

will be introduced. The controls are located under the **Forms** tab of the Toolbox.

You will notice in the section below that often times the HTML code for a particular control has some text that appears within curly braces (e.g. {IncludePage1}). The curly braces are indication that when the page is run live, the dynamic content will be substituted for the content in curly braces. When editing the HTML content manually or using a third party editor, take care not to alter the content in curly braces otherwise the page will not function as intended.

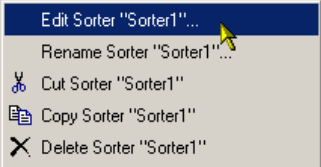
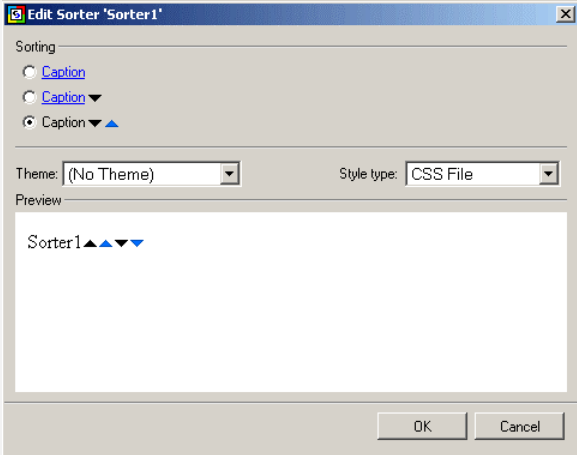
**Include Page**

<b>HTML</b>	{IncludePage1 }
<b>Image</b>	{IncludePage1 }
<b>Description</b>	Certain pages can be created specifically for being included in other pages. The Include Page control is used to include such pages, a good example of which are the header and footer pages that appear in many sites.


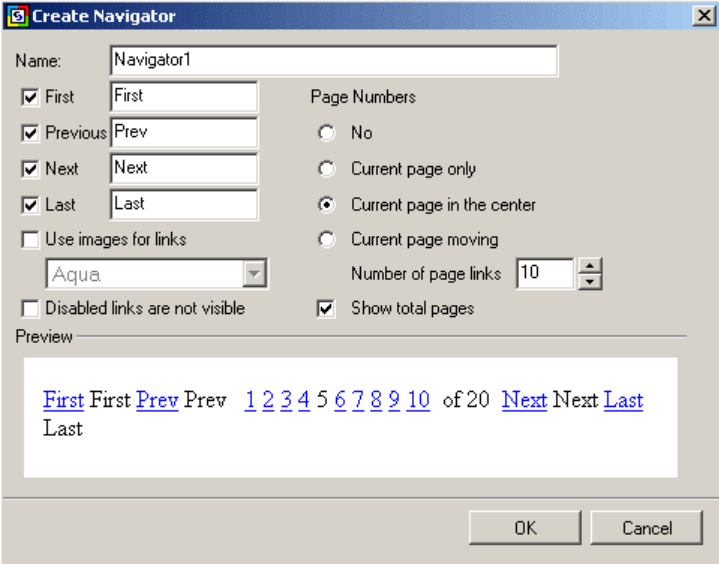
**Label**

<b>HTML</b>	{Label1 }
<b>Image</b>	{Label1 }
<b>Description</b>	<p>A label field is probably the simplest of the Controls although its flexibility allows it to be used to represent a wide variety of content. It is important to understand that Label fields are not HTML controls such as Text Box or Checkbox fields. A Label is simply plain text and therefore cannot be used to submit a value in a form.</p> <p>Due to their read-only nature, Labels are used to present content that should not be altered. As such, grid forms are mainly composed of Label fields.</p>

Sorter

Image	sorter▲▲▼▼
Description	<p>A sorter is used exclusively on grid forms. It allows the data within the grid to be sorted according to a particular field. Usually a sorter is placed in the top row of a HTML table where grid data is displayed. After adding a sorter, you can customize it by right-clicking on it and selecting the <b>Edit Sorter</b> option.</p> <div data-bbox="699 326 1016 493">A context menu is shown with a mouse cursor pointing at the 'Edit Sorter "Sorter1"...' option. The menu includes: 'Edit Sorter "Sorter1"...', 'Rename Sorter "Sorter1"...', 'Cut Sorter "Sorter1"', 'Copy Sorter "Sorter1"', and 'Delete Sorter "Sorter1"'.</div> <p>You can then use the options available in the <b>Edit Sorter</b> window to customize the sorter.</p> <div data-bbox="571 569 1142 1025">The 'Edit Sorter "Sorter1"' dialog box is shown. It has a 'Sorting' section with three radio buttons: 'Caption' (selected), 'Caption▼', and 'Caption▼▲'. Below this are 'Theme' and 'Style type' dropdown menus. The 'Preview' section shows the sorter control 'Sorter1▲▲▼▼'. At the bottom are 'OK' and 'Cancel' buttons.</div> <ul style="list-style-type: none"><li>• <a href="#">Caption</a> : The sorter has a link that can be used to initiate sorting. Clicking on the link consecutively will toggle the order of the sort.</li><li>• <a href="#">Caption▼</a> : The sorter has a link to initiate sorting. An image is used to indicate the order in which the sorting is currently done.</li><li>• <a href="#">Caption▼▲</a> : The sorter has four images, two that can be clicked to initiate sorting in ascending or descending order and two that indicate the current order of the sort.</li><li>• <b>Theme</b>: Select a theme that will determine the appearance of the sorter.</li><li>• <b>Style type</b>: Select how the theme will be implement. [HTML, CSS File, CSS Inline, CSS Page].</li></ul>

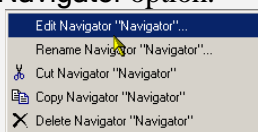
# Navigator

Image	
Description	<p>Like the Sorter control, the navigator is used exclusively on grid forms. A navigator provides functionality that allows the database records within a grid to be browsed. A navigator can be of varying complexity ranging from very simple navigators with single links to move back and forth, to complex navigators that have links for individual pages and display the total range of pages.</p> <p>When you click on the Navigator Icon in the Toolbox, the <b>Create Navigator</b> window appears where you can configure the properties of the Navigator.</p> 
	<ul style="list-style-type: none"><li>• <b>Name:</b> Assign a name for the navigator control</li><li>• <b>[First, Prev, Next, Last]:</b> Activate the corresponding checkbox if you want to have a link for the option in the navigator.</li><li>• <b>Use Images for links:</b> Select this option if you want images to be used in place of the text for the four links above. You can then use the listbox to select one of the available image themes.</li><li>• <b>Disabled links are not visible</b> – If selected, links for pages that cannot be navigated to are not displayed e.g. when displaying page 1, you cannot use the link for the First page so it should not be displayed.</li></ul>

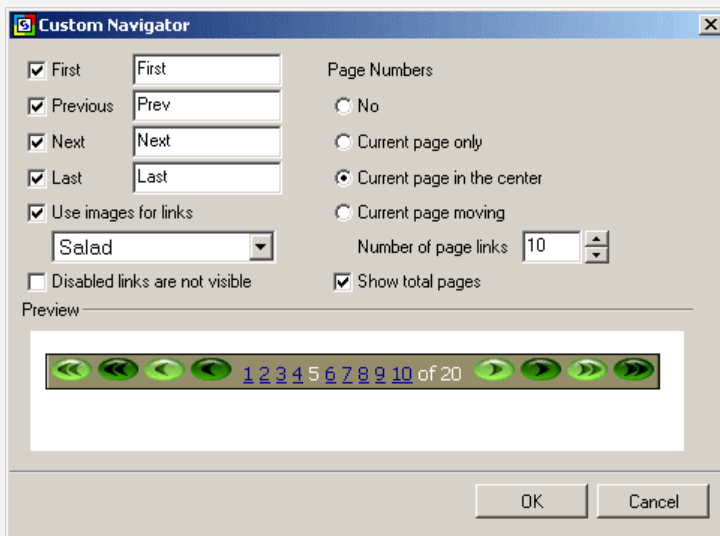
## Page Numbers

- **No:** No Page numbers are shown
- **Current page only:** Only the current page number is shown
- **Current page in the center:** A list of pages is shown with the current page appearing in the center.
- **Current page moving:** A list of pages is shown with the current page appearing in plain text relative to the other pages.
- **Number of page links:** The number of pages that should be shown in the list of pages.
- **Show total pages:** Show the total number of pages that can be browsed.

After adding a navigator control to a grid, you can adjust it using the “Custom Navigator” window. To access the window, right-click anywhere within the navigator and select the Edit Navigator option.



You can then proceed to configure the various options in the **Custom Navigator** window that is similar to the Create Navigator window shown above.



## Link

HTML	<code>&lt;a href="{Link1_Src}"&gt;{Link1}&lt;/a&gt;</code>
Image	<a href="#">{Link1}</a>



<b>Description</b>	This creates a hyperlink to a resource whose URL comes from a database column or expression.
--------------------	--

## Text Box

<b>HTML</b>	<code>&lt;input value="{ TextBox1 }" name="TextBox1"&gt;</code>
<b>Image</b>	
<b>Description</b>	A text box is used to provide single line input. It is usually used when a small amount of data is to be entered.

## Hidden

<b>HTML</b>	<code>&lt;input type="hidden" value="{ Hidden1 }" name="Hidden1"&gt;</code>
<b>Image</b>	(only visible in design mode)
<b>Description</b>	A Hidden field is the equivalent of a Text Box except that a hidden field has no visual representation on the rendered page. Unless the user views the source HTML for a page, there is no visual indication of the existence of a Hidden field.

## Text Area

<b>HTML</b>	<code>&lt;textarea name="TextArea1"&gt;{TextArea1}&lt;/textarea&gt;</code>
<b>Image</b>	
<b>Description</b>	A Text Area field is used to when a large amount of data needs to be entered. It is possible to set the number of rows as well as columns to be displayed. The control also has scrollbars to allow adding text beyond the specified rows and columns.

## Checkbox


<b>HTML</b>	<code>&lt;input type="checkbox" name="Checkbox1" value="1" {Checkbox1}&gt;</code>
<b>Image</b>	
<b>Description</b>	A checkbox is used to toggle the selection of an option. It can otherwise be viewed as a Boolean or On/Off control.

## Image

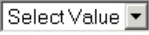
<b>HTML</b>	<code>&lt;img src="{ Image1 }"&gt;</code>
<b>Image</b>	

Description	This is used to embed an image in the current page.
-------------	---


## RadioButton

HTML	<!-- BEGIN RadioButton RadioButton1 --><input {Check} type="radio" name="RadioButton1" value="{Value}">{Description}<!-- END RadioButton RadioButton1 -->
Image	 {Description}
Description	Radio button controls are used for the selection of mutually exclusive options. A number of radio buttons with the same name are presented and out of these, the user can select only one.


## List Box

HTML	<select name="ListBox1"><option value="">Select Value</option>{ListBox1_Options}</select>
Image	
Description	A List Box is used to present a list of options from which the user can select one. The user clicks on the dropdown arrow to view all the options available. Using its format properties, a List Box can be configured to display more than one value at a time as well as allow the user to select multiple values.

## ImageLink

HTML	<a href="{ImageLink1}"></a>
Image	
Description	This is similar to an ordinary Link control but instead of text being used to represent the link, an image is used.

## Button

HTML	<!-- BEGIN Button Button1 --><input type="submit" value="Button1" name="Button1"><!-- END Button Button1 -->
Image	
Description	When placed within a HTML form, a submit button serves the purpose of submitting all the fields within the form to the designated action page.

# VI. Forms

Within CodeCharge Studio, forms are the main building blocks of a page. Although possible, it is highly unlikely that you will have a page without any form on it. This section deals with the mechanics of the various forms used within CodeCharge.

In CodeCharge Studio, a form refers to a section of a page containing normal content, markup, labels and optionally may contain a HTML form element with form controls in it. A CodeCharge Studio form is visually demarcated using one or more HTML tables with various content appearing in the table rows and columns. Note that a CodeCharge Studio form is not the same as a HTML Form element which is represented using the tags `<Form></Form>`. A CodeCharge Studio form may or may not contain a HTML Form element.

The following are the different types of forms you can build within CodeCharge Studio:

- Grid Form
- Record Form
- Search Form
- Login Form (Authentication)

The above forms can be constructed manually by putting together various Controls and HTML elements or alternatively you can use one of the builders to automate the process.

## Grid Form

As the name suggests, grid forms are used to display multiple records of data. In addition to displaying data, a grid form can be enhanced to allow the user to browse through multiple records as well as change the order in which the records are displayed.

Of all the forms that can be created within CodeCharge Studio, the grid form is the only form that does not have a HTML form element by default. This is because grid forms are intended to display data as opposed to altering it. However, it is possible to manually add a Form element to a grid form by editing the HTML code.

The illustration below shows a basic grid form. This form is created when the **Grid** option in the **Forms** tab of the **Toolbox** is clicked. The form has a title, a row for the field captions and another row where the database content is displayed.

NewGrid1	
Label1	Label2
{Label1}	{Label2}

## Record Form

Record forms are used to add, edit or delete database content. By default, a record form has a HTML Form element, a number of controls that can submit values as well as four buttons (Insert, Update, Delete and Cancel).

The display of the four buttons on a record form is controlled programmatically such that only the applicable buttons appear at the right time when the page is viewed live. For instance, if no record data is retrieved from the database, only the Insert and Cancel buttons will appear since you can't update or delete a non-existent record. On the other hand, if a record is retrieved and displayed in the form controls, the Update, Delete and Cancel buttons are shown but not the Insert button.

The illustration below shows a basic record form. This form is created when the **Record** option in the **Forms** tab of the **Toolbox** is clicked. The form has a title, a column for the field captions and another column with controls where the database content is displayed and can be altered. Notice too that unlike the grid form, a record form has a HTML form element.



The screenshot shows a web form titled "NewRecord1" enclosed in a "form" tag. Inside the form, there is an "{Error}" label, followed by two text input fields labeled "TextBox1" and "TextBox2". Below these fields are four buttons: "Insert", "Update", "Delete", and "Cancel". The form is closed with a "/form" tag.

## Using the Builders to create forms

In the interest of saving time and effort, the process of creating forms can be greatly automated by using builders. A builder is essentially a series of dialogs spanning the process of creating a page or form. The user configures the various options available in the series of dialogs, at the end of which the builder generates forms based on the selections made.

Generally speaking, you will find it much easier to create forms using the builders then later on edit the forms as desired, rather than create the forms from scratch manually. This is especially the case if you would like to maintain a consistent look since the builders offer the option of creating forms based on preset style and image themes.

The following are the different builders that can be used to create forms:

- Application Builder
- Grid Builder
- Record Builder
- Search Builder

- Authentication Builder
- Grid and Record Builder

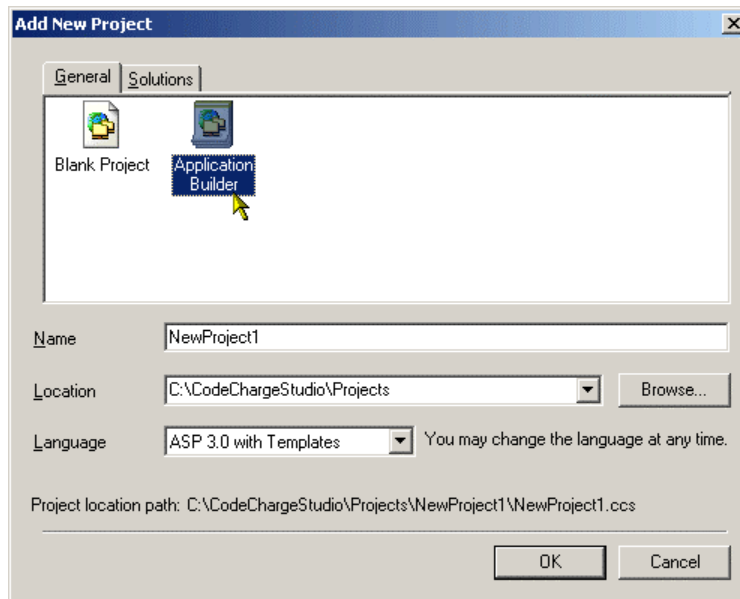
In this section, we shall go through the process of creating different forms using each of the builders.

## Application Builder

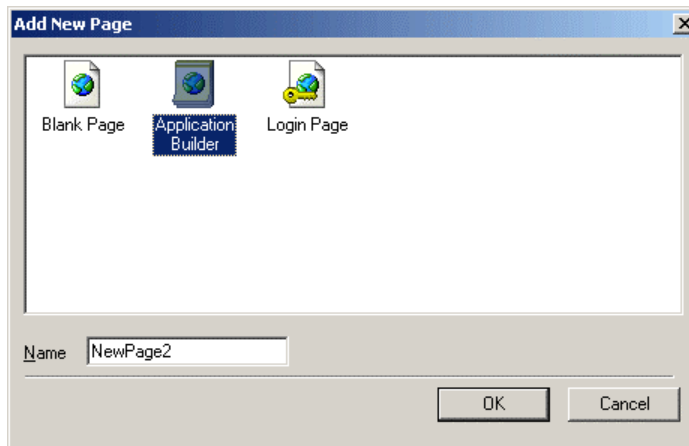
The Application Builder can be used to create multiple project pages based on the tables or queries in a designated database connection. In some instances, the Application Builder can be used to create all the pages needed for a project. The different phases of the Application Builder allow you to create grid, record and search forms for some or all the tables within a connection. You can then publish and use the generated pages or edit them to include custom functionality.

The steps below show how to create multiple project pages using the Application Builder.

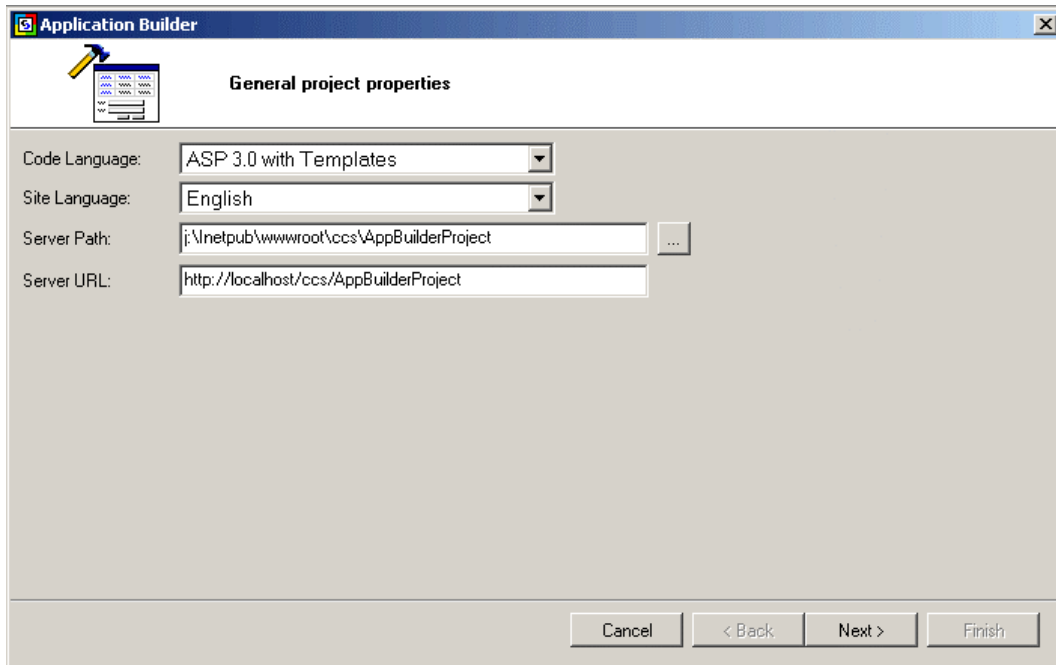
1. There are two ways of accessing the Application Builder. If you want to create an entirely new project using Application Builder, click on the **File** menu and select the **New → Project...** option. The **Add New Project** window pops up where you should select the **Application Builder** option then click OK.



Alternatively, if you want to add multiple pages to an existing project using the Application Builder, click on the **File** menu and select the **New → Page...** option. The **Add New Page** window pops up where you should select the **Application Builder** option, then click OK.



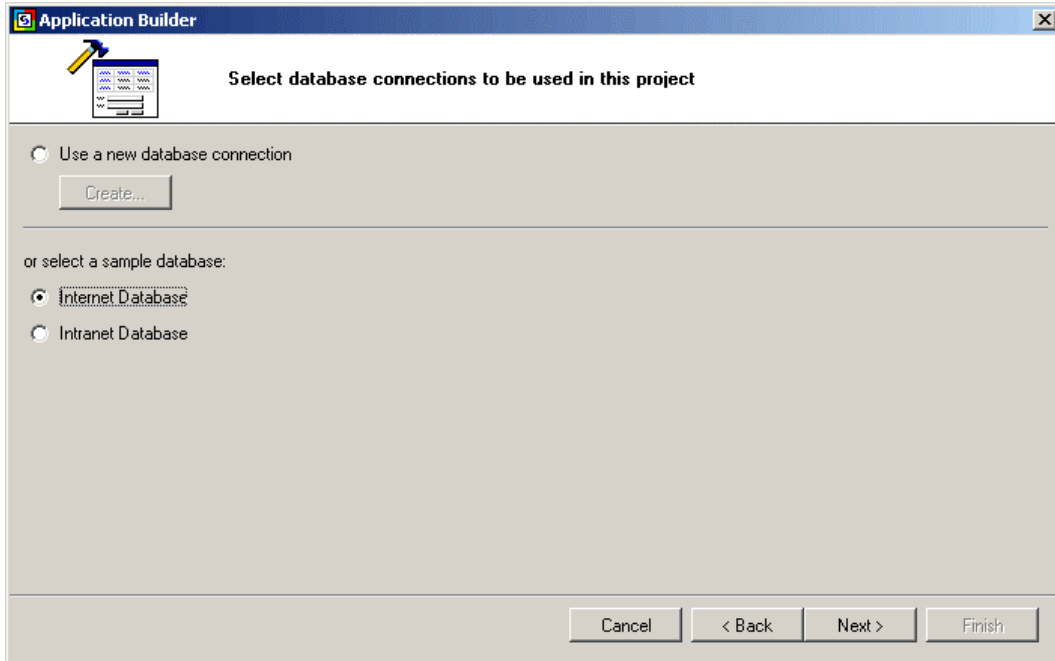
2. If you are building a new project, the first window of the Application Builder has four project-wide settings that need to be configured.



Property	Description
Code Language	Select the programming language in which the pages will be generated.
Site Language	Select the spoken language for the project pages
Server Path	Specify the server path where the pages will be published.
URL Path	Specify the URL that maps to the Server Path specified above. This is the URL you would type into a browser in order to access the pages.

Once done, click Next to proceed.

3. You now have to configure a new database connection for the project or select one of the sample databases.



Property	Description
Use a new database connection	Select this option if you want to configure a new database connection. You can then click on the <b>Create...</b> button to create the connection.
Select a sample database connection	Select either the Internet or Intranet sample database to use for the project.

If you opted to create a new database connection, click on the **Create...** button to bring up the **Add New Connection** window. You can then proceed to create a database connection based on [ODBC DSN](#) or a [Connection string](#).

Once you have selected or configured a database connection, click **Next** to proceed.

4. In this step, there are a number of options that will determine how the builder will generate the pages.

Property	Description
Enable smart naming	With this option selected, the Builder will generate English like captions for the database table and field names. For example, the grid form for a database table called “ <i>employees</i> ” will have a caption “ <i>List of Employees</i> ”. Additionally, fields with underscores in them will be converted to column captions without the underscore e.g. “ <i>task_name</i> ” will be shown as “ <i>Task Name</i> ”.
Autoincremented primary keys	This option should be checked if the tables in your database have autoincremented primary keys. With autoincremented primary keys, new records are automatically assigned unique primary key values by the database. Note that both of the sample databases have autoincremented primary keys.
Use single keyword field for text and memo fields on Search	Select this option if you want search forms to have a single field that can be used to search against all the text and memo fields in the corresponding grid form. If this option is not selected, the search form will have a field for each to the text and memo fields.
Publish the site after Builder completes creating pages	Select this option if you want the generated pages to be published to the Server Path immediately after the Builder completes creating the pages. The Server Path was specified in Step 1 above.



5. The next step involves setting up the security mechanism of the project.

**Application Builder**

**Access Authentication**

☒ **Use authentication:**

User Table:

User ID Field:

Login Field:

Password Field:

Level/Group Field:

**Security Groups**

Group ID	Group Name
1	User
2	Admin

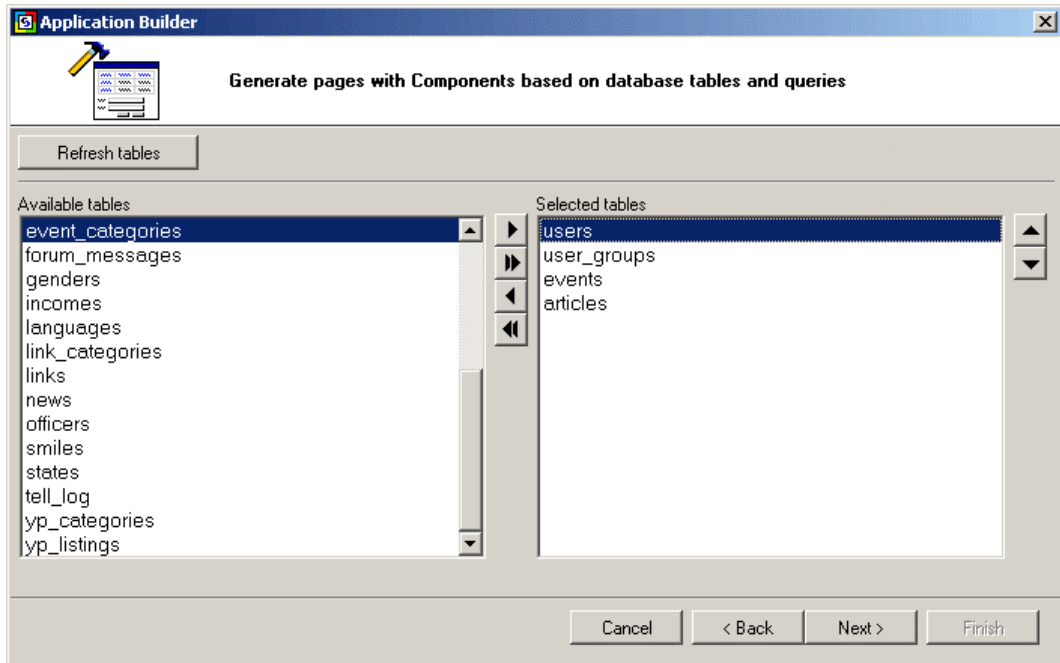
☒ Higher Level Inclusive of Lower Levels







Buttons: Add... Modify... Remove... Cancel < Back Next > Finish

Property	Description
Use authentication	Select this option if you want your pages to have security authentication.
User Table	Select the table where user authentication information is contained.
User ID Field	Select the database field with the unique user ID's.
Login Field	Select the database field with the Login name of the users.
Password Field	Select the database field with the user passwords.
Level/Group Field	Select the field with the user security levels.
Security Groups	This table contains a list of all the security groups that registered users can belong to.
Group ID	The ID of the security group. This also serves as the level of the group.
Group Name	The descriptive name for the security group.
Higher Level Inclusive of Lower Levels	If this option is selected, users with higher level security groups will be able to access pages with lower security levels e.g. a user in security group 3 will be able to access pages with security levels 2 or 1 but not 4.
Add...	Click this button to add a new security group.
Modify...	Click this button to modify an existing security group.
Remove	Click this button to remove an existing security group.

Once you are done configuring the security options, click **Next** to proceed.

6. In this step, you are presented with a list of the tables in the database and out of these, you select those for which you want to create pages.




Property	Description
Refresh tables	If you make changes to the database during the process of using the builder, click this button to refresh the connection and retrieve the latest state of the tables.
Available tables	This is a list of the tables available in the database.
Selected table	This is a list of the tables you have selected and based on which you will be able to create pages with forms.
	Select an Available table then click this button to add it to the list of selected tables
	Click this button to add ALL Available tables to the Selected tables list.
	Select a Selected table then click this button to remove it from the list.
	Click this button to remove all Selected tables from the list.
	Select a table then click this button to move it upwards in the order of appearance.
	Select a table then click this button to move it downwards in the order of appearance.

Once done, click Next to proceed.

7. This is the pivotal step in the process and involves selecting the types of forms to be created for each of the tables you selected in the previous step.

Application Builder



Generate pages with Components based on database tables and queries

Table	Grid pages			Record maintenance/view page		
(Click for details) <a href="#">All</a>	Grid <input checked="" type="checkbox"/>	Search <input checked="" type="checkbox"/>	Security Level	Record <input checked="" type="checkbox"/>	Updateable <input checked="" type="checkbox"/>	Security Level
<a href="#">users</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
<a href="#">user_groups</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin
<a href="#">events</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User
<a href="#">articles</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	User

Cancel

< Back







Next >

Finish

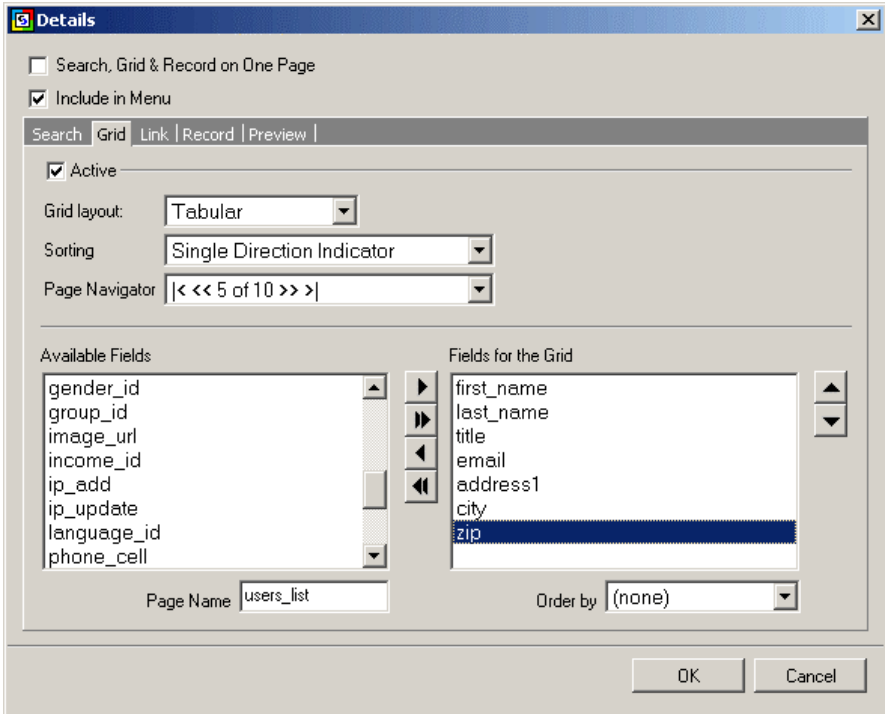
Property	Description
Tables	A list of the tables you selected. Click on a particular table to access more configuration details.
Grid Pages	
Grid	Activate this option if you want to have a grid form for the table.
Search	Activate this option if you want to have a search form for the table.
Security Level	Specify the security level required to access the form.
Record maintenance/view page	
Record	Activate this option if you want to have a record form for the table.
Updateable	Activate this option if you want the record form to be able to update records.
Security Level	Specify the security level required to access the form.

If you click on a particular table name under the **Table** column, the **Details** window shown below appears using which you can configure more details for the table. The first tab in the **Details** window is the **Search** tab where you can configure an optional search form.








Property	Description
Search, Grid & Record form on One Page	Select this option if you want the Search, Grid and Record forms to appear on the same page.
Include in Menu	If selected, a link to this page will be added in the navigation menu page.
Active	Check this option to indicate your intention to create a search form.
Sorting list box	If selected, the search form will have two listboxes for sorting. The first one will contain a list of all the available fields so that the user can select the field to sort by. The user will also be able to indicate the desired sort order (ASC, DESC) using the second listbox.
Records per page list box	If selected the search form will have a listbox that can be used to specify the number of records to be returned per page after the search is conducted
Orientation	Specify whether the search form should be vertically or horizontally oriented
Search Type	Specify the SQL operator to be used for the search. If AND is selected, the search form will have an input form control for all the fields. Otherwise, if OR is selected, the search form will have only one input form control. Any value entered into the single control will be used to search against all the fields in the form. You can use the preview tab to see how the form looks like with either option selected.
Available Fields	This listbox shows all the fields available in the table or view.



Search Fields	These are the fields to be shown in the search form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.
	Click this button to add ALL Available Fields to the form
	Select a search form field then click this button to remove it from the form
	Click this button to remove all fields from the search form.
	Select a form field then click this button to move it upwards in the order of appearance.
	Select a form field then click this button to move it downwards in the order of appearance.
Control Type	Specify the control type to be used for each of the fields in the form.

After configuring the search form, you can access configuration options for the grid form by clicking on the **Grid** tab of the **Details** window.

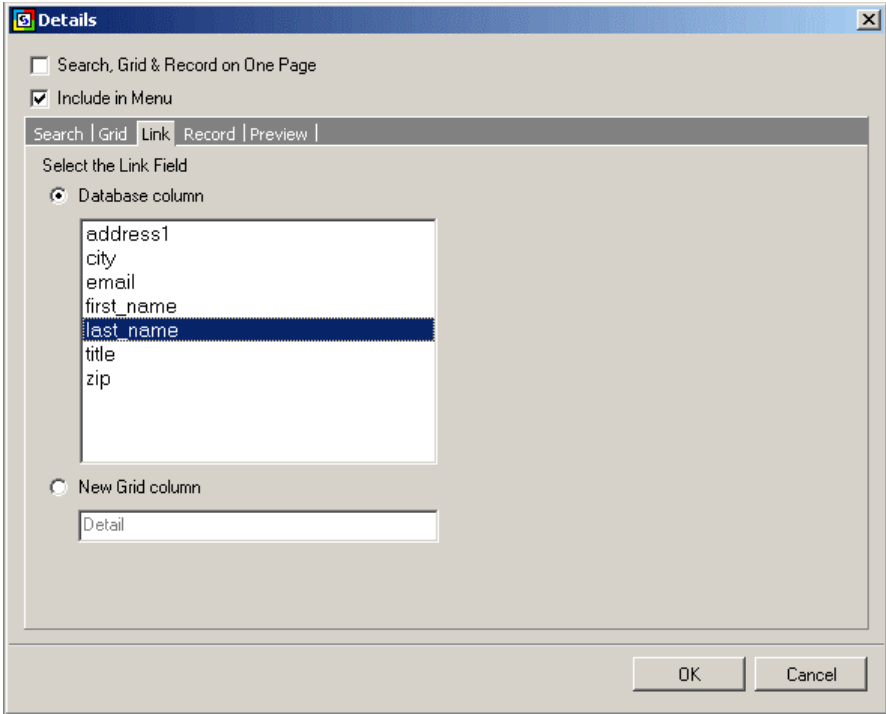


Property	Description
Active	Check this option to indicate your intention to create a grid form.

Grid Layout	<p>Select the desired layout of the grid form:</p> <ul style="list-style-type: none"> <li>▪ <b>Tabular</b></li> </ul> <p><b>members</b></p>  <ul style="list-style-type: none"> <li>▪ <b>Columnar</b></li> </ul> <p><b>members</b></p>  <ul style="list-style-type: none"> <li>▪ <b>Justified</b></li> </ul> <p><b>members</b></p> 
Sorting	Select the type of sorting controls you want to have on the grid form.
Page Navigator	Select the type of navigation controls you want to have on the grid form.
Available Fields	This listbox shows all the fields available in the table or view.
Fields for the Grid	These are the fields to be shown in the grid form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.
	Click this button to add ALL Available Fields to the form.
	Select a form field then click this button to remove it from the form.
	Click this button to remove all fields from the form.







	Select a form field then click this button to move it upwards in the order of appearance.
	Select a form field then click this button to move it downwards in the order of appearance.
Page Name	Specify the name of the page where the grid and search form will appear.
Order by	Select a field by which the records will be ordered by when initially displayed.

In the **Links** tab of the **Details** window, you can specify how the records in the grid form will be linked to the records in the record form.



Property	Description
Database column	Select this option if you want to use one of the existing form fields as the link field. You can then specify the field to be used from among the available fields.
New Grid column	You can also create a new column in the Grid form that will specifically be used to link to the Record form. Enter some text of your choice to be used for this new grid column.

After configuring the link to the record form, you can now proceed to configure options for the record form itself. This is done under the **Record** tab of the **Details** window.

Property	Description
Active	Activate this option to indicate your intention to create a record form.
Primary Key	Select the primary key field in the table.
Available Fields	This listbox shows all the fields available in the table.
Fields for the Record	These are the fields to be shown in the record form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form
	Click this button to add ALL Available Fields to the form
	Select a form field then click this button to remove it from the form
	Click this button to remove all fields from the form
	Select a form field then click this button to move it upwards in the order of appearance
	Select a form field then click this button to move it downwards in the order of appearance
Control Type	Specify the control type to be used for each of the fields in the form.
Page Name	Specify the name of the page where the record form will appear. This option is not available when the grid, search and record forms appear on the same page.

Finally, after configuring all the forms, you can preview how they will look like when generated by using the **Preview** tab of the **Details** window. You will have a chance to select a



style at a later stage but at the moment, you can see the layout of the three forms based on the selections you made.

The screenshot shows a 'Details' window with a tabbed interface. The 'Preview' tab is active, displaying a form titled 'Search Users'. The form contains several text input fields: 'First Name' with placeholder '{s\_first\_name}', 'Last Name' with '{s\_last\_name}', 'Email' with '{s\_email}', 'Address1' with '{s\_address1}', 'City' with '{s\_city}', and 'Zip' with '{s\_zip}'. A 'Search' button is located to the right of the 'Zip' field. Below the form is a section labeled 'List of Users' with a horizontal scrollbar. At the top of the window, there are two checkboxes: 'Search, Grid & Record on One Page' (unchecked) and 'Include in Menu' (checked). The window has 'OK' and 'Cancel' buttons at the bottom right.

Once you are satisfied with the appearance of the forms, click on the **OK** button to close the **Details** window and return to the **Application Builder** window. If need be, you can click on another table name and access the **Details** window for that table. You can also click on the **All** link in the **Table** column to apply global options for all the forms to be created:

This screenshot shows the 'Details' window with the 'Search' tab selected. The 'Search, Grid & Record on One Page' checkbox is checked, and the 'Include in Menu' checkbox is also checked. The 'OK' and 'Cancel' buttons are visible at the bottom right of the window.

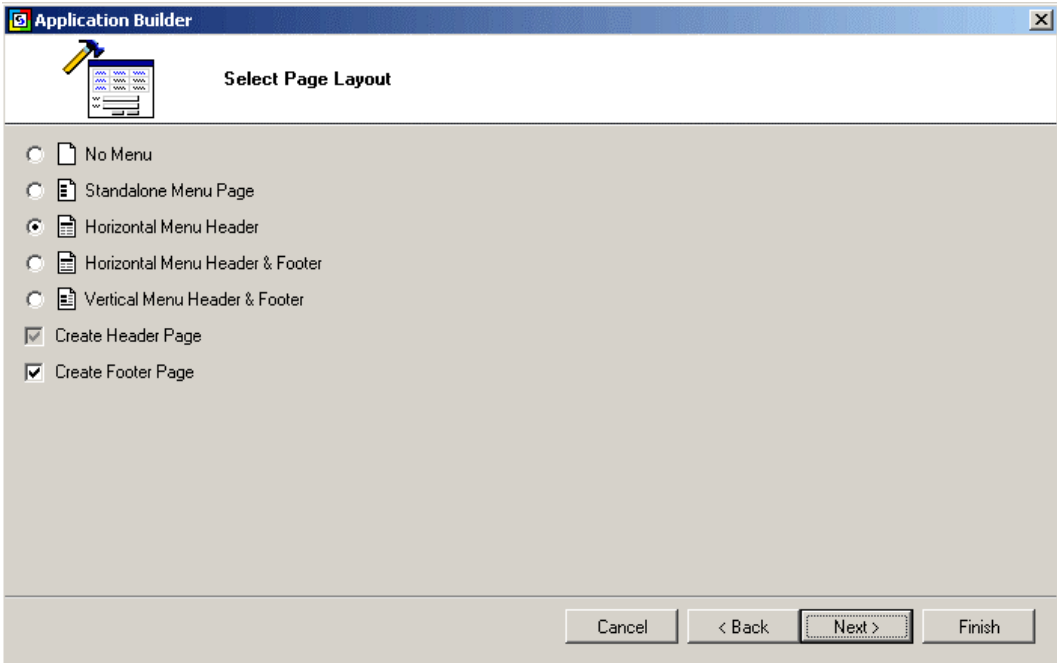
**Property**

**Description**

Search, Grid & Record form on One Page	Select this option if you want the Search, Grid and Record forms to appear on the same page for all pages.
Include in Menu	If selected, a link to all pages will be added in the navigation menu page.

Once done, click on the **Next** button in the **Application Builder** to proceed.

8. You now have options to customize the page layout.

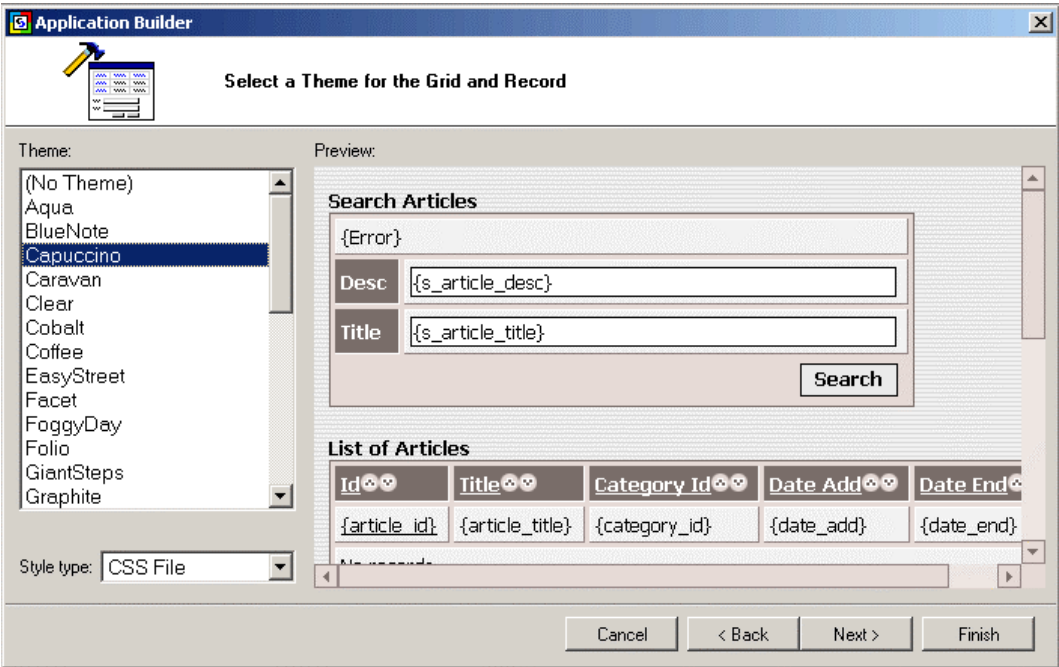


Property	Description
No Menu	Select this option if you don't want to have a menu page.
Standalone Menu Page	If this option is selected, the menu will be placed in a standalone page from where the user will be able to navigate to the linked pages. This means that the individual pages will not have a menu and neither will the header & footer pages.
Horizontal Menu Header	Select this option to create a horizontal menu of the Header page.
Horizontal Menu Header & Footer	With this option, a horizontal menu is placed on the Header & Footer pages that subsequently appear at the top and bottom of all the other pages.
Vertical Menu Header & Footer	With this option, a vertical menu is placed on the Header & Footer pages that subsequently appear at the top and bottom of all the other pages.
Create Header Page	Select this option if you want all pages to have a header page.

Create Footer Page	Select this option if you want all pages to have a footer page.
--------------------	---

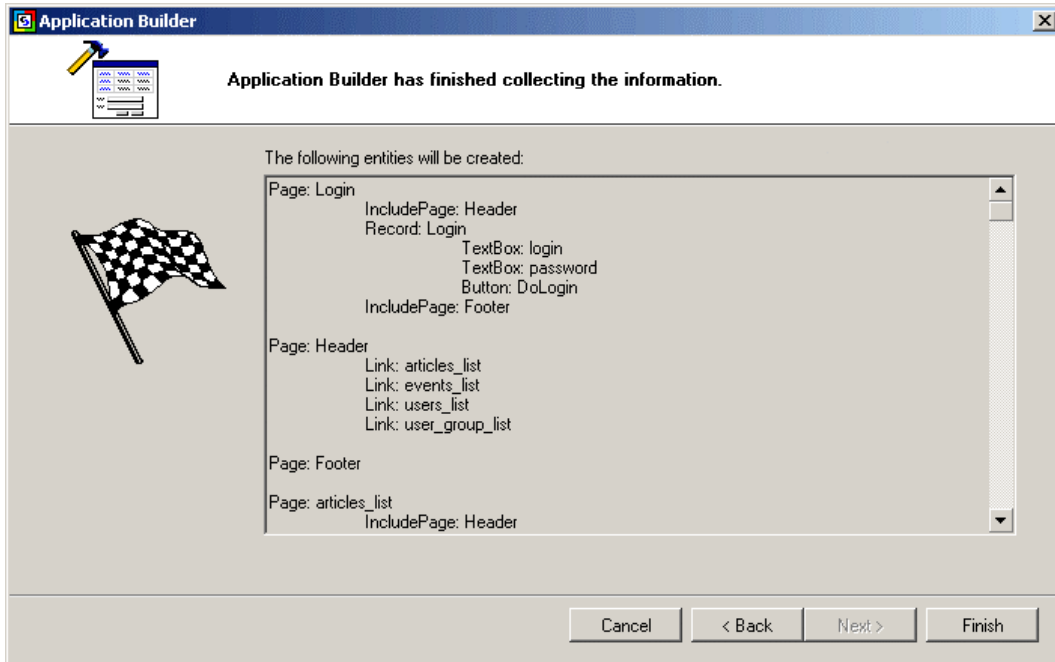
After configuring the page layout, click **Next** to proceed.

- With all the other configurations in place, you now have the chance to select a theme to be used for the forms

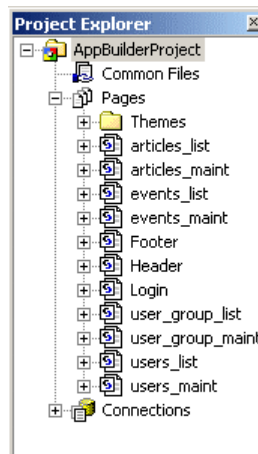


Property	Description
Theme	This is a list of available themes. Select the (No Theme) option if you don't want to use a theme.
Preview	This window shows how the forms would look like based on the selected theme.
Style Type	Select how the style will be applied: <ul style="list-style-type: none"> <li>HTML – Styles are defined using HTML attributes and properties.</li> <li>CSS Page – Cascading Style Sheet definitions are placed at the top of the HTML pages in the &lt;HEAD&gt; section.</li> <li>CSS File – Cascading Style Sheet definitions are placed in a separate file that is linked to the HTML pages.</li> <li>CSS Inline – Cascading Style Sheet definitions are placed within the HTML tags.</li> </ul>

- After deciding on a theme to use, click **Next** to advance to the last step. This window displays a summary of the configurations that have been made to this point. Click **Finish** to generate the pages and optionally publish them to the specified Server Path.



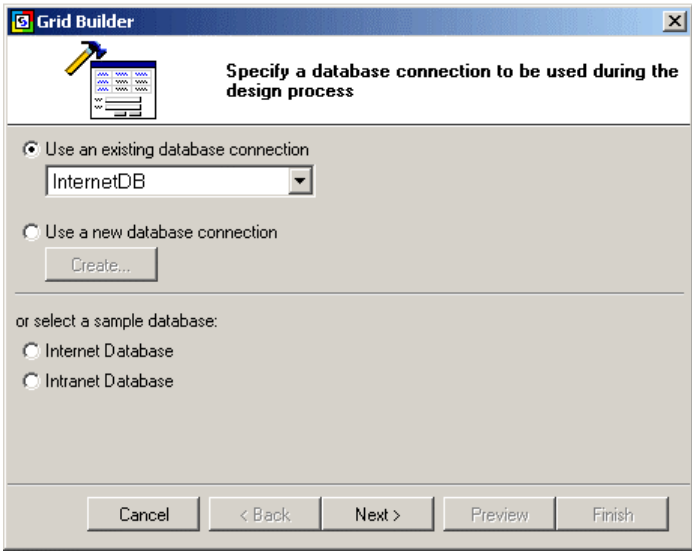
Once the pages have been generated, the Project Explorer window will show all the pages for the project and you can then open them and make any adjustments you want.



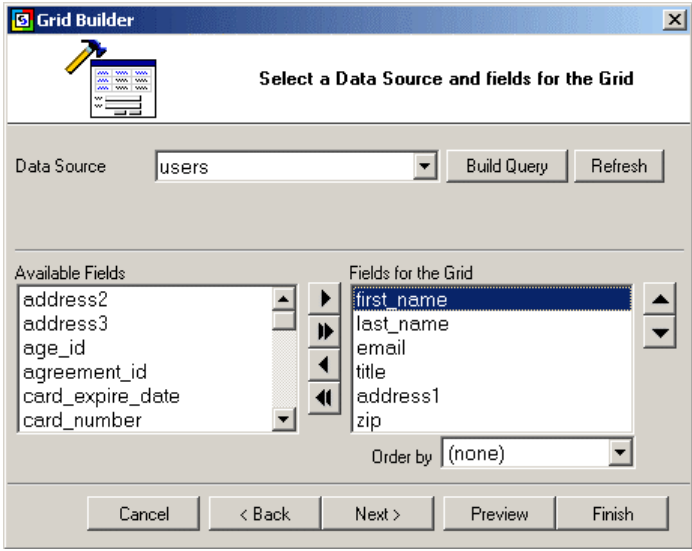
## Grid Builder

Use the Grid builder to create grid forms that can optionally have a related search form. As will become evident as we proceed through the process of creating the grid forms, the builder provides many options that would otherwise require a considerable amount of time to add manually.







1. To begin the process, click on the Grid Builder option in the **Builders** tab of the **Toolbox**.
2. The first step involves selecting or creating a database connection from where the data shown in the grid will be retrieved. You can select one of the existing database connections, create a new connection or select one of the sample databases. Once done click **Next** to proceed.



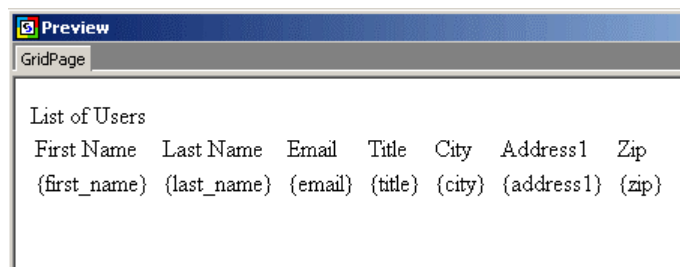
3. Having selected a connection, you now have to select a data source as well as the fields to be included in the grid.



Property	Description
Data Source	Select the table or view from where the data will come from.






Refresh	Click this button to refresh the connection if you make changes to the database after having begun using the builder.
Build Query	Clicking this button will bring up the <a href="#">Visual Query Builder</a> window where you can setup a SQL query that will act as the grid data source.
Available Fields	This listbox shows all the fields available in the table or view.
Fields for the Grid	These are the fields to be shown in the grid. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the grid.
	Click this button to add ALL Available Fields to the grid.
	Select a grid field then click this button to remove it from the grid.
	Click this button to remove all fields from the grid.
	Select a grid field then click this button to move it upwards in the order of appearance
	Select a grid field then click this button to move it downwards in the order of appearance
Order by	Select a field by which the records will be ordered by when initially displayed


After having added fields to the grid and optionally set the Order by field, you can click on the **Preview** button to see how the grid looks like at this point. For the above setup, the illustration below shows how the grid looks like in its current state of construction.



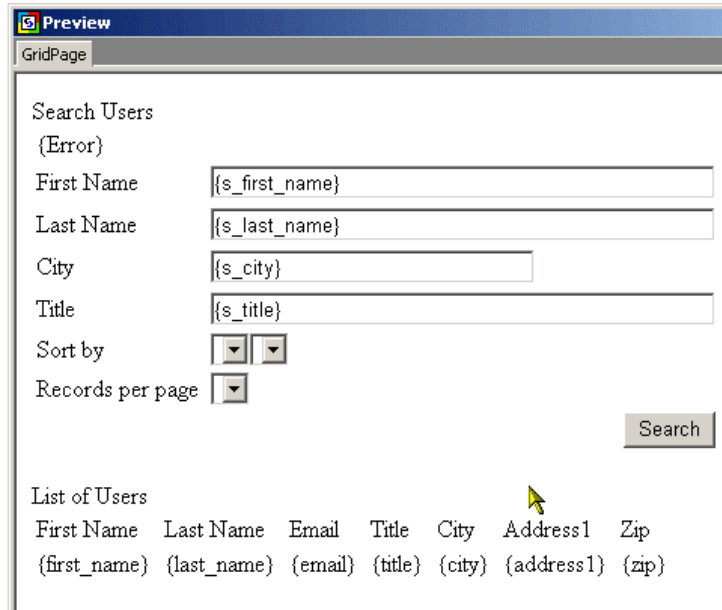
After previewing the grid, close the preview window and click on the **Next** button to proceed. Note that if you are satisfied with the form at this point, you can click on the **Finish** button to close the builder and generate the form.

- 4. The next step involves adding a search form that can be used to filter the records displayed in the grid form. This step is optional and if you don't want a corresponding search form for the grid, you can simply click the **Next** button to proceed to the next step.

Property	Description
Create Search/Filter	Activate this option to indicate your intention to create a search form.
Sorting list box	If selected, the search form will have two listboxes for sorting. The first one will contain a list of all the available fields so that the user can select the field to sort by. The user will also be able to indicate the desired sort order (ASC, DESC) using the second listbox.
Records per page list box	If selected, the search form will have a listbox that can be used to specify the number of records to be returned per page after the search is conducted.
Orientation	Specify whether the search form should be vertically or horizontally oriented.
Search Type	Specify the SQL operator to be used for the search. If AND is selected, the search form will have an input form control for all the fields. Otherwise, if OR is selected, the search form will have only one input form control. Any value entered into the single control will be used to search against all the fields in the form. You can use the preview button to see how the form looks like with either option selected.
Available Fields	This listbox shows all the fields available in the table or view.
Search Fields	These are the fields to be shown in the search form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.
	Click this button to add ALL Available Fields to the form.
	Select a form field then click this button to remove it from the form.
	Click this button to remove all fields from the form.
	Select a form field then click this button to move it upwards in the order of appearance.

	Select a form field then click this button to move it downwards in the order of appearance.
Control Type	Specify the control type to be used for each of the fields in the form.

Once again, during the process of creating the search form, you should use the Preview button to see how the form looks with the selected options. You can therefore try out various options and preview them before deciding on the appropriate one.



Preview

GridPage

Search Users



{Error}


First Name {s\_first\_name}

Last Name {s\_last\_name}

City {s\_city}

Title {s\_title}

Sort by  

Records per page 

Search

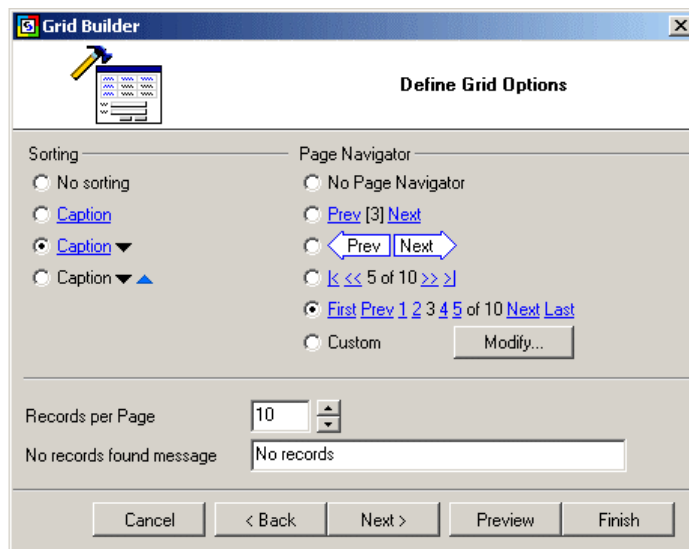
List of Users

First Name	Last Name	Email	Title	City	Address1	Zip
{first_name}	{last_name}	{email}	{title}	{city}	{address1}	{zip}

After configuring the form satisfactorily, click on the **Next** button to proceed.

- At this juncture, you can further customize the grid form by selecting the caption to be used for sorting as well as the navigational controls.





**Grid Builder**

**Define Grid Options**

Sorting

- ☐ No sorting
- ☐ Caption
- ☒ Caption ▼
- ☐ Caption ▼ ▲

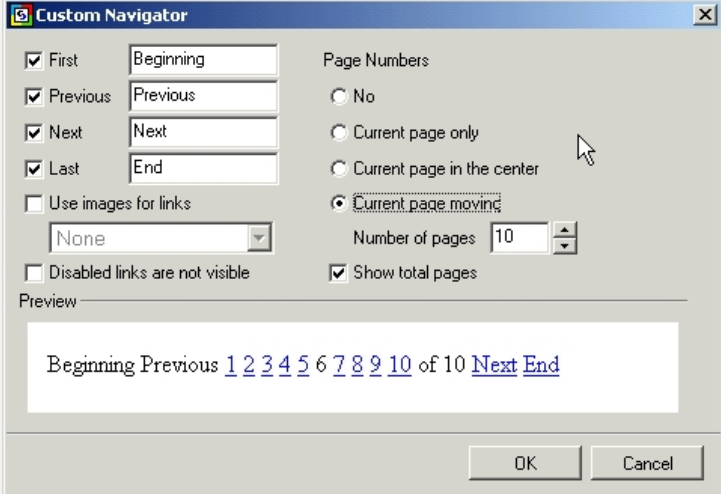
Page Navigator

- ☐ No Page Navigator
- ☐ Prev [3] Next
- ☐ ◀ Prev Next ▶
- ☐ k << 5 of 10 >> j
- ☒ First Prev 1 2 3 4 5 of 10 Next Last
- ☐ Custom Modify...

Records per Page: 10

No records found message: No records

Cancel < Back Next > Preview Finish

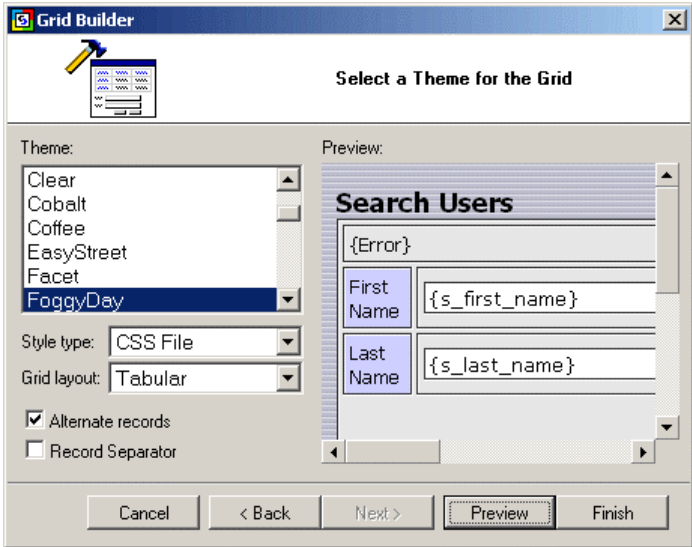
Property	Description
Sorting	Select the type of sorting controls that you want to have in the grid.
Page Navigator	Select the type of Navigation controls that you want to have in the grid.
Modify	<p>If you want to have custom navigation controls on the form, click on this button to open the <b>Custom Navigator</b> window where you can customize the controls to suit your needs.</p>  <p><b>Custom Navigator</b></p> <p> <input checked="" type="checkbox"/> First Beginning  <input checked="" type="checkbox"/> Previous Previous  <input checked="" type="checkbox"/> Next Next  <input checked="" type="checkbox"/> Last End  <input type="checkbox"/> Use images for links  None  <input type="checkbox"/> Disabled links are not visible </p> <p>Page Numbers</p> <ul style="list-style-type: none"> <li><input type="radio"/> No</li> <li><input type="radio"/> Current page only</li> <li><input type="radio"/> Current page in the center</li> <li><input checked="" type="radio"/> Current page moving</li> </ul> <p>Number of pages: 10</p> <p><input checked="" type="checkbox"/> Show total pages</p> <p>Preview</p> <p>Beginning Previous 1 2 3 4 5 6 7 8 9 10 of 10 Next End</p> <p>OK Cancel</p>

		Property	Description
		First	Text for the link to the first record
		Previous	Text for the link to the previous record
		Next	Text for the link to the next record
		Last	Text for the link to the last record
		Use images for links	Select this option if you want to have images instead of text for the navigation links
		Disabled links are not visible	If this option is selected, links to pages that cannot logically be browsed are not shown e.g. if you are on page 1, you cannot browse using the first or previous links.
		Page Numbers	Apart from having the four links mentioned above, specify whether you also want to have links to the individual pages and how they should appear.
		Number of Pages	Specify how many page links should be displayed.
		Show total pages	Select this option if you want to display the total number of pages that can be browsed.
Records per page	Specify the default number of records to be shown per page		
No records found message	Specify the message that will be displayed in the grid form if there are no records to be shown.		




Once you have made all the selections, you can use the preview button to see how the form looks like and if necessary, make adjustments.

The screenshot shows a 'Preview' window with a 'GridPage' tab. It displays a web form for searching users. The form includes a 'Search Users' section with an '(Error)' message, input fields for 'First Name' and 'Last Name' (both containing placeholder text like '{s\_first\_name}'), and a 'Search' button. Below the search section is a 'List of Users' section. It features a header row with blue, underlined links for 'First Name', 'Last Name', 'Email', 'Title', 'City', 'Address1', and 'Zip'. Below this is a row of placeholder text: '{first\_name} {last\_name} {email} {title} {city} {address1} {zip}'. At the bottom, it says 'No records' and shows a pagination control with a blue, underlined link for '{Page Number}', followed by '{Page Number} of {Total Pages}' and navigation arrows.

6. The final step is optional and involves selecting a theme for the form.



Property	Description
Theme	This listbox contains a list of themes from which you can select one.
Style Type	<ul style="list-style-type: none"><li>HTML – Styles are defined using HTML attributes and properties.</li><li>CSS Page – Cascading Style Sheet definitions are placed at the top of the HTML pages in the &lt;HEAD&gt; section.</li><li>CSS File – Cascading Style Sheet definitions are placed in a separate file that is linked to the HTML pages.</li><li>CSS Inline – Cascading Style Sheet definitions are placed within the HTML tags.</li></ul>

Grid Layout	<p>Specify the layout of the grid form</p> <ul style="list-style-type: none"><li>▪ <b>Tabular</b></li></ul> <p><b>members</b></p>  <p>The tabular layout shows a grid with two columns. The first column has a header 'first_name' and a data row '{first_name}'. The second column has a header 'last_name' and a data row '{last_name}'. Below the data rows is a 'No records' message. At the bottom is a navigation bar with four icons: a left arrow, a double left arrow, a double right arrow, and a right arrow.</p> <ul style="list-style-type: none"><li>▪ <b>Columnar</b></li></ul> <p><b>members</b></p>  <p>The columnar layout shows a grid with two rows. The first row has a header 'first_name' and a data cell '{first_name}'. The second row has a header 'last_name' and a data cell '{last_name}'. Below the data rows is a 'No records' message. At the bottom is a navigation bar with four icons: a left arrow, a double left arrow, a double right arrow, and a right arrow.</p> <ul style="list-style-type: none"><li>▪ <b>Justified</b></li></ul> <p><b>members</b></p>  <p>The justified layout shows a grid with two columns. The first column has a header 'first_name' and a data row '{first_name}'. The second column has a header 'last_name' and a data row '{last_name}'. Below the data rows is a 'No records' message. At the bottom is a navigation bar with four icons: a left arrow, a double left arrow, a double right arrow, and a right arrow.</p>
Alternate records	If this option is selected, every other record in the grid will appear in a row with a different background color than the preceding record.
Record Separator	With this option selected, each row where a record appears is followed by an empty row that separates it from the next row with a record.

Clicking the preview button now will show the forms as they will look when generated. The search form appears at the top with the grid right below it.

**Search Users**

{Error}

First Name {s\_first\_name}

Last Name {s\_last\_name}

Search

**List of Users**

First Name▲▼	Last Name▲▼	Email▲▼	Title▲▼	City▲▼	Address
{first_name}	{last_name}	{email}	{title}	{city}	{address}
{Alt_first_name}	{Alt_last_name}	{Alt_email}	{Alt_title}	{Alt_city}	{Alt_add}

No records

First First Prev Prev {Page Number} {Page Number} of {Total Pages} Next

## Record Builder

The Record Builder can be used to automate the process of creating record forms where database content can be inserted, edited or deleted.

1. To begin the process, click on the **Record builder** option in the **Builders** tab of the **Toolbox**.
2. The first step involves selecting or creating a database connection from where the data to be edited or added using the record form is stored. You can select one of the existing database connections, create a new connection or select one of the sample databases.

**Record Builder**

Specify a database connection to be used during the design process

☒ Use an existing database connection

internet

☐ Use a new database connection

Create...

or select a sample database:

☐ Internet Database

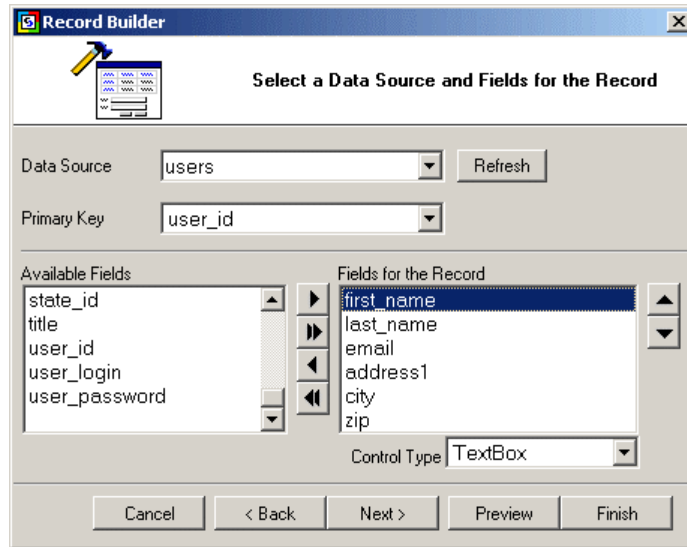
☐ Intranet Database







☒ Autoincremented primary keys

Cancel < Back Next > Preview Finish

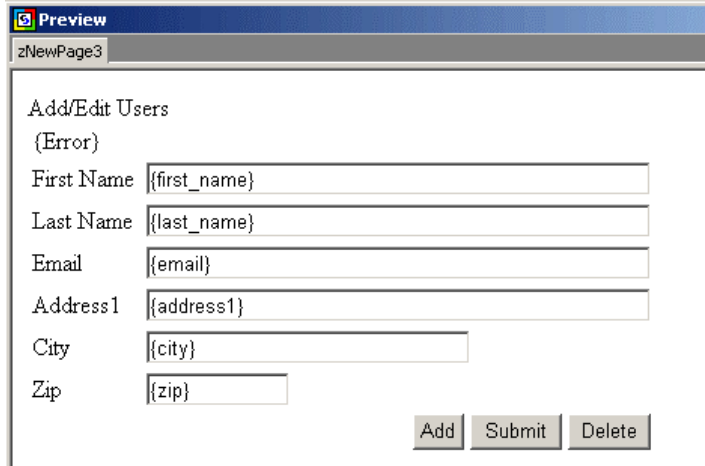
Once done click **Next** to proceed.

3. Having selected a connection, you now have to select a data source as well as the fields to be included in the record form



Property	Description
Data Source	Select the table from where the data will come from.
Refresh	Click this button to refresh the connection if you make changes to the database after having begun using the builder.
Primary Key	Select the primary key field in the table.
Available Fields	This listbox shows all the fields available in the table
Fields for the Record	These are the fields to be shown in the record form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.
	Click this button to add ALL Available Fields to the form.
	Select a form field then click this button to remove it from the form.
	Click this button to remove all fields from the form.
	Select a form field then click this button to move it upwards in the order of appearance.
	Select a form field then click this button to move it downwards in the order of appearance.
Control Type	Specify the control type to be used for each of the fields in the form.

Once you have made the selections, you can click on the preview button to see how the form looks like and maybe make some adjustments. Once done, click **Next** to proceed



Preview

zNewPage3

Add/Edit Users

{Error}

First Name {first\_name}

Last Name {last\_name}

Email {email}

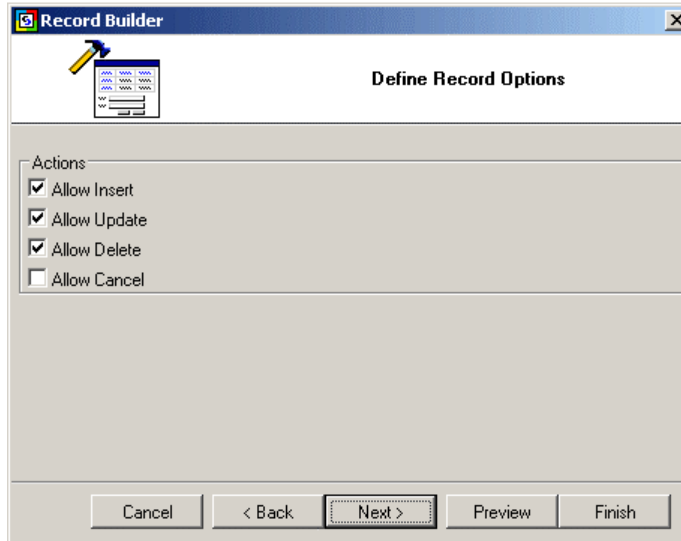
Address1 {address1}

City {city}

Zip {zip}

Add Submit Delete

4. You now have to decide the kind of functionality that will be available in the record form. You can uncheck any of the options if you don't want the form to have the corresponding functionality.



Record Builder

Define Record Options

Actions

☒ Allow Insert

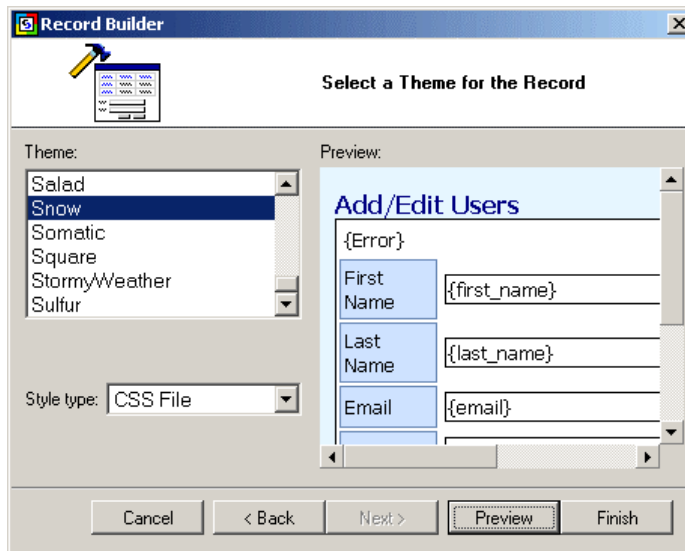
☒ Allow Update

☒ Allow Delete

☐ Allow Cancel

Cancel < Back Next > Preview Finish

5. In the final step, you optionally select a theme for the record form. The Preview portion of the window shows how the form would look like with the currently selected theme.



Once done, click on the **Finish** button to close the builder and generate the form. The illustration below shows the final form based on the configurations made above.

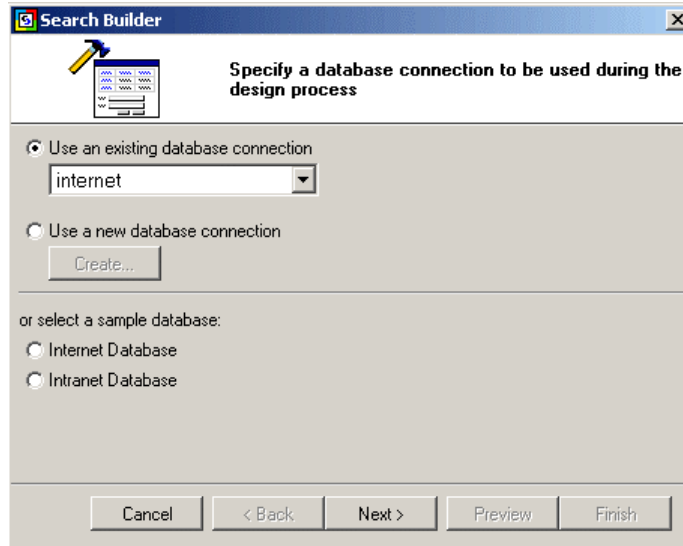
## Search Builder

Search forms are used to enter values into controls such that when the values are submitted, they are used to search and retrieve records from the database. Search forms are used in conjunction with grid or record forms whereby the search form is used to submit the search criteria and the grid or record form displays the retrieved records that match the search criteria.

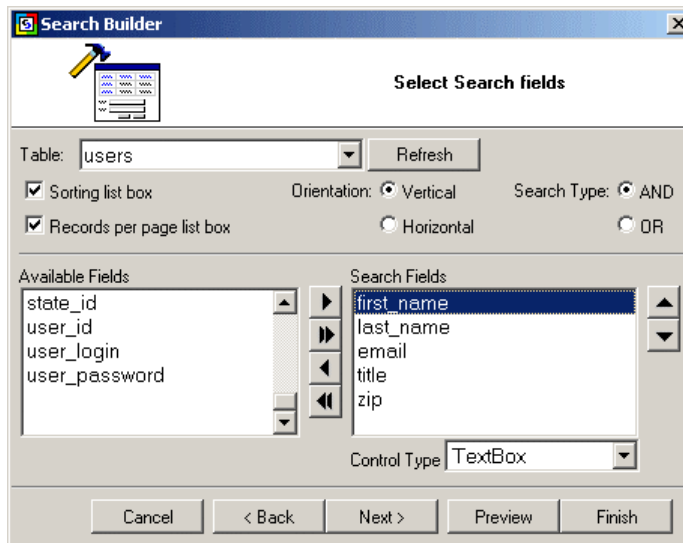
In their simplest form, search forms simply submit search criteria. However, you can use the Search Builder to create complex search forms that also determine the number of records to be displayed per page, the order in which the records are displayed as well as the SQL operator to be used to perform the search.









1. To begin the process of creating a search form, click on the **Search builder** option in **Builders** tab of the **Toolbox**.
2. Select or create the database connection whose fields will be used in the search. You can select one of the existing database connections, create a new connection or select one of the sample databases. Once done click **Next** to proceed.

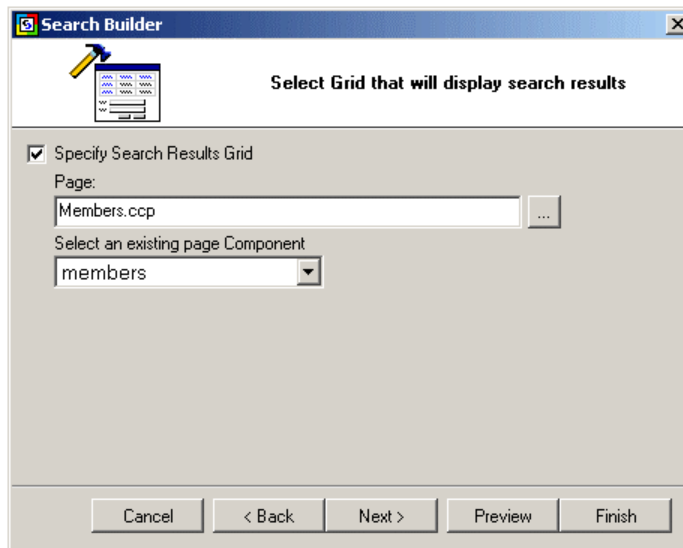


3. This is the main step in creating the search form and involves selecting fields to be shown in the form as well selecting other display options.

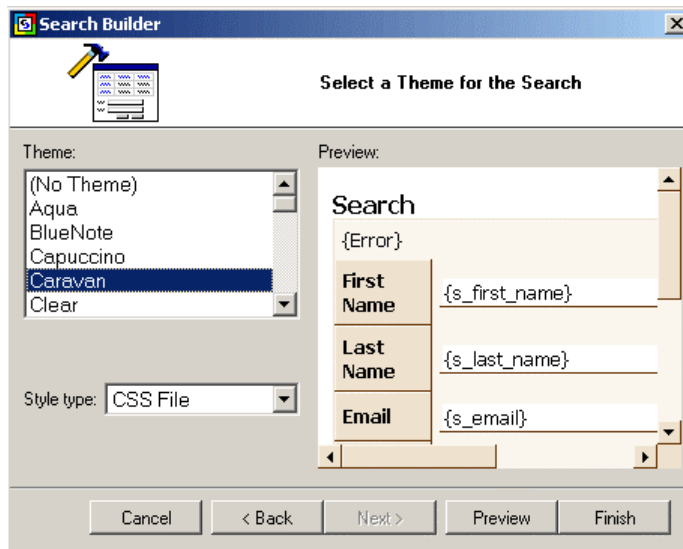


Property	Description
Table	Select the database tables whose fields will be used in the search form.
Sorting list box	If selected, the search form will have two listboxes for sorting. The first one will contain a list of all the available fields so that the user can select the field to sort by. The user will also be able to indicate the desired sort order (ASC, DESC) using the second listbox.
Records per page list box	If selected the search form will have a listbox that can be used to specify the number of records to be returned per page after the search is conducted.
Orientation	Specify whether the search form should be vertically or horizontally oriented.
Search Type	Specify the SQL operator to be used for the search. If AND is selected, the search form will have an input form control for all the fields. Otherwise, if OR is selected, the search form will have only one input form control. Any value entered in the single control will be used to search against all the fields in the form. You can use the preview button to see how the form looks like with either option selected.
Available Fields	This listbox shows all the fields available in the table or view.
Search Fields	These are the fields to be shown in the search form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.
	Click this button to add ALL Available Fields to the form.
	Select a form field then click this button to remove it from the form.
	Click this button to remove all fields from the search form.
	Select a form field then click this button to move it upwards in the order of appearance.
	Select a form field then click this button to move it downwards in the order of appearance.
Control Type	Specify the control type to be used for each of the fields in the form.

- Next, you can optionally specify a page and a grid form to which the search will apply. You can jump this step if the form is not yet available or you intend to use the search form for a different purpose.



5. In the final step, you can optionally select a theme to be used to display the form.



Once done, click the **Finish** button to close the builder and generate the search form.

Search

{Error}

First Name	<input type="text" value="{s_first_name}"/>
Last Name	<input type="text" value="{s_last_name}"/>
Email	<input type="text" value="{s_email}"/>
Title	<input type="text" value="{s_title}"/>
Zip	<input type="text" value="{s_zip}"/>
Sort by	<input type="button" value="v"/> <input type="button" value="v"/>
Records per page	<input type="button" value="v"/>

Search

Authentication Builder

A login or authentication form is used to input authentication details (Login and Password) that are required to grant access to users in sites that implement security.

The Authentication builder is used to add login/logout functionality to an existing page. Click on the **Authentication builder** option in the **Builders** tab of the **Toolbox** to begin the process.

1. In the first step, you have to select the type of functionality you want to add to the page.

Authentication Builder

Select the authentication feature to be added

☒ Login Form
 ☐ Logout Link to this Page
 ☐ Logout Link to custom Page
 ☐ Logout Action on this Page

Cancel

< Back

Next >

Preview

Finish

Option	Description
Login Form	Select this option to add a login form to the current page.

Logout Link to this Page	Select this option to add a logout link in the current page. The link points to the current page.
Logout Link to custom Page	Select this option to add a logout link in the current page. The link points to the page you select in the next step.
Logout action on this Page	Select this option to add a logout action to the current page. The action is added in the After Initialize event of the page.

If you opt to add a Login form, clicking the preview button will show a basic form similar to that shown below.

Login

{Error}

Login

Password

Login

2. If in the first step you opted to add a Login form, the next step allows you to specify whether you want the login field in the login form to have focus when the page loads. If the field is in focus, the user can begin typing into it immediately without having to first click on it.

Authentication Builder

Please select Login options

☒ Focus on Login field

Cancel

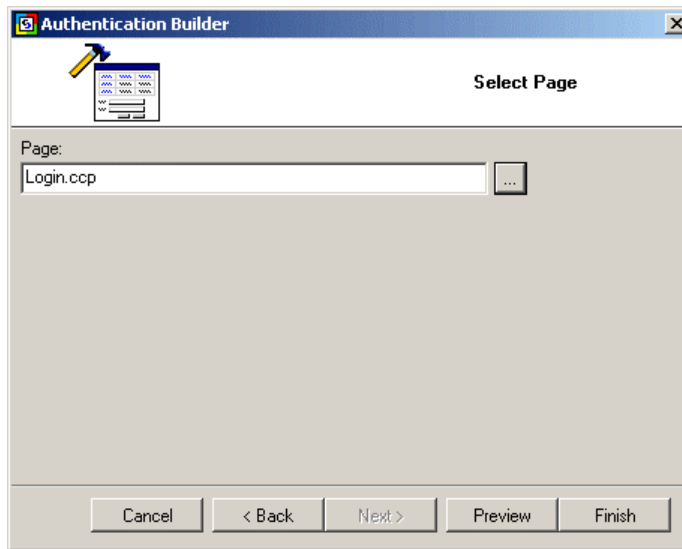
< Back

Next >

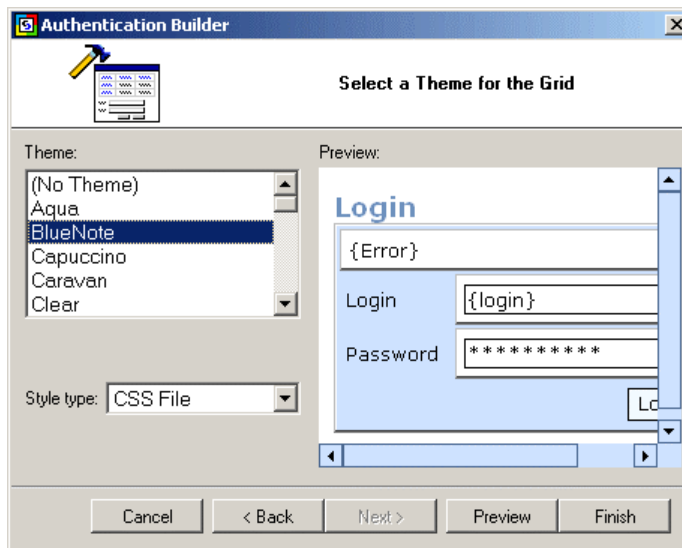
Preview

Finish

Alternatively, if in the first step you opted to add a Logout Link to a Custom Page, you now have to select the custom page to where the link will refer.



3. The final step that applies only if you are building a login form involves the selection of a theme for the form.



As usual, you can use the Preview button to see how the form looks like. Click the **Finish** to close the builder and generate the form.

**Login**

{Error}

Login {login}

Password \*\*\*\*\*

Login

## Grid and Record Builder

Often times a grid form is used in conjunction with a record form such that the grid form is used to display multiple records and a link is provided from the grid form records to the record form where the details of the record can be viewed and edited. For this purpose, the Grid and Record builder can be used to produce a grid form with an attendant record form.

1. Click on the **Grid and Record Builder** option in the **Builder** tab of the **Toolbox** to begin the process.
2. Select or create the database connection whose fields will be used in the two forms. You can select one of the existing database connections, create a new connection or select one of the sample databases.

**Grid and Record Components Builder**

Specify a database connection to be used during the design process

☒ Use an existing database connection  
internet

☐ Use a new database connection  
Create...

or select a sample database:

☐ Internet Database  
☐ Intranet Database

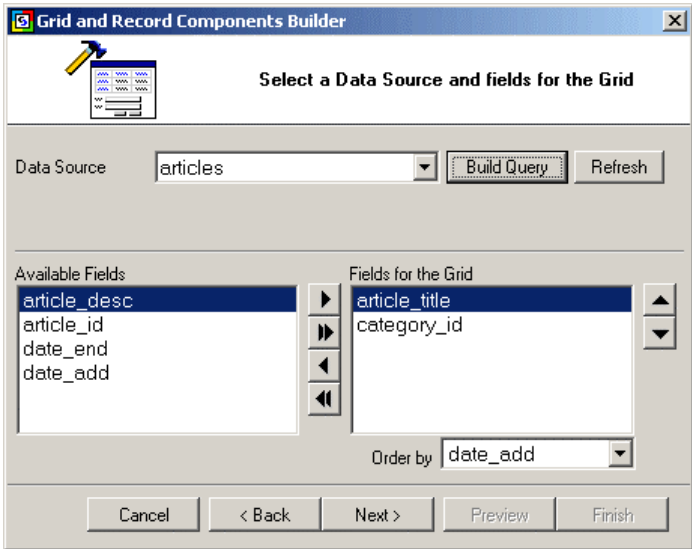
☒ Autoincremented primary keys







Cancel < Back Next > Preview Finish

Property	Description
----------	-------------

Autoincremented primary keys	Select this option if the database table has an autoincremented primary key field. This option prevents the builder from creating a record form with the primary key field since it will be provided by the database when a new record is inserted.
------------------------------	---

3. Now select the fields to be shown in the grid form.

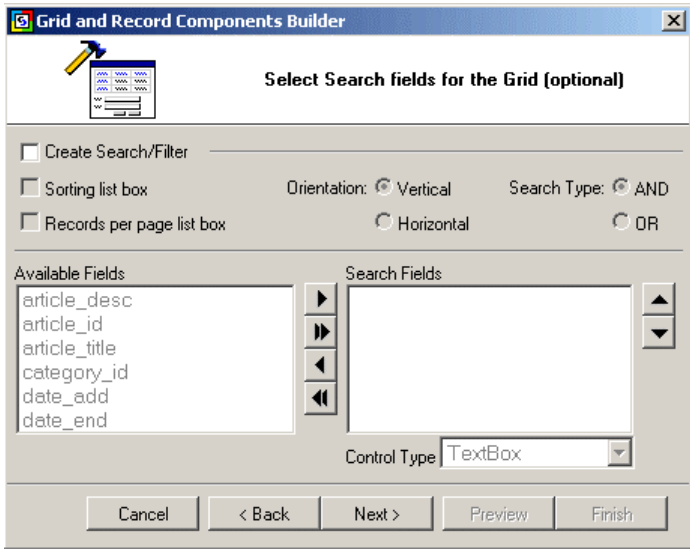



Property	Description
Data Source	Select the table from where the data will be retrieved.
Build Query	Clicking this button will bring up the <a href="#">Visual Query Builder</a> window where you can setup a SQL query that will act as the grid data source.
Refresh	Click this button to refresh the connection if you make changes to the database after having begun using the builder.
Available Fields	This listbox shows all the fields available in the table.
Fields for the Grid	These are the fields to be shown in the grid. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the grid.
	Click this button to add ALL Available Fields to the grid.
	Select a grid field then click this button to remove it from the grid.
	Click this button to remove all fields from the grid.
	Select a grid field then click this button to move it upwards in the order of appearance.
	Select a grid field then click this button to move it downwards in the order of appearance.








Order by	Optionally specify a field by which the records in the grid will be ordered by when initially displayed.
----------	--

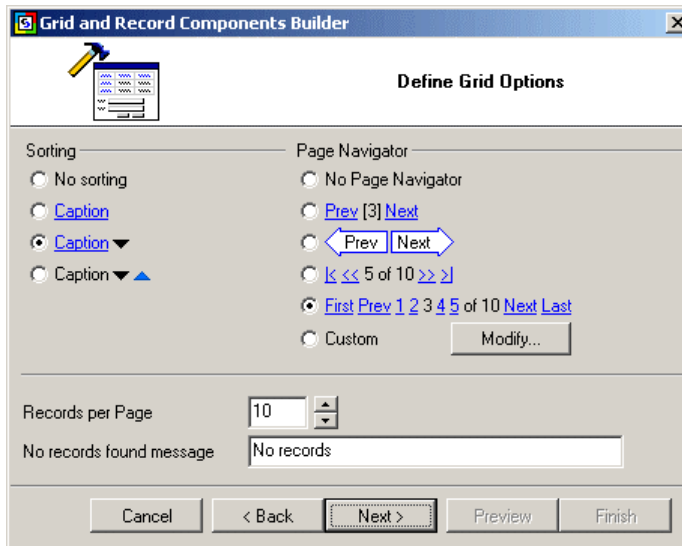
4. The next step is optional and should only be used if you want to have a search form that can be used to filter the records displayed in the grid. If a search form is not needed, simply click the **Next** button to proceed.



Property	Description
Create Search/Filter	Select this option to indicate your intention to create a search form.
Sorting list box	If selected, the search form will have two listboxes for sorting. The first one will contain a list of all the available fields so that the user can select the field to sort by. The user will also be able to indicate the desired sort order (ASC, DESC) using the second listbox.
Records per page list box	If selected the search form will have a listbox that can be used to specify the number of records to be returned per page after the search is conducted.
Orientation	Specify whether the search form should be vertically or horizontally oriented
Search Type	Specify the SQL operator to be used for the search. If AND is selected, the search form will have a form control for all the fields. Otherwise, if OR is selected, the search form will have only one form control. Any value entered in the single control will be used to search against all the fields in the form. You can use the preview button to see how the form looks like with either option selected.
Available Fields	This listbox shows all the fields available in the table or view.
Search Fields	These are the fields to be shown in the search form. They are selected from among the Available Fields.
	Select an Available Field then click this button to add it to the form.

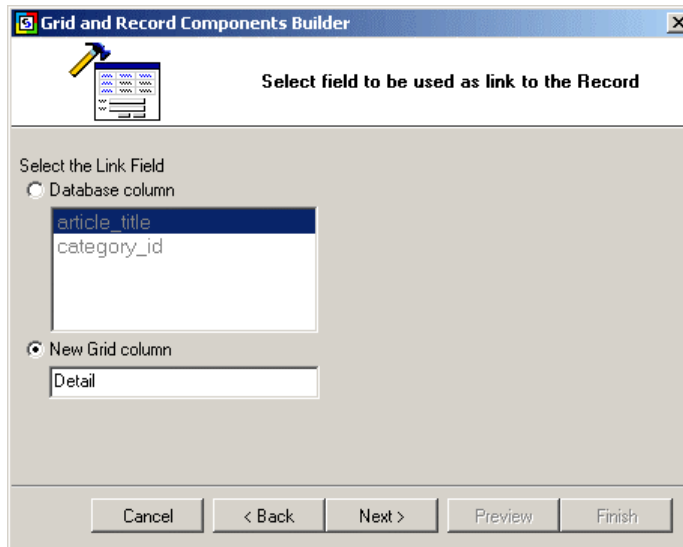
	Click this button to add ALL Available Fields to the form.
	Select a form field then click this button to remove it from the form.
	Click this button to remove all fields from the form.
	Select a form field then click this button to move it upwards in the order of appearance.
	Select a form field then click this button to move it downwards in the order of appearance.
Control Type	Specify the control type to be used for each of the fields in the form.

5. You can now specify the type of sorting and navigation controls to appear in the grid form.

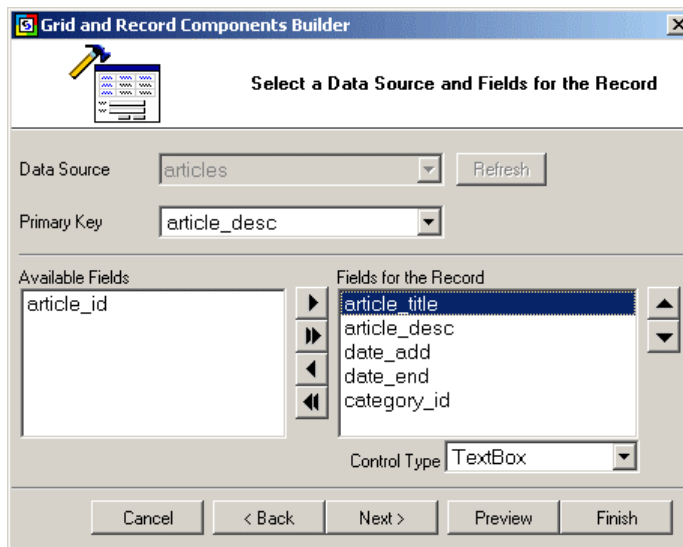


Property	Description
Sorting	Select the type of sorting controls that you want to have in the grid.
Page Navigator	Select the type of Navigation controls that you want to have in the grid.
Modify	If you want to have custom navigation controls on the form, click on this button to open the <b>Custom Navigator</b> window where you can customize the controls to suit your needs.
Records per page	Specify the default number of records to be shown per page.
No records found message	Specify the message that will be displayed in the grid if there are no records to be shown.

6. The last step in configuring the grid involves selecting the field to be used to link the records in the grid to the details in the record form. You can select any one of the fields that are already in the grid or alternatively, you can specify a new column whose sole purpose will be to act as the link to the record form.



7. The next set of windows pertains to the configuration of the record form. The record form will be based on the same table as was the grid form but you have the option of selecting different fields to appear in the record form. After adding the fields, you should also select the Control Type to be used for each of the fields.



At this point, you can use the Preview button to see how the forms look.

List of Articles

[Title](#) ▲▼ [Category Id](#) ▲▼

[Detail](#) {article\_title} {category\_id}

No records

[Add New](#) ◀ ▶ ▶▶

Add/Edit Articles

{Error}

Title

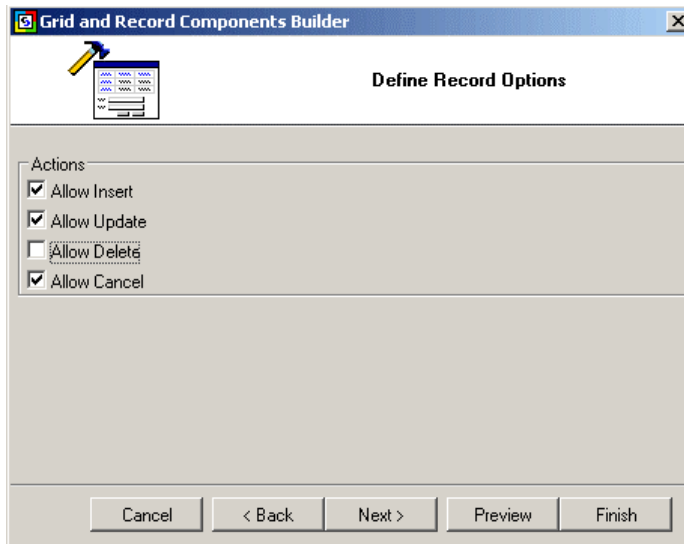
Desc

Date Add

Date End

Category Id

8. Next, select the type of operations you want the user to be able to perform using the record form. In the illustration below, the **Allow Delete** option is disabled meaning that the record form will not allow the user to delete any existing records.



**Grid and Record Components Builder**

**Define Record Options**

Actions

☒ Allow Insert

☒ Allow Update




☐ Allow Delete

☒ Allow Cancel

9. In the last step, select a theme to be used to display both forms.



Property	Description
Theme	This listbox contains a list of themes from which you can select.
Style Type	<ul style="list-style-type: none"> <li>HTML – Styles are defined using HTML attributes and properties.</li> <li>CSS Page – Cascading Style Sheet definitions are placed at the top of the HTML pages in the &lt;HEAD&gt; section.</li> <li>CSS File – Cascading Style Sheet definitions are placed in a separate file that is linked to the HTML pages.</li> <li>CSS Inline – Cascading Style Sheet definitions are placed within the HTML tags.</li> </ul>

Grid Layout	<p>Specify the layout of the grid form</p> <ul style="list-style-type: none"><li>▪ <b>Tabular</b></li></ul> <p><b>members</b></p>  <p>The tabular layout shows a table with two columns: 'first_name' and 'last_name'. Below the table is a 'No records' message and a set of navigation buttons (back, forward, search, etc.).</p> <ul style="list-style-type: none"><li>▪ <b>Columnar</b></li></ul> <p><b>members</b></p>  <p>The columnar layout shows a single column with labels 'first_name' and 'last_name' above their respective input fields. Below the fields is a 'No records' message and a set of navigation buttons.</p> <ul style="list-style-type: none"><li>▪ <b>Justified</b></li></ul> <p><b>members</b></p>  <p>The justified layout shows a single column with labels 'first_name' and 'last_name' above their respective input fields. Below the fields is a 'No records' message and a set of navigation buttons.</p>
Alternate records	If this option is selected, every other record in the grid will appear in a row with a different background color than the preceding record.
Record Separator	With this option selected, each row where a record appears is followed by an empty row that separates it from the next row with a record.

You can use the **Preview** button to see how the forms will look like then click on the **Finish** button to close the builder and generate the forms.

List of Articles

	Title	Category Id
Detail	{article_title}	{category_id}
No records		
<a href="#">Add New</a>		

Add/Edit Articles

{Error}

Title	{article_title}
Desc	{article_desc}
Date Add	{date_add}
Date End	{date_end}
Category Id	{category_id}

Add

Submit

Cancel

## VII. Object Properties

This section deals with the various properties exposed by the **Properties** Window. The process of creating and customizing pages revolves around configuring different settings for these properties.

Many of the properties encountered are applicable to multiple objects. For instance, a Text Box control shares many similar properties with a Hidden Field control. On the other hand, a Navigator control bears very few similarities with any other object.

### *Project Properties*

Clicking on the project name within the **Project Explorer** window will expose the project properties under the **Data** tab of the **Properties** window. Note that a project does not have any properties under the **Format** or **Event** tabs. In addition, most of the project properties can be accessed using the **Project** → **Settings...** menu option that opens the [Project Settings](#) window.

Property	Description
Code Language	Select the programming language in which the code will be generated.
Site Language	Specify the spoken language for the site.
Server URL:	The HTTP URL to be used to access the published pages.
Secured Server URL	Specify the SSL HTTP URL to be used to access the published pages. This option is only used if you want to access the pages over a SSL connection and the server support SSL.
Server Path	The filing system path to where the pages will be published.
Location:	Specify whether the generated pages will be published to a local computer or network or whether they will be published to a remote computer via FTP.
FTP Server	The address of the FTP host machine.
FTP Folder	The path to the folder in the FTP host machine where the pages will be uploaded.
FTP Login	The login name required to access the FTP host machine.
FTP Password	The password required to access the FTP host machine.
FTP Server URL	The HTTP URL that maps to the location on the FTP host where the pages are uploaded.
FTP Passive Mode	Whether or not the FTP session should use passive mode, for instance if connecting from behind a firewall.
Default Date Format	Select the format in which you want date values to be displayed in the project. If no selection is made, date values will be displayed in the default database format



Security Type	<ul style="list-style-type: none"> <li>• None: No site security is implemented</li> <li>• Table: Security will be based on information contained in a database table</li> <li>• Database: The inbuilt database security will be used for the site</li> </ul>
Login Page	Select the project page that will be used to enter authentication details by the users.
Convert URL To	Specify whether URLs should be automatically converted to absolute URLs or secure URLs for the SSL protocol (https://).
User ID Stored As	Select the method to be used to store the User ID after a user logs in (Session, URL, Cookies).
Security Connection	Select the connection from where security information will be retrieved.
User Table	Select the database table containing user security information.
User ID Field	Select the database table field containing the unique ID that identifies the users.
User Login Field	Select the database table field containing the login usernames of the users.
User Password Field	Select the database table field containing the login passwords of the users.
Level/Group Field	If applicable, select the database table field containing the security level of the users.
User ID Variable	Enter a name to be used for the user ID variable for a logged in user.
User Login Variable	Enter a name to be used for the user login name variable for a logged in user.
User Password Variable	Enter a name to be used for the user password variable for a logged in user.
Group ID Variable	Enter a name to be used for the group ID variable for a logged in user.
Level Inclusion	If this option is selected, users with a higher security level will be able to access pages designated with a lower security level. Otherwise, an exact match will be needed in order to access any page.
Character Set CodePage	Set the value for the code page used for setting the body of the HTTP request. This can be necessary when sending data in the body using a character set other than the system default.
Geographic Local LCID	The LCID is a locale identifier that identifies the language in use.

## Form Properties

The different types of forms can be divided into two main categories. Those that submit value (i.e. have a HTML <Form> element) and those that don't. Record, Search and Login forms belong in the first category while the second is composed solely of grid forms. Record, Search and Login forms therefore have similar properties and events while grid forms have their own distinct properties.

## Grid Properties

Property	Description
Name	The name of the form.

Connection	The database connection from where information will be retrieved.
Source Type	The type of data source or the method used to retrieve the data.
Data Source	The database column, SQL Query or other expression from where the data will be retrieved.
Restricted	Specify where users must be logged in before accessing the form.
Records Per Page	The numbers of records to be displayed per page by default.
Page Size Limit	The maximum number of records a user may opt to display per page if using a search form that allows control over the number of records displayed.

## Record, Search and Login Form Properties

Property	Description
Name	The name of the form.
Connection	The database connection from which information will be retrieved.
Source Type	The type of data source or the method used to retrieve the data.
Data Source	The database column, SQL Query or other expression from where the data will be retrieved.
Source Columns	The columns in the data source that will be available for usage in the form.
Return Page	The page to which the user is directed to after the form has been submitted successfully.
Convert URL To	Specify whether URLs should be automatically converted to absolute URLs or secure URLs for the SSL protocol (https://).
Restricted	Specify where users must be logged in before accessing the form.
Allow Insert	Specify where the form can be used to insert new data.
Allow Update	Specify where the form can be used to update existing data.
Allow Delete	Specify where the form can be used to delete existing data.
Preserve Parameters	Specify whether Get or Post parameters should be preserved.
Custom Insert Type	If you wish to override the default insert mechanism, select the type of method to be used to perform the insert operation.
Custom Insert	Specify the procedure to be used to insert data.
Custom Update Type	If you wish to override the default update mechanism, select the type of method to be used to perform the update operation.
Custom Update	Specify the procedure to be used to update data.
Custom Delete Type	If you wish to override the default delete mechanism, select the type of method to be used to perform the delete operation.
Custom Delete	Specify the procedure to be used to delete data.

## Control Properties

This section examines the properties for the various controls found under the **Control** tab of the

**Toolbox.** Many of the Controls share similar properties and the table below presents all the possible properties in alphabetic order. The third column indicates the controls that possess the property.

Property	Description	Applicable Controls
Bound Column	Each option in a ListBox or RadioButton control is represented by a pair of values whereby one value is displayed to the user and another is submitted. Bound Column refers to the data source for the values that are submitted.	ListBox, RadioButton
Caption	The name used to refer to the control in an error message.	TextBox, TextArea, ListBox, RadioButton, Hidden
Checked Value	The value submitted by a Checkbox Control when it is checked.	Checkbox
Column	For a Sorter control, this specifies the Grid column that the sorter applies.	Sorter
Connection	The database connection from where information will be retrieved.	ListBox, RadioButton
Control Source	Specify the source of data for the control e.g. the name of a database column	TextBox, TextArea, ListBox, Checkbox, RadioButton, Label, Image, Hidden, Link, ImageLink
Convert URL To	Specify whether the URL should be converted to an absolute URL or a secure (https://) URL.	Button, ImageLink, Link
Data Type	The type of data that will be contained in the control: <ul style="list-style-type: none"> <li>• Text</li> <li>• Integer</li> <li>• Float</li> <li>• Date Boolean</li> <li>• Memo</li> </ul>	TextBox, TextArea, Checkbox, RadioButton, ListBox, Label, Hidden, Link, ImageLink

DBFormat	Specify the format in which you want the control value to be entered into the database e.g. the user may enter a date value in one format but you want it to be entered into the database in a different format.	TextBox, TextArea, Label, Hidden
Default Button	[Yes/No] If you select 'Yes' the button will be the default button on the form so that clicking on the Enter key will be equivalent to clicking on the button.	Button
Default [Value]	Specify a default value to be used for the control if no value is otherwise specified.	TextBox, TextArea, Checkbox, RadioButton, ListBox, Label, Image, Hidden, Link, ImageLink
Enable Validation	[Yes/No] Specify whether validation should be enabled when the button is clicked.	Button
Error Control	Specify the name of a control that will be used to display any error messages pertaining to the	TextBox, TextArea, Checkbox, RadioButton, ListBox, Hidden
Format	Depending on the Data Type property, select the format of the data to be contained in the control.	TextBox, TextArea, Label, Hidden, Link, ImageLink
Href Source	The page path, database column or other expression containing the URL of the hyperlink.	ImageLink
Href Type	The type of source where the URL value will come from.	ImageLink, Link
HTML	[Yes/No] If 'Yes' is selected, the control value will not be converted to text before being displayed. As such, if the value is HTML markup, it will be appear as regular HTML markup on the page.	TextArea, TextArea, Label, Link, ImageLink
Input Verification	Select a regular expression to be used to check the validity of the data entered in the control.	TextBox, Hidden

List Data Source	Specify the actual source of data for the list of values depending on the List Source Type.	ListBox, RadioButton
List Source Type	Specify the type of source of data for the list of values to be shown in the control e.g. the name of a database column.	ListBox, RadioButton
Name	The name of the control.	ALL
Operation	The type of operation performed when the button is clicked [Insert, Update, Delete, Cancel or Search]	Button
Order By	Specify an Order By clause for the SQL statement used to retrieve values for the control.	RadioButton, ListBox
Page Links	Specify the number of page links to be shown on the Navigator. The default is 10.	Navigator
Preserve Parameters	Specify whether GET or Post parameters should be preserved.	ImageLink, Link
Remove Parameters	Specify a semicolon-separated list of parameters that should be removed from the hyperlink.	ImageLink, Link
Required	[Yes/No] Select 'Yes' if a value must be entered for the control in order for the form to be submitted successfully.	TextBox, TextArea, RadioButton, ListBox
Return Page	Specify the page where the user should be directed after the button operation has been performed.	Button
Source Type	The type of data source that will provide data for the control: <ul style="list-style-type: none"> <li>• Database Column</li> <li>• Database Expression</li> <li>• Code Expression</li> </ul>	TextBox, TextArea, ListBox, Checkbox, RadioButton, Label, Image, Hidden, Link, ImageLink
Source/Parameters	This opens the <b>Href Source</b> window where you can specify the page to be linked to as well as any parameters that should be transmitted via the URL.	Link
Text Column	Each option in a ListBox or RadioButton control is represented by a pair of values whereby one value is displayed to the user and another is submitted. Text Column refers to the data source for the values that are displayed.	Listbox, RadioButton
Type	Select the type of page links to be shown in the Navigator control.	Navigator

Unchecked Value	The value submitted by a Checkbox Control when it is unchecked.	Checkbox
Unique	[Yes/No] Select 'Yes' if you want a check to be made to ensure that any value entered in the control does not already exist in the database.	TextBox, TextArea, RadioButton, ListBox, Hidden
Validation Rule	Specify an expression that will be used to check the validity of the data entered in the control.	TextBox, TextArea, Checkbox, RadioButton, ListBox, Hidden
Validation Text	Specify the text that will be displayed if the data entered in the control does not meet the validation rule.	TextBox, TextArea, Checkbox, RadioButton, ListBox, Hidden
Where	Specify a Where clause for the SQL statement used to retrieve values for the control.	RadioButton, ListBox

## ***HTML Properties***

The properties that appear under the **Data** tab of the **Properties** window pertain to objects that are created using the **Forms** tab of the **Toolbox**. However, the **Format** tab of the **Properties** windows also exposes many other properties that apply to HTML elements in general. When the **Document** window is in **HTML** mode, the **Format** tab of the **Properties** window displays HTML properties for the HTML element where the cursor is currently located.

Different HTML elements have different properties, with some properties being common to multiple elements. You will also notice that along with the properties, there are JavaScript Events located at the bottom of the **Properties** window **Format** tab. Events vary according to the selected element and can be used to execute JavaScript code in response to certain actions.

Both the properties and events under the **Format** tab are standard HTML properties and events and as such, you can refer to suitable HTML documentation for their meaning and usage. The following are some reference that you may find useful:

Microsoft DHTML Reference	<a href="http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtmlrefs.a">http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/dhtmlrefs.a</a>
W3C TML 4.01 ecification	<a href="http://www.w3.org/TR/html401/">http://www.w3.org/TR/html401/</a>

## Connection Properties

Selecting a project connection in the **Project Explorer** window will expose its properties under the **Data** tab of the **Properties** window. Connections have no Events nor **Format** properties. The following are the available connection properties.

Property	Description
Name	The name of the connection. Has to be unique.
Design Database Type	The type of database being accessed during design time. If your database type is not listed, select the ANSI SQL-92 option.
Connection Type	The type of connection. ODBC or Connection String.
Connection String	Depending on the type of connection, this can be either an ODBC DSN or a connection string.
Login	The user login name needed to access the database.
Password	The user password needed to access the database.
Date Format	The format in which date values should be entered into the database.
Boolean Format	The format in which Boolean values should be entered into the database.
Catalog	Specify a database catalog from which to display objects during design time.
Schema	Specify a database schema from which to display objects during design time.
Views	[Yes/No] Select 'Yes' to enable the display of database views during design time.
Synonyms	[Yes/No] Select 'Yes' to enable the display of database Synonyms during design time.
System Tables	[Yes/No] Select 'Yes' to enable the display of database System Tables during design time.
System Views	[Yes/No] Select 'Yes' to enable the display of database System Views during design time.
Aliases	[Yes/No] Select 'Yes' to enable the display of database aliases during design time.
Server Same As Design	[Yes/No] Select 'Yes' if the same connection will be used for design time as well as on the server.
Server Connection Type	If the server connection is different from the design connection, specify whether the server connection will use an ODBC DSN or connection string.
Server Connection String	If the server connection is different from the design connection, specify the ODBC DSN or connection string to be used.

Server Login	If the server connection is different from the design connection, enter the user login name needed to access the server database.
Server Password	If the server connection is different from the design connection, enter the user password needed to access the server database.



# VIII. Working with Documents

A good deal of the time spent using CodeCharge Studio will involve the use of the document window. The document window serves multiple purposes depending on the mode in which it is currently set to. The five different modes in which the document window can be set are:

- **Design:** This is a visual representation of the HTML content of a page as it would appear if rendered by a web browser. The user is able to add, modify and remove different forms of the page. The Design mode is also known as ‘What You See Is What You Get’ (WYSIWYG).
- **HTML:** This shows the actual HTML code for a page as well as any style and Javascript code that might be on the page. If you are familiar with HTML, you can add your own HTML code or adjust the existing code as desired. Switching to Design mode will show the appearance of the HTML code when rendered in a browser.
- **Code:** This shows the language specific code for a page. For instance, if using the ASP & Templates language, this mode will display the ASP/VBScript code for a page.
- **Preview:** This mode is used to preview a page without actually running it off the server. The page is shown without the special tags or borders that are visible in Design mode. The user cannot make any changes to the page while in Preview mode.
- **Live Page:** This involves actually executing the page from the server. Any database content, include pages and styles are displayed in their entirety. The page is fully functional and can be used to navigate, post or adjust database content or perform any other function that the page is designed to have. Error messages are displayed if problems are encountered with any of the page code.

## Design Mode



















Design mode allows the user to construct a page while being able to immediately see the appearance of the different components on the page. As the name implies, Design mode is more concerned with the appearance of the page as opposed to it functionality.

While in design mode, you can use the Toolbox to add various components to the page such as form controls, plain text and images. Tables and cascading style sheets can be used to format the layout of the page.

The **Properties** window displays the pertinent properties based on the component that is selected in the document window. You can set or adjust these properties as needed.

While in design mode, a number of icons are used to give a visual representation of otherwise non-visual elements. The following table shows the meaning of these icons.

Icon	Represents
------	------------

 	Beginning and end paragraph tags <p> </p>.
	Line break tag  .
 	Beginning and end form tags <form> </form>.
 	Submit button <input type="submit">.
 	An Include page.
 	Mark the beginning and end of a Grid form .
 	Mark the beginning and end of a Record, Login or Search form. These forms submit values and have a HTML form (<form></form>) within them.
 	A Label control based on database content.
 	A Link control based on database content.
	Represents an ImageLink or Image control. The actual image will be visible when the page is viewed in Live Page mode.

While in design mode, you can toggle the display of the Paragraph and Line break icons by pressing the *ctrl*+/ *shortcut key or using the View → Reveal Tags menu option.*

You will also notice that some of the text displayed will be enclosed within curly braces (e.g. {article}). The content in curly braces are termed as **Blocks** and are indication that the text will be substituted for database content when the page is viewed live. Care should be observed when handling any blocks since if they are inadvertently changed, then the corresponding database content would not be displayed. As a rule, you should always use the **Properties** window or the **Project Explorer** window to change the properties of this content or if you need to rename or remove it. Note that you can change the position of the blocks relative to the other page contents without causing any harm. What could be potentially harmful is changing the block itself i.e. spelling, or deleting a block.

## HTML Mode

This displays the raw HTML code for a page. Along with HTML content, there could be Javascript code that was added manually or using the events tab of the **Properties** window. For pages that make use of Cascading Style Sheets, style content will also be visible depending on the method used to apply the style.

Just as was the case in **Design** mode, you will see content within the HTML code that is enclosed within curly braces. Caution should be observed when editing this content and this should only be done using the **Properties** window or **Project Explorer** window. In HTML mode, there is another form of block that is not evident in Design mode. This occurs in the form of HTML comments and is used to mark out the beginning and end of content that can be viewed as a logical unit.

```
<!--BEGIN blockname-->
```

```
Block contents go here
```

```
<!-- END blockname -->
```

There are numerous types of logical units that are represented by blocks. The common denominator is that there is a HTML comment to mark their beginning and another to mark their end. Additionally, it is possible to have nested blocks such that one block is within another. In such scenarios, the naming of the blocks makes it clear where each of the blocks begins and ends. It is therefore important to make sure that none of the block comments are altered or removed otherwise the page would not function as correctly.

Any client side events that are added using the **Events** tab of the **Properties** window are also visible in HTML mode. The script code appears at the top of the page in the <head> </head> section. The illustration below shows the code that is added for an On Load event of a page to set focus to a field in a form.

```
<script language="JavaScript">
//Begin CCS script
function page_OnLoad() { //page_OnLoad @1-1C723708

//Set Focus @5-84B91275
    document.Form1.TextBox1.focus();
//End Set Focus

} //Close page_OnLoad @1-FCB6E20C

//bind_events @1-F4D605D7
function bind_events() {
    page_OnLoad();
}
//End bind_events

window.onload = bind_events; //Assign bind_events @1-19F7B649

//End CCS script
</script>
```

Notice that event code is displayed with a different background color compared to the rest of the HTML content. Usually the background color is light gray and signifies that the code was generated by CodeCharge Studio and has not been altered. If you modify or add your own code anywhere within the gray background, the background will change to the default color of the rest of the document. This is an indication that the code is different from what CodeCharge Studio generated.

## Code Mode

Code Mode displays the programming code for a page. By default, the code in a page is displayed against a gray background to indicate that it has not been altered from its original state. Making changes to the code will cause the background to change to white by default. In addition to this, all the code will be color coded

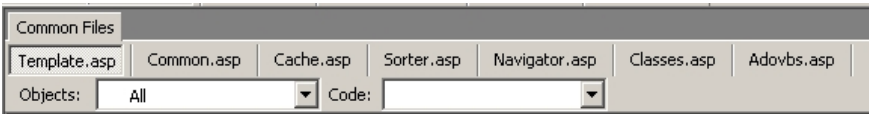
in order to distinguish various parts of the code. For instance, commented code will appear in a different color compared to Strings or even reserved words.

It is usual to have pages with upwards of a thousand lines of code. To assist in navigating large amounts of code, the document window makes it possible to jump to specific points in the code by selecting an object then a member of the object to bring into focus.



Whenever appropriate, the two listbox fields at the top left of the document window will display the Objects in the page as well as the members of that object if any. Selecting a particular **Object** then its member in the **Code** field will cause the window to scroll to the location where the relevant code is located.

In each project, regardless of coding language, there is always a set of **Common Files**. These files contain functionality that is needed in multiple pages and hence it is placed in central files. In order to access these files, click on the **Common Files** Icon in the **Project Explorer**. The files will be opened in the document window and you will be able to access a specific file by clicking on the tab with its name.



The number and names of the Common Files will vary depending on the coding language being used. The illustration above shows the common files for the ASP language.

## Preview Mode

Preview mode shows how the static HTML part of a page would look like when viewed in a browser. None of the programming code is executed or published to the server. Preview mode is intended to show the layout of the page without the editing features that are visible in HTML mode. It is not possible to make changes to a page while in Preview mode.

## Live Page

In Live Page Mode, the page is published to the server and executed such that dynamic content is made available for viewing. Before attempting to view a page in Live Page mode, you should make sure that the Project Publishing settings have been properly configured so that the pages are generated to the correct location.

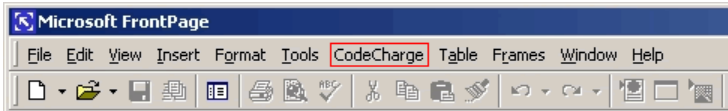
In this mode, you can navigate to other pages and manipulate database content as you would normally do if the page were viewed using a browser window.

If problems are encountered during the process of generating or publishing the page, the **Messages** window will show appropriate error messages. Additionally, if the generated pages themselves are not properly created, the document window will show the error message returned from the server.

### Using the Microsoft FrontPage Add-In

If you installed the FrontPage Add-in, you can take advantage of the robust HTML and code editing functionality provided by FrontPage to enhance the development process. The Add-In has the effect of adding the main CodeCharge Studio functionality to FrontPage so that you can create entire projects using the combined power of CodeCharge Studio and FrontPage.



When you open FrontPage, you will notice that the menu bar has a menu for CodeCharge as shown in the illustration below.



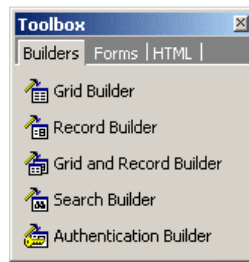
In order to bring up the other CodeCharge Studio toolboxes and windows, you need to open an existing project or create a new one. Once you open or create a project, the **Project Explorer** and **Properties** windows will become visible as well as other CodeCharge Studio toolbars. Using the CodeCharge → Windows menu option, you can also make the **Message** and **Toolbox** windows visible.

With the exception of two commands that are unique to the FrontPage add-in toolbar, the rest of the commands in the toolbar shown below are similar to the commands within CodeCharge Studio.



Command	Description
	Use this button to open a window with the programming code for a particular page.
	Use this button to generate the current page and view it in live page mode.

The add-in also features the CodeCharge Studio **Toolbox** with the three tabs for Builders, Forms and HTML controls. The options available in the toolbox are similar to those available within CodeCharge Studio itself.

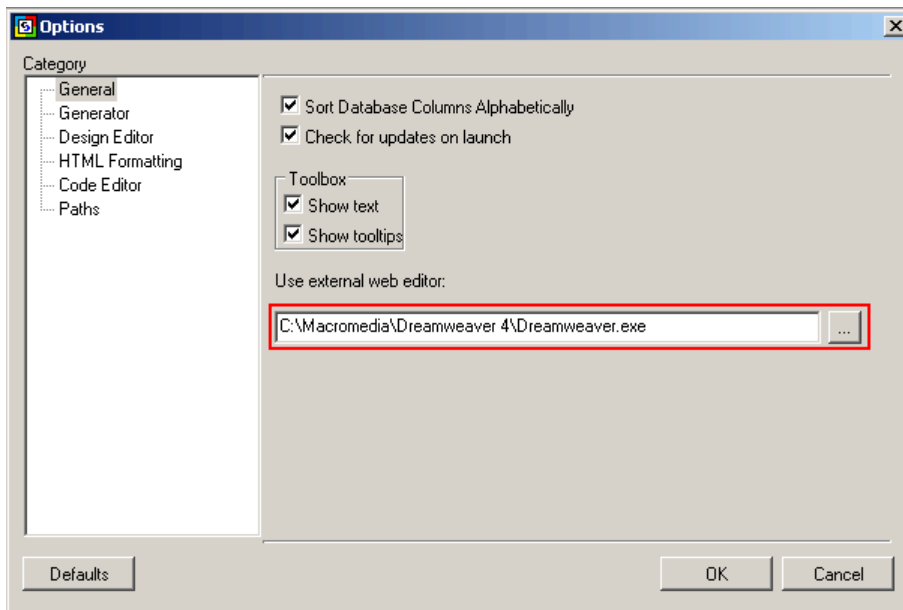


Note that FrontPage workspace is very customizable and you can arrange the various windows as well as toolbars in the manner that best accommodates your working requirements.

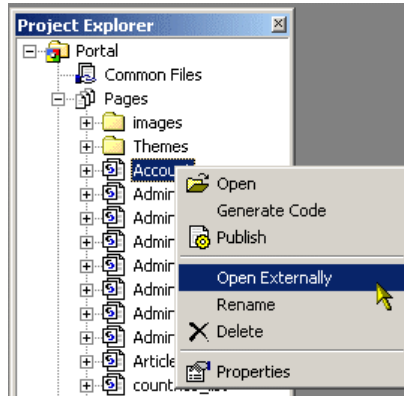
While using the FrontPage add-in confers considerable leverage to performing various design tasks that would otherwise be tedious within CodeCharge Studio, caution should be exercised so as not to alter the blocks of code that have to remain intact in order for the pages to function properly. Any content within curly braces {example}, should not be directly edited and neither should the HTML nor the code comments.

## ***Using External HTML Editors***

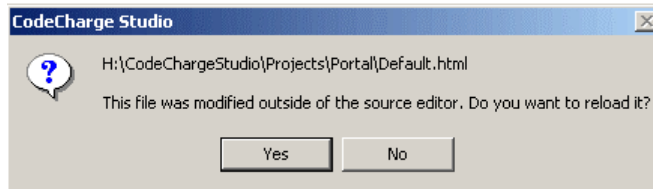
Besides using Microsoft FrontPage to edit HTML content, you could also configure CodeCharge Studio to work in conjunction with the HTML editor of your choice. You can do so by using the **Tools→Options...** menu to specify the location of the program you want to use.



Once you specify the external web editor to be used, you can then open a page in that editor by right clicking on the page name within the **Project Explorer** and selecting the **Open Externally** option.



With the document open in the external editor, you can proceed to make changes and adjustments to the document. While doing so, be cautious so as not to alter any of the CodeCharge Studio blocks that appear within curly braces or in HTML comments. When you are ready to return to CodeCharge Studio, save and close the document in the external editor then switch back to CodeCharge Studio. A window will appear prompting you that the document was altered outside CodeCharge Studio. Click on the **Yes** button to reload the document with the changes made in the external editor.



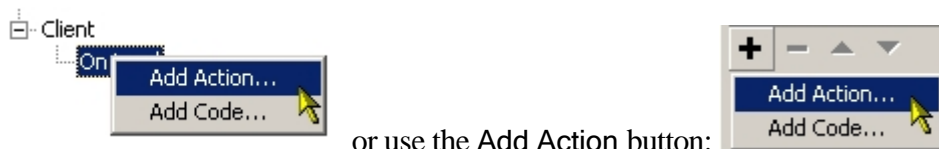
## IX. Client Side Event Actions

The **Properties** window features two categories of events. There are the **Server** events that are implemented in programming language and there are **Client** events that are implemented using browser-scripting languages. The former are executed entirely on the server before a page is sent back to the requesting browser while the latter are executed by the browser of the client machine that requested the page.

By their nature, client side events are used mainly to provide interactivity without having to post feedback to the server. A client side script can respond to a user action and continuously provide feedback entirely on the client side. This is made possible by the event-driven nature of HTML objects that provide a mechanism to respond to particular actions using event handlers. In fact, all the client events are event handlers for a corresponding browser document object. The event names are not unique to CodeCharge Studio but rather are the same names used in the document object model (DOM) for a browser document.

There are many common tasks often performed in response to events and for this reason, CodeCharge Studio provides a number of **Actions** that are essentially snippets of code that implement often used functionality. Instead of having to hand code the functionality each time it is needed, you can simply add an Action and customize it for the particular situation.

Actions are added via the events listed under the **Events** tab of the **Properties** window. You can either right-click on a client event then select the Add Action... option:



or use the Add Action button:

Where an action is placed will affect its execution or failure to execute. When placing an action, you should bear in mind the event that you anticipate to trigger the action. If the event does not occur, then the action will not be executed.

### ***Set Focus***

This action is used to set focus to a particular object in the window. For a input field, having focus means that anything typed by the user will be entered into the field. For a button, having focus means that pressing the enter key is equivalent to clicking on the button.

Use this action to set focus to a particular field for the convenience of the user or in cases where you want input to begin at a particular field.



When you add this action, there are two properties you have to provide.

Property	Description
Form	The name of the form in which the control to be focused on is located.
Control	The name of the control to receive focus.

## ***Validate Entry***

This action is used to setup one or more validation conditions for a field. Multiple fields within a form can have their own Validate Entry actions. After the Validate Entry actions have been setup, the Validate Form action is used to execute all the validations.

Property	Description
Required	Specify whether a field value is required or not.
Minimum Length	Specify the minimum acceptable field value.
Maximum Length	Specify the maximum acceptable field value.
Input Mask	Specify an input mask to be used to determine the validity of the field value.
Regular Expression	Specify a regular expression to be used to determine the validity of the field value.
Error Message	Specify the error message to be displayed if an invalid entry is made.
Validation Function	Specify the name of a custom validation function to be executed.

## ***Validate Form***

This action makes use of the inbuilt CodeCharge Studio validation functionality contained in a Jscript include file called functions.js. Within this file, a function is called to validate any conditions that have been set for the fields within the forms using the Validate Entry action. The possible conditions are:

- Required field value.
- Minimum field value.
- Maximum Field value.
- Check against a regular expression.
- Call a custom function to validate the field.

If any of the conditions are not satisfied, the function displays an appropriate error message.

## ***Validate Maximum Length***

This action is usually used in conjunction with input fields and serves to ensure that the value entered in a field does not exceed a specified number of characters. Among other places, this action could be placed in the On Click event of a button in the form or the On Change event of the field being validated.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Maximum Length	The maximum allowable length of input in the control.
Error Message	The error message to be displayed if the maximum length is exceeded.

### ***Validate Minimum Length***

This action is usually used in conjunction with input fields and serves to ensure that the value entered in a field is at least a specified number of characters long. Among other places, this action could be placed in the On Click event of a button in the form or the On Change event of the field being validated.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Minimum Length	The minimum allowable length of input in the control.
Error Message	The error message to be displayed if the input is less than the minimum length.

### ***Validate Maximum Value***

This action is usually used in conjunction with input fields that receive numeric values. The action ensures that the value entered is less than a specified maximum value. Among other places, this action could be placed in the On Click event of a button in the form or the On Change event of the field being validated.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Maximum Value	The maximum allowable input value.
Error Message	The error message to be displayed if the input value exceeds the maximum value.

### ***Validate Minimum Value***

This action is usually used in conjunction with input fields that receive numeric values. The action ensures that the value entered is more than a specified minimum value. Among other places, this action could be placed in the On Click event of a button in the form or the On Change event of the field being

validated.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Minimum Value	The minimum allowable input value.
Error Message	The error message to be displayed if the input value exceeds the maximum value.

### ***Validate Required Value***

This action is usually used for any control that accepts input and ensures that the control has a value. The action can be placed in the On Click event of a button among other places.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Error Message	The error message to be displayed if the control has no value.

### ***Regular Expression Validation***

This action is used to validate the input in a control based on a specified regular expression.

The following are the properties that need to be configured for the action.

Property	Description
Control Name	The name of the control to be validated.
Regular Expression	The regular expression to be used for validation.
Error Message	The error message to be displayed if the control has no value.

### ***Custom Code***

This action essentially allows you to enter your own code to be executed when the event occurs.

```
//Custom Code @41-2A29BDB7
// -----
// Write your own code here.
// -----
//End Custom Code
```

## X. Appendix A

### *Java and JSP Implementation Architecture*

CodeCharge Studio comes with its own extensive framework that makes up the core of the generated Java web applications. The framework is based on the Model-View-Controller paradigm described at [http://java.sun.com/blueprints/patterns/j2ee\\_patterns/model\\_view\\_controller/](http://java.sun.com/blueprints/patterns/j2ee_patterns/model_view_controller/). The model utilized in CodeCharge Studio generated applications consists of Web Controls implemented as Java Beans. The Controller layer is constructed from the ControllerServlet class and page Actions. The View is represented by the ViewerServlet class (in Servlet implementation) or by JSP pages (in JSP implementation).

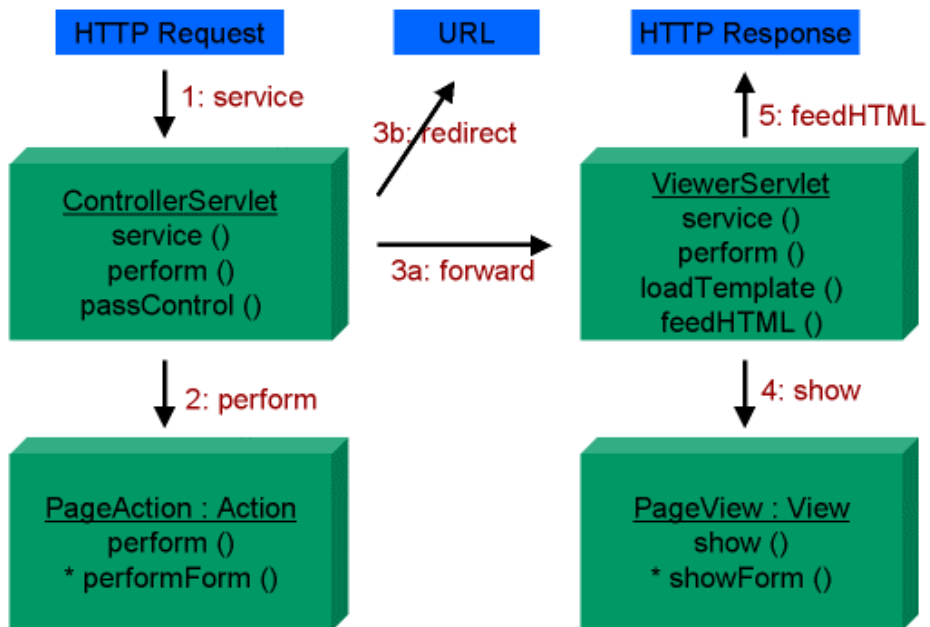
While other similar frameworks are usually implemented as a “closed” environment, the CodeCharge Studio framework is fully extensible and allows the developer to modify the included components (Web Controls) or even create new ones.

From a single project model, CodeCharge Studio generates two types of Java code, both of which support the total separation of code from the design aspect:

- **Java Servlets with HTML Templates:** Once on the server, the Servlet loads the appropriate HTML Template and replaces the static HTML components with dynamic database output.
- **JSP and XML model:** XML files contain configuration parameters for each Web Control included on the page. The JSP pages use these parameters to output the appropriate HTML.

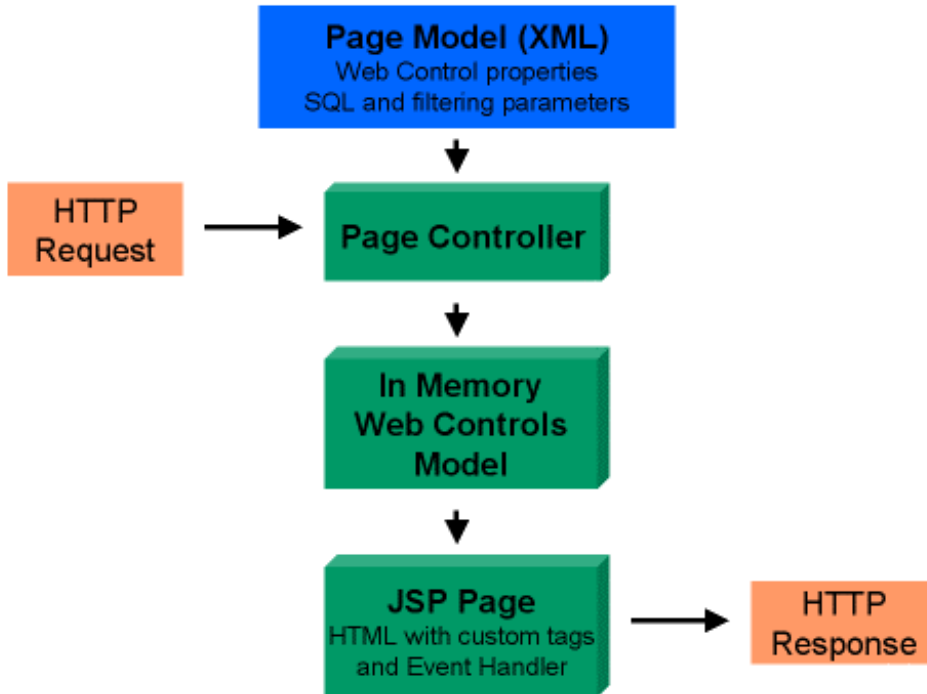
### **Java Servlet Control Flow**

The diagram below shows how a component’s state (Model) proceeds from the moment a request is made to the point where the page is presented for viewing. PageAction action and PageView view have access to the loaded HTML Template, HttpServletRequest and DBConnectionManager (connection pooling) objects.



## JSP and XML Model

The diagram below shows how the page model stored in XML format is translated into an HTML page (HTTP Response) with the use of the Page Controller that populates and updates the In-Memory Model object. A JSP page reads the In-Memory Model and outputs the data in HTML format.



## Approaches

### Model-View-Controller (MVC)

Framework design is based on the Model-View-Controller paradigm.

### Connection Pooling

Allows increase in performance by saving database connections in a pool. In case the server's drivers don't support pooling, the CodeCharge Studio framework offers its own pooling class.

### Events

All CodeCharge Studio Web Controls utilize events that allow users to extend the application's functionality by implementing event listeners.

### Data Access layer

Implemented as a Java Bean, business logic is separated from data. This facilitates an easy switch to other Data Sources.

### Templates

The design aspect is separated from the programming code.

### XML Model

JSP uses XML representation of the Page model. This describes all controls' properties and how to retrieve their data from the database. By editing the appropriate XML files, users can change the properties and behavior of the page and its controls.

## Features

### Logger

Allows logging of application's activity with different levels of verbosity. The user can easily detect application errors and where they occur. The logger can be disabled to increase performance, or its output can be redirected to a file.

### Data Source JDBC extension

Allows the use of predefined connections stored in JNDI. Users don't need to bother with connection details such as database name, driver class, login, and other parameters of a JDBC connection. This allows for easy integration with most application servers.

### Data formatting and validation

Supports Date, Numeric, Boolean and other formats through the utilization of format patterns. This allows for easy implementation of validation operations such as: required, unique, min/max length, min/max value, regular expression, date, etc.

### Passing of parameters

Allows for easy interaction between pages and forms. The Developer can indicate what parameters to preserve (QueryString, POST, ALL) and which to remove (by name). This is especially helpful in preserving the state of the sorter, navigator and search criteria between pages.

### Default values and Code Expressions

SQL parameters can be obtained from different sources: Request, Session, Application, CodeExpression (any java expression). If a parameter is empty, the default value is applied.

### Auto-construction of SQL statements based on parameter values

Automatically constructs the appropriate SQL statements depending on the values of parameters. This saves time in implementing complex SQL queries.

### Predefined Actions

Semi-generated code snippets that can be used to store and retrieve session values, validate data input, perform Login and Logout actions, etc. The user is therefore spared the task of implementing the functionality manually, yet still retains the option to include custom actions when necessary.

## Components

### Web Controls

Web Controls represent html controls and other higher level components such as Grid and Record forms, Navigator and Sorter. They are implemented as Java Beans and allow for manipulation of the control's behavior on the server, during run-time.

### Tags

Simplify the structure of JSP pages and provide reusability.

### ControllerServlet

Decides how to dispatch requests to an Action or View based on the RequestURL.

### Action abstract class

Provides base functionality for Page Action. Simplifies the development of new Actions.

### View abstract class

Provides the base functionality for Page View. Stores the responses and other objects needed to render the View.

### JDBCConnection class

Helper class that encapsulates the use of the JDBC API and provides an interface for different Data Sources (Pooled Connection or JDBC 2.0 Data Source connection).



## Requirements

- JDK 1.3 or 1.4 (<http://java.sun.com/j2se/1.3/download.html> )
- Servlet 2.2 and JSP 1.1 (<http://java.sun.com/products/servlet/download.html> )
- Ant 1.4 (<http://jakarta.apache.org/ant/index.html> )
- JDBC 2.0 Optional package (<http://java.sun.com/products/jdbc/download.html> ) if you choose to use the Data Source extension
- JAXP 1.1 and SAX 2.0 if you choose to generate JSP (<http://xml.apache.org/crimson/index.html>) (<http://java.sun.com/xml/index.html> )

The JAVA\_HOME environment variable should indicate the path to the JDK that should be used to build the projects. The ANT\_HOME environment variable or Ant Home site property should indicate the path to the Ant build tool installation directory.

## CLASS PATH REQUIREMENTS (BOTH COMPILATION TIME AND RUN TIME)

### JSP

The Class Path should contain Servlet 2.2 API library, JAXP 1.1 library, SAX 2.0 parser (such as crimson or xerces) and JDBC 2.0 extensions library if you want to use Data Sources.

### Servlet Templates

Class Path should contain Servlet 2.2 API library and JDBC 2.0 extensions library if you want to use Data Sources.

## Tomcat 4.0 scenarios

Check that the DB drivers are in the class path of your server. Choose from one of the following scenarios and customize the underlined properties:

### Scenario 1. Auto-deployment from webapps directory (DataSource extension is not used)

- Before generating the code, ensure that the “Use Data Source” site property is set to “No”. This is in the Project Explorer window.
- Fill in connection properties to indicate Driver class name, Database url, user login and password.
- Ensure that your database driver is in your application server class path.
- Place the generated war file at the location where the appBase attribute of the Virtual Host configuration element points. Usually, it is <TOMCAT\_HOME>/webapps.

### Scenario 2. Deployment through Context configuration (DataSource extension is not used)

- Add the Context element as a child to the Host element in your server.xml configuration file.

```
<Context path="/JSPPortal" docBase="JSPPortal.war">
</Context>
```

The path attribute is the context path of your application while docBase is the absolute or relative path to the war file from the appBase.

### Scenario 3. Deployment through Context configuration (DataSource extension is used)

- Before code generation, verify that the “Use Data Source” site property is set to “Yes”.
- Set the connection “DataBase URL” property to “java:comp/env/jdbc/Connection1” where Connection1 is the name of your connection.
- Define Context element as described in the previous scenario.
- Add a Resource child element with ResourceParams to the Context element:

```
<Resource name="jdbc/Connection1" auth="Container"
type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/Connection1">
  <parameter><name>user</name><value>User Name</value></parameter>
  <parameter><name>password</name><value>User
Password</value></parameter>
  <parameter><name>driverClassName</name>
  <value>Driver Class Name</value></parameter>
  <parameter><name>driverName</name>
  <value>Database Connection URL</value></parameter>
</ResourceParams>
```

The name attribute is the connection name as indicated in your project, with the “jdbc/” prefix prepended.

## Resin 2.0 scenarios

Check that the DB drivers are in the class path of your server. Choose from one of the following scenarios and customize the underlined properties:

### Scenario 1. Auto-deployment from webapps directory (DataSource extension is not used)

- Before code generation, verify that the “Use Data Source” site property is set to “No”.
- Fill in connection properties to indicate Driver class name, Database url, user login and password.
- Ensure that your database driver is in your application server class path.
- Place the generated war file at the location where the war-dir element of the Virtual Host configuration element points. Usually, it is <RESIN\_HOME>/webapps.

## Scenario 2. Deployment through web-app configuration (DataSource extension is not used)

- Add the Context element as a child to the Host element in your resin.conf configuration file.

```
<web-app id="/JSPPortal"  
  app-dir="c:/projects/Portal/JSPPortal.war"/>
```

The id attribute is the context path of your application while app-dir is the absolute or relative path to the war file from the app-dir.

## Scenario 3. Deployment through web-app configuration (DataSource extension is used)

- Before code generation, verify that the “Use Data Source” site property is set to “Yes”.
- Set the connection “DataBase URL” property to “java:comp/env/jdbc/Connection1”, where Connection1 is name of your connection.
- Add the resource-ref element as the child of the web-app element:

```
<resource-ref>  
  <res-ref-name>jdbc/Connection1</res-ref-name>  
  <res-type>javax.sql.DataSource</res-type>  
  <init-param driver-name="Driver Class Name" />  
  <init-param url="Database connection URL" />  
  <init-param user="User Name" />  
  <init-param password="User Password" />  
  <init-param max-connections="20" />  
  <init-param max-idle-time="30" />  
  <init-param enable-transaction="false" />  
</resource-ref>
```

The res-ref-name attribute is the Connection name as indicated in your project with the “jdbc” prefix prepended.

## WebLogic 7.0 scenarios

Check that the DB drivers are in the class path of your server. Choose from one of the following scenarios and customize the underlined properties:

### Scenario 1. Auto-deployment from applications directory (DataSource extension is not used)

- Before code generation, verify that the “Use Data Source” site property is set to “No”.
- Fill in connection properties to indicate Driver class name, Database url, user login and password.
- You can quickly deploy your application on the administration server if auto-deploy is enabled by copying the war file into the \applications directory of the administration server.

## Scenario 2. Deployment through the Administration Console

- Set the “Use Data Source” site property appropriately to indicate whether you want to use a Data Sources in your code.
- If you want to work with Data sources and you have not configured it already, add and configure the JDBC Pool and DataSource to your server as described in (<http://e-docs.bea.com/wls/docs70/ConsoleHelp/jdbc.html#1104939>). Bear in mind that the WebLogic JNDIName property should be equal to the “DataBase URL” property of your CodeCharge Studio connection. Install your application as described in (<http://e-docs.bea.com/wls/docs70/ConsoleHelp/applications.html#1104914>).

## ***End User License Agreement***

End-User License Agreement for CodeCharge Studio 1.0 ("Software")

Copyright (c) 2002 YesSoftware, Inc.

All Rights Reserved.

**BY DOWNLOADING AND/OR IMPLEMENTING THIS SOFTWARE YOU AGREE TO THE FOLLOWING LICENSE:**

Definitions: "You" and "LICENSEE" refers to the person, entity or organization which is using the software known as "CodeCharge Studio", and any successor or assignee of same.

**AGREEMENT.** After reading this agreement carefully, if you ("Customer") do not agree to all of the terms of this agreement, you may not use this Software. Unless you have a different license agreement signed by YesSoftware that covers this copy of the Software, your use of this Software indicates your acceptance of this license agreement and warranty. All updates to the Software shall be considered part of the Software and subject to the terms of this Agreement. Changes to this Agreement may accompany updates to the Software, in which case by installing such update Customer accepts the terms of the Agreement as changed. The Agreement is not otherwise subject to addition, amendment, modification, or exception unless in writing signed by an officer of both Customer and YesSoftware. This Software is owned by YesSoftware and is protected by national copyright laws and international copyright treaties.

**1. GRANT OF LICENSE AND PROHIBITIONS.** This Software is licensed to you. You are not obtaining title to the Software or any copyrights. You may not sublicense, rent, lease, convey, modify, translate, convert to another programming language, decompile, or disassemble the Software for any purpose. The license may be transferred to another if you keep no copies of the Software. Permission must be obtained before mirroring or redistributing the evaluation copies of the Software. You may not convert CodeCharge Studio or its parts to a different computer language or environment, either manually, or using an automated conversion tool, such that CodeCharge Studio or any modification thereof will run under any language, software, or program other than that implemented by YesSoftware. You agree that any modifications made to CodeCharge Studio belong to YesSoftware and are permitted for your exclusive use during the period of this License Agreement, and may not be transferred, sold or licensed to another entity.

**2. USE AND EVALUATION PERIOD.** You may use one copy of this Software on one client computer. A copy of this Software is considered in use when loaded into temporary memory (i.e., RAM) and/or installed on a permanent storage device (i.e., hard disk, CD-ROM, etc.). You may also use a copy of the Software on a home or portable computer, provided only one copy of the Software is in use at a time. You may use an evaluation copy of the Software for only thirty (30) days in order to determine whether to purchase the Software.

3. **MULTI-COMPUTER LICENSES.** If this is a multi-computer or network license, you may make, install, and use additional copies of this Software up to the number of copies authorized in your registration documentation. If this is an educational license, use is restricted to non-commercial use by students, faculty and staff using campus facilities and to off-campus non-commercial use by students, faculty and staff primarily for school business.

4. **LIMITED WARRANTY.** THE SOFTWARE IS PROVIDED AS IS AND YESSOFTWARE DISCLAIMS ALL WARRANTIES RELATING TO THIS SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. **LIMITATION ON CONSEQUENTIAL DAMAGES.** NEITHER YESSOFTWARE NOR ANYONE INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE EVEN IF YESSOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. IN NO EVENT SHALL YESSOFTWARE'S LIABILITY FOR ANY DAMAGES EXCEED THE PRICE PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF CLAIM. THE PERSON USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

IN NO EVENT WILL YESSOFTWARE BE LIABLE FOR ANY AMOUNT GREATER THAN WHAT YOU ACTUALLY PAID FOR THE SOFTWARE.

6. **TERMINATION.** This Agreement terminates on the date of the first occurrence of either of the following events: (1) The expiration of one (1) month from written notice of termination from Customer to YesSoftware; or (2) One party materially breaches any terms of this Agreement or any terms of any other agreement between Customer and YesSoftware, that are either uncorrectable or that the breaching party fails to correct within one (1) month after written notification by the other party.

**TERMINATION.** This Agreement is effective until terminated. This Agreement will terminate automatically without notice from YesSoftware if you fail to comply with any provision of this Agreement. Upon termination you shall destroy the written materials and all copies of the Software, including modified copies, if any.

You agree that monetary damages alone is not an adequate and just relief resulting from any breach of this License, that a court order prohibiting any further breach of this License is necessary to prevent further damages, and that you will not oppose any reasonable request for a temporary restraining order, preliminary injunction, or other relief sought by YesSoftware in the event of a breach of this License. YesSoftware shall not be required to notify you of any breach, nor make any demand or claim against you resulting from any such breach, or for a demand to stop any use or distribution in violation of the terms of this License, and you agree that any breach of this License and damages resulting thereof shall relate back to the first and earliest breach thereof. Failure of YesSoftware to enforce its rights pursuant to this License shall not constitute a waiver of such rights, and shall not prejudice YesSoftware in any later enforcement of its rights or rights to seek damages thereof.

7. **UPGRADES.** If you acquired this Software as an upgrade of a previous version, this Agreement replaces and supercedes any prior Agreements. You may continue to use the previous version of the Software, provided that both the previous version and the upgrade are installed on the same computer

at all times. You may not have a previous version and the related upgrade version installed on separate computers at any time.

#### 8. ENTIRE AGREEMENT

This End-User Agreement is the entire agreement between you and YesSoftware relating to the Licensed Software, and supercedes all prior written or oral statements, promises, representations and agreements.

9. GOVERNING LAW. The agreement shall be governed by the laws of the State of California. Any action or proceeding brought by either party against the other arising out of or related to this agreement shall be brought only in a state or federal court of competent jurisdiction located in San Mateo County, California. The parties hereby consent to the personal jurisdiction of such courts.

10. U.S. GOVERNMENT RESTRICTED RIGHTS. This Software is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software -- Restricted Rights clause at 48 CFR 52.227-19, as applicable. Manufacturer is:

YesSoftware

2226 A Westborough Blvd. #405

South San Francisco, CA 94080

USA

e-mail: sales@codecharge.com, sales@codechargestudio.com

## ***Copyright***

© **YesSoftware** Incorporated. All rights reserved.

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by YesSoftware Incorporated. YesSoftware Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of YesSoftware Incorporated.

YesSoftware, the YesSoftware logo, CodeCharge and the CodeCharge Logo are either registered trademarks or trademarks of YesSoftware Incorporated in the United States and/or other countries. ASP, IIS, Microsoft, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. JavaScript and JSP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.