

CODECHARGE 2.0

Charge Your Code With CodeCharge



CodeCharge 2.0 Users Guide

CODECHARGE V2.0

Users Guide

© YesSoftware, Inc.
2226 A. Westborough Blvd. #405
South San Francisco, CA 94080
Phone (650) 754-9810 • (415) 820-7566

www.codecharge.com

Table of Contents

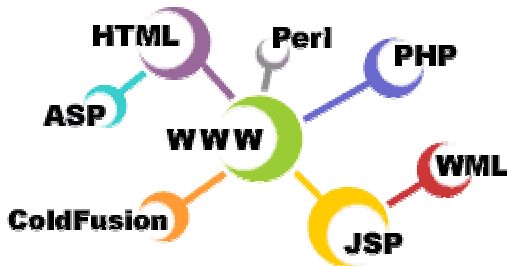
I.	Introduction.....	4
1.	Produce Overview.....	4
1.1	Welcome.....	4
1.2	Features.....	4
1.3	Application	4
1.4	Generated code.....	5
1.5	Sample Sites.....	5
1.6	Pre-Defined Styles.....	5
1.7	Online Tour.....	5
1.8	System Requirements	5
II.	Getting Started with CodeCharge.....	6
1.	Setting up the CodeCharge Environment.....	6
1.1	Downloading CodeCharge	6
1.2	Installing CodeCharge	6
1.3	Quick Start.....	9
III.	Users Guide.....	14
1.	The Basics	14
1.1	Starting CodeCharge	14
1.2	Site Menu.....	14
2.	Properties.....	15
2.1	General Tab.....	15
2.2	Publishing Tab.....	17
2.3	Database Tab	19
2.4	Security Tab.....	31
2.5	Pages Tab	32
2.6	Files Tab.....	32
3.	Style	33
3.1	CSS Styles.....	35
3.2	HTML Styles	35
3.3	Choosing the Type of Style	36
4.	Pages.....	36
4.1	Page Settings	37
4.2	Page Properties.....	38
4.3	Form Settings	41
4.4	Form Properties.....	42
4.5	Field Settings.....	50
4.6	Field Properties.....	51
5.	Diagram.....	55
6.	Modules and Events	56
7.	Generator	58
8.	Toolbar	58
9.	Options.....	59
10.	Diagram Mode	60
11.	Live Data Mode.....	60
IV.	Events.....	62
1.	Page Events.....	62
2.	Form Events.....	63

2.1	Form Types.....	63
3.	Function List.....	67
3.1	ASP functions	67
3.2	PHP functions	67
3.3	Perl functions	68
3.4	JSP functions.....	69
3.5	Cold Fusion tags.....	70
V.	Forms Overview.....	71
1.	Grid Form	71
2.	Record Form	73
3.	Search Form	74
4.	Menu Form.....	75
5.	Tree Form	77
6.	Login Form	78
VI.	Using the New Page Wizard	80
1.	Grid Page Wizard.....	80
2.	Record Page Wizard.....	84
3.	Grid and Edit Page Wizard	86
4.	Grid and Edit (one page) Page Wizard	90
5.	Tree View Page Wizard	94
6.	Master/Detail Page Wizard	100
7.	Master/Detail (four pages) Page Wizard.....	107
VII.	Appendix A:	115
1.	Uninstalling CodeCharge	115
2.	CodeCharge Extensions	116
	Copyright	117

I. Introduction

1. Produce Overview

1.1 Welcome



Welcome and thank you for choosing CodeCharge™.

With CodeCharge at hand, the ground is now set for the easy creation of dynamic websites in record time. CodeCharge is a revolutionary code-generating tool designed for web developers, Internet consultants, application programmers and webmasters who need to speed up the deployment of database-driven web applications. With some basic database knowledge, you can now deploy databases on the web using CodeCharge.

CodeCharge is bar none, the easiest code generation tool available; allowing you to quickly generate dynamic, bug-free websites in most web scripting languages of merit. CodeCharge automates the creation of Web applications that access and update data in real time, taking full advantage of such mainstream technologies as ASP/ASP.Net, JSP, PHP, Perl, ColdFusion and other server technologies.

Click [here](#) for an overview of the various CodeCharge features or if you want to get started on a website right away, check out the [Quick Start](#) instructions which will get you up and going in a couple of minutes.

1.2 Features

CodeCharge has a wide span of applicability, allowing you to quickly:

- Web-enable databases and applications using an easy assembly process.
- Build new web applications fast.
- Create e-commerce components such as online stores and shopping carts, classifieds or auctions.
- Add dynamic content to your site, such as message boards, surveys or catalogs.
- Build powerful, user-friendly intranet and Internet applications with much less manual coding, if any.
- Develop dynamic, database-driven sites in a matter of hours or days instead of weeks or months.
- Create customized message boards, classifieds, online catalogs, and shops, auctions, directories or other web systems.
- Generate code in ASP, ASP.Net/C#, ColdFusion, JSP, PHP, Perl and other languages.
- Generate clean and robust code that is compatible with many databases, operating systems and browsers.

The following list highlights some of the main features and capabilities of CodeCharge.

1.3 Application

- ✓ Works with most databases, including Microsoft Excel, Microsoft Access, Microsoft SQL, MySQL, Oracle, PostgreSQL and Sybase
- ✓ Generates source code in many programming languages including ASP/VBScript, ColdFusion, PHP, JSP, Perl and ASP.NET
- ✓ HTML and WAP output support
- ✓ Displays visual site diagram
- ✓ Supports Extensions/Components
- ✓ Live HTML page template preview
- ✓ Instant Live Site preview
- ✓ Supports file locking
- ✓ Custom HTML and Code editor

1.4 Generated code

On the other end, the code generated by CodeCharge:

- ✓ Is robust and easy to modify
- ✓ Optionally utilizes Styles/CSS to customize the look of the site
- ✓ Runs on Windows and UNIX (all flavors) platforms
- ✓ Supports multi-level security (administrator, editor, user, etc.)
- ✓ Optionally, code can be generated separately for HTML template files which can then be easily edited to change the layout.

1.5 Sample Sites

A number of sample sites that have been developed using CodeCharge are available for viewing at the CodeCharge website (<http://www.codecharge.com>)

1.6 Pre-Defined Styles

A number of pre-defined style files for use with CodeCharge generated websites are available for download at the CodeCharge website as well. You can use these styles to set a consistent theme for your site as well as edit any of the style elements to your liking. CodeCharge allows you to save different site styles so that you can reuse them at a later point.

1.7 Online Tour

We also invite you to browse through the CodeCharge online tour located at <http://www.codecharge.com/tour/>.

1.8 System Requirements

The following is a list of system requirements for generating Web sites using CodeCharge.

1.8.1 DESKTOP APPLICATION REQUIREMENTS (FOR GENERATING WEB SITES)

- 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 or 2000
- Database application (MS Access, MS SQL, MySQL, etc.)
- MS Internet Explorer 5.5 or higher

The following is a list of system requirements of the servers for hosting Web sites generated by CodeCharge.

1.8.2 SERVER REQUIREMENTS (FOR HOSTING WEB SITES)

- ASP 2.0 or 3.0, ColdFusion 4.0, JSP 1.1, PHP 3.0 / 4.0 or Perl 5.0 server or higher
- CodeCharge generated code runs on most systems including Windows and UNIX

II. Getting Started with CodeCharge

1. Setting up the CodeCharge Environment

Installing CodeCharge is simple and easy. The following is an outline of the steps involved in setting up a full installation of CodeCharge.

1.1 Downloading CodeCharge

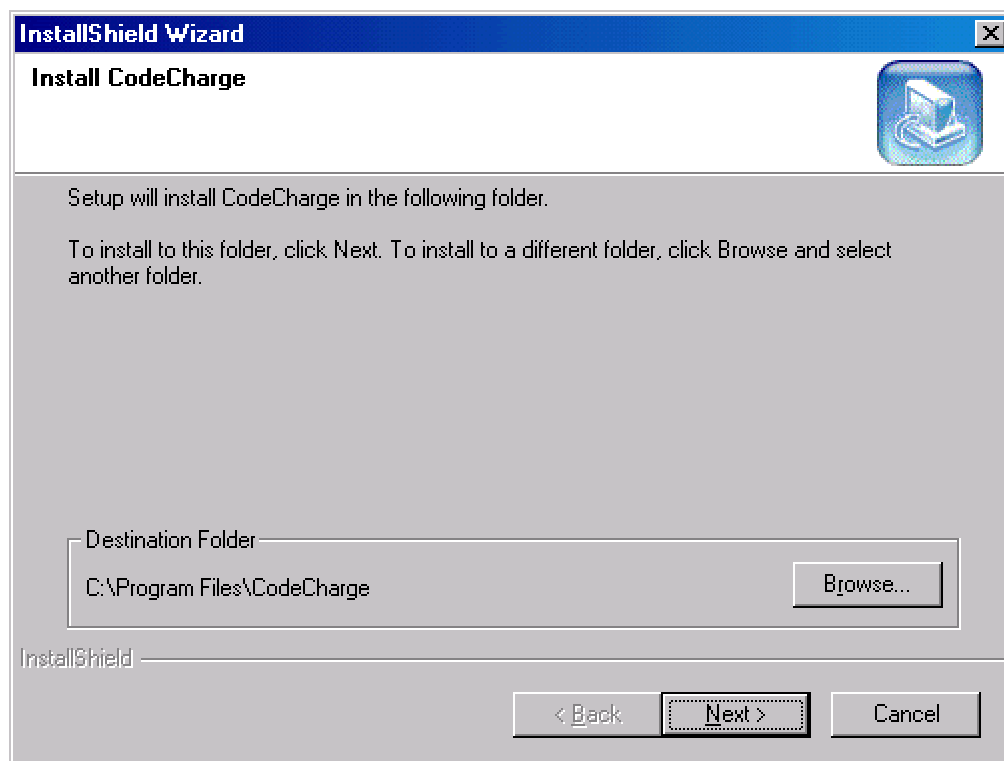
To begin with, you need to acquire a copy of CodeCharge. To download the latest version of CodeCharge for purchase or trial, use your browser to navigate to the CodeCharge website (<http://www.codecharge.com/>) and navigate to the download section.

1.2 Installing CodeCharge

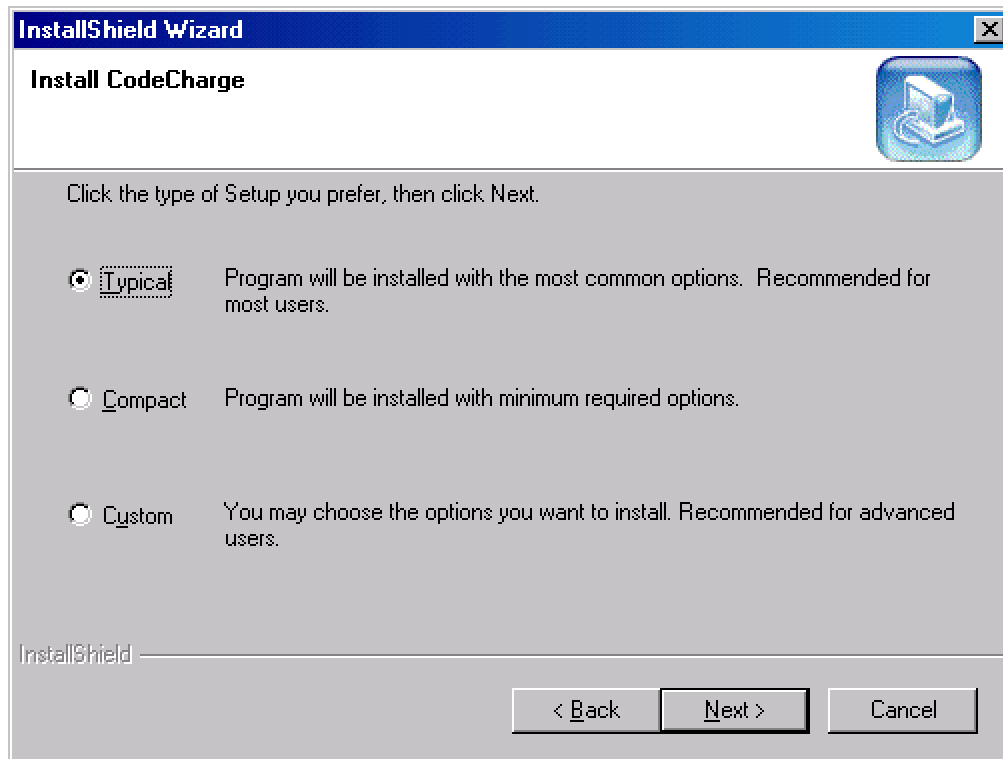
Once you have downloaded CodeCharge, locate the downloaded package file on your hard drive. The installation program will take you through the steps of installing CodeCharge and registering the necessary components on your machine.

The steps to follow are as shown below:

1. To begin the installation process, double-click on the self-extracting executable file CodeCharge.exe
2. In the ensuing popup window, confirm or specify the destination folder where you want CodeCharge to be installed. The default recommended path for installing CodeCharge is **C:\Program Files\CodeCharge**. However, if you wish to change this path, click **Browse** and specify an alternate path.



3. Click **Next** to proceed to the next screen where you will be presented with the option of choosing from three different types of setup.



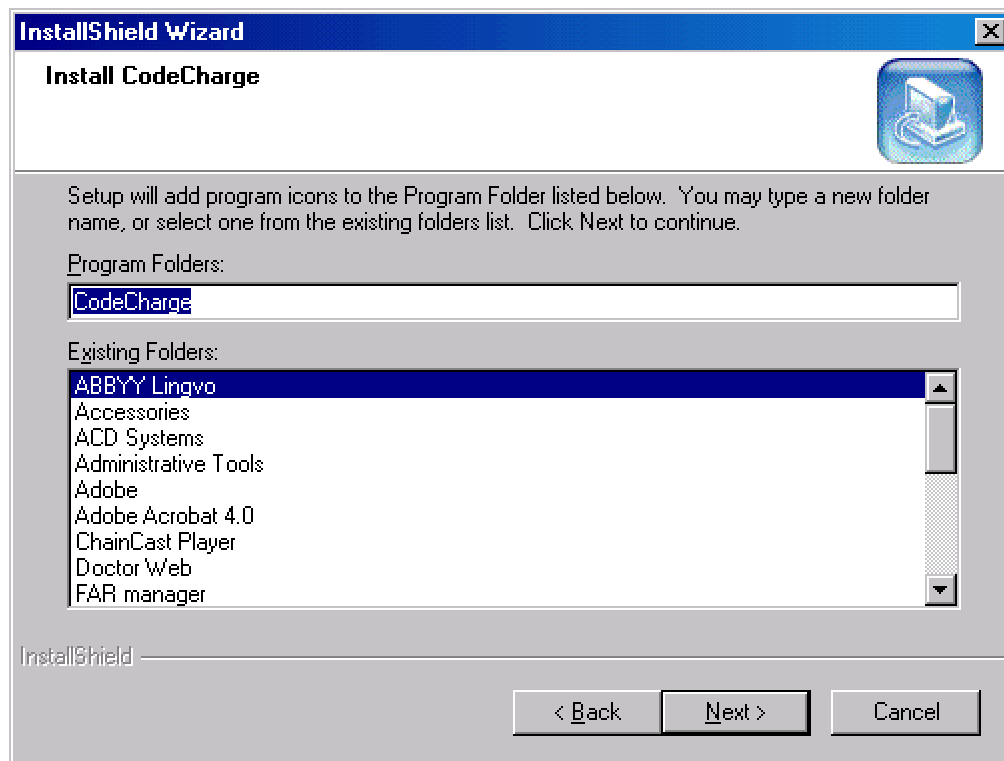
Typical: This is the recommended option for most users. CodeCharge will be installed with the most common options. These options are detailed below:

- Patterns
 - ASP
 - ColdFusion
 - JSP
 - Perl
 - PHP
- Styles
- Examples
 - Bookstore
 - BugTrack
 - Classifieds
 - EmpIDir
 - Events
 - Forum
 - Ledger
 - Links
 - NewsWAP
 - Portal
 - RegForm
 - TaskMan
 - TellAFriend
 - YellowPages
- Help
- Jet 4.0

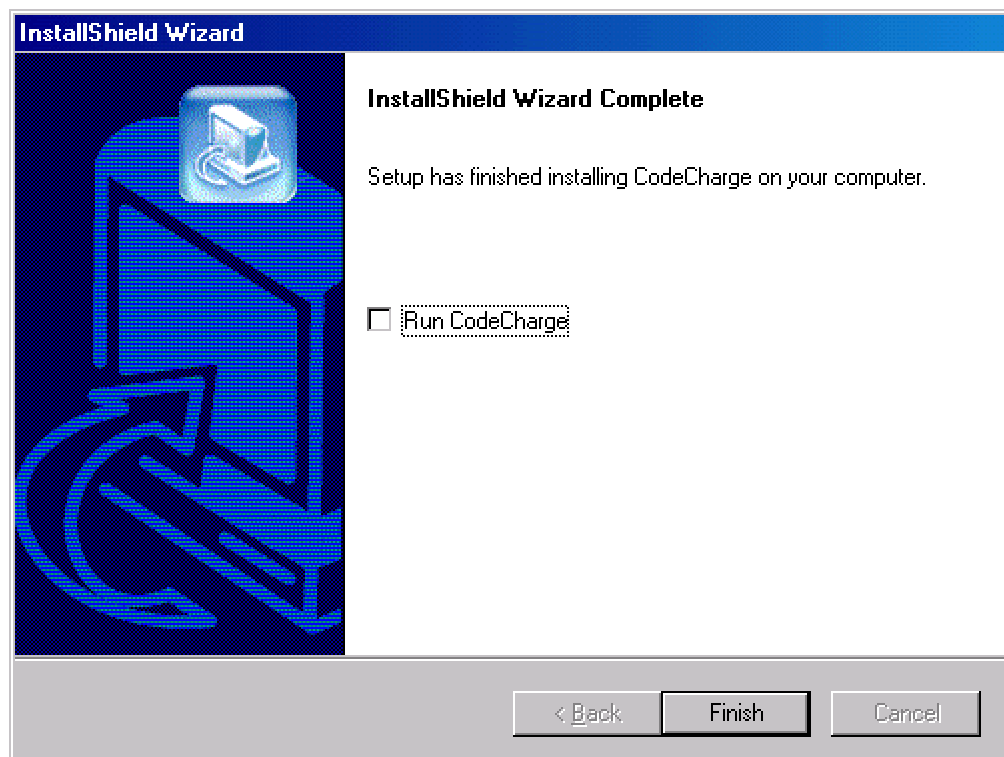
Compact: CodeCharge will be installed with the minimum required options.

Custom: Recommended for advanced users. Use this option to select the components that you want installed and omit those you don't need.

4. Assuming that you have selected the default option (Typical), click **Next** to move to the next screen where you can specify the Program Folder where setup will add the program icons. The default Program Folder is **CodeCharge**.



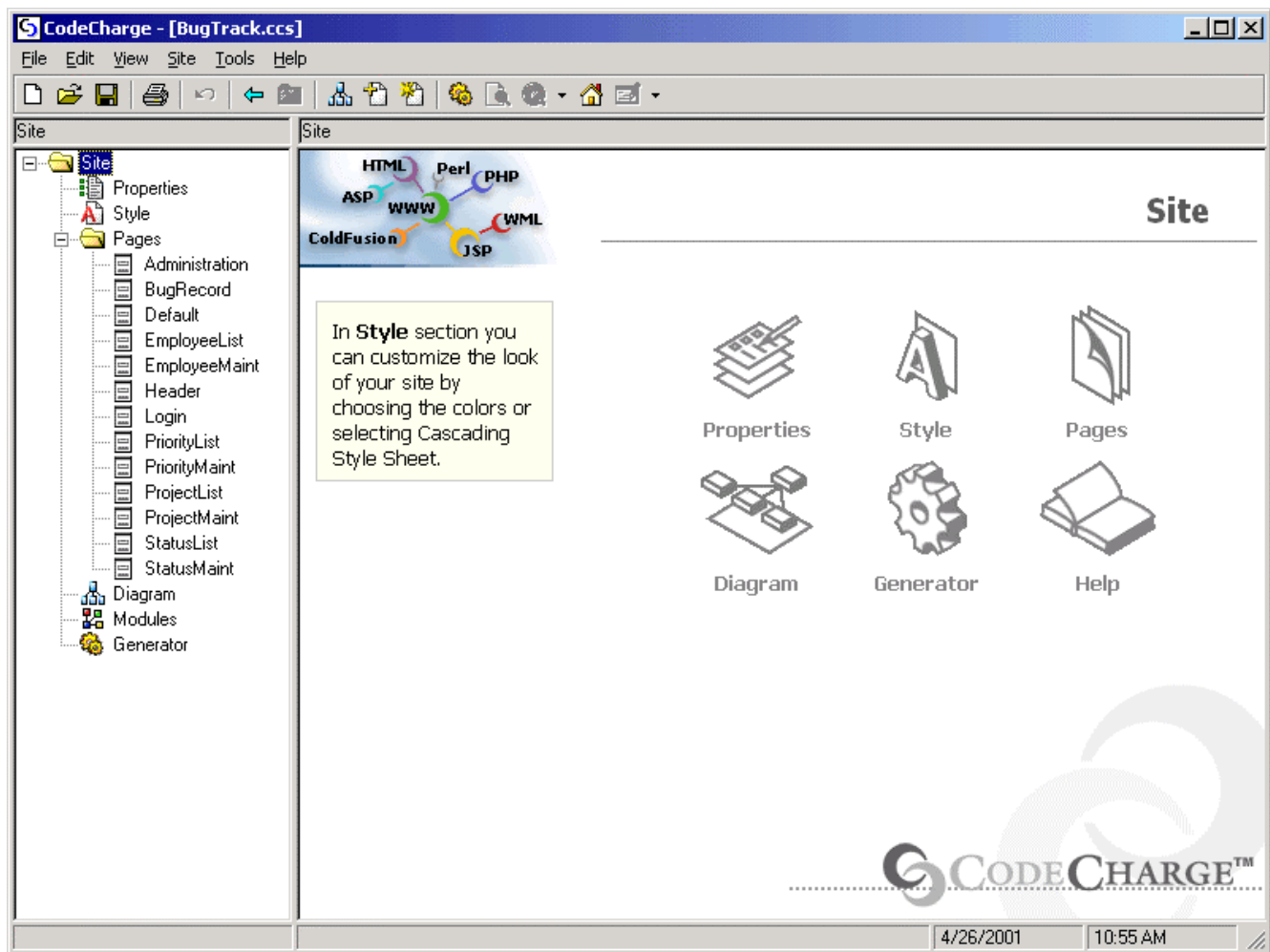
5. At this point, click **Next** to begin the actual installation of the software. Once setup has finished copying the files and creating the necessary icons, you will be presented with the following screen upon successful completion of the setup process. To run CodeCharge, click on the **Run CodeCharge** checkbox then click **Finish**.



1.3 Quick Start

The following section is intended for those users who are eager to delve right into CodeCharge and start creating websites. The section addresses the essential features of CodeCharge which you will typically encounter in the process of creating websites. Even if you do not intend to get started with CodeCharge immediately, you are encouraged to skim through this section so as to get acquainted with some of the features that will be addressed later on in more detail.

We begin with the main CodeCharge window which, as you can see from the diagram below, is divided into two main sections. The left panel contains a dropdown menu featuring the main project components that you work with within the CodeCharge IDE. The composition of the right panel changes according to the selection that you have made in the left panel. For instance, the diagram below shows the left and the right panel when the 'Site' component is selected in the dropdown menu.



Let us now step through the process involved in creating a basic website.

Step	Description
1. Setup up the global project properties.	This is done within the "Properties" option of the main site menu.
2. Set the Project name.	The 'General' tab contains a field where you enter the name of the Project as shown below (Bug Management). This name is also going to be entered as the default Title in all the pages you add to the project later on.

3. Choose the Programming Language	General Publishing Database Security Pages Files Project Bug Management Language ASP 2.0 & Templates Default <head> section <pre><meta http-equiv="pragma" content="no-cache"> <meta http-equiv="expires" content="0"> <meta http-equiv="cache-control" content="no-ca</pre> Under the “General” tab, you also have to specify the language in which your project pages will be generated. You do so by selecting the language of your choice from the “Language” listbox. General Publishing Database Security Pages Files Project Bug Management Language - ASP 2.0 & Templates - ASP 2.0 - ASP.NET C# - ColdFusion 4.0 & Templates - ColdFusion 4.0 - JSP 1.1 & Templates - JSP 1.1 & WML 1.1 - JSP 1.1 - PHP 3.0 & Templates - PHP 3.0 - PHP 4.0 & Templates Under the “Publishing” tab, you have to specify the path to a folder in your web server to where your project pages will be generated. This can be a local folder, a network folder or a remote location accessed by FTP. In the ‘Live Site URL’ field, enter the URL that maps to the location you specified in the ‘Output Code Path’ field.
4. Specify the Publishing location.	

5. Setup database connection

Properties

General Publishing Database Security Pages Files

Location

Local/Network

Output Code Path

C:\inetpub\wwwroot\bugtrack

Output File Format

Windows

Live Site URL

localhost/bugtrack

Use the “Database” tab to configure design time as well as server side connection strings to your database. Configuring a database connection is a necessary prerequisite for creating project pages. Your choice of connection type will be influenced by the language you chose earlier on as well as the database product you are using.

Data Link Properties

Provider Connection Advanced All

Select the data you want to connect to:

OLE DB Provider(s)

Microsoft ISAM 1.1 OLE DB Provider
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services

Next >>

OK Cancel Help

5. Choose a Graphic Scheme

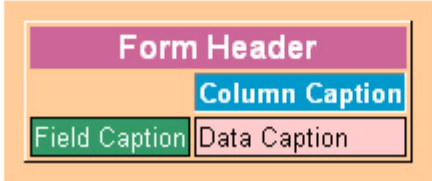
In the left panel of the CodeCharge IDE, select the "Style" component. You will be presented with a screen where you can select and preview various pre-defined styles. You can adjust the style elements to suite your design needs.

Type
 ☐ **Generate File containing Styles**

Scheme _____

Item

<BODY Style="
 
" >



6. Assemble project pages

Once you have selected the style move on to the "Pages" component. It is here that you will spend the bulk of your project design time. In this workspace, you create pages composed of different types of forms (search, grid, record, login and so on) and the forms will subsequently have different types of fields which may or may not expose database contents.

Page 

Name   

Title

Security Level

Columnar Layout ☐

Forms    

Search  **Bugs**

Name  **Type**

Form Title

Table 

Fields  

Key Field

 ☐ 

☐ 

List of Fields

Fields

- assigned_by
- assigned_to
- bug_desc
- bug_id
- bug_name
- date_assigned

7. Generate Source Code and HTML files for your Web site

Once you are done creating your project pages, move on to the “Generator” section to generate the pages to the web server.

Generator

Generate Pages ☒ ☐

- ☒ Administration
- ☒ BugRecord
- ☒ Default
- ☒ EmployeeList
- ☒ EmployeeMaint
- ☒ Login
- ☒ Page
- ☒ PriorityList
- ☒ PriorityMaint
- ☒ ProjectList
- ☒ ProjectMaint
- ☒ StatusList
- ☒ StatusMaint

If you use a template version of a language such as ASP & Templates or PHP & Templates, you can import the generated HTML files into a HTML editor and further customize the look of your site. You can modify HTML layouts generated by CodeCharge using any HTML editor such as DreamWeaver, HomeSite, FrontPage and so on. However, if you do so, make sure that you do not remove or alter any of the text that appears between curly braces e.g. {SearchFormName}

The final step would be to view the generated pages and make sure that they look and perform as intended. To do so, simply click on the ‘Go Live’ or ‘Open Default Page’ button on the toolbar then sit back and enjoy the results of your effort!

III. Users Guide

The following sections shall address the various CodeCharge features in more detail.

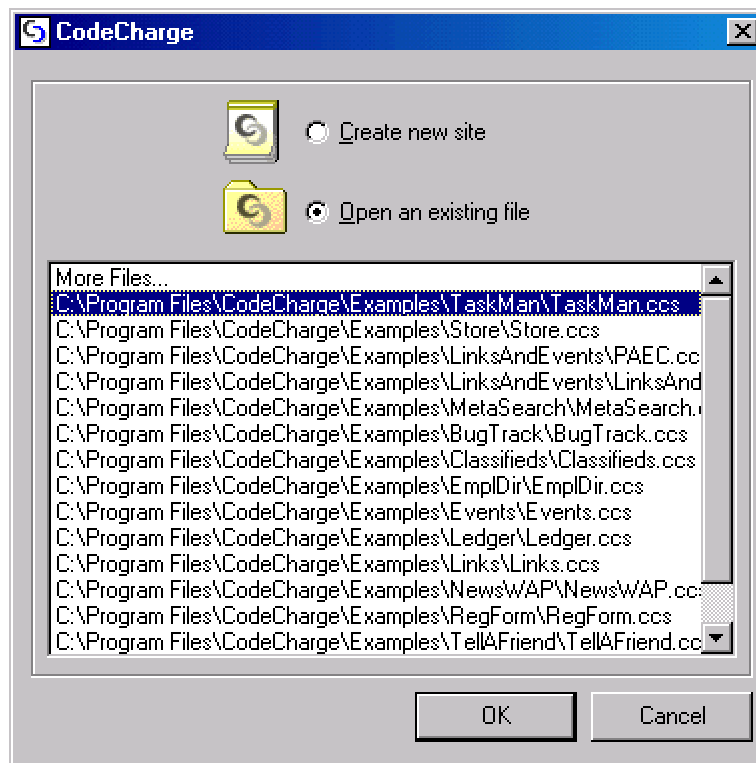
1. The Basics

1.1 Starting CodeCharge

To start CodeCharge, click on the “Start” button on your Windows desktop and select Programs ► CodeCharge ► CodeCharge

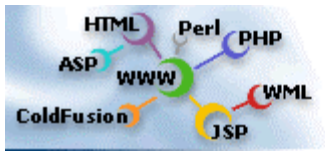
Once the program starts up, you will be presented with a screen where you can select one of two options:

- Create new site – this will create a new project
- Open an existing site – this will open a previously saved project. If you installed the sample applications, the screen will also contain a list of the application project files.



1.2 Site Menu

At the top of the CodeCharge dropdown menu hierarchy is the “Site” option under which you will find other options such as Properties, Style, Pages, Diagram, Generator and Help. Simply click on one of the options to access its features.



Site

Pages section allows you to create web pages by selecting and assembling database fields onto forms.



Properties



Style



Pages



Diagram



Generator



Help



The Site/Menu includes icons for the following sections

- ✓ Properties
- ✓ Style
- ✓ Pages
- ✓ Diagram
- ✓ Generator
- ✓ Help

Each of the above sections will be addressed in more detail in the sections to come.

2. Properties

The **Properties** section of CodeCharge allows you to setup global project settings such as Project Name, Language, Database settings, Publishing and Security.

2.1 General Tab

The first of the property categories is contained within the “General” tab as shown below:

General Publishing Database Security Pages Files

Project

Language

Default <head> section

```
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="cache-control" content="no-cache">
```

An explanation of each of the properties follows:

Parameter	Description
Project	This is the name of your Project. This name will also be used as the default "Title" for all the project pages that you will later on add.
Site Language Pattern	Of pivotal importance is the Language you select for your project. Usually, this is a decision you make before hand and is influenced by a number of factors, prime among them being the nature and capability of the server you intend to deploy your project to as well as the database containing your dynamic data. To this end, you are provided with the following selection to choose from: - ASP 2.0 & Templates – for generating ASP/VBScript code as .asp files and .html files that contain HTML code - ASP 2.0 – for generating ASP/VBScript code as .asp files that include HTML - ASP.Net C# - for generating .Net compliant C# code as .aspx and .cs file - Cold Fusion 4.0 & Templates - for generating CFML (ColdFusion Markup Language) code as .cfm files and .html files that contain HTML code - Cold Fusion 4.0 – for generating CFML code as .cfm files that include HTML - JSP 1.1 & Templates – for generating JSP (Java Server Pages) code as .jsp files and .html files containing HTML code - JSP 1.1 & WML 1.1 – for generating JSP code as .jsp files for Wireless Access Protocol (WAP) - JSP 1.1 – for generating JSP code as .jsp files including HTML - PHP 3.0 & Templates - for generating PHP 3.0 code as .php3 files and.html files containing HTML code - PHP 3.0 - for generating PHP 3.0 code as .php3 files including HTML - PHP 4.0 & Templates – for generating PHP 4.0 code as .php files and.html files containing HTML code - PHP 4.0 & WML 1.1 – for generating PHP code as .php files for Wireless Access Protocol (WAP) - PHP 4.0 – for generating PHP 4.0 code as .php files including HTML - Perl 5.0 & Templates – for generating Perl code as .pl/.cgi files and .html files containing HTML code - Perl 5.0 – for generating Perl code as .pl/.cgi files including HTML You probably have noticed that some of the languages have a Templates option while others don't. Unless you are an advanced user or don't see the Template option for your language, we recommend that you select the Templates version. Templates are HTML files that are separate from the programming code that executes on the server side. Templates allow

Default <Head> section	programmers and designers to work independently such that the programmers can continue to manually modify the code, while the designers load the HTML templates into an html editor and modify them by adding graphics, text, changing colors, etc. Use this option to specify HTML meta tags or other code to be included at the top of all the generated html templates. Note that later on, you will also have the option of specifying different meta tags at the page level. You can also use this section to specify frameset code if you intend to utilize HTML frames.
------------------------------	--

In addition to the above three options, if you are using the PHP or Perl languages, you will also have the option of specifying the file extension to be used for the code files. For instance, your web server can be configured to recognize PHP files if they have the .php extension. In that case, you would configure the extension options as shown below:

General Publishing Database Security Pages Files

Project

Language **Extension**

Default <head> section

2.2 Publishing Tab

As the name suggests, you use the “Publishing” tab to specify where your project files will be generated.

Properties

General Publishing Database Security Pages Files

Location

Output Code Path

Output File Format

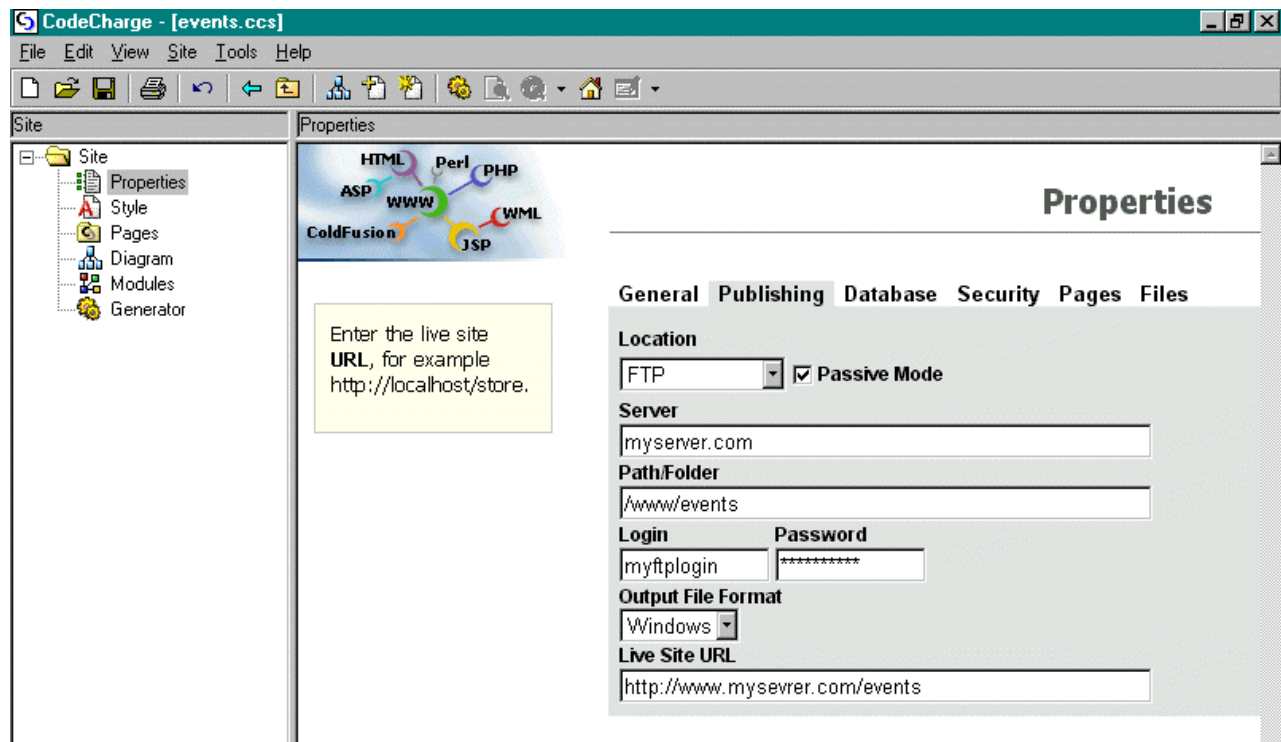
Live Site URL

Note: To assist in specifying the desired path, you can click on the File icon  and browse to the location where you want the files to be generated. .

The table below describes the various parameters:

Parameter	Description
Location	You can deploy your files to a local or network folder or use the FTP protocol to deploy your files to a remote server.
Output Code Path	This path represents the absolute location where your files will be generated. The format of the path will depend on the platform concerned as well as the method used to publish the files to the server.
Output Files Format	Windows example: C:\inetpub\wwwroot\bugtrack Unix example: \usr\apache\htdocs\bugtrack This option allows you to specify the output format for your files depending on the platform of the server machine. The "Windows" format is the default but if you are deploying to a Linux/Unix server, change the format to "Unix". If using Perl, this value also determines the nature of the hash bang (shebang) directive included at the top of the generated code files.
Live Site URL	Enter the URL which maps to the location you specified above for the "Output Code Path" field. Essentially, this is the URL you would normally type into the address bar of a browser so as to access the files that will be generated to the "Output Code Path". If you are deploying to the same machine, you can use 'localhost' as in "http://localhost/bugtrack".

If it so happens that during the development phase or later you'd like to generate and publish your files to an external server, change the "Location" field to 'FTP' and specify the rest of the publishing settings as follows:



The following is an explanation of the available options:

Parameter	Description
Passive Mode	This is a commonly used method of transferring files to the server using FTP. Normally, when you connect to an FTP site, the site establishes the data connection to your PC (the client). However, if the site allows passive transfers, you can have your PC establish the data connection. Select this option if transfers are erratic, you keep getting failed data channel errors or your PC is behind a router-based firewall or gateway requiring passive

Server	transfers.
Path/Folder	This is the address of your server usually provided by the server administrator or hosting company if applicable. You can enter the server's qualified name or IP address.
Login and Password	This is the location on the server where you'd like to place your files.
Live Site URL	Enter your authentication details necessary to login to the server via FTP.
	The URL that you enter in the address bar of a browser to view the generated pages.

2.3 Database Tab

CodeCharge requires you to specify how it should connect to the database while in design mode (Design connection), as well as how the generated pages should connect to the database (Server Connection). Working within CodeCharge revolves around database content and as such, it is important to have a firm grasp of the database connection settings.

The following are some of the common fields that you will encounter in the process of creating a database connection string.

Parameter	Description
Type	Custom or DSN. The "Custom" option entails building a connection string that will be used to connect to the database using an OLE DB or other provider. On the other hand, selecting the "DSN" type essentially plugs into the systems ODBC control where you can specify an ODBC DSN to be used or you can create one.
Design Connection String	This is a string containing a set of name-value pairs that specify how the connection to the database is to be made. The nature of the connection string depends on a variety of factors including the database driver, provider and product being connected to. This connection string is used to make the connection to the database from within CodeCharge.
Login and Password	If applicable, the authentication details needed to access the database.
Server Connection String	This serves a similar function as the "Design connection string" only that this string is used by the generated pages to connect to the production database. Usually, this will be a string very similar to "Design Database Connection" string, but the database name can be different or the database location can be different and so on. You would specify this optionally if you would like to use a different database or a database at a different location than the Design database. Specifically, if you plan to deploy your code to a different machine,

DSN Login and Password Database Type Persistent Connection Database Driver	you can copy and paste the “Design Database Connection” string then modify it to fit the server configuration. If you are using the Cold Fusion language, you have to specify the Data Source Name (DSN) to be used to connect to the production database. Chances are that the authentication details needed to connect to the database during design time are not similar to those needed to connect to the production database. In this case, you need to specify a separate set of authentication details. This option is available for the PHP and Cold Fusion languages. Use it to specify the database product that is being connected to. General Publishing Database Security Pages Files Design Type Data Source Name DSN MS Access Database Login Password Server Location: Local/Network Database Name Host bugtrack localhost Login Password admin ***** Database Type MySQL ☒ Persistent Connection JOIN Style ANSI SQL (Left Join/Inner Join) This option is only available for the PHP language and if specified, will cause PHP to create and use a persistent connection to the database. Persistent connections do not close when the execution of a script ends. When a persistent connection is requested, PHP checks if there's already an identical persistent connection (that remained open from earlier) - and if it exists, it uses it. If it does not exist, it creates the link. This option is only available for the JSP language. Use it to specify the database driver so be used to connect to the database.
---	---

	Properties General Publishing Database Security Pages Files Design Connection String Provider=MSDASQL.1;Persist Security Info=False;Data S Login Password Admin Server Location: Local/Network Connection String jdbc:odbc:taskman Login Password admin ***** Database Driver sun.jdbc.odbc.JdbcOdbcDriver
Use Custom HTML Encoding	This option, which is only available in the ASP language, can be used instead of the standard ASP HTMLEncode function to achieve better compatibility with International characters.
LongReadLen value for DBI Module	This is option is only available in the Perl language and corresponds to the LongReadLen attribute of Perl DBI modules. The LongReadLen attribute is used to control the maximum length of long fields (blob, memo, etc.) which the driver will read from the database automatically when it fetches each row of data.
Join Style	ANSI SQL (Left Join/Inner Join) Oracle (+/=) This allows you to specify the syntax to be used when extracting related data that is stored in two or more tables. This can be done using the ANSI SQL style or the Oracle style: ANSI SQL style example: <pre>SELECT P.patient_name, W.ward_name FROM Wards W JOIN Patients P ON W.ward_id = P.ward_id</pre> Oracle style example: <pre>SELECT P.patient_name, W.ward_name FROM Wards W, Patients P WHERE W.ward_id = P.ward_id</pre> Oracle style joins are performed in the WHERE clause while ANSI SQL style joins are performed in the FROM clause. You can also have a WHERE clause in the ANSI SQL style to filter the result set.

2.3.1 CONNECTION TYPES

There are several standard methods of connecting to databases, such as Open Database Connectivity (ODBC) and Jet database engine, otherwise known as **DSN** and **DSN-less** connections respectively. DSN-less/JET connections are often used when connecting to MS Access, Excel or other file or text based databases. On the other hand, a DSN connection is used with almost all other proprietary databases such as MS SQL, MySQL, DB2, Oracle, Sybase, etc. As a prerequisite, DSN connections require the appropriate ODBC driver installed on your computer. Most database vendors provide ODBC drivers along with their products and usually, the drivers are installed together with the database software. If this is not the case, you can usually find an appropriate ODBC driver at the vendor's site or other sites where freeware drivers are posted.

There are conflicting reports about the advantages of DSN connections over DSN-less connections. However, if you use MS Access and need to make a choice between the two, here is some information that could aid you in making the correct decision:

A **DSN-less** connection is a little easier to setup in CodeCharge than a DSN/ODBC connection because you don't need to define the Data Source Name (DSN) in the system Control Panel. A DSN-less connection can also come in handy on the server side since some hosting companies place a limit on the number of DSN connections a client can have. DSN-less connections are not subject to such a limit.

Although it is disputable whether **DSN connections** performs better than DSN-less connections, the key advantage of using a DSN is that you can setup the same DSN on multiple machines, for example the development and the production machines, and your programs will work on the configured machines without any changes. You would simply copy the generated files from one computer to another and they should function correctly as long as you defined the same DSN on both computers. On the other hand, if you use a DSN-less connection, you may need to modify your connection string for each computer, either directly in CodeCharge or in the **Common.asp** file generated by CodeCharge. Often you will use one connection type during design phase and a different one for production.

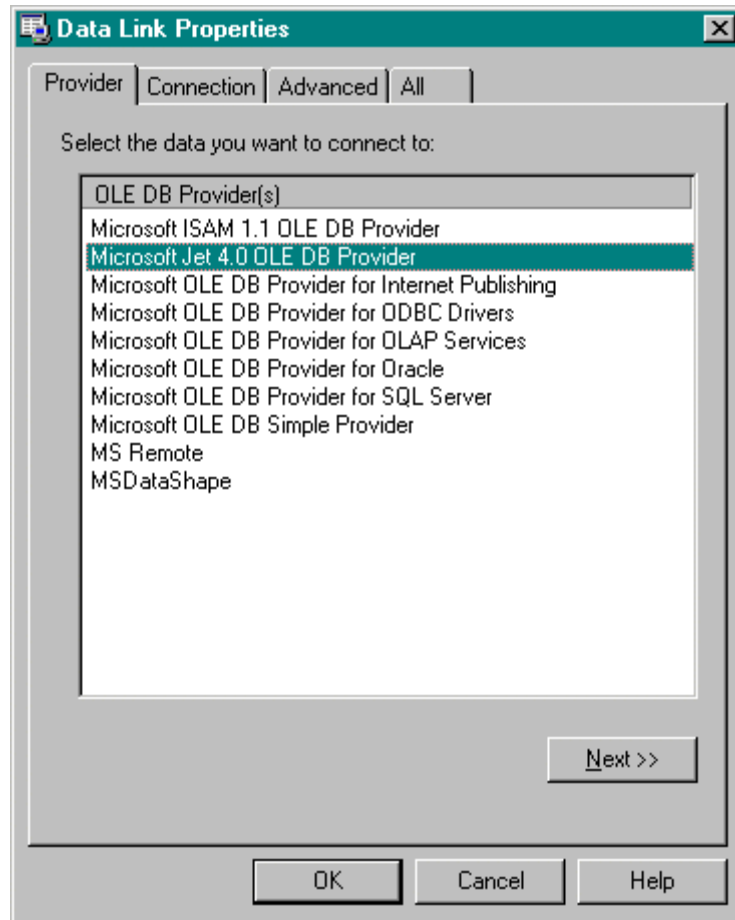
If you use MySQL during design time, you have to use a DSN/ODBC connection when working within CodeCharge. However, the generated web pages don't have to use a DSN/ODBC connection. In the case of PHP, CodeCharge generates PHPLib compliant code which allows for direct connection to the database without ODBC.

2.3.2 DESIGN CONNECTION STRING

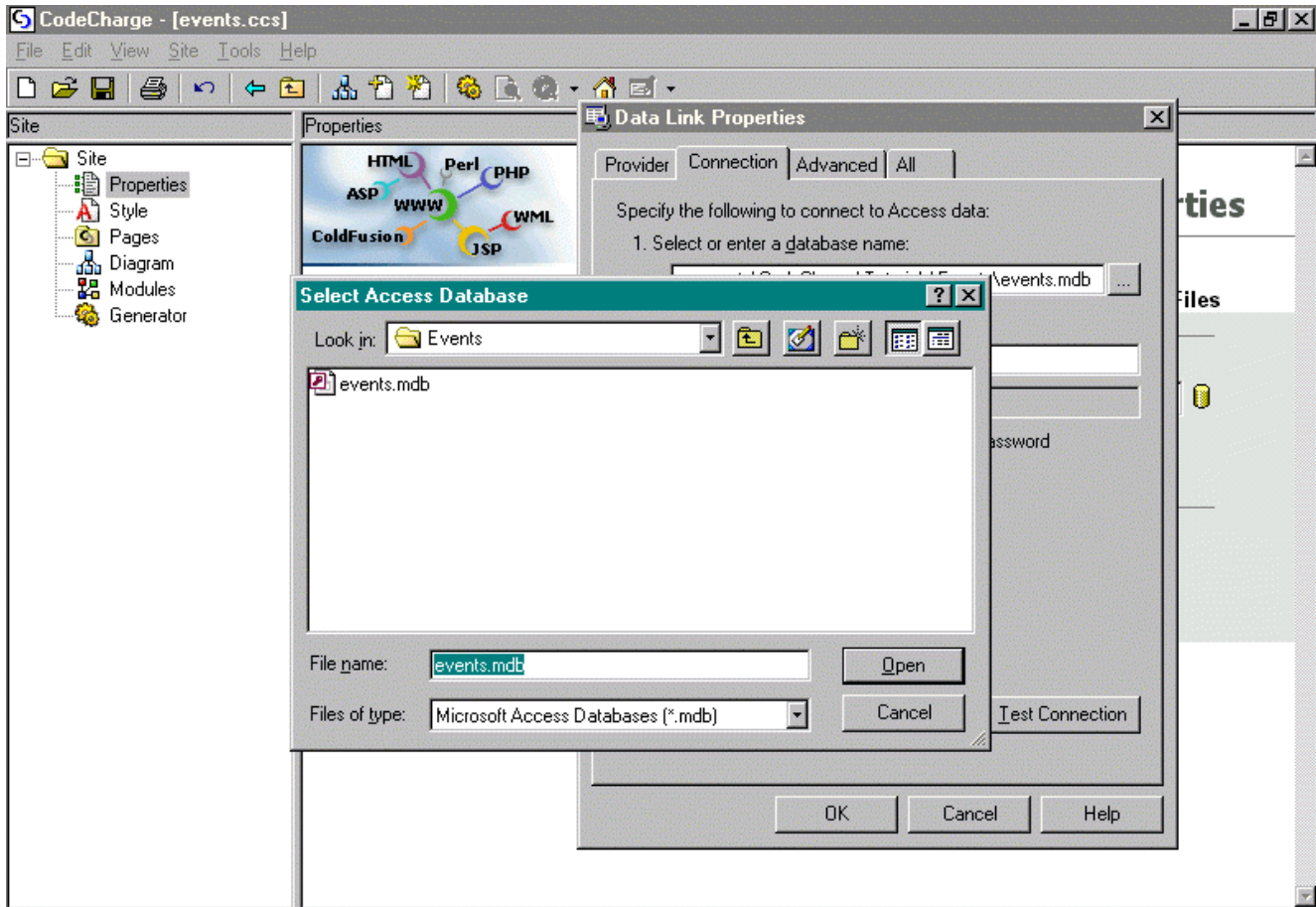
The design connection string tells CodeCharge how to connect to the database during design phase i.e. while you are creating your site. If you use MS Access, you have the choice of using a DSN or DSN-less connection, both of which we will explain how to create. Otherwise, if for example you use MySQL database, you have to use an ODBC connection type (MyODBC).

DSN-less connection / JET

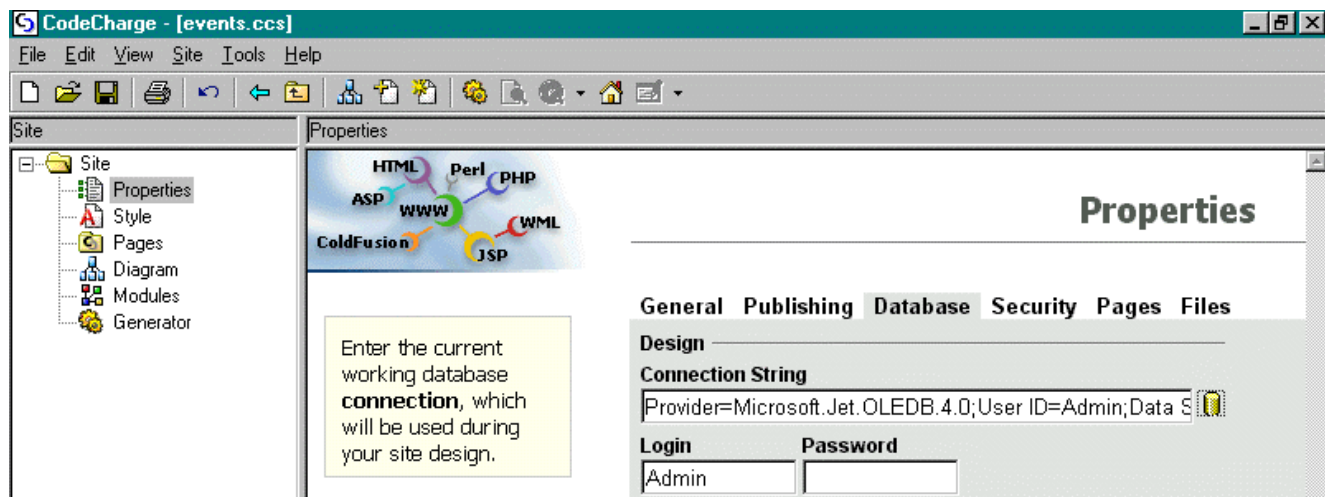
To setup a DSN-less database connection string, click on the yellow icon  next to the 'Connection String' field. This is under the "Database" tab of site "Properties". The "Data Link Properties" window pops up as shown below.



Under the "Provider" tab will be listed the available database Providers. The choice of provider will depend on the database you want to connect to. For instance, to connect to a Microsoft Access database, select the 'Microsoft Jet 4.0 OLE DB Provider' whereas if you want to connect to an Oracle database, select the 'Microsoft OLE DB Provider for Oracle'. Once you are done selecting the appropriate OLE DB provider, click on the "Connection" tab. Once again, the options available under the connection tab will vary depending on your choice of provider. If you are connecting to an Access database using the Jet provider, then you would proceed to select the .mdb file containing your data. Optionally, you can specify authentication details.



After having configured/selected the database to connect to, click on the “Test Connection” button to confirm that your settings are correct. The message “Connection Succeeded” should appear unless there is something wrong with your setup. Once you close the “Data Link Properties” window, notice that the “Connection String” field is now populated with a string that represents the configuration that you just made.



This completes the setup of the Design Connection String and at this point you should be able to view the tables and fields belonging to the database within CodeCharge.

DSN connection / ODBC

If you use MySQL, MS SQL, Access, Oracle, etc. you can setup your design connection using ODBC. But before you do this, you might also need to download and install the appropriate ODBC driver. The following explains how to setup the MyODBC driver that is often used with MySQL. An extensive tutorial is also available at <http://support.codecharge.com/tutorials/myodbc.html>.

Setting Up MyODBC for MySQL

To connect to MySQL database on the Windows platform, you will need to download the MyODBC Windows driver from <http://www.mysql.com/downloads/api-myodbc.html>. Once you install it, you will be prompted to enter the information needed to connect to your local or remote MySQL server. You only need to enter the following options:

- Windows DSN name – any name that you want to give to your Data Source Name, in this case, let's name it 'events'.
- MySQL host (name or IP) – the IP address of your MySQL database, which you can usually obtain from your hosting company. If your MySQL database is installed on the same computer as CodeCharge, enter '*localhost*' or the loopback IP address 127.0.0.1.
- User – this is the login that you or your hosting company setup to access your database.
- Password – this is the password for your database.

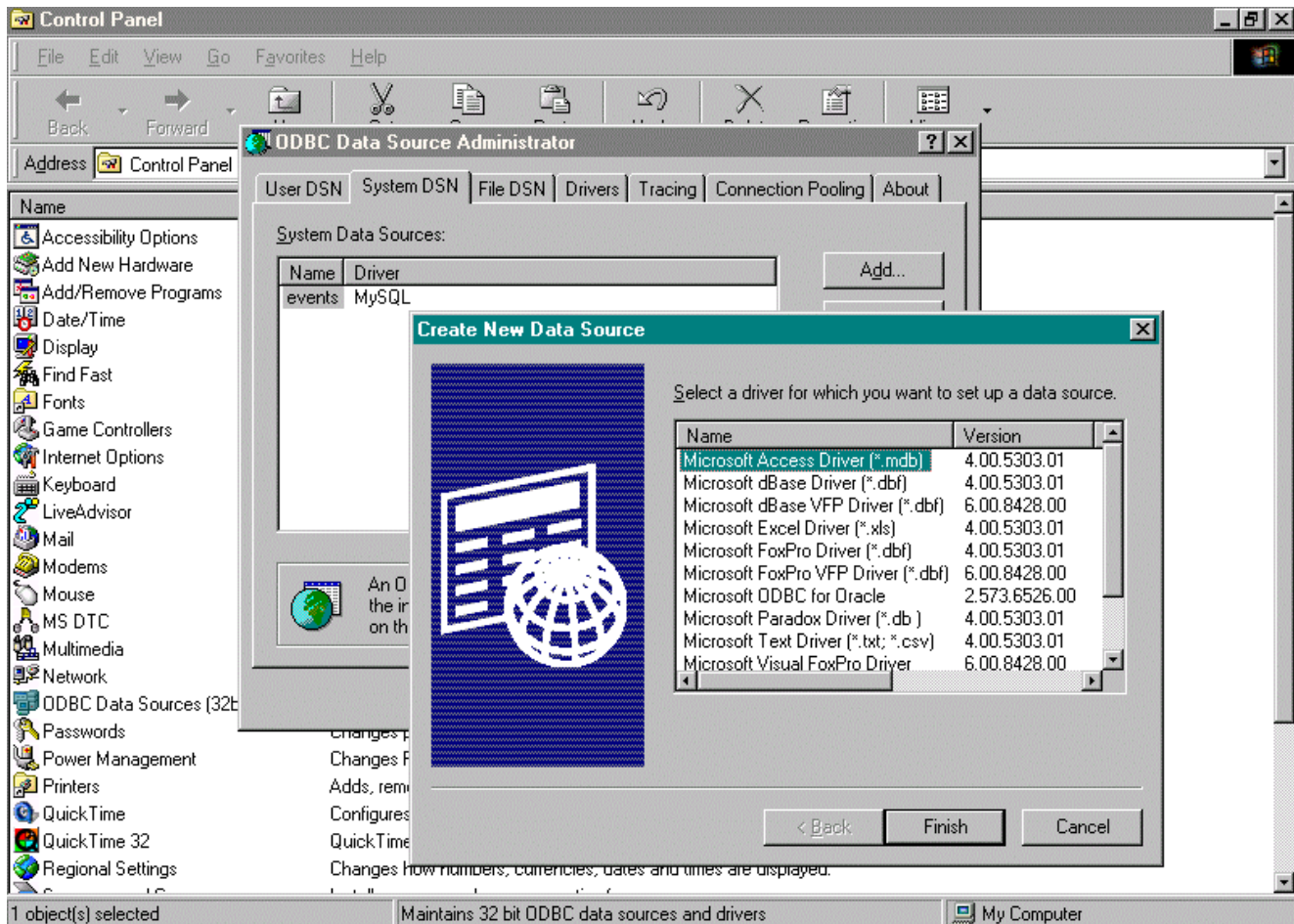
The screenshot below shows a sample MyODBC setup:

The screenshot shows a Windows dialog box titled "TDX mysql Driver default configuration". The dialog box has a standard Windows title bar with a close button. The main content area is light gray and contains the following elements:

- A disclaimer: "This is in public domain and comes with NO WARRANTY of any kind".
- A label: "Enter a database and options for connect".
- Input fields for:
 - Windows DSN name: "events"
 - MySQL host (name or IP): "123.123.123.123"
 - MySQL database name: "my_events"
 - User: "my_login"
 - Password: "xxxxxxxxx"
 - Port (if not 3306): (empty)
 - SQL command on connect: (empty)
- A section titled "Options that affects the behaviour of MyODBC" containing two columns of checkboxes:
 - Column 1:
 - ☐ Don't optimize column width
 - ☐ Return matching rows
 - ☐ Trace MyODBC
 - ☐ Allow BIG results
 - ☐ Don't prompt on connect
 - ☐ Simulate ODBC 1.0
 - ☐ Ignore # in #.table
 - ☐ Use manager cursors (exp)
 - ☐ Don't use setlocale
 - Column 2:
 - ☐ Pad CHAR to full length
 - ☐ Return table names in SQLDescribeCol
 - ☐ Use compressed protocol
 - ☐ Ignore space after function names
 - ☐ Force use of named pipes
 - ☐ Change BIGINT columns to INT
 - ☐ No catalog (exp)
 - ☐ Read options from C:\my.cnf
 - ☐ Safety (Check this if you have problems)
- At the bottom, there are two buttons: "OK" and "Cancel".

Creating a DSN (Data Source Name)

In case you need to change or create a new DSN, you can do so by going to Control Panel on your computer (Start ► Settings ► Control Panel) and selecting ODBC Data Sources or equivalent option. Once there, select the System DSN tab and click Add or Configure to setup a database DSN provided you have the appropriate drivers installed on your computer:



Troubleshooting Database Connection issues

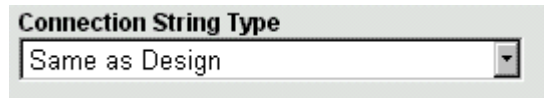
In case you run into problems in the process of connecting to the database while using CodeCharge, you might need to install the latest connectivity drivers from Microsoft, called MDAC. The latest MDAC 2.6 drivers are available at http://www.microsoft.com/data/download_260rtm.htm. Also make sure that you have a version of Internet Explorer later than or equal to v5.5.

2.3.3 SERVER CONNECTION STRING

The server connection string will be included in the generated pages to allow them to connect to the production database. As with the Design Connection String, the Server Connection String depends on the language and database to be used.

Server Connection Strings for use with ASP

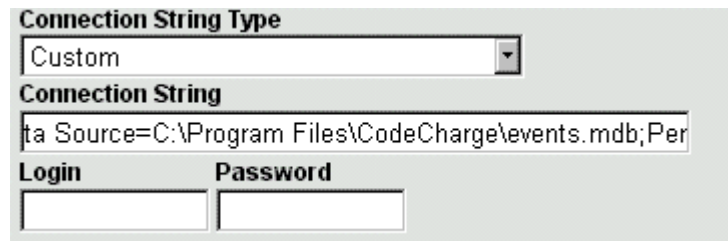
Same as Design



A screenshot of a web application configuration interface. It features a dropdown menu titled "Connection String Type" with the option "Same as Design" selected.

This database connection setting will simply include the design connection string in the generated pages, causing them to use the exact same database as you used during the design phase. You should use it only when you generate programs on the same computer where you installed CodeCharge or you have setup similar data source names (DSN) on the design as well as the production machine.

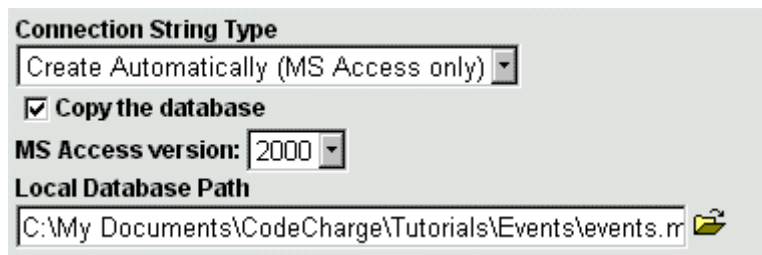
Custom



A screenshot of a web application configuration interface for a custom connection string. It includes a dropdown menu for "Connection String Type" set to "Custom", a text field for "Connection String" containing "ta Source=C:\Program Files\CodeCharge\events.mdb;Per", and two empty text fields for "Login" and "Password".

You can use this setting when you know the connection string that you want to use. For example, you can manually copy the design connection string here and modify the path where your database is located.

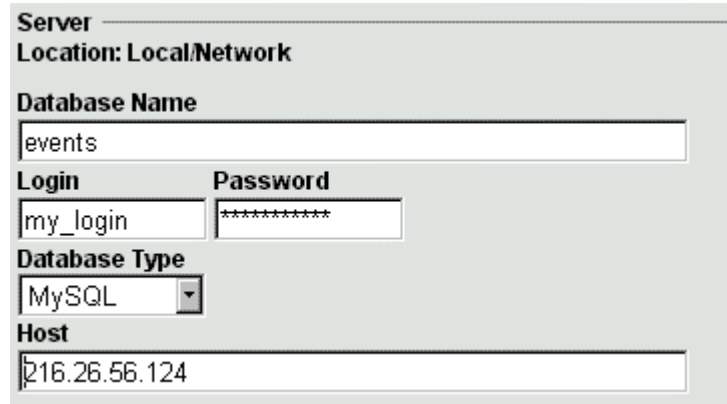
Create Automatically (MS Access only)



A screenshot of a web application configuration interface for creating a connection string automatically for MS Access. It includes a dropdown menu for "Connection String Type" set to "Create Automatically (MS Access only)", a checked checkbox for "Copy the database", a dropdown menu for "MS Access version" set to "2000", and a text field for "Local Database Path" containing "C:\My Documents\CodeCharge\Tutorials\Events\events.m".

If you are publishing your programs to an external server, you might want to use this option to automate the database publishing. In this case CodeCharge will automatically assume that your database file (.mdb) is located in the same folder as the rest of the programs on the server, and it will create the appropriate connection string. However, you can also select 'Copy the Database' option and provide a 'Local Database Path' at least for the first time you publish your site, so that CodeCharge can copy your database to the server and ensure that the created connection string is valid.

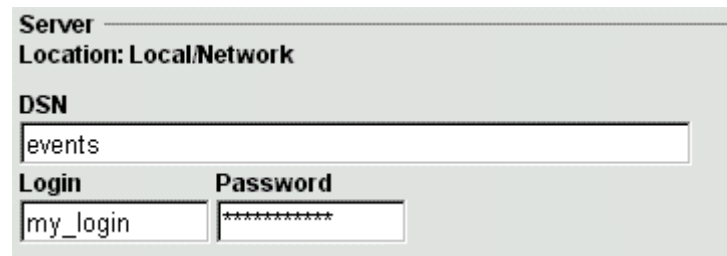
Server Connection for PHP



The screenshot shows a configuration window titled 'Server' with a sub-header 'Location: Local/Network'. It contains several input fields: 'Database Name' with the value 'events', 'Login' with 'my_login', 'Password' with a masked value '*****', 'Database Type' with a dropdown menu set to 'MySQL', and 'Host' with the IP address '216.26.56.124'.

The server connection string for PHP is pretty straightforward. You only need to enter the basic information about your server, such as the database name, login and password. You also need to provide the IP address of the machine on which the database resides, which is often the same as the IP address of your domain name, or 'localhost' if your database is hosted locally. When generating PHP programs, CodeCharge will also generate the PHPLib library with connection settings based on the above setup. Your server doesn't need ODBC or other drivers. This is only needed during design time.

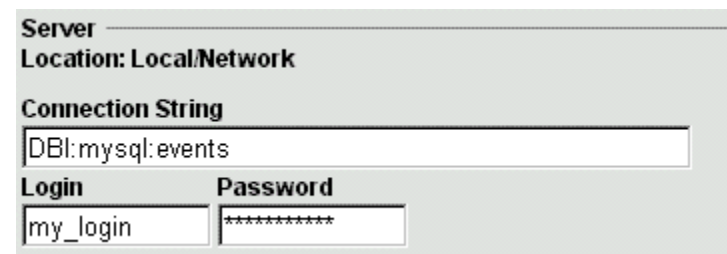
Server Connection for ColdFusion



The screenshot shows a configuration window titled 'Server' with a sub-header 'Location: Local/Network'. It contains input fields for 'DSN' with the value 'events', 'Login' with 'my_login', and 'Password' with a masked value '*****'.

When using ColdFusion, you only need to know your Data Source Name (DSN) and optional Login and Password as shown above. If you don't already have a DSN setup, you can use the ODBC option in Control Panels or the ColdFusion Administrator (<http://coldfusionserver/CFIDE/administrator>) to create the DSN.

Server Connection String for Perl



The screenshot shows a configuration window titled 'Server' with a sub-header 'Location: Local/Network'. It contains input fields for 'Connection String' with the value 'DBI:mysql:events', 'Login' with 'my_login', and 'Password' with a masked value '*****'.

Perl uses a database interface module called DBI to connect to databases. In CodeCharge, you will need to supply a simple connection string specifying the DBI as the connection method, the database type (mysql, odbc, etc.) and the name of the database. Refer to <http://dbi.symbolstone.org/> for more information on DBI or read the short guide to DBI at <http://www.perl.com/pub/1999/10/DBI.html>.

Server Connection String for JSP

Server
Location: Local/Network

Connection String
jdbc:odbc:events

Login Password
my_login *****

Database Driver
sun.jdbc.odbc.JdbcOdbcDriver

When using JSP, it is very common to use the Java Database Connectivity (JDBC) driver to connect to the database. The proper database connection string, also known as URL is usually 'jdbc:(driver type):(database)'. You can also use ODBC in combination with JDBC and use a connection string like **jdbc:odbc:events** as shown above whereby "event" is the name of a DSN. In addition, you will need to supply the class name of your driver, for example **sun.jdbc.odbc.JdbcOdbcDriver**. For more information on establishing JDBC connections, visit <http://java.sun.com/docs/books/tutorial/jdbc/basics/connecting.html>.

Sample Database Settings:

ColdFusion with MySQL
database "tutorial":

General Publishing Database Security Pages Files

Design

Connection String
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=f:\ColdFusion\wwwroot\cfdocs\Tutorial\Tutorial.mdb

Login Password
admin *****

Server
Location: Local/Network

DSN
tutorial

Login Password
admin *****

Database Type
ODBC

JSP connected to an ODBC DSN called "taskman". In this case the JDBC:ODBC bridge is being used. The JDBC ODBC driver is that provided by Sun.

Perl connected to an ODBC DSN called "MyODBCdsn" using the ODBC DBI driver module:

Properties

General Publishing Database Security Pages Files

Design

Connection String

Provider=MSDASQL.1;Persist Security Info=False;Data S 

Login

Admin

Password

Server

Location: Local/Network

Connection String

jdbc:odbc:taskman

Login

admin

Password

Database Driver

sun.jdbc.odbc.JdbcOdbcDriver

General Publishing Database Security Pages Files

Design

Connection String

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=f:\Cor 

Login

admin

Password

Server

Location: Local/Network

Connection String

DBI:ODBC:MyODBCdsn

Login

admin

Password

LongReadLen value for DBI module

1000

PHP with MySQL database
“phpDatabase”:

General Publishing Database Security Pages Files

Design

Connection String

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=f:\Coc

Login Password

admin *****

Server

Location: Local/Network

Database Name

phpDatabase

Login Password

admin *****

Database Type

MySQL ☒ Persistent Connection

Host

127.0.0.1

2.4 Security Tab

As the name suggests, the “Security” tab is used to specify the table and fields to be used in the process of authenticating users. The options under this tab are available to the users who elect to use the standard CodeCharge authentication scheme which defines three levels of security. As such, as a prerequisite to using this scheme, your database has to contain a table with the necessary fields.

Properties

General Publishing Database Security Pages Files

Use Authentication ☒

User access table (table containing user logins)

employees

Primary Key (usually user_id)

employee_id

Login field

login

Password field

pass

Security Level field (optional)

security_level

Parameter	Description
Use Authentication	Select this option to indicate your intention to use the standard CodeCharge authentication scheme whereby you will be able to specify that only certain users can

User access table	access certain pages.
Primary Key	Choose the table from your database that contains user login and password information. Choose the Primary Key field within the User table specified above. This field is normally numeric and each entry is unique (autoincrement or index). Possible field names are user_id, UserID, MemberID and so on.
Login field	Select the field within your User table that contains user login information.
Password field	Select the field within your User table that contains user password information.
Security Level field	Optionally, select the field that contains user Security Levels. This field is used to regulate access to project pages depending on the users security level. For instance, an administrator with a security level of 3 will be able to access all pages but a guest with security level 1 will only be able to access a limited subset of the project pages.

2.5 Pages Tab

The “Pages” tab is used to designate web pages which play a central role in the structure of the website.

Properties

The table below describes the various options:

Parameter	Description
Default Page	This is the main page of the Web site, for example Index (.asp, .php, etc.) or Default (.asp, .php, etc.), etc. This page is called up when you click on the ‘Home’ icon in the top toolbar.
Header Page	The Header page is automatically pre-pended to the top of all the other pages in the website. You can therefore use it to add global content such as a navigation bar or site-wide links.
Footer Page	The Footer page is automatically appended to the bottom of all the other pages in the website. It can similarly be used to add global content.
Login Page	Specify the <i>Page</i> that will be used as the default Login page when users need to login to access a secured page.

Note that when you initially create your project, you will not be able to specify all of the above options since there is likely to be no pages in the project as that point. As you create more pages, you can return to this tab to set the appropriate fields. If you selected the “Use Authentication” option in the “Security” tab, then before you generate and view your project, remember to return to the “Page” tab and designate an appropriate page as the Login page since this is obviously needed when a page requires user authentication.

2.6 Files Tab

Use the “Files” tab to specify a folder which you want to be copied to the production machine when the project is generated. For instance, you might have a folder that contains pictures and media that is used in the project. To ensure that the same resources are available in the production machine, use this tab to specify the folder in question.

General Publishing Database Security Pages **Files**

☒ Copy/Publish Files to the server

Source Folder
 

Destination/Server Folder

The table below describes the various settings:

Parameter	Description
Copy/Publishing Files to the server	Select this checkbox if you want a certain folder containing project resources such as images to be copied to the server when the project is generated. Once the folder and its contents have been copied to the server, you can deselect this checkbox so that the files don't have to be copied again every time the project is generated.
Source Folder	Specify the source path to be copied to the server. You can click on the yellow file icon to open a window and navigate to the desired folder e.g. 'Images'.
Destination/Server Folder	Specify the name of the folder that will be created on the server to contain the contents of the source folder. Usually, this is the same name as that of the Source Folder.

3. Style

Use the style option to setup the general look of the generated pages. CodeCharge comes with a set of predefined styles which you can use as-is or modify to suit your needs. Designating a style is a hierarchical process which begins at the Site level and trickles down to the Page then the Form level.

Note that if at the site level you specify 'None' for the "Type" field, you will not be able to specify page nor form styles later on. By choosing 'None' you are in essence saying that you don't want to use any style elements in your website. The diagram below illustrates this option.

Type
  ☐ Generate File containing Styles

Scheme

Item
 

<BODY

 >

Form Header

Column Caption

Field Caption Data Caption

The above does not mean that you have to specify a particular style at the Site level. If in the “Type” field, you select either ‘HTML’ or ‘CSS’, you will subsequently be able to specify unique styles at the page as well as the form level. As long as the “Type” field has one of the two afore mentioned options, you need not specify a particular Site level style and yet you will still be able to specify Page as well as Form Level styles.

The style “Type” that you select at the Site level will act as a precedent for all the other styles you will specify later. When choosing styles later on, whether at the page level or at the form level, you will have to maintain consistency as far as style “Type” is concerned. This is to say that if you select HTML style “Type” at the site level (e.g. Fiesta(HTML)), then you can’t subsequently select a CSS style “Type” at the Page or Form level (e.g. Amazon(CSS)). You have to maintain the same style “Type” throughout the site. However, this should not be a handicap in any manner since all the styles come in both formats.

The screenshot shows the CodeCharge configuration window. The 'Type' dropdown menu is currently set to 'HTML' and is highlighted with a red rectangular box. To the right of this dropdown is a checkbox labeled 'Generate File containing Styles'. Below the 'Type' section is the 'Scheme' section, which includes a text input field and two buttons: 'Load...' and 'Save As...'. Underneath the 'Scheme' section is the 'Item' section, featuring a dropdown menu currently showing 'PageBODY'. Below the 'Item' section is the '<BODY>' section, which contains a large text area for editing the body content, flanked by help icons. At the bottom of the window, there are four labels: 'Form Header', 'Column Caption', 'Field Caption', and 'Data Caption'.

When you select a particular style, you can view and edit the style elements.

Parameter	Description
Type	Choose type of the Style that CodeCharge should generate. The different Style types are discussed below in detail.
Scheme	Specify the name of the preset Scheme containing color, table and font settings of the site. You can load another Scheme by clicking Load . You can also save the scheme after changing the settings then clicking Save As...
Generate File containing Styles	If you would like the site layout settings to be generated to a separate file such as “site.css” or “site.inc”, select this option.
Item	This is a list of style elements which you can choose and modify. For example, to specify blue, thick Table borders, select Item “FormTABLE” and enter value: <i>border=3 bordercolor="blue" cellpadding=0</i>
<Body>	When you make a selection in the “Item” field, the corresponding item values/properties are displayed here. You can change the values as you wish or add additional elements to enhance the style. Knowledge of Cascading Style Sheets and HTML is an asset here.

The available tag options within the “Item” listbox are:

- ✓ PageBODY
- ✓ FormTABLE
- ✓ FormHeaderTD
- ✓ FormHeaderFONT

- ✓ ColumnCaptionTD
- ✓ ColumnCaptionFONT
- ✓ FieldCaptionTD
- ✓ FieldCaptionFONT
- ✓ RecordSeparatorTD
- ✓ DataTD
- ✓ DataFONT

3.1 CSS Styles

3.1.1 INCLUDED IN HTML FILES

If you opt to use a CSS style type for your site, for example by loading one of the predefined CSS styles such as Artic(CSS), CodeCharge will produce HTML files that include code such as that shown below. This particular code pertains to the FieldCaptionTD and FieldCaptionFONT items.

```
<td style='background-color: #CCCCCC; color: #000000; border-style: outset; border-width: 1'>
<font style='font-size: 10pt; color: #000000'>
Field Caption Text
</font>
</td>
```

The above output will be repeated for each item that appears on the page even if similar elements are used multiple times on the same page. This will cause the HTML files to become large and consequently take longer to load.

3.1.2 PLACED IN SEPARATE CSS FILE

In the interest of speed and size, you can optimize CSS output by selecting the option 'Generate Files containing Styles'. In this case CodeCharge will create a separate file called **Site.css** containing style tags, which includes the definition for each item, for example:

```
.FieldCaptionTD {background-color: #CCCCCC; color: #000000; border-style: outset; border-width: 1}
.FieldCaptionFONT {font-size: 10pt; color: #000000}
```

The generated HTML files will reference the above item definitions wherever applicable without including multiple instances of the definitions for each occurrence of the item.

```
<td class='FieldCaptionTD'>
<font class='FieldCaptionFONT'>
Field Caption Text
</font>
</td>
```

The browser will use the definitions such as 'FieldCaptionTD' or 'FieldCaptionFONT' from the .css file in place of style tag names in .html files. This makes for much less traffic between the server and the browser because each set of properties is defined only once.

3.2 HTML Styles

3.2.1 INCLUDED IN HTML FILES

The HTML style option is very similar to the CSS style option when both are used without the 'Generate Files containing Styles' option. The HTML output will include detailed properties for each item, for example:

```
<td bgcolor='#C0C0C0'>
```

```
<font style='font-size: 10pt; color: #000000; font-family: Arial, Tahoma, Verdana, Helvetica'>
Field Caption Text
</font>
</td>
```

If there are multiple items on the page, the same set of properties will be repeated for each of the items and this could make the HTML files pretty big.

3.2.2 PLACED IN SEPARATE HTML STYLE FILES

When using the HTML style type with the option 'Generate Files containing Styles' selected, the output visible to the user via the browser will be no different than when not using this option. However, the difference will be on the server side. The web pages will be generated using a separate file called **Styles.inc** which will contain HTML tag definitions such as:

```
styleFieldCaptionTD = 'bgcolor="#C0C0C0"'
styleFieldCaptionFONT = 'style="font-size: 10pt; color: #000000; font-family: Arial, Tahoma, Verdana, Helvetica"'
```

And the HTML file on the server will include simplified tags, like:

```
<td {FieldCaptionTD}>
<font {FieldCaptionFONT}>
Project
</font>
</td>
```

When the web page is requested by the browser, the server will read the HTML template file for that page as well as the **Styles.inc** file and will combine them to create output that is the same as when using a single-HTML file. As such, the browser will receive identical HTML content as that shown earlier.

```
<td bgcolor='#C0C0C0'>
<font style='font-size: 10pt; color: #000000; font-family: Arial, Tahoma, Verdana, Helvetica'>
Field Caption Text
</font>
</td>
```

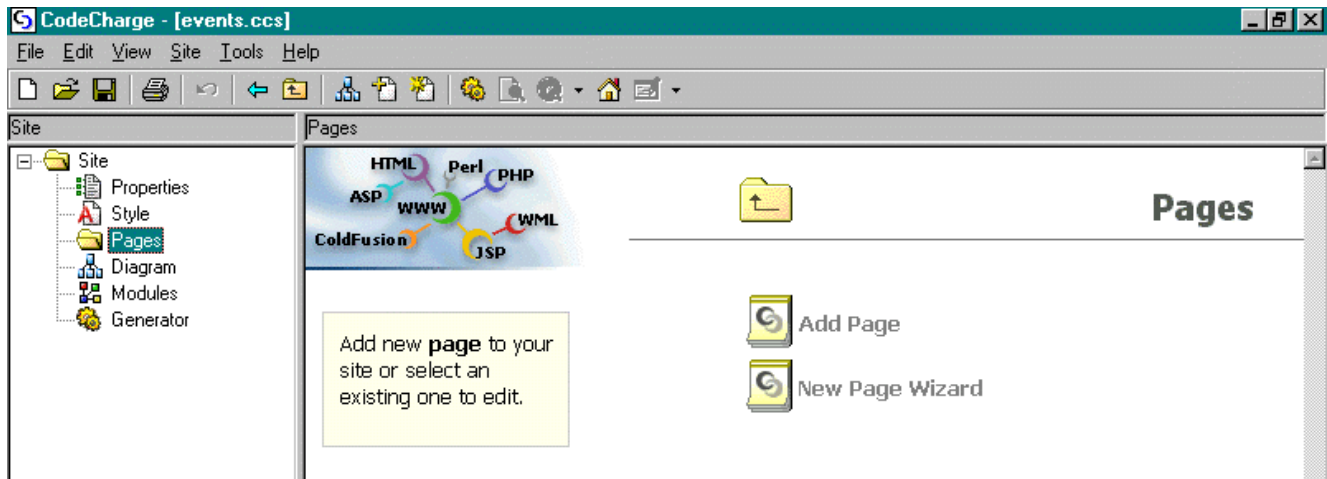
The advantage however, is that the various style items are defined only once in the 'Styles.inc' file.

3.3 Choosing the Type of Style

Based on the above information, it is evident that using CSS styles with the 'Generate File containing Style' option is the most efficient choice. It allows for more compact HTML files by keeping the definitions of various elements in a separate .css file. The only potential issue is that some older browsers may not be capable of handling Cascading Style Sheets correctly. If you have web designers on hand, they may have their own preferences too. Web designers may wish to modify the generated HTML templates to their liking in which case using the HTML styles is a potentially good option.

4. Pages

This is the most commonly used section of CodeCharge, where you add and configure web pages and change existing ones. You can create a page manually by specifying all it necessary options or use the "New Page Wizard" to automate the process of creating various kinds of pages.



After creating a new Page, you can add Forms and select Form Fields to be included on each Page. The example below includes 2 Forms: Search and Bugs on a page titled 'Bug Management'. The web page "Name" which is 'Default' in the example below will be used as the file name. It is therefore important to ensure that each of the project pages have unique names.

Page ✖

Name  

Title 

Security Level

Columnar Layout ☐

Forms    ✖

Search ▶ **Bugs**

Name  **Type**

Form Title **Form Action**

Table 

Fields  

	Key Field	Caption	Type	Size	
↑ ↓	☐ project_id	Project	ListBox		 ✖
	☐ assigned_to	Assigned To	ListBox		 ✖
	☐ priority_id	Priority	ListBox		 ✖
	☐ status_id	Status	ListBox		 ✖

4.1 Page Settings

The top section of the Page contains general Page settings and Form layout preview in icon format:

Page 

Name    

Title

Security Level

Columnar Layout ☐

The table below describes the various parameters:

Parameter	Description
Name	The name of the file that will be generated. In this example, generated files will be named Default.asp and Default.html (when using ASP & Templates).
Title	This name will also be used on the Site Diagram to identify the page. Enter the Title of the Page to be displayed within the browser when users view this page. The title will be included between <code><TITLE></code> and <code></TITLE></code> HTML tags within the generated code.
Security Level	If you chose Authentication in the Security tab of Site Properties , specify the security level needed to access this page.
Columnar Layout	Choosing this option causes the forms in the page to appear in table columns such that the forms are aligned vertically.

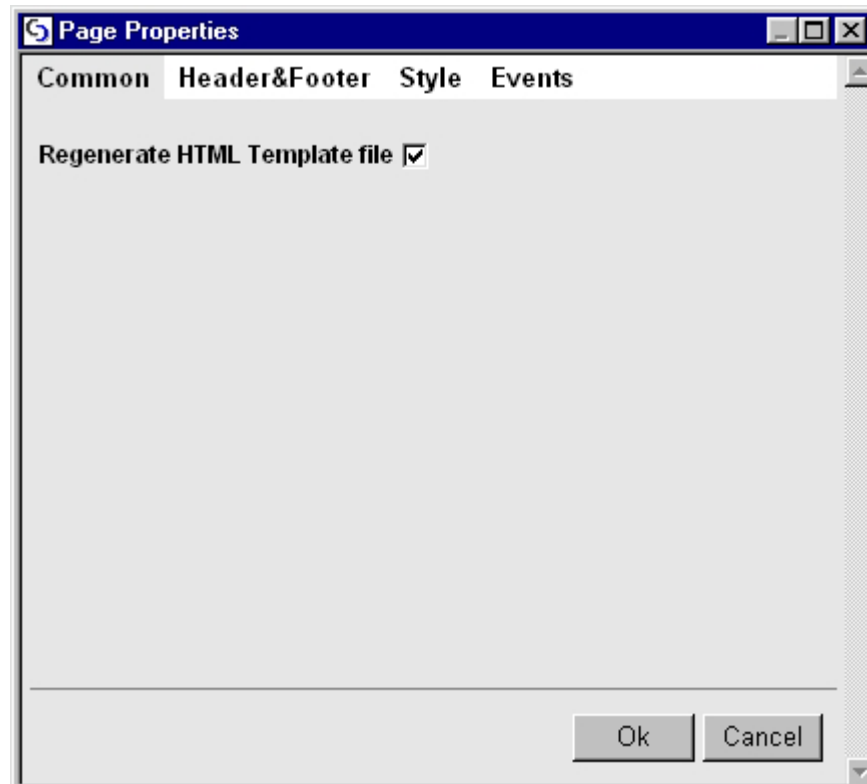
Additionally, the following icons can be used:

Icon	Description
	Click on this icon to specify additional Properties for the page. This opens the "Page Properties" window which is discussed below.
 **Add pic**	Click on this icon to open the Page Info window which contains comprehensive information about the entire site at a glance
	Click on this icon to delete current page.
	Clicking on this icon will generate source files for the current page. For example, <i>Default.asp</i> and <i>Default.html</i> .
	This is a preview of the current page based on the Forms appearing on the page.

4.2 Page Properties

When you click on the  icon, you will access the **Page Properties** window where you can setup more advanced page features.

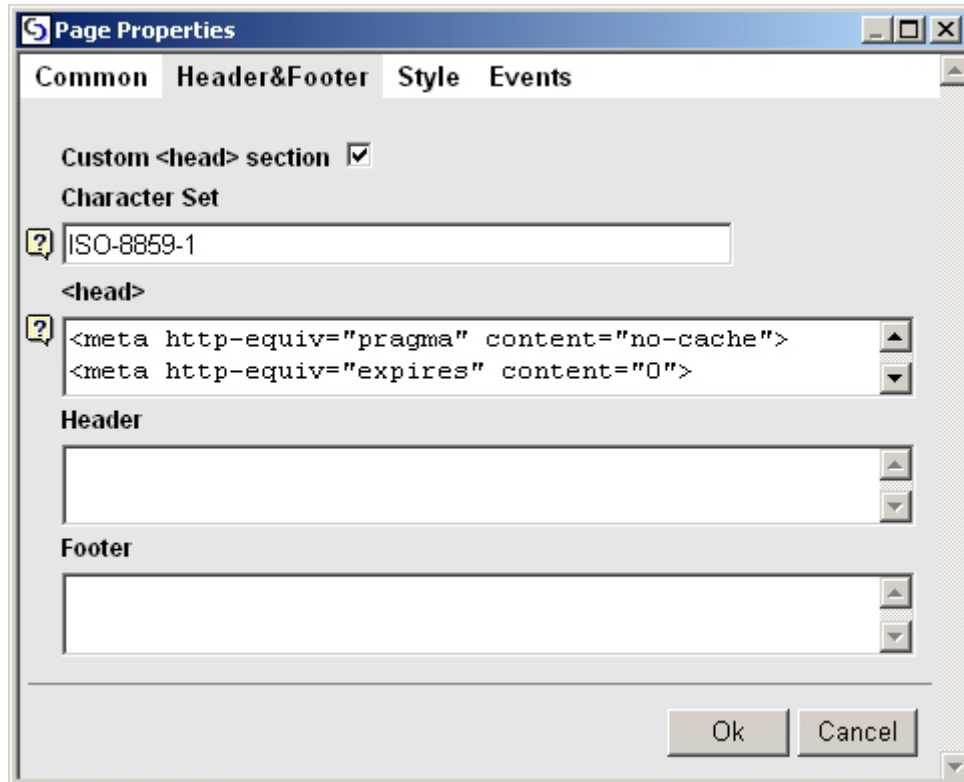
4.2.1 COMMON TAB



Regenerate HTML Template File

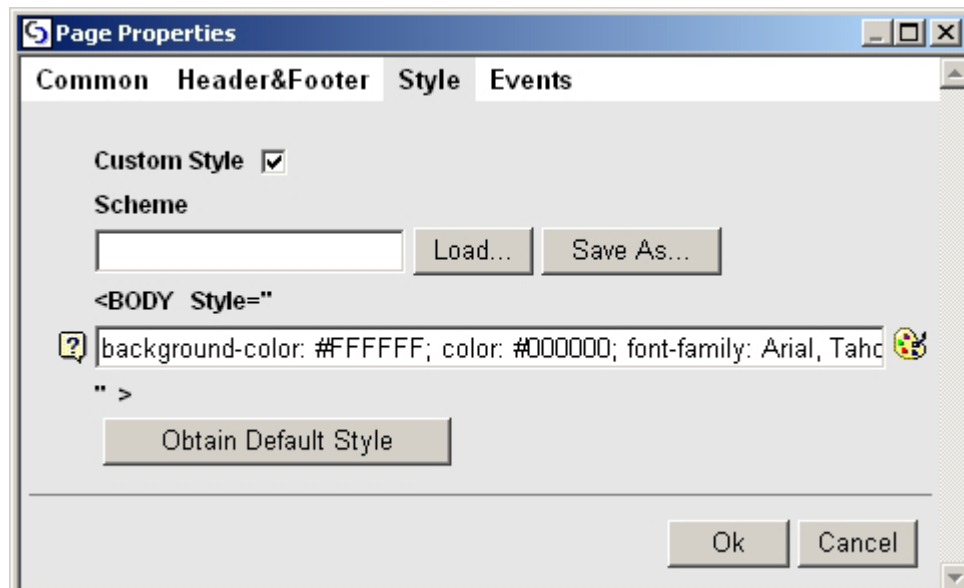
If you want to keep your templates unchanged while regenerating the code you should uncheck the **Regenerate HTML Template File** checkbox.

4.2.2 HEADER&FOOTER TAB



You can also create a custom Header and Footer for each Page. If you select the “Custom <head> section” option, you will be able to specify the ISO encoding for a character set as well as custom html meta tags to appear at the top of the page. This usually comes in handy when you want to localize the language that the web page will be displayed in.

4.2.3 STYLE TAB



Optionally, you can use the style tab to specify a style other than the one selected at the site level. However, remember that the style type selected here has to be consistent with the type that was selected at the site level.

4.2.4 EVENTS TAB

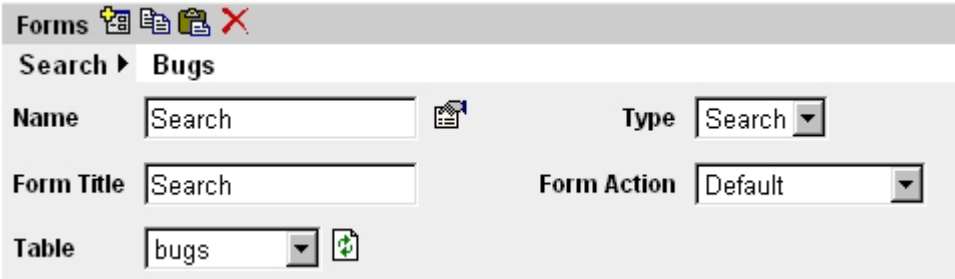


You can also create, save or load custom Events for each Page. This allows you to specify custom code to be executed at certain points during the rendering of the page.

4.3 **Form Settings**

The center section of the **Page** screen allows you to add, modify and configure **Forms** to be included on each page.

For example, a page can contain a Search and a Grid **Form** as shown below:



Icon	Description
	Add a new form to the page
	Copy a form to the clipboard
	Paste a form from the clipboard to the current page
	Delete current Form
	View and enter additional Form Properties. This is described below in detail.

The table below describes the remaining options:

Parameter	Description
Name	The name of the form as it will appear in the generated HTML. The name must not contain spaces or special characters.
Type	CodeCharge includes the following Form Types:

	• Grid (Display multiple database records) • Record (View single record details as well as add and edit records) • Search (Enter criteria to perform a search) • Tree (Display related records Hierarchically) • Menu (Display links to other pages or resources) • Login (Prompt user for authentication details)
Form Title (optional)	Enter text to be displayed at the top of the Form.
Form Action	Select the Page to be opened when the users submit the current Form. For example, this can be the page to be activated when users click the Submit or Search buttons.
Table	You can choose the current Form as the Form Action if you do not want users to leave current page after submitting the Form. Select the database table that contains the data to be displayed or manipulated in the current Form. For example, you can select Form Type: "Grid" and Table: "employees" to create an Employee Listing.

The following are the available icons and their respective use:

Icon	Description
	Refreshes the listbox containing the list of tables (for example, if you make changes to your database)
	Shows the list of fields within the database table, from which you can select fields you would like to include on the Form
	Adds an empty field to the Field section below. You can then configure the field to display data from a particular table column.

4.4 Form Properties

When you click on  icon next to the "Form" field, you will access the Form Properties window where you can setup more advanced Form functions and features.

Form Properties are functionally divided into tabs as follows:

- ✓ Common
- ✓ Input
- ✓ SQL
- ✓ Header&Footer
- ✓ Style
- ✓ Events

The following sections explain these tabs in detail:

4.4.1 COMMON TAB

The common tab contains different options depending on the Form Type you are using.

Grid Form Type

Form Properties

Common Input SQL Header&Footer Style Events

Name:

Allow Sorting: ☒

Allow Insert: ☐

Grid Type: Separate Blocks: ☐

Page Navigation: ☒ Records:

Show First: ☐

Show Prev: ☒

Show Next: ☒

Show Last: ☐

Page Numbers: ☒ Type:

Show Total Pages: ☒

Select the type of **Page Navigator**.

Ok Cancel

A **Grid Form** has the following options:

Parameter	Description
Name	The name of the form as entered on the main screen.
Allow Sorting	Specify if you would like users to sort the table content by clicking on Column Heading (Field Caption)
Allow Insert	Specify if you would like to have a link from this form to the record form where users can add new records to the database. You can also enter the Caption for the link to be displayed. If you select Allow Insert , you must specify Form Action on the main screen which refers to the page with the record form that shall be used to add, edit or delete records.
Grid Type	Select either tabular, columnar or justified layout for the form fields.
Separate Blocks	This option is only available with columnar or justified grid layouts. Usually, with these two layouts, all the records appear within one HTML table. However, with this option selected, each record appears in a separate table. You can modify the RecordSeparatorTD style element to customize the space that appears between the tables.
Page Navigation	Select this option if you want your form to have the capability to navigate through records. With this option selected, you will be able to specify more options regarding how you want the navigation controls to appear. For instance, you can change the text that appears for the controls, the number of records to appear per form, the number of pages to be displayed on the control as well as the order the various controls appear in.

Tree Form Type

The screenshot shows a 'Form Properties' dialog box with the 'Common' tab selected. The 'Name' field contains 'Bugs'. The 'Tree Page' dropdown is set to 'BugRecord'. The 'Items Page', 'Category ID', 'Category Name', and 'Parent Category ID' fields are empty. A yellow box at the bottom left contains the text 'Form Name.'. The 'Ok' and 'Cancel' buttons are at the bottom right.

A **Tree** Form has the following options:

Parameter	Description
Name	The name of the form as entered on the main screen
Tree Page	Select destination page where the full Category List is displayed (Browse page)
Items Page	This is the page that will be used when users drill down to the lowest Category
Category ID	Select the key field within the Categories table
Category Name	Specify database field containing Category names
Parent Category ID	Specify the database field containing the parent category (for multi-level category structures)

Record Form Type

Form Properties

Common Input SQL Header&Footer Style Events

Name: Bugs

Create Validation Rules: ☒

Allow Insert: ☒ Insert

Allow Update: ☒ Update

Allow Delete: ☒ Delete

Allow Cancel: ☒ Cancel

Data Transfer Method: POST

Form Name.

Ok Cancel

A **Record** Form has the following options:

Parameter	Description
Name	The name of the form as entered on the main screen.
Create Validation Rules	Select this option if you would like the program to verify data entered by the user, according to the field properties. The generated code will check to ensure that any values entered in the field comply with the specified length and type attributes.
Allow Insert	Specify if you would like to allow users to add new records to the database from this form. You can also enter the caption to be displayed on the Insert button. If you select Allow Insert, you need to specify a Form Action on the previous (Pages) screen, which will be displayed after users add a new record.
Allow Update	Specify if you would like to allow users to modify database records using this form. You can also enter the caption that will be displayed on the Update/Submit button.
Allow Delete	Specify if you would like to allow users to delete database records using this form. You can also enter the caption that will be displayed on the Delete button.
Allow Cancel	Specify if you would like to show the Cancel/Reset button on this form.
Data Transfer Method	Specify whether the 'Get' or the 'Post' form action method should be used to submit the form values. The 'Get' method uses the URL variables to submit values contained in a form. URL variables are name-value pairs that are sent to a web page appended to the end of the URL. For instance in the URL http://www.codecharge.com?action=deploy , the URL variable pair would be action=deploy. The down side about using the 'Get' method is that with large forms, the URL variables could get out of hand and also any sensitive information in the form would be visible to the user. Additionally, some older browsers are unable to process more than 255 characters in the URL even though most recent browsers don't have this limitation.
	The 'Post' method sends values through HTTP headers and hence the information is not visible to the user. Of the two, the 'Post' method is the preferable one when submitting sensitive information. The modality of retrieving values when either the 'Get' or the 'Post' method is used depends on the particular language being used.

Search Form Type

Form Properties

Common Input Header&Footer Style Events

Name Search

Orientation Vertical

Search Button Caption Search

Data Transfer Method POST

Form Name.

Ok Cancel

If using the **Search** Form Type, you can specify:

Parameter	Description
Name	The name of the form as entered on the main screen.
Orientation	Select "Horizontal" to display search fields within a row; Select "Vertical" to display search fields within a column.
Search Button Caption	Enter caption to be shown on the Search button
Data Transfer Method	Specify whether the 'Get' or the 'Post' form action method should be used to submit the form values.

Menu Form Type

The screenshot shows the 'Form Properties' dialog box with the 'Common' tab selected. The 'Name' field contains the text 'Menu'. The 'Orientation' dropdown menu is set to 'Horizontal'. Below these fields is a yellow rectangular box with the text 'Form Name.'. At the bottom right of the dialog are 'Ok' and 'Cancel' buttons.

When configuring **Menu** Form Type, you can specify:

Parameter	Description
Name	The name of the form as entered on the main screen.
Orientation	Select “Horizontal” to display menu items within a row; Select “Vertical” to display menu items within a column.

4.4.2 INPUT TAB

Use the Input Tab to specify input parameters that the form shall expect or be required to receive.

The screenshot shows the 'Form Properties' dialog box with the 'Input' tab selected. On the left, there is a 'Parameters' list with a plus and minus icon; it contains one item, 'bug_id'. On the right, there are configuration fields: 'Field' (dropdown set to 'bug_id'), 'Variable Name' (text field with 'bug_id'), 'Type' (dropdown set to 'Param'), 'Data Type' (dropdown set to 'Number'), 'Operation' (dropdown set to '='), 'Required' (checkbox, unchecked), 'Transfer' (checkbox, checked), and 'Default Value' (empty text field).

For example, a Search form usually works in conjunction with a Grid form to display the results of the search. To achieve this, the Grid form needs to receive the search criteria and filter the records to be displayed accordingly. The incoming parameters are specified in the Input Tab

The table below describes the various options available in this tab:

Input Parameter	Description
Field	Select the database field used by this Form that should be matched to the Session or Parameter value passed to the current Form.
Variable Name	The name of the incoming parameter.
Type	The type of the incoming parameter. If the variable is submitted over the URL or through form submission the Type should be 'Param'. Select 'Session' if the parameter is a session variable.
Data Type	Specify the data type of the incoming parameter: • Number: Received value will be numeric • Date: Received value will be a date • Text: Received value will be textual • Memo: Received value will contain a large amount of text and correspond to a large text input field in the database (Access memo fields or SQL server text fields)
Operation	Specify the method of comparing the value of the incoming parameter with the value from the database. • =; >; >=; <; <=; <>: These common operators are used to select appropriate records from the table. • LIKE '...%': Received values will be searched for within the database by matching the value to the field using the LIKE (value)% or LIKE (value)* method. This operator is often used for displaying search results. • LIKE '%...%': Received values will be searched for within the database by matching the value to the field using the LIKE %(value)% or LIKE *(value)* method. This operator is often used for displaying search results. The LIKE '%..%' method is usually slower than LIKE '...%' method, because it cannot take advantage of database indexes.
Required Transfer	If the form MUST receive a particular parameter, select this checkbox. Specify if the parameter value should be passed to the next form or self (depending on Form Action)
	Delete an existing input parameter
	Add a new input parameter

4.4.3 SQL TAB

Common Input **SQL** Header&Footer Style Events

Where

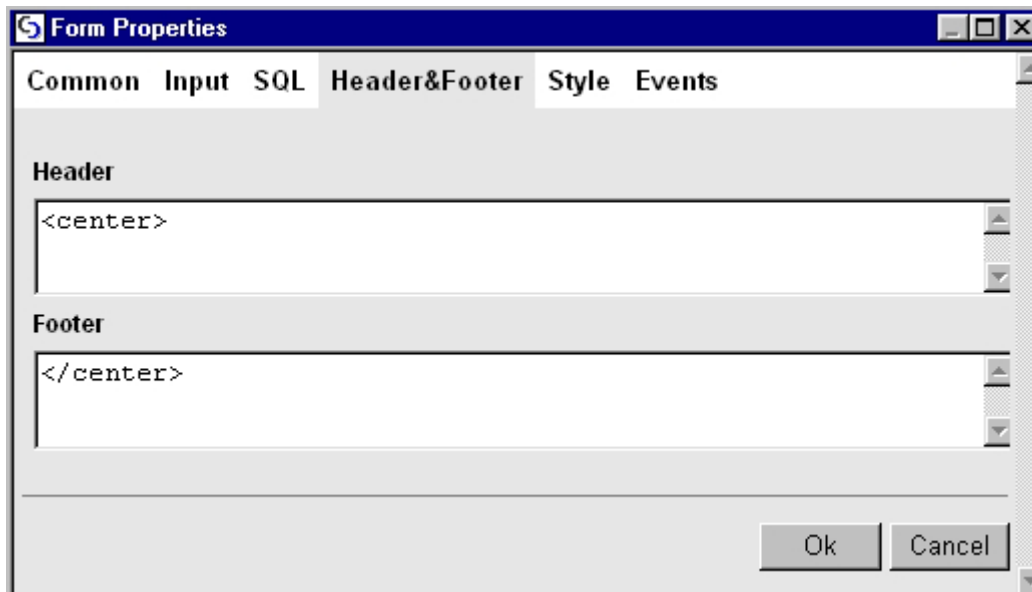
Order by

☒ Use SQL string:

```
select priority_id, name from priorities
```

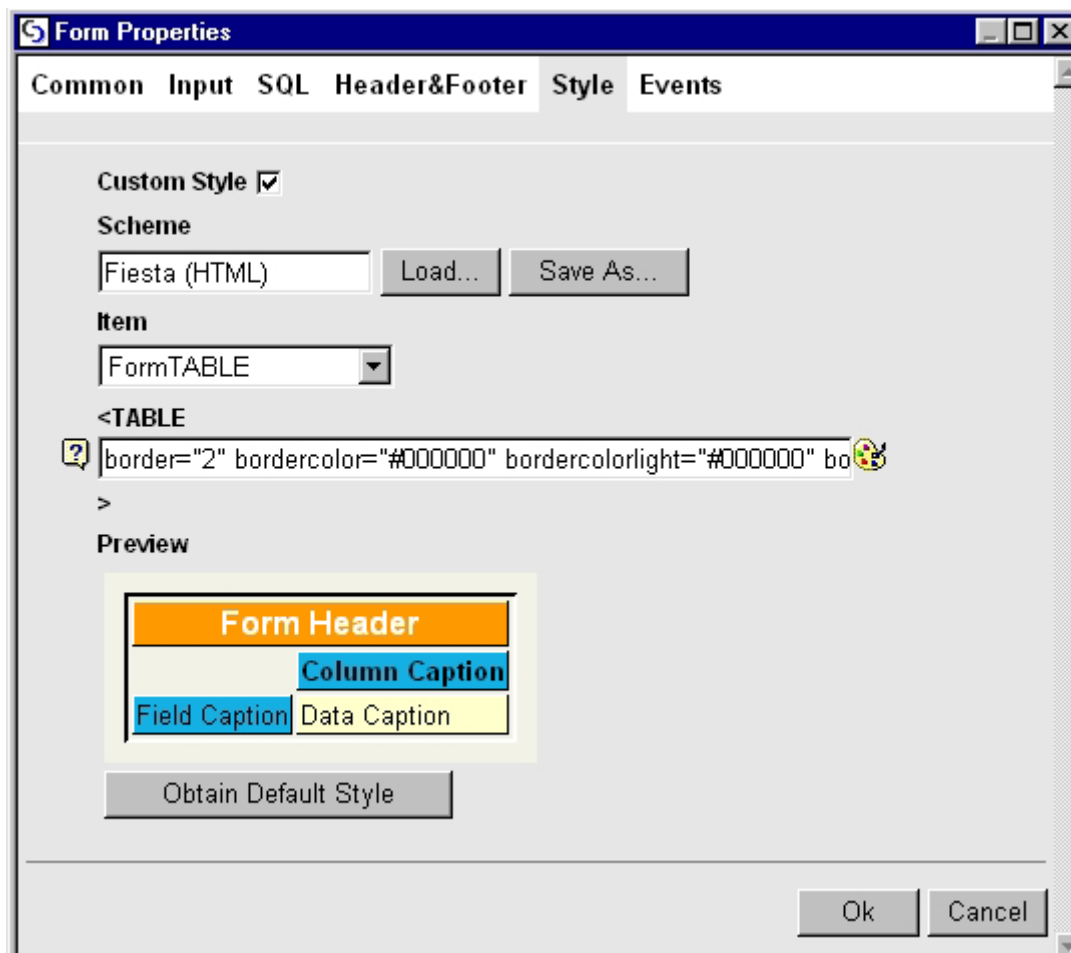
In this tab you can customize the form SQL by entering values for the where clause or order by clause or alternatively entering the entire SQL statement to be used for the form.

4.4.4 HEADER&FOOTER TAB



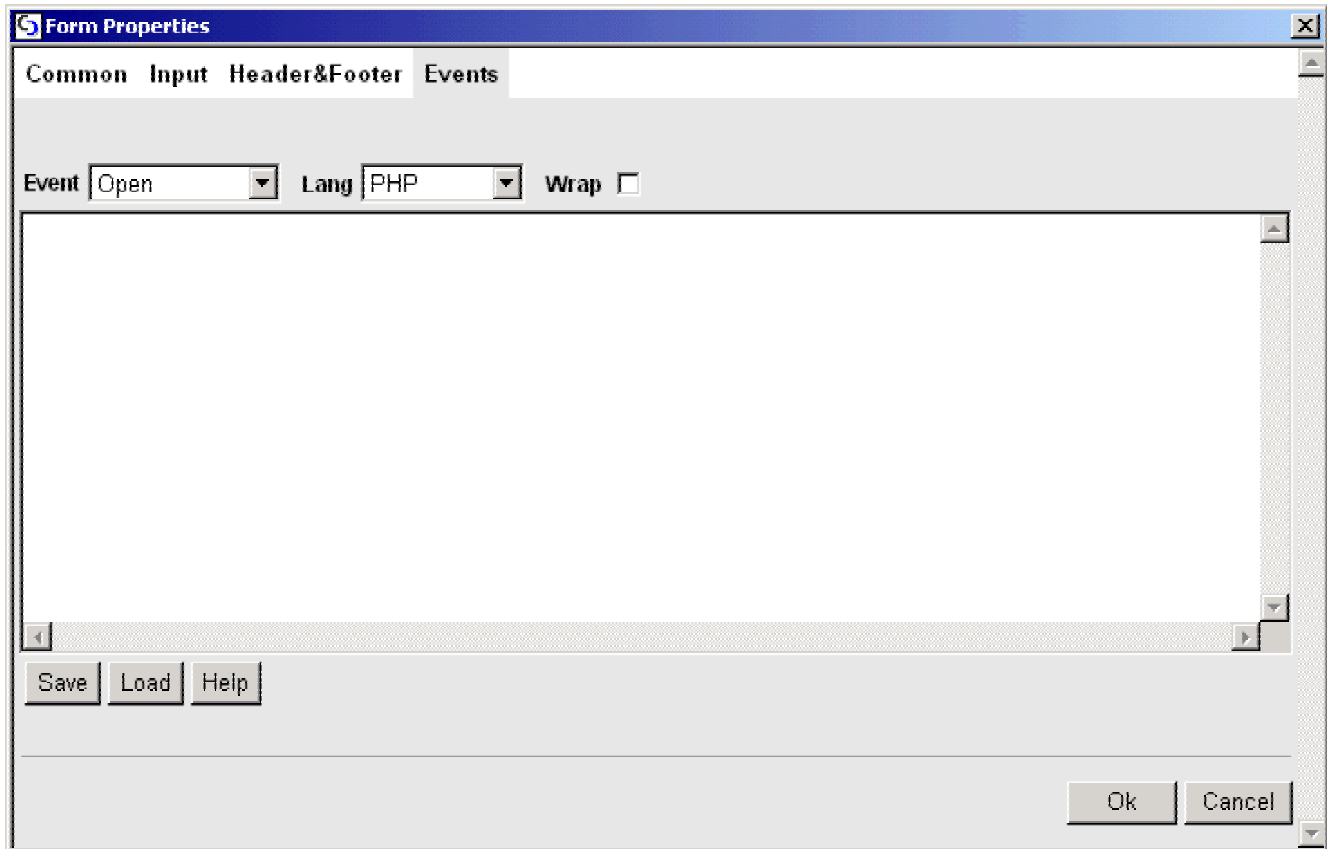
Within this tab, you can customize your form Header and Footer using HTML, JavaScript or language code.

4.4.5 STYLE TAB



Use this tab to select a form style that is different from the previously selected site style or page style. Note that you have to maintain consistency as far as style Type is concerned. If you selected a HTML style at the Site level, you cannot subsequently select a CSS style type at the Page or the Form level.

4.4.6 EVENTS TAB



This screen allows you to enter programming code that will be executed during specific Form events such as Open, Close, Before Show, Custom Show, On Login, On Logout, Before Insert, Before Update, and Before Delete. Refer to section [IV](#) for more information about Page and Form events.

A particular project file can contain event code for different languages. When you switch the language in the “General” tab of site properties, the event code for the selected language becomes active. Later on when you generate the project, only the event code pertaining to the currently selected language is generated.

You can also save event code to a file and similarly load previously saved event code from a file. This comes in handy when you want to swap out a piece of code for another and still keep a copy of the previous version, for instance when you are debugging your project.

When you select any of the ‘Custom’ events such as ‘Custom Show’, ‘Custom Login’, ‘Custom Delete’, etc a button entitled “Obtain Generated Code” will appear. Use this button to acquire the code that is generated by default then manipulate it to meet your requirements. This feature is meant to save you the time and effort of coding the ‘Custom’ event in its entirety.

4.5 Field Settings

The lower section of the **Page** template allows you to add, modify and configure **Fields** to be included within each Form on the page.

The following is an example of a Grid form with five fields:

Key Field	Field Name	Caption	Type	Size
☒	item_id	item_id	Hidden	0
☐	name	Name	TextBox	30
☐	category_id	Category	ListBox	
☐	price	Price	TextBox	10
☐	notes	Notes	TextArea	30

The table below describes the field settings:

Field Setting	Description
Key	For some Forms, like Record form, you must specify the primary key field in the database table being used with this Form.
Field	Select the database field that you can want to display or use with this Form.
Caption	Enter the text to be displayed on the page to identify this field. Note that you can enter HTML code into this field.
Type	Choose one of the following Types: • TextBox: An entry box allowing users to enter one line of text • URL: Same as Label field, but links to another page within the project • Label: Plain text output showing the values of the database field • Hidden: Field that is not shown on the page, but used for example as the key to the current table • ListBox: Drop-down menu field linked to another table and showing values from another table • TextArea: An entry box allowing users to enter several lines of text, usually used for maintaining Memo fields within the database • CheckBox: A single checkbox linked to the True/False or Yes/No type fields within the database • RadioList: A group of mutually exclusively boolean controls whereby only one can be 'on'.
↑ ↓	Change order of fields; click on the up-arrow to move the current field up and on the down-arrow to move the field down.
	View and modify Field Properties, as described below.
	Delete current field.

4.6 Field Properties

When you click on the icon, you will access the Field Properties window where you can setup more advanced field features.

Field Properties are divided functionally into tabs as follows:

- ✓ Common
- ✓ Output
- ✓ Join / List

The following sections explain these tabs in detail:

4.6.1 COMMON TAB

The screenshot shows the 'Field Properties' dialog box with the 'Common' tab selected. On the left, a list of fields includes 'Home', 'Reg' (highlighted), 'Shop', and 'Admin'. To the right, the 'Name' field is populated with 'Reg'. The 'Default Value' field is empty. The 'Data Type' is set to 'Number' via a dropdown menu. The 'Page Link' is set to 'Registration' via a dropdown menu. A yellow warning box at the bottom center reads: 'Enter the field name. This field cannot be blank'. At the bottom right are 'Ok' and 'Cancel' buttons.

The available options in this tab depend on the particular type of field.

The table below describes the various field options:

Field Property	Description
Name	The field name used to uniquely identify the field
Default Value	The default value to be used for the field if a database or parameter value does not exist.
Data Type	The type of data contained in the field (number, date, text or memo)
URL Field Type	
Page Link	The destination page that the field should be linked to
Use URL Field	Specify if the URL values are contained in a database field
URL Field	Select the field that contains the destination URL values
TextBox Field Type	
Max Length	Maximum number of characters allowed in the field
Password	Check this option to designate a password field whereby the text entered will be shown as asterisks (*)
Required	Specify if this field is required. When Form Validation is turned on, a message will be displayed if users do not enter any information for this field
Label Field Type	
HTML	Specify if the data within this field contains HTML code that should be displayed as HTML

ListBox Field Type	
Height	Number of rows of the listbox that should be visible
Custom Initial Value	This refers to the value that appears at the top of the listbox and is usually selected by default. Activate this option if you wish to specify the value to be displayed
Display Value	Enter text that will be displayed for the custom initial value
Submit Value	The corresponding value that is submitted when the custom initial display value is selected
TextArea Field Type	
Rows	Number of visible rows for the textarea control
Required	Specify if this field is required. When Form Validation is turned on, a message will be displayed if users do not enter any information for this field
Checkbox Field Type	
Required	Specify if field is required. When Form Validation is turned on, a message will be displayed if users do not enter any information for this field
Checked/Unchecked Value	Enter values to be returned when users check or uncheck checkbox
RadioList Field Type	
Orientation	Specify whether the radio buttons are to be aligned vertically or horizontally
Required	Specify if this field is required. When Form Validation is turned on, a message will be displayed if users do not enter any information for this field
Unique	With this option selected, a check is made before database content is altered to ensure that the operation will not result in a duplicate value

4.6.2 LIST TAB

Field Properties

Fields: project_id, project_name, **employee_id**

Common List

Table: employees

Primary Key: employee_id

Showing: employee_name

Lookup SQL:

List Of Values:

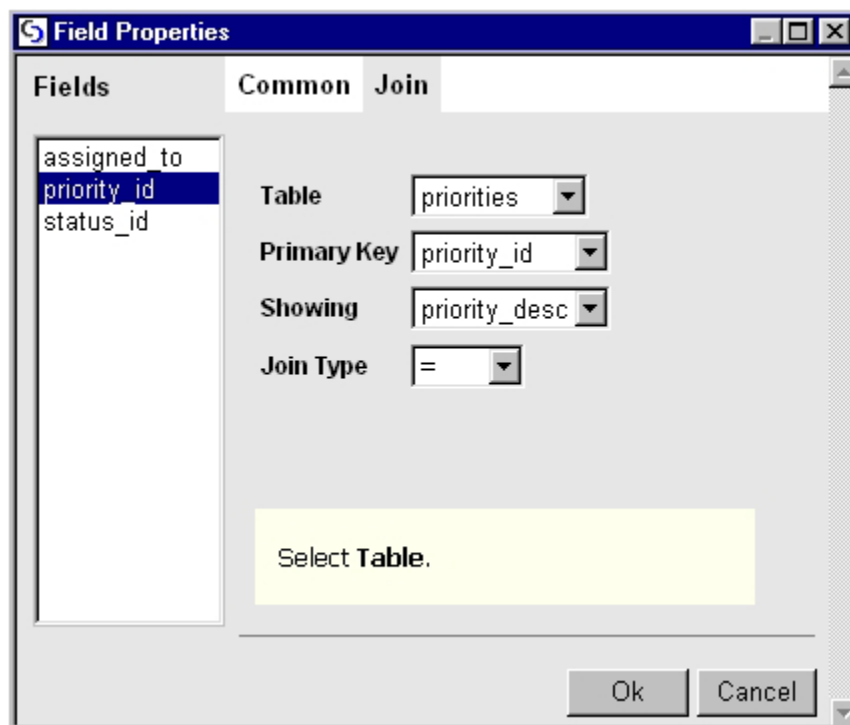
Select Table

Ok Cancel

This tab is displayed for **ListBox** and **RadioList** type fields only. The table below describes the various options:

Parameter	Description
Table	Select the database table from which the related values will be retrieved
Primary Key	Choose the Primary Key field for the table selected above
Showing	Select the field to be displayed within the drop-down menu
Lookup SQL	Enter custom SQL for populating the drop-down menu. An important point to note when writing the Lookup SQL is that you must select two fields from the database. The first field should be the primary key of the table and the other should be the field that the user shall actually see in the listbox. For example, the following SQL statement would create a listbox similar to that which would be created using the field settings shown in the diagram above. <i>select employee_id, employee_name from employees</i>
List of Values	Enter a custom list of values for populating the listbox. The syntax for doing this is <i>submitValue;displayValue</i> whereby the submitValue is what is submitted when the submit button is clicked and the displayValue is the value shown to the user. An example would be <i>1;John;2;Patrick;3;Sally;4;Jen</i>

4.6.3 JOIN TAB

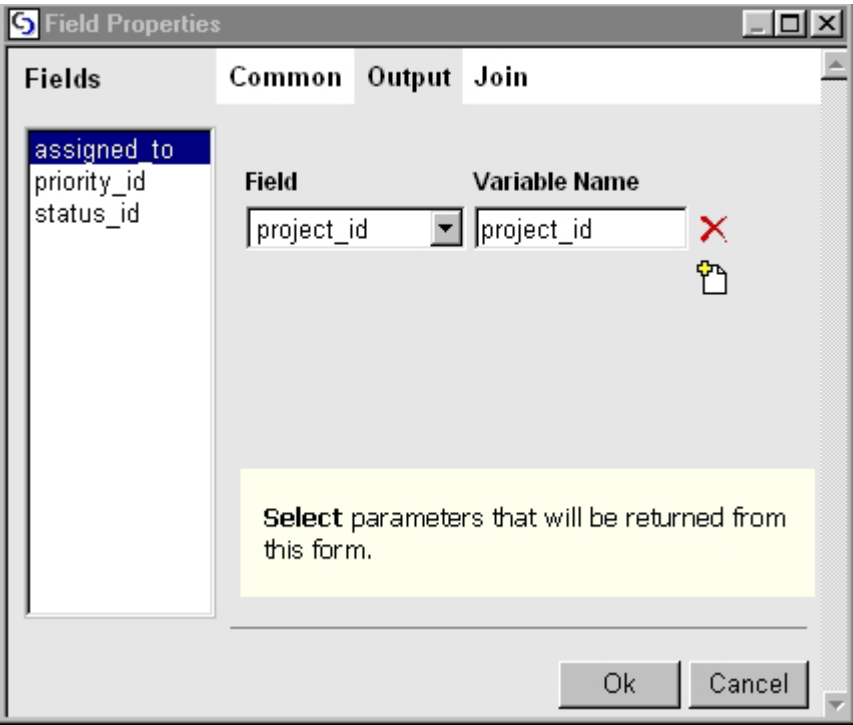


Join Tab is displayed with **URL** and **Label** field types only. The table below describes the various options:

Parameter	Description
Table	Select the database table from which the related values will be retrieved
Primary Key	Choose the Primary Key field for the table selected above
Showing	Select the field to be displayed within this field
Left Join	If this option is selected, a SQL left join will be performed such that unmatched records from the first (left) table are included but any unmatched records from the second (right) table are not included in the result set.

4.6.4 **OUTPUT TAB**

This tab is only available for URL type fields. You can setup output parameters to be transferred to the page that is linked to.



Parameter	Description
Field	Select a field from among the list of available database table fields whose value will be passed as a parameter to the receiving page.
Variable Name	You can assign a name of your choice to the parameter that will be passed. This will not affect the actual value passed.
	Delete an already existent parameter
	Add a new output parameter

5. Diagram

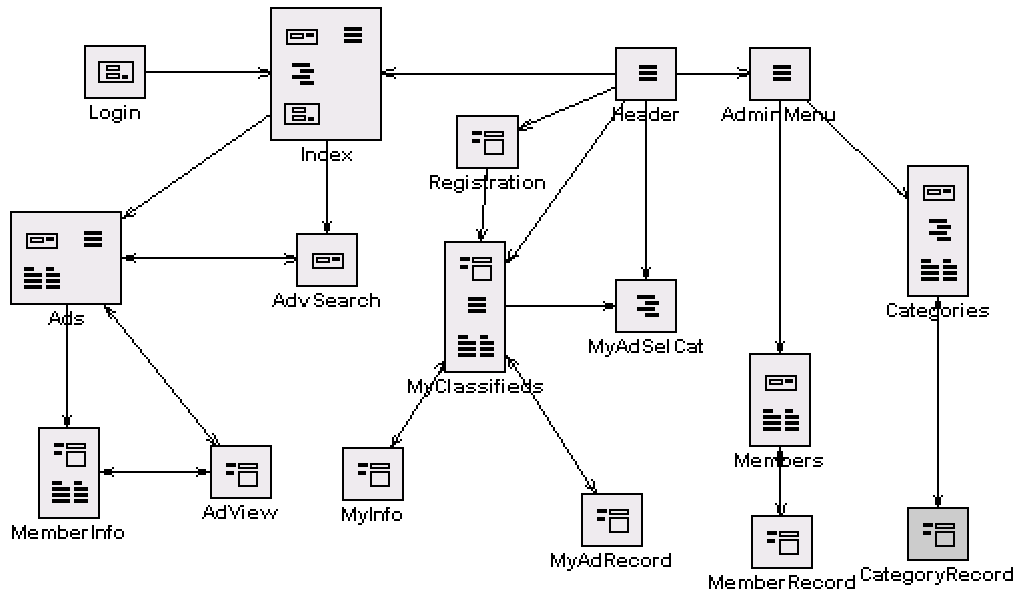
The **Diagram** section within CodeCharge allows you to view the site layout after you have setup the pages for your site.

CodeCharge automatically creates this diagram based on your project pages, Form Input and Field Output settings.

Using the site diagram component you can perform the following operations:

- Right click on a page icon to open a drop down menu with a variety of options. Note that if you select the delete option, the concerned page will be deleted from the site entirely and not just from the diagram
- Right click on any part of the window to add a new page or open the new page wizard
- Double-click on the Web page icon to view its configuration screen
- Move page icons on the screen for better visual presentation of all Web components

- Zoom in and out the diagram by right-clicking on it and choosing the appropriate option
- Print the diagram for review, documentation or other purposes



6. Modules and Events

Events are functions and snippets of code executed throughout various locations in the code. By creating Events you can override and customize the generated code or add your own functions. Events that are global and apply to all pages are categorized as Modules, while other Events apply to individual Pages and Forms.

Note: The Events are language specific. Depending on your needs, you might want to recreate the same event in ASP, PHP, JSP, Perl, etc.

The “Modules” section allows for the customization of various features in the code generated by CodeCharge. Here you can enter your own functions that will be applied to all pages on your site. For example, you can add an Encryption Function in the **Global Functions** section and then use it in individual pages or you can override the current security mechanism by creating your own **Custom Authentication** Event.

Events

Event	Description
Global Functions	Allows for the addition of custom functions which will be placed in the common.[ext] file (for example Common.asp when using ASP). As such, these functions will be accessible to all pages since all pages import the common.[ext] file.
Common	The code placed in this event essentially replaces all the default code that is placed in the common.[ext] file. Use this event with caution since the common.[ext] file is pivotal to the functioning of the entire site
Custom Connection	Use this event to write code that overrides the default database connection mechanism
Open Any Page	Any code placed in this event will be executed at the beginning of all the pages in the site. This event can further be overridden by the Open event at the page level
Custom Authentication	Use this event to write code that overrides the default authentication mechanism in sites which implement security

Variables

Variable	Description
cn	ADO Connection Object.
sAppPath	The physical path on the server to the generated files of your site.
sHeaderFileName*	The name of the file which contains a template for the header.
sFooterFileName*	The name of the file which contains a template for the footer
strConn	ADO Connection String as per the settings configured under the Database tab of site properties
UserID**	The User ID taken from the security table for the logged user.
UserRights**	The User Permissions taken from the security table for the logged user.

* Template version only

** Session variables which are available in sites that implement security. The session variables are created after a user logs in successfully

Examples

Custom Connection (ASP)

This example demonstrates how to connect to an ORACLE database using custom login and password values stored in session variables. The session variables can be set in the **OnLogin** Event of the Login Form.

```
if Session("Login") <> "" and Session("Password") <> "" then
  set cn = Server.CreateObject("ADODB.Connection")
  on error resume next
  cn.open "DSN=Test_ORACLE_DB", Session("Login"), Session("Password")
  if cn.errors.count > 0 then bLogged = False
  on error resume 0
end if
```

Open Any Page (ASP)

This example demonstrates how to increase page Hit count when each page is opened.

```
Session("Hits")=Session("Hits")+1
```

Custom Authentication

The following example shows how to call your own function to authenticate each page.

```
MyVerification(PageSecurityLevel)
```

Global Functions

The following example shows how to call your own function to authenticate each page.

```
Sub MyVerification(PageSecurityLevel)
  if PageSecurityLevel < Session("Level") then
    response.write("Access denied")
    response.end
  endif
End sub
```

7. Generator

The Generator section allows you to specify which files are to be generated. If you edit a file outside CodeCharge, you can deselect the checkbox next to the page so that the file is not generated again thereby overwriting your modifications.

To generate the site pages, click **Generate**.

Generator

Generate Pages ☒ ☐

☒ Administration

☒ BugRecord

☒ Default

☒ EmployeeList

☒ EmployeeMaint

☒ Login

☒ Page

☒ PriorityList

☒ PriorityMaint

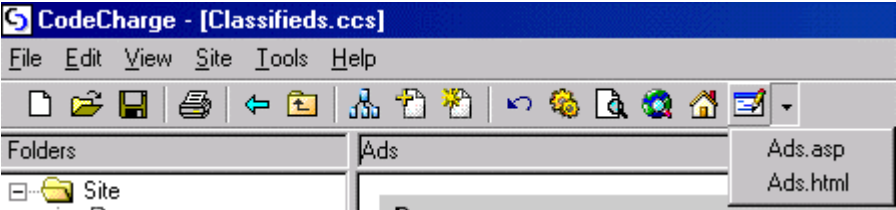
☒ ProjectList

☒ ProjectMaint

☒ StatusList

☒ StatusMaint

8. Toolbar



The CodeCharge toolbar contains the following icons/features:

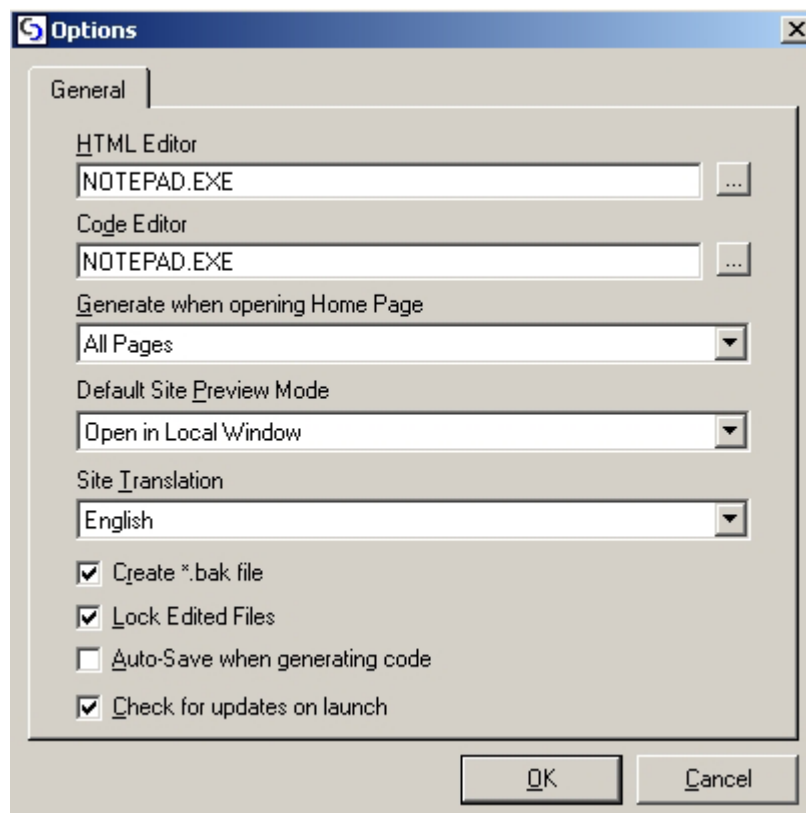
Icon	Description
	Create a New CodeCharge project

	Open an existing CodeCharge Project Save the current project. You can also use the 'Save As...' option in the <u>F</u>ile Menu to assign a new name to the project Print the contents of the current window Navigate to the previous selection Navigate to a higher selection in the hierarchy of options View site diagram Add a New Page Open the New Page Wizard Undo last operation Generate the site Preview current page layout Go Live – generate code and open current page in the Web browser Go to Home Page – generate code and open the Default page in the Web browser Edit a file using the editor specified in the <u>T</u>ools ► Options menu
---	---

9. Options

Use Tools ► Options menu to configure operational options for CodeCharge.

The following are the available options:

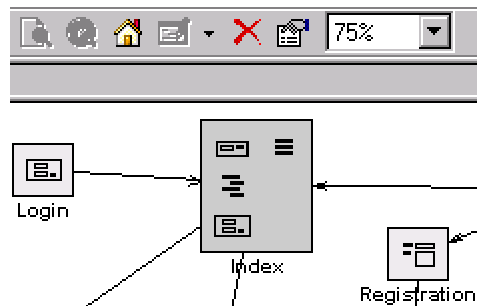


The table below describes the various options:

Option	Description
HTML Editor	Specify an external HTML editor (Microsoft Front Page, Macromedia DreamWeaver and so on) that you would like to use to modify generated HTML templates
Code Editor	Specify an external editor that you like to use to make code modifications
Generate when opening Home Page	Specify which pages should be generated before opening the Home page. You can specify None, All or just the current page
Default Site	Specify if you would like to open a new window (launch external browser) or use the current CodeCharge window to preview site pages
Preview Mode	Choose the language of your choice. If you don't see an appropriate choice, you can create an XML file for your language and place it in the translations folder under your CodeCharge installation. Use the available XML files as a base for the file you create
Site Translation	Check this options so that a backup copy of your project file is made when you open and modify a project in CodeCharge
Create *.bak file	Specify if you would like to lock pages that you manually edit to prevent them from being overwritten by CodeCharge
Lock edited files	With this option selected, the project will always be saved prior to code generation
Auto-Save when generating code	With this option selected, CodeCharge will check for updates over the web and prompt you if any new updates are available. This check takes place whenever you launch CodeCharge.
Check for updates on launch	

10. Diagram Mode

When in Diagram mode, some more options are available in the Toolbar as shown below:

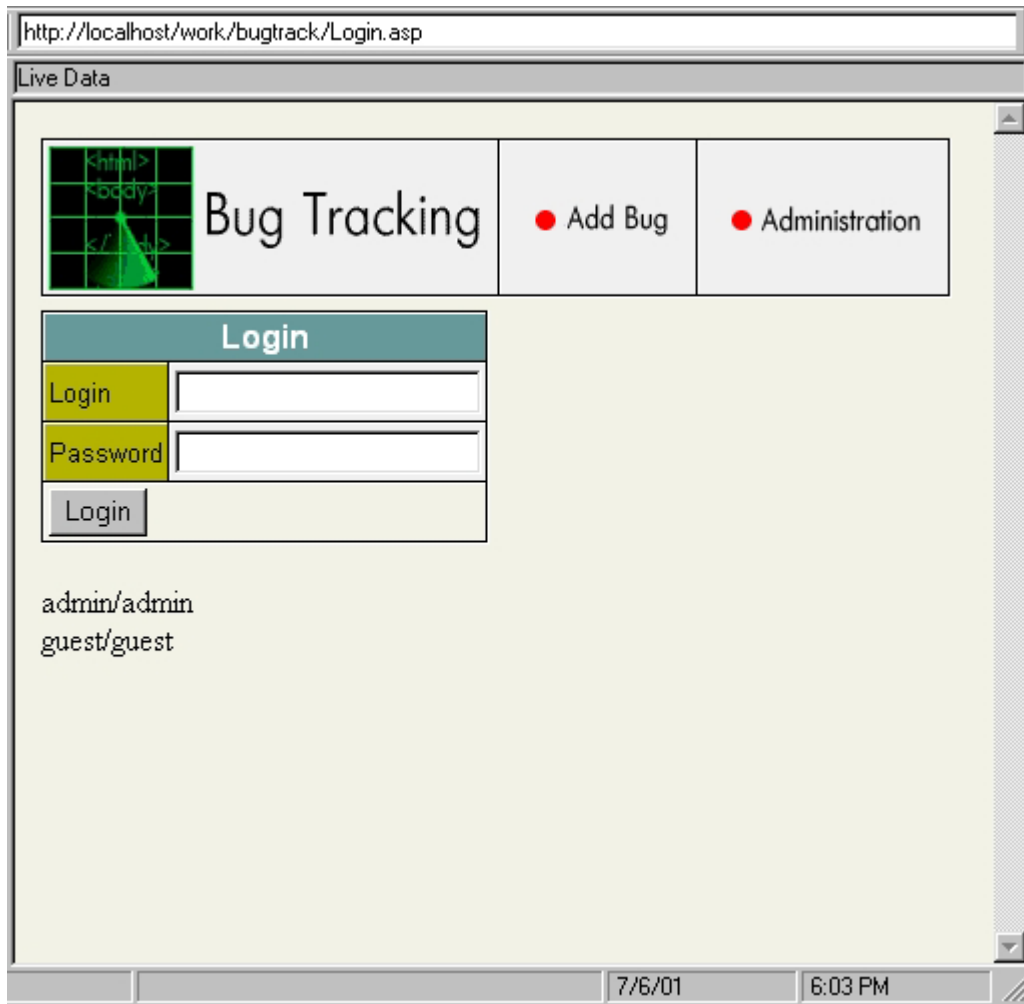


The table below describes the various options:

Option	Description
	Delete the currently selected page
	Edit the currently selected page
	Specify the zoom aspect of the site diagram

11. Live Data Mode

When you click on the  'Go live' icon, the Internet Explorer bar appears on the [Tool Bar](#) and you can view the generated page as it would appear in a Browser window.



When previewing a page, an address bar appears at the top of the page and shows the URL that maps to the page. You can also manually enter the path to any generated page within the **Bar** and click on  **Go** button to view it.

The table below describes the options available in the navigation bar:

Option	Description
	Go Back
	Go Forward
	Stop the page loading process
	Refresh the page

IV. Events

1. Page Events

The following are the events that are available at the Page level. These events can be accessed under the Events tab of the Page Properties window. An important point to remember about the events that begin with the word *Custom* is that you can use the "Obtain Generated Code" button to retrieve the code that would be generated by CodeCharge by default. You can then use this code as a basis to make your modifications.

Events

Event type	Description
Custom Includes	Use this event to specify files that should be included into the current page. By default, all the pages include the common.[ext] file and optionally include the header and footer files if they are defined for the site.
Custom Security	Use this event to customize or otherwise modify how security is handled in a specific page
Open	This event occurs at the onset of the actual page production process and can be used to set or alter variables that are used in the rest of the page such as the variables that hold the names of the header and footer pages
Custom Show Page	This event essentially determines how the entire page is displayed and as such, caution is advised when using it.
Close	This event occurs after all the page forms have been produced and is usually used to do cleanup tasks such as updating database tables or appending other files to the end of the current page

Variables

Variable	Description
sFileName	The current file name
sTemplateFileName*	Contains the name of the HTML template for the current page
sHeaderFileName	Contains the name of the file to be used as the header
sFooterFileName	Contains the name of the file to be used as the footer

* Template version only

Examples

Custom Security

The code below is an example of verifying user privileges against the database based on group access.

```
Rs=cn.execute("select * from groups, users where users.groups_id=groups.group_id and user=" & session("User"))
If rs("rights") = "Developer" then
    Response.write("You are not authorized to access this page")
    Response.end
End if
```

2. Form Events

2.1 Form Types

Forms

Type	Description
Grid	Generally used to display multiple records or the results of a previously conducted search.
Record	Used to display single record details. Used to add, edit or delete records.
Search	Used to facilitate a search of other records based on the fields included in the form.
Menu	Used to create navigational content such as links to other pages in the site.
Tree	Enables you to include Categories on your page or any hierarchically structured content.
Login	Contains fields into which authentication details are entered.

2.1.1 GRID FORM EVENTS

Events

Event type	Description
Open	Executed before the form is processed
Close	Executed after the form is processed
Before Show	Executed before the form output is shown
Custom Show	Allows for overriding the default form output
Custom Navigation	Allows you to code your own record navigation mechanism

Variables

Variable	Description
sAction	The current form action being executed
sForm	Contains the name of the form
sFormTitle	Contains the title of the form
sSQL	SQL statement to be executed
sWhere	WHERE statement based on the Form Input parameters or WHERE statement entered in the SQL tab. If you don't use joins or custom filters, then the "WHERE" statement is usually placed in the sSQL variable
sOrder	ORDER BY statement based on the Form Sorting options. You can modify sOrder to add or override sort fields
fld<FieldName>	All field values are stored in variables that have the prefix "fld" as in fld<FieldName>. Available only in Before Show event
rs	The data source for the grid. For example Recordset object in ASP. Please see the generated code for more details. Available only in Before Show event

Examples

Applying a custom filter

Open Form Event

```
if Session("UserID") = "" then
  sWhere="category_id=5 and user_id=" & Session("UserID")
else
  sWhere="category_id is Null"
end if
```

Printing the total amount of items which are placed in the user-shopping cart.

Before Show Event

```
fldprice="$" & fldprice
iTotal = iTotal & fldprice
```

Close Form Event (ASP without templates)

Response.Write "<hr>Total: \$" & iTotal

2.1.2 RECORD FORM EVENTS

Events

Event type	Description
Open	Executed before the form is processed.
Close	Executed after the form is processed
Before Show	Executed before the form output is shown. You can use fld<field_name> variables
Before Show Insert	Executed before creating the blank record form. Allows for modification of the initial values of the fields.
Before Show Update	Recommended for modifying or using fld<field_name> variables. Executed before showing the record from the database on the form. Recommended for modifying or using fld<field_name> variables.
Custom Show	Allows for overriding the default form output.
Before Execute	Executed before the code for any of the form actions is executed.
Before Insert	Executed before inserting records into the database. Often used to modify the SQL query or the data. You can use sSQL, fld<field_name>, s<Form_Name>Err variables
Before Update	Executed before updating records in the database. Often used to modify the SQL query or the data. You can use sSQL, fld<field_name>, s<Form_Name>Err variables
Before Delete	Executed before deleting records from the database. Often used to modify the SQL query or the data. You can use sSQL, fld<field_name>, s<Form_Name>Err variables
Before Cancel	Executed before the cancel action code is executed.
After Insert	Executed after records are successfully inserted into the database. You can use sSQL, fld<field_name>, sWhere variables
After Update	Executed after records are successfully updated in the database. You can use sSQL, fld<field_name>, sWhere variables
After Delete	Executed after records are successfully deleted from the database. You can use sSQL, fld<field_name>, sWhere variables
Custom Action	Allows overriding of the standard form action processing
Validation	Allows for custom data validation. You can use s<Form_Name>Err, and fld<Field_Name> variables

Variables

Variable	Description
sForm	Form Name
sAction	The current action. During insert action, this variable will contain "insert", during update it will contain "update", etc.
sFormTitle	Contains the title of the form
s<FormName>Err	The description of the error that occurred while processing the form action.
sSQL	SQL statement to be executed.
sWhere	WHERE statement for update and delete actions.
bPK	Boolean. The value is True if the form receives the Primary Key and False otherwise.
p<FieldName>	The value of the Primary Key.
fld<FieldName>	The value of the table or query field usually sent to the page using URL variables or data posted by a form. If the variable s<FormName>Err is not blank (""), then fld<FieldName> contains the value of the form field. See the generated code for more details.
sActionFileName	Name of the file name where the page will be redirected.

Examples

Setting the default value for fldDate field.

Before Show Insert Event

```
fldDate=now()
```

Validating E-mail field before the insert or update. The function IsEmail returns True if e-mail is correct. It can be defined in the Global Functions module of the site.

Validation Event

```
if not IsEmail(fldemail) then  
    sFormNameErr = sFormNameErr & "Email is wrong"  
end if
```

Execute stored procedure.

Before Insert

```
sSQL="stored_procedure_add_user(" & fldUserName & ")
```

Check if user can delete records from the current table.

Before Delete

```
rs=cn.execute("select * from groups, users where users.groups_id=groups.group_id and user=" &  
session("User"))  
If rs("rights") = "Developer" then  
    sFormNameErr="You are not authorized to delete records"  
End if
```

Populate a log table with information about database operations that have been performed.

After Insert

```
cn.execute("insert into log (user_id, operation_id, log_date) values (" & Session("UserID") & ", "delete"," &  
now() & ")")
```

2.1.3 SEARCH FORM EVENTS

Events

Event type	Description
Open	Executed before the form is processed.
Close	Executed after the form is processed
Before Show	Executed before the form output is shown.
Custom Show	Allows for overriding the default form output.

Variables

Variable	Description
sFormTitle	Contains the title of the form
sActionFileName	Name of the file name where the page will be redirected.
fld<FieldName>	The value of variables which are used in Search.

2.1.4 MENU FORM EVENTS

Event type	Description
Open	Executed before the form is processed.
Before Show	Executed before the form output is shown.
Custom Show	Allows for overriding the default form output.

Variables

Variable	Description
sForm	Form Name
sAction	The current form action being executed.
sFormTitle	Contains the title of the form
fld<FieldName>	Contains the value of the field called <FieldName>. Available only in the Before Show event.

2.1.5 TREE FORM EVENTS

Events

Event type	Description
Open	Executed before the form is processed.
Before Show	Executed before the form output is shown.
Show Category	Override the display of a category.
Close	Executed after the form is processed
Custom Action	Specify and execute a custom action.
Custom Show	Allows for overriding the default form output.

Variables

Variable	Description
sForm	Form Name
sAction	The current form action being executed.
sFormTitle	Contains the title of the form
sSQL	SQL statement to be executed
sWhere	Where clause to be used in the SQL statement

2.1.6 LOGIN FORM EVENTS

Events

Event type	Description
Open	Executed before the form is processed.
Close	Executed after the form is processed
Before Show	Executed before the form output is shown.
Custom Show	Allows for overriding the default form output.
On Login	Executed after obtaining the Login and Password, but before validating it. You can use sLogin and sPassword variables.
Custom Login	Executed instead of the Login process. Allows for overriding of the standard user login form action.
On Logout	Executed directly before the Logout process.
Custom Logout	Executed instead of the Logout process. Allows for overriding of the standard user logout form action.

Variables

Variable	Description
sForm	Form Name
sAction	The current form action being executed.
sFormTitle	Contains the title of the form
S<FormName>Err	The description of the error that occurred while processing form action.
sActionFileName	Name of the file name where the page will be redirected.
sLogin	User login.
sPassword	User password.

bPassed	Boolean value. True if user login and password was found in the user table. False otherwise.
---------	---

Examples

Use built-in security of the SQL database.

Custom Login

```
sLogin = GetParam("Login")
sPassword = GetParam("Password")
on error resume next
cn.open strConn, sLogin, sPassword
if cn.error.count > 0 or Err.Description <> "" then
    sLoginErr="Connectin Failed." & Err.Description
else
    Session("User")=sLogin
end if
```

Custom Logout

```
Session("Login") = ""
Session("Password") = ""
```

3. Function List

3.1 ASP functions

Name	Description
openrs(rs, sql)	Creates a forward-only recordset using the current database connection and the SQL statement passed to the function
openStaticRS(rs, sql)	Creates a static recordset using the current database connection and the SQL statement passed to the function
dlookup(table, field, condition)	Retrieves a field value from a database table based on the criteria provided in the condition input parameter
ToHTML(strValue)	Converts special characters to HTML using the HTMLEncode method of the server object
ToURL(strValue)	Makes a string URL format compliant using the URLEncode method of the server object
GetValueHTML(rs, strFieldName)	Obtains a HTML encoded value of a database field
GetValue(rs, strFieldName)	Obtains a database field value from a recordset
GetParam(ParamName)	Retrieves URL or Form parameters
ToSQL(Value, sType)	Makes a value SQL format compatible
getCheckBoxValue(sVal, CheckedValue, UnCheckedValue, sType)	Obtains a checkbox value depending on the user specified values and field type
getValFromLOV(sVal, aArr)	Obtains a lookup value from an array containing a List Of Values
ProcessError	Checks whether an error has occurred and return an error message if one occurred
CheckSecurity(level)	Checks whether a user is allowed to access a page and redirect to the login page if needed

3.2 PHP functions

dlookup(table, field, condition)	Retrieves a field value from a database table based on the criteria provided in the condition input parameter
tohtml(\$strValue)	Converts non-standard characters to HTML

tourl(\$strValue) \$db->f("field_name") \$db->query(\$sql) \$db->next_record() get_param(\$param_name) tosql(\$value, \$type) is_number(\$string_value) get_checkbox_value(\$value, \$checked_value, \$unchecked_value, \$type) get_lov_value(\$value, \$array) check_security(\$security_level) get_session(\$param_name) set_session(\$param_name) strip(\$value) db_fill_array(\$sql_query) get_db_value(\$sql)	Converts value to the URL compatible format Obtains the database field value Executes the SQL command Moves the pointer to the next record and returns false if the recordset end is reached Obtains the specified URL or Form Parameter Makes a value SQL format compatible Checks if a value is numeric Obtains a checkbox value depending on the user specified values and field type Obtains a lookup value from an array containing a List Of Values Checks whether a user is allowed to access a page and redirect to the login page if needed Gets the session variable value Sets the session variable value Returns a string with the backslashes stripped off Returns a name->value array resulting from the SQL statement. Field list should consist of two fields: name and value Looks up a single value in a database
--	---

3.3 Perl functions

dlookup(\$table, \$field, \$condition) ToHTML(\$strValue) ToURL(\$strValue) GetValueHTML(\$rehash, \$strFieldName) GetValue(\$ref_rec, \$strFieldName) GetParam(\$ParamName) ToSQL(\$Value, \$sType) IsNumeric(\$value) getCheckBoxValue(\$sVal, \$CheckedValue, \$UnCheckedValue, \$sType) getValFromLOV(\$sVal, \$aArr) ProcessError CheckSecurity(level) exec_sql(\$sql) openrs(\$sSQL openrs_arr(\$sSQL) GetCookieValue(\$name) GenerateSessionID GetSessionVar(\$name) SetSessionVar(\$name,value) DelExpiredSessions replace(\$source,\$find,\$replace) get_form_parameters http_redirect	Retrieves a field value from a database table based on the criteria provided in the \$condition input parameter Converts non-standard characters to HTML Converts value to the URL compatible format Obtains the HTML value of a field Obtains the database field value Obtains the specified URL or Form Parameter Makes a value SQL format compatible Checks if a value is numeric Obtains a checkbox value depending on the user specified values and field type Obtains a lookup value from an array containing a List Of Values Checks whether an error has occurred and return an error message if one occurred Checks whether a user is allowed to access a page and redirect to the login page if needed Executes the SQL statement Executes the SQL SELECT statement and fills an array with the resulting records. Returns an array of references to hashes. Each hash is one record (field_name->field_value) Executes the SQL SELECT statement and fills an array with the resulting records. Returns an array of the references to arrays. Each second-level array is one record (field_number->field_value) Returns a value for the cookie specified by its name Returns the just generated session ID Returns the session variable specified by the name Sets the session variable Deletes the temporary session files for the expired sessions Finds and replaces a substring in the string Parses the HTTP parameters and returns a hash containing all of them Redirects the script

3.4 JSP functions

dLookup(table, field, condition)	Retrieves a field value from a database table based on the criteria provided in the condition input parameter
String toHTML(String str)	Converts non-standard characters to HTML
String toURL(String strValue)	Converts value to the URL compatible format
String getValueHTML(ResultSet rs, String fieldName)	Obtains a HTML encoded value of a database field
String getValue(ResultSet rs, String fieldName)	Obtains a database field value
String getParam(HttpServletRequest req, String paramName, int paramType)	Obtains a specified URL or Form Parameter
String toSQL (String value, in type)	Makes a value SQL format compatible
boolean isNumber (String param)	Checks if a value is numeric
getCheckBoxValue(sVal, CheckedValue, UnCheckedValue, sType)	Obtains a checkbox value depending on the user specified values and field type
String getValFromLOV(String selectedValue , String sLOV)	Obtains a lookup value from an array containing a List Of Values
String ProcessError (ServletResponse response, Exception e)	Checks whether an error has occurred and return an error message if one occurred
String CheckSecurity(int iLevel, HttpSession session, HttpServletResponse response, HttpServletRequest request, String scriptName)	Checks whether a user is allowed to access a page and redirect to the login page if needed
cn()	Returns "sendRedirect" if the check was unsuccessful or "" otherwise
openrs(java.sql.Statement stat, String sql)	Returns a database connection object
openStaticRS(java.sql.Statement stat, String sql)	Returns a resultset
absolute(java.sql.ResultSet rs, int row)	Returns a static resultset
absoluteRS(java.sql.ResultSet rs, int row)	Moves a resultset pointer to a given position
getFieldsName(ResultSet rs)	Moves a resultset pointer to a given position. Returns false if the position is out of range.
getRecordToHash(ResultSet rs, HashTable h, String [] fields)	Returns an array of field names
getLongCountRec(String sql)	Puts the record into a HashTable object
isEmpty (String val)	Gets the number of the selected records
toWhereSQL(String fieldName, String fieldVal, int type)	Checks if the String variable has no value (null, empty or undefined)
replace(String str, String pattern, String replace)	Returns a SQL compatible where clause
getOptions(Connection conn, String sql, boolean isInitValue, String selectedValue, String sSubmitValue, String sDisplayValue)	Finds and replaces a substring in a string.
getRadio(Connection conn, String sql, boolean isInitValue,	Returns a string containing the list of options for a <SELECT> tag

String selectedValue, String sSubmitValue, String sDisplayValue, String name, String sStyle,boolean isLayout) getOptionsLOV(String sLOV, boolean isInitValue, String selectedValue, String sSubmitValue, String sDisplayValue) getRadioLOV(String sLOV, boolean isInitValue, String selectedValue, String sSubmitValue, String sDisplayValue, String name , String sStyle,boolean isLayout) getAppPath() stringToInt (String str, int defaultValue)	Returns the html options for the <SELECT> tag based on a List of Values Based on a List of Values, this function returns a string containing the value of a Radio control Returns the application path on the server Converts a string to an integer
---	---

3.5 Cold Fusion tags

<CF_DlookUP APP_NAME TABLE sField sWhere> <CF_TOHTML Value> <CF_TOSQL VALUE> <CF_getCheckBoxValue VALUE CheckedValue UncheckedValue TYPE> <CF_getValFromLOV Val Ar> <CF_CHECKSECURITY APP_NAME LEVEL> <CF_OPENRS APP_NAME NAME SQL>	Retrieves a field value from a database table based on the criteria provided in the sWhere parameter Converts non-standard characters to HTML Makes a value SQL format compatible Obtains a checkbox value depending on the user specified values and field type Obtains a lookup value from an array containing a List Of Values Checks whether a user is allowed to access a page and redirect to the login page if needed Opens a recordset based on the provided SQL statement

V. Forms Overview

To efficiently use CodeCharge, you should understand the basic steps and requirements of setting up the following types of forms in site pages:

- ✓ Grid
- ✓ Record
- ✓ Search
- ✓ Menu
- ✓ Tree
- ✓ Login

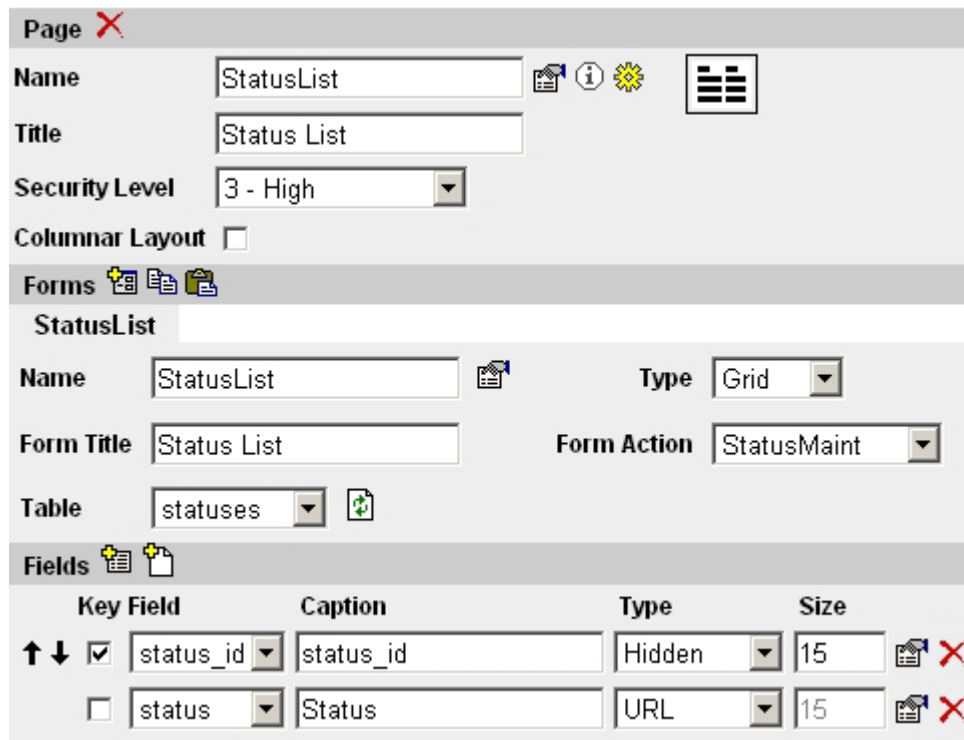
The following section discusses the various form types and shows how they are typically used:

1. Grid Form

A Grid form is used to display multiple records from a database table. If the Grid form receives any input parameters, then the contents of the grid are filtered according to the input parameters. This scenario usually occurs when a Grid form is used in conjunction with a Search form.

To create a Grid form, select the Grid option from the Type field in the form section. You can then enter various form options such as Name, Title and Form Action. The grid contents will depend on the database table that is selected in the Table field.

Use the  icon to open a window with a list of the available table fields and select the fields you want to show in the grid.



The screenshot shows the CodeCharge form configuration interface. At the top, there's a 'Page' header with a red 'X' icon. Below it, the 'Name' field is set to 'StatusList', 'Title' is 'Status List', and 'Security Level' is '3 - High'. There's a 'Columnar Layout' checkbox which is unchecked. Below this is a 'Forms' section with a 'StatusList' sub-section. In this sub-section, 'Name' is 'StatusList', 'Type' is 'Grid', 'Form Title' is 'Status List', 'Form Action' is 'StatusMaint', and 'Table' is 'statuses'. Below the 'Forms' section is a 'Fields' section with a table of fields. The table has columns for 'Key Field', 'Caption', 'Type', and 'Size'. There are two rows of fields: one for 'status_id' (Hidden, 15) and one for 'status' (URL, 15). Each row has a checkbox for 'Key Field' and a 'Status' icon.

Key Field	Caption	Type	Size
☒ status_id	status_id	Hidden	15
☐ status	Status	URL	15

Additionally, you can click on the  icon to open the Form Properties window and further customize the form.

Once you add fields to a form, you can change the field Type depending on what function the field should serve as well as specify a field Size if appropriate. More field options can be accessed in the Field Properties window which varies depending on the type of the field.

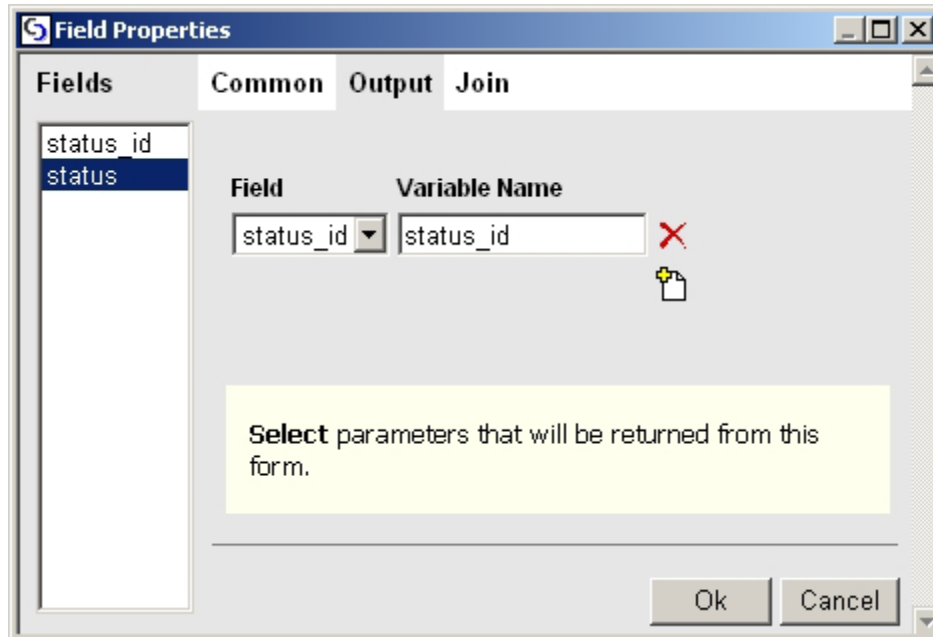
For example, the diagram below shows the Field Properties window for a URL field:

The image shows a 'Field Properties' dialog box with four tabs: 'Fields', 'Common', 'Output', and 'Join'. The 'Fields' tab is active, displaying a list of fields on the left: 'status_id' and 'status'. The 'status' field is selected. To the right of the list, various properties are configured for the selected field:

- Name:** status
- Default Value:** (empty text box)
- Data Type:** Text (dropdown menu)
- Page Link:** StatusMaint (dropdown menu)
- Use URL Field:** ☐
- URL Field:** (empty dropdown menu)
- Html:** ☐
- Add Transit Params:** ☒

A yellow warning box at the bottom states: 'Enter the field name. This field cannot be blank.' At the bottom right are 'Ok' and 'Cancel' buttons.

The diagram above shows that the URL field is linked to the "StatusMaint" page. As is common with URL fields, you can also specify output parameters to be passed to the linked page. The diagram below illustrates this:



As per the settings above, the “StatusMaint” page will receive an input parameter in the form of a name-value pair transmitted via the URL (e.g status_id=4).

The screenshot below shows the Grid Form generated by CodeCharge:

Status List
Status
Open
On hold
Closed
In progress
Add New Status

2. Record Form

Record forms are used to add, edit or delete database records or create forms that display single records.

You can create a simple Employee Maintenance form as follows:

Select **Record** Form type, specify that you will use the **employees** table, then select the fields to be included in your Form:

Page ✖

Name EmployeeMaint   

Title Employee Maintenance

Security Level 3 - High

Columnar Layout ☐

Forms   

Employee

Name Employee  **Type** Record

Form Title Add/Edit Employee **Form Action** EmployeeList

Table employees 

Fields  

	Key Field	Caption	Type	Size		
↑ ↓	☒ employee_id	employee_id	Hidden	5		✖
	☐ employee_name	Name	TextBox	30		✖
	☐ email	Email	TextBox	30		✖
	☐ login	Login	TextBox	15		✖
	☐ pass	Password	TextBox	15		✖
	☐ security_level	Security Level	ListBox	5		✖

CodeCharge subsequently generates the following form:

Add/Edit Employee	
Name	Terry Lincoln
Email	terry@nowhere.com
Login	terry
Password	*****
Security Level	Developer
Update Delete	

3. Search Form

A search form is usually used in conjunction with a Grid form whereby the search form contains the fields used to specify the search criteria and the Grid form displays the results based on the search criteria.

The diagram below shows a sample Search form setup whereby four ListBox fields will determine the search criteria:

Page ✖

Name Default    

Title Bug Management

Security Level No authentication ▼

Columnar Layout ☐

Forms    ✖

Search ▶ **Bugs**

Name Search  **Type** Search ▼

Form Title Search **Form Action** Default ▼

Table bugs ▼ 

Fields  

	Key Field	Caption	Type	Size	
↑ ↓	☐ project_id ▼	Project	ListBox ▼		 ✖
	☐ assigned_to ▼	Assigned To	ListBox ▼		 ✖
	☐ priority_id ▼	Priority	ListBox ▼		 ✖
	☐ status_id ▼	Status	ListBox ▼		 ✖

CodeCharge subsequently generates a form such as the one shown below:

Search	
Project	CodeCharge ▼
Assigned To	Administrator ▼
Priority	High ▼
Status	Closed ▼
Search	

4. Menu Form

As the name implies, the Menu form is a pretty straightforward form that usually contains navigational controls to other pages.

To create a Menu form, select **Menu** Form Type and add several **URL** Type Fields, one for each Web page we want to link to. The diagram below shows an example setup:

Page ✖

Name Administration   

Title Administration

Security Level 3 - High

Columnar Layout ☐

Forms    ✖

Administration ▶ **Login**

Name Administration  **Type** Menu

Form Title Administration **Form Action**

Table

Fields  

	Key Field	Caption	Type	Size	
↑ ↓	☐	Employees	URL		 ✖
	☐	Priorities	URL		 ✖
	☐	Projects	URL		 ✖
	☐	Statuses	URL		 ✖

For each Menu field specify the target Page as shown below:

Field Properties

Fields **Common** **Output**

employees
priorities
projects
status

Name employees

Default Value

Data Type Number

Page Link EmployeeList

Use URL Field ☐

URL Field 

Html ☐

Add Transit Params ☒

Using the above settings, CodeCharge will generate a form similar to that shown below:

Administration
Employees
Priorities
Projects
Statuses

5. Tree Form

The **Tree** form enables you to include categorical or hierarchically structured content on your page. To use this feature, you must first create a database table having a field as the parent of another field. An example is shown below using the Microsoft Access database:

categories : Table		
	Field Name	Data Type
	category_id	AutoNumber
	name	Text
	par_category_id	Number

The next step is to create a Page that contains the Tree Type Form:

Page ✖

Name

Default

Title

Links

Security Level

No authentication

Columnar Layout

☐

Forms

✖

Search

◀ Tree ▶

Lates

Name

Tree

Type

Tree

Form Title

Categories

Form Action

BrowseCategories

Table

categories

Once you select the form Type and the database table to be used, open the Form Properties window and specify the relationship between the various fields to be displayed:

Name

Tree

Tree Page

BrowseCategories

Items Page

BrowseCategories

Category ID

category_id

Category Name

category_desc

Parent Category ID

parent_category_id

CodeCharge subsequently generates a form such as the one shown below:

[Top > Employment](#)
[Administrative and Corporate](#)
[Media, Arts, and Design](#)
[Facilities Management and Maintenance](#)
[Financial Services](#)
[Health Care](#)
[Information Technology](#)
[Engineering](#)

6. Login Form

The commonly used Login Form is one of the easiest to setup in CodeCharge.

To create a Login Page (with Login Form), within Page Editor, simply select **Login** in the **Type** dropdown menu:

The screenshot shows the CodeCharge Page Editor interface. At the top, there's a 'Page' header with a red 'X' icon. Below it, the 'Name' field is set to 'Login', the 'Title' field is set to 'Links', and the 'Security Level' is set to 'No authentication'. The 'Columnar Layout' checkbox is unchecked. Below this, there's a 'Forms' section with a plus icon and a trash icon. The 'Login' form is selected, and its configuration is shown below. The 'Name' field is set to 'Login', the 'Type' dropdown is set to 'Login', the 'Form Title' field is set to 'Login', and the 'Form Action' dropdown is set to 'AdminMenu'. The 'Table' field is empty, and there's a plus icon next to it.

That is all that is need for the Login form. However, remember that you need to have previously configured security settings in the **Site ► Properties ► Security** tab as shown below:

General Database Security Pages

Use Authentication ☒

User access table (table containing user logins)
employees

Primary Key (usually user_id)
employee_id

Login field
login

Password field
pass

Security Level field (optional)
security_level

With that done, CodeCharge will generate the following fully functional Login page:

Login

Login

Password

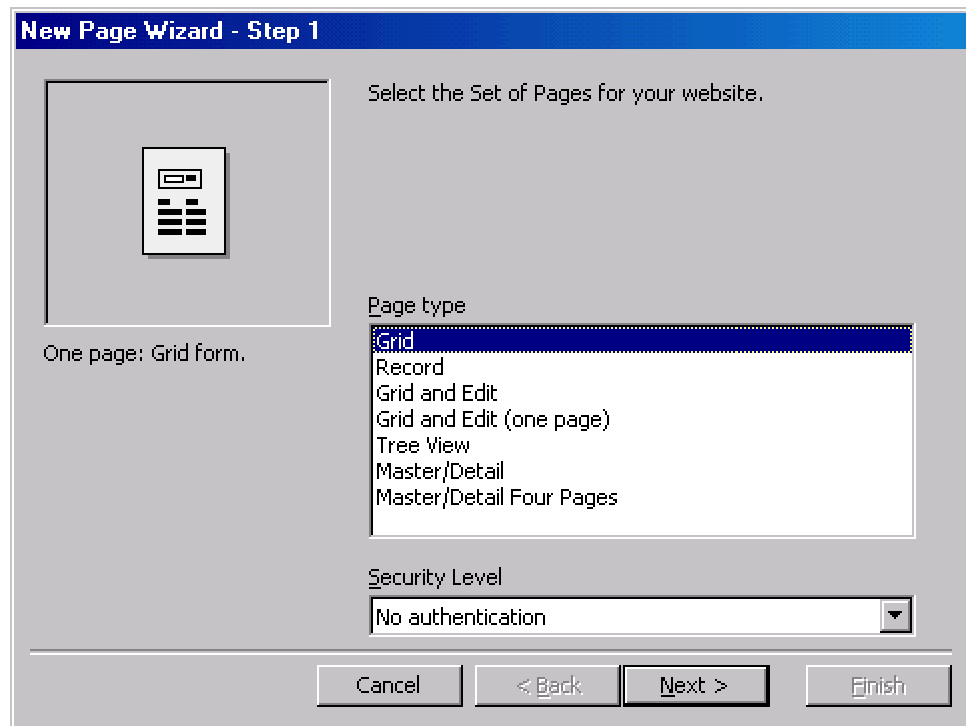
Login

This Page will be displayed automatically to users trying to access web pages that require authentication.

VI. Using the New Page Wizard

You can use the New Page Wizard to design pages for your Web site.

To start the New Page Wizard, click on the  icon on the Toolbar. CodeCharge in turn displays the following dialog:



Step 1 includes choosing the appropriate Page Type for your site. The options include:

- ✓ Grid
- ✓ Record
- ✓ Grid and Edit
- ✓ Grid and Edit (one page)
- ✓ Tree View
- ✓ Master/Detail
- ✓ Master/Detail Four Pages

You can also choose the appropriate security level from the following options:

- ✓ No authentication
- ✓ 1 – Low
- ✓ 2 – Medium
- ✓ 3 – High

The following section shows how to create the various pages types using the wizard.

1. Grid Page Wizard

To create a Grid page using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Grid** for **Page Type** and specify an appropriate **Security Level** then click on the **Next** button

New Page Wizard - Step 1

Select the Set of Pages for your website.



One page: Grid form.

Page type

- Grid
- Record
- Grid and Edit
- Grid and Edit (one page)
- Tree View
- Master/Detail
- Master/Detail Four Pages

Security Level

No authentication

Cancel < Back Next > Finish

2. Select the table and fields to be displayed in the form.

New Page Wizard - Step 2

Select parameters for the Grid form.



Record Source Refresh

ads

Primary Key ad_id

Available Fields

- date_posted
- member_id
- ad_id

Fields for Grid form

- name
- category_id
- ad_description
- price

Order by name

Cancel < Back Next > Finish

Note: Use the   **Right** and   **Left** arrows to choose the fields to display in the **Grid**, and the  **Up** and  **Down** arrows to reorder the fields in the **Form**. Use the  **Refresh** button to update the database tables and fields if you make changes to the database structure.

3. Click Next and optionally add search functionality.

New Page Wizard - Step 3

Select search fields for Grid form.

☒ Use search

Available Fields

- ad_id
- category_id
- date_posted
- member_id

Selected fields

- name
- ad_description
- price

Search type

☒ And

☐ Or

Cancel < Back Next > Finish

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.
- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

4. Click Next and specify additional functionality that should be included in the form.

New Page Wizard - Step 4

Select Grid form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

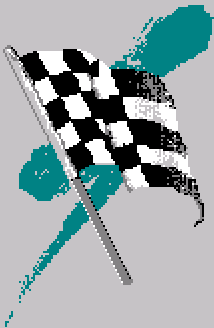
Records per Page

20

Cancel < Back Next > Finish

5. Click Next and review the configuration details.

New Page Wizard - Finished!



The New Page Wizard is finished collecting the information.

Press Finish to build your Pages.

Will be created:

Page: adsGrid

Form: Search

Form: ads

Cancel < Back Next > Finish

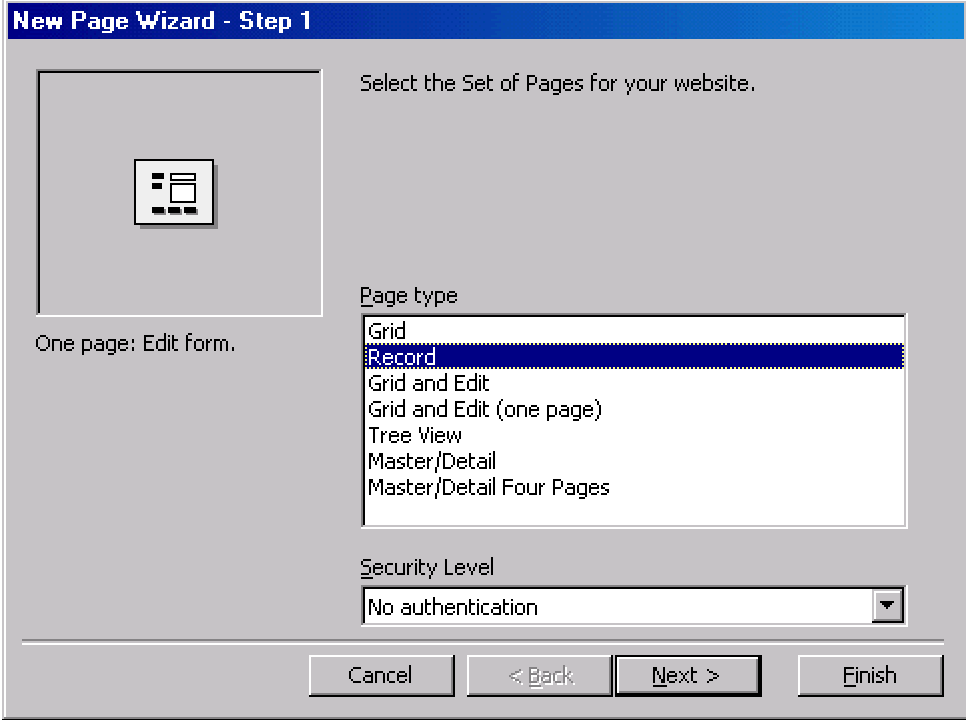
Click on **'Finish'** to generate the new page.

See also [Pages](#).

2. Record Page Wizard

To create a Record page using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Record** in the **Page Type** window and set the **Security Level**.



New Page Wizard - Step 1

Select the Set of Pages for your website.

One page: Edit form.

Page type

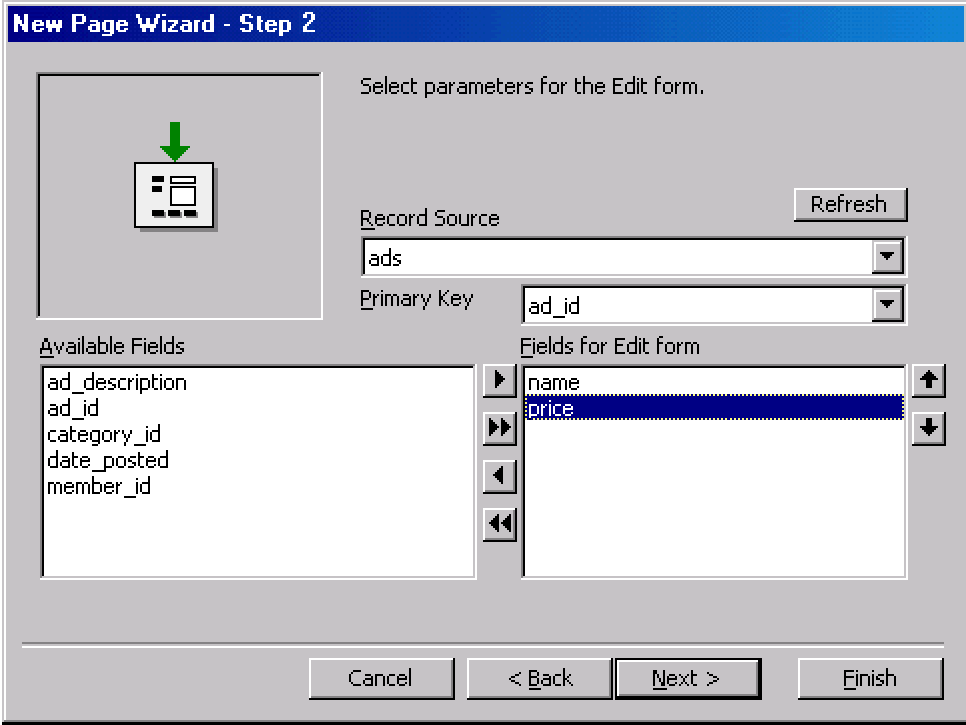
- Grid
- Record**
- Grid and Edit
- Grid and Edit (one page)
- Tree View
- Master/Detail
- Master/Detail Four Pages

Security Level

No authentication

Cancel < Back Next > Finish

2. Select the table and fields to be used in the form.



New Page Wizard - Step 2

Select parameters for the Edit form.

Record Source Refresh

ads

Primary Key ad_id

Available Fields

- ad_description
- ad_id
- category_id
- date_posted
- member_id

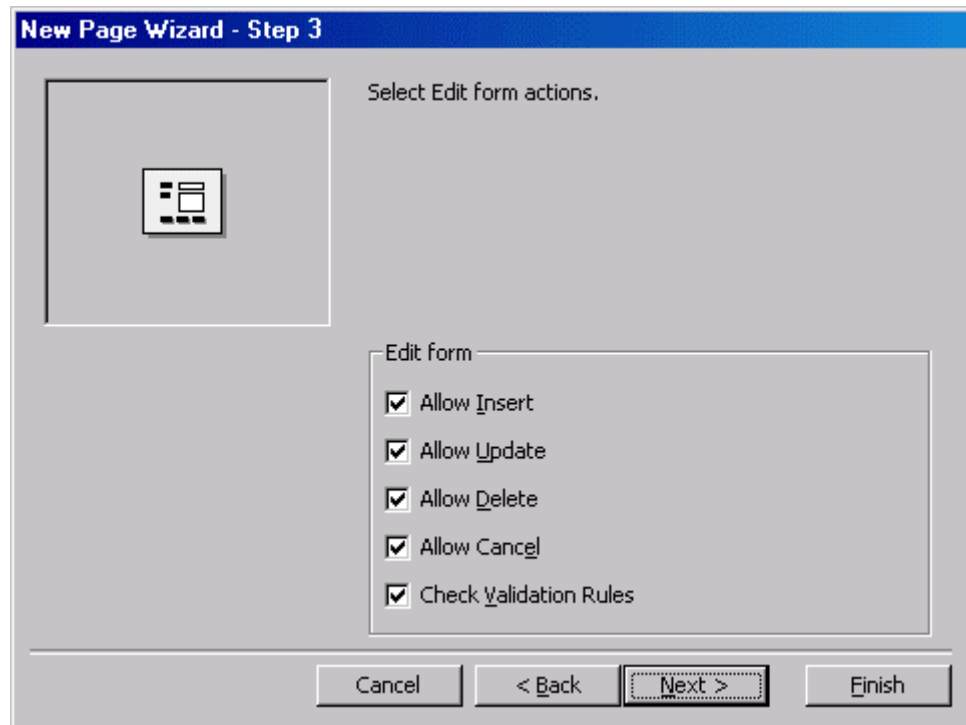
Fields for Edit form

- name
- price**

Cancel < Back Next > Finish

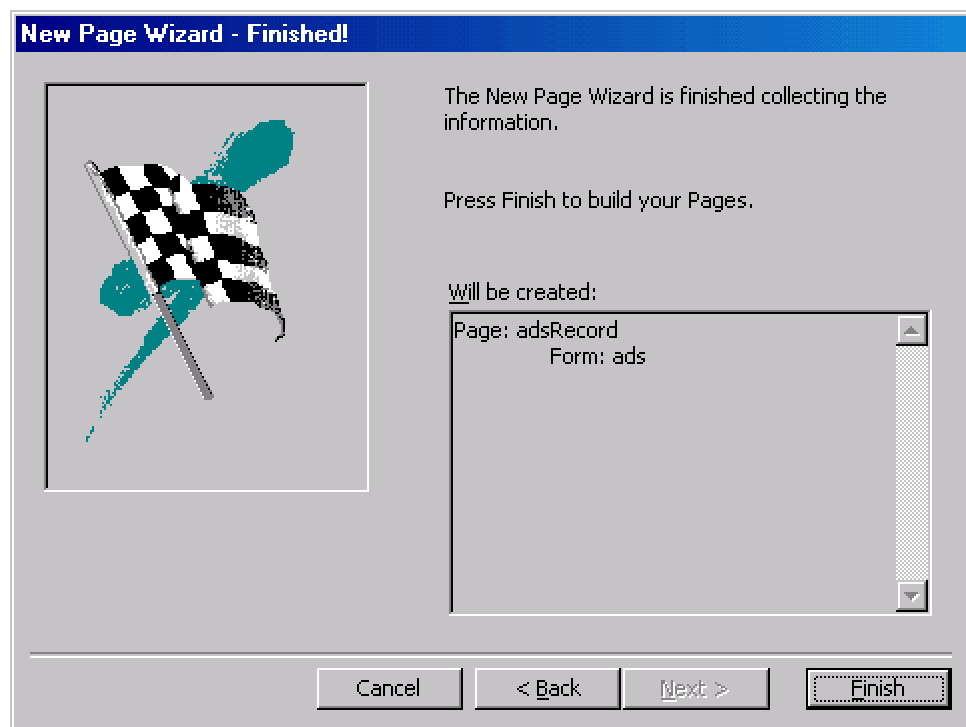
Note: Use the  **Right** and  **Left** arrows to choose the fields to display in the **Grid**, and the  **Up** and  **Down** arrows to reorder the fields in the **Form**. Use the  **Refresh** button to update the database tables and fields if you make changes to the database structure.

3. Select the operations that you want the user to be able to perform in the Edit form.



The dialog box is titled "New Page Wizard - Step 3". It contains a section titled "Select Edit form actions." with a list of checkboxes under the heading "Edit form". The checkboxes are: "Allow Insert", "Allow Update", "Allow Delete", "Allow Cancel", and "Check Validation Rules". All checkboxes are checked. At the bottom, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

4. Review the configuration details then click **Finish** to generate the page.



The dialog box is titled "New Page Wizard - Finished!". It contains a section with a checkered flag icon and the text "The New Page Wizard is finished collecting the information." and "Press Finish to build your Pages." Below this, there is a section titled "Will be created:" with a list box containing "Page: adsRecord" and "Form: ads". At the bottom, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

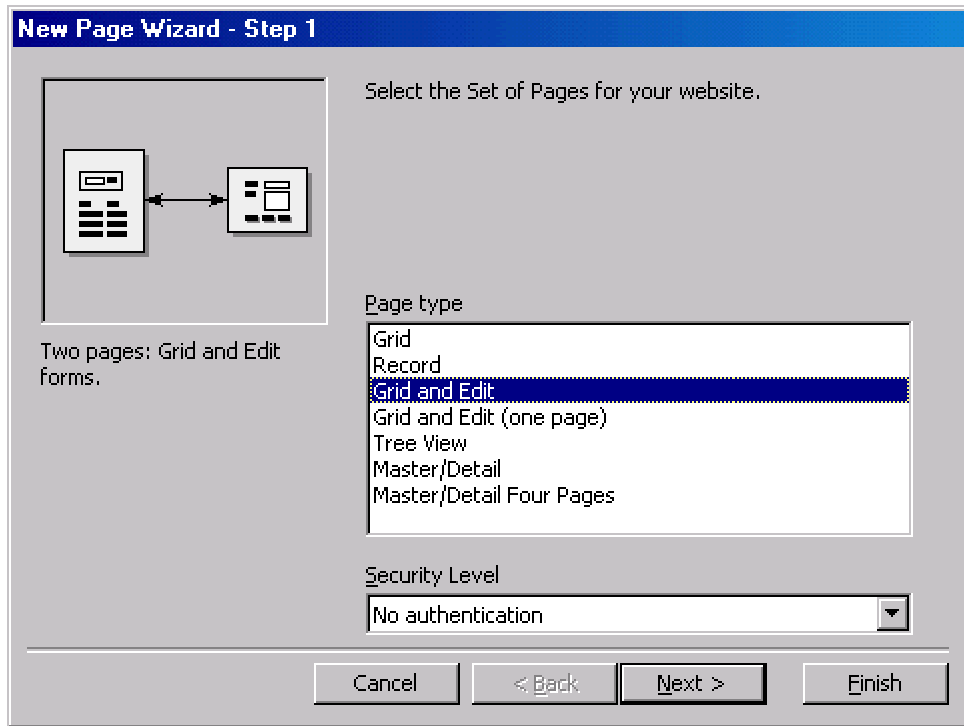
See also [Pages](#).

3. Grid and Edit Page Wizard

The Grid and Edit page wizard creates two pages that each contain a form. The first page contains a Grid form that has a list of records. The second page contains a record type form that is used to display the details of any of the records selected in the Grid form. The record form can also be used to add, edit or delete records.

To create Grid and Edit pages using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Grid and Edit** in the **Page Type** window and set the **Security Level**.

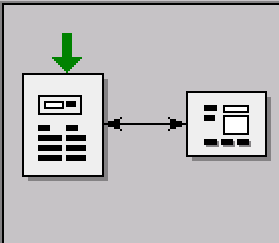


The image shows a dialog box titled "New Page Wizard - Step 1". It contains a diagram on the left showing two pages connected by a double-headed arrow, with the text "Two pages: Grid and Edit forms." below it. On the right, there is a text label "Select the Set of Pages for your website." followed by a "Page type" list box. The list box contains the following options: "Grid", "Record", "Grid and Edit" (which is highlighted with a blue selection bar), "Grid and Edit (one page)", "Tree View", "Master/Detail", and "Master/Detail Four Pages". Below the list box is a "Security Level" label and a dropdown menu currently showing "No authentication". At the bottom of the dialog are four buttons: "Cancel", "< Back", "Next >", and "Finish".

1. Select the table and fields to be used in the Grid form.

New Page Wizard - Step 2

Select parameters for the Grid form.



Record Source Refresh

ads

Primary Key ad_id

Available Fields

- ad_description
- category_id
- date_posted
- member_id
- price

Fields for Grid form

- ad_id
- name

Order by (none)

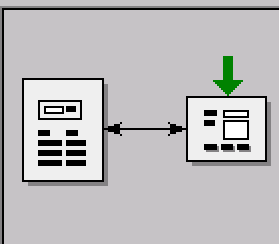
Cancel < Back Next > Finish

Note: Use the   **Right** and   **Left** arrows to choose the fields to display in the **Grid**, and the  **Up** and  **Down** arrows to reorder the fields in the **Form**. Use the Refresh **Refresh** button to update the database tables and fields if you make changes to the database structure.

2. Select the fields to be used in the Edit form

New Page Wizard - Step 3

Select parameters for the Edit form.



Available Fields

- ad_description
- ad_id
- date_posted
- member_id
- name
- price

Fields for Edit form

- category_id

Order by

Cancel < Back Next > Finish

3. Optionally add a Search form and select the fields to be used in the search.

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

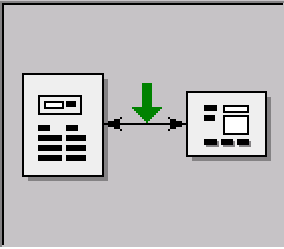
You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.
- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

4. Select the field in the Grid form that will link to the Record page.

New Page Wizard - Step 5

Select the field, which will link to the Edit form.



Select the URL field

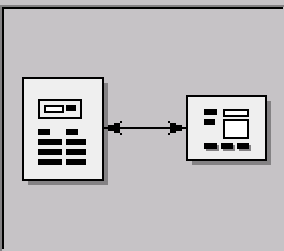
name

Cancel < Back Next > Finish

- Specify the operations that you would like to be available in both the Grid form as well as the Edit form.

New Page Wizard - Step 6

Select Grid and Edit form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page

20

Edit form

☒ Allow Insert

☒ Allow Update

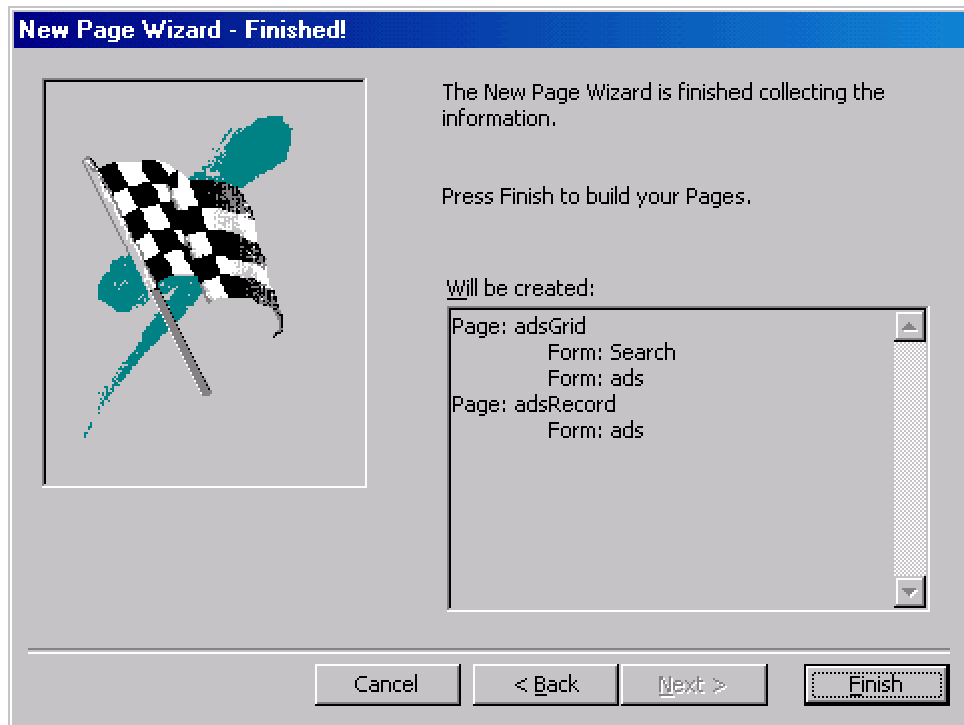
☒ Allow Delete

☒ Allow Cancel

☒ Check Validation Rules

Cancel < Back Next > Finish

- Finally, review the configuration details and click **Finish** to generate the page.



See also [Pages](#).

4. Grid and Edit (one page) Page Wizard

The Grid and Edit (one page) option is very similar to the Grid and Edit page option discussed above with the exception that the Grid and Edit (one page) creates a single page that contains both the Grid form and the Edit/Record form.

To create a Grid and Edit(one page) page using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Grid and Edit (one page)** in the **Page Type** window and set the **Security Level**.

New Page Wizard - Step 1

Select the Set of Pages for your website.



One page: Grid and Edit forms.

Page type

- Grid
- Record
- Grid and Edit
- Grid and Edit (one page)**
- Tree View
- Master/Detail
- Master/Detail Four Pages

Security Level

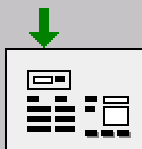
No authentication

Cancel < Back Next > Finish

2. Select the table and fields to be used in the Grid form.

New Page Wizard - Step 2

Select parameters for the Grid form.



Record Source Refresh

ads

Primary Key ad_id

Available Fields

- ad_description
- ad_id**
- category_id
- date_posted
- member_id

Fields for Grid form

- name
- price**

Order by (none)

Cancel < Back Next > Finish

Note: Use the   **Right** and   **Left** arrows to choose the fields to display in the **Grid**, and the  **Up** and  **Down** arrows to reorder the fields in the **Form**. Use the  **Refresh** button to update the database tables and fields if you make changes to the database structure.

3. Select the fields to be included in the Edit form.

The screenshot shows the 'New Page Wizard - Step 3' dialog box. The title bar is blue with the text 'New Page Wizard - Step 3'. The main area has a light gray background. On the left, there is a preview icon of a form with a green arrow pointing to it. To the right of the icon, the text 'Select parameters for the Edit form.' is displayed. Below this, there are two list boxes. The first list box, labeled 'Available Fields', contains 'category_id' and 'date_posted'. The second list box, labeled 'Fields for Edit form', contains 'ad_description', 'ad_id', 'member_id', 'name', and 'price'. Between the two list boxes are four arrow buttons: a single right arrow, a double right arrow, a single left arrow, and a double left arrow. At the bottom of the dialog, there are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

4. Optionally add a Search form and select the fields to be used in the search.

The screenshot shows the 'New Page Wizard - Step 4' dialog box. The title bar is blue with the text 'New Page Wizard - Step 4'. The main area has a light gray background. On the left, there is a preview icon of a form with a green arrow pointing to it. To the right of the icon, the text 'Select search fields for Grid form.' is displayed. Below this, there is a checkbox labeled 'Use search' which is checked. To the left of the checkbox, there is a 'Search type' section with two radio buttons: 'And' (selected) and 'Or'. Below the checkbox, there are two list boxes. The first list box, labeled 'Available Fields', contains 'ad_description', 'ad_id', 'category_id', and 'date_posted'. The second list box, labeled 'Selected fields', contains 'member_id', 'name', and 'price'. Between the two list boxes are four arrow buttons: a single right arrow, a double right arrow, a single left arrow, and a double left arrow. At the bottom of the dialog, there are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.
- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

5. Select the field in the **Grid Form** that will link to the **Edit Form**.

New Page Wizard - Step 5

Select the field, which will link to the Edit form.

Select the URL field

name

Cancel < Back Next > Finish

7. Specify the operations that you would like to be available in both the Grid form as well as the Edit form.

New Page Wizard - Step 6

Select Grid and Edit form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page:

Edit form

☒ Allow Insert

☒ Allow Update

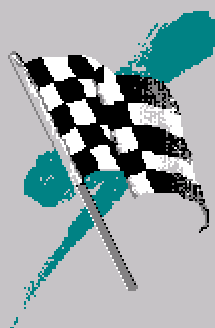
☒ Allow Delete

☒ Allow Cancel

☒ Check Validation Rules

6. Review the configuration details and click **Finish** to generate the page.

New Page Wizard - Finished!



The New Page Wizard is finished collecting the information.

Press Finish to build your Pages.

Will be created:

Page: adsRecord

Form: Search

Form: ads

Form: ads1

See also [Pages](#).

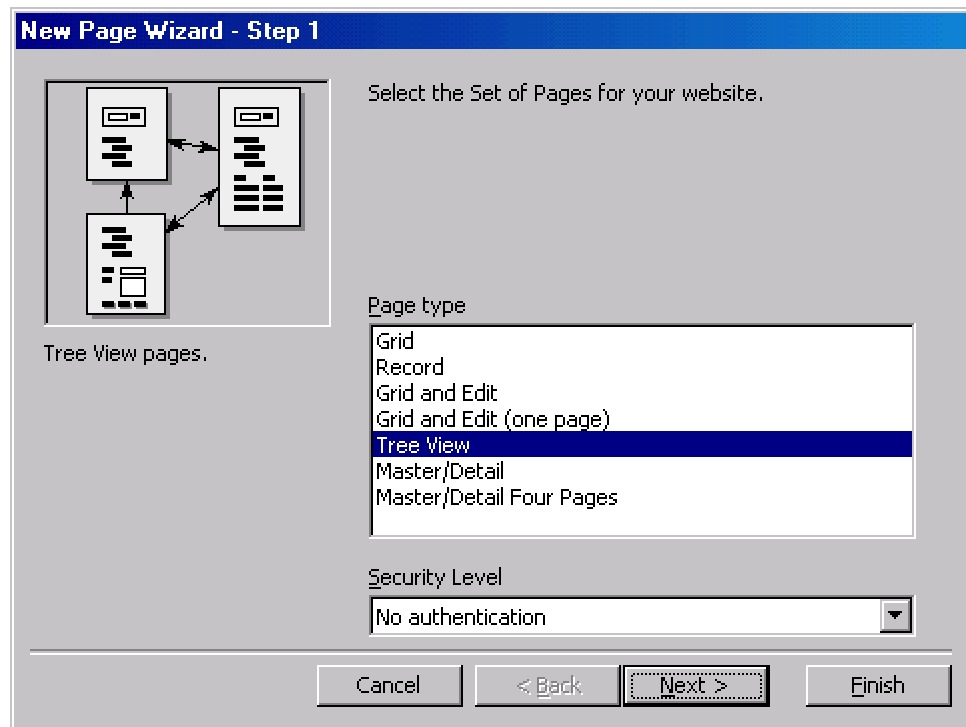
5. Tree View Page Wizard

The Tree View wizard creates three pages that are geared towards displaying and editing hierarchically structured data. Data of this kind is usually composed of categories whereby there is a root or parent category

that has one or more sub-categories. The sub-categories could also have child categories of their own. Apart from displaying the hierarchical structure, a grid form is also included such that as you traverse the hierarchy, other records that pertain to the selected category are displayed in the Grid.

To create Tree View pages using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

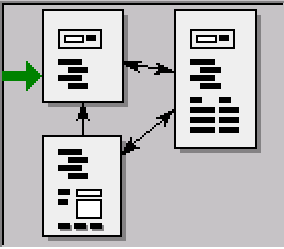
1. Select **Tree View** in the **Page Type** window and set the **Security Level**.



2. Select the table and fields to be used in the form. Note that there are three particular fields that are required
 - Category ID: Identifies each of the categories
 - Category Name: Name associated with each of the categories
 - Parent Category ID: For a child category, identifies the immediate parent category

New Page Wizard - Step 2

Select parameters for the Tree View form.



Record Source Refresh
 ads

Category ID Field
 ad_id

Category Name Field
 name

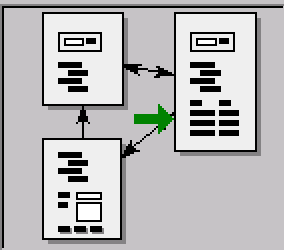
Parent Category ID Field
 category_id

Cancel < Back Next > Finish

3. Select the table and fields to be used in the Grid form.

New Page Wizard - Step 3

Select parameters for the Grid form.



Record Source Refresh
 categories

Primary Key
 category_id

Available Fields

Fields for Grid form

category_id
 name
 par_category_id

Order by (none)

Cancel < Back Next > Finish

4. Select the table and fields to be used in the Edit form. The Edit form is used to edit the details of the categories.

New Page Wizard - Step 4

Select parameters for the Edit form.

Available Fields

Fields for Edit form

Order by

Cancel < Back Next > Finish

Note: Use the **Right** and **Left arrows** to choose the fields to display in the **Grid**, and the **Up** and **Down** arrows to reorder the fields in the **Form**. Use the **Refresh** button to update the database tables and fields if you make changes to the database structure.

- Optionally add a Search form and select the fields to be used in the search.

New Page Wizard - Step 5

Select search fields for Grid form.

☒ Use search

Search type

☒ And

☐ Or

Available Fields

Selected fields

Cancel < Back Next > Finish

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.
- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

6. Select the field in the **Grid Form** that will link to the **Edit Form**:

New Page Wizard - Step 6

Select the field, which will link to the Edit form.

Select the URL field

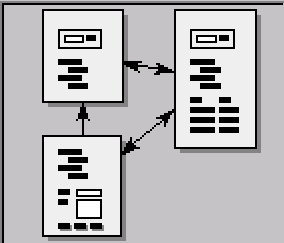
name

Cancel < Back Next > Finish

7. Specify the operations that you would like to be available in both the Grid form as well as the Edit form.

New Page Wizard - Step 7

Select Grid and Edit form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page:

Edit form

☒ Allow Insert

☒ Allow Update

☒ Allow Delete

☒ Allow Cancel

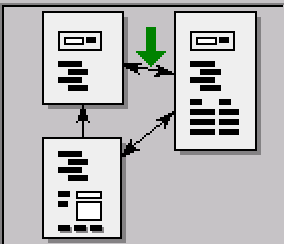
☒ Check Validation Rules

- Select the fields that define the relationship between the categories. This is a one-to-many relationship whereby a category could have many records associated with it.

New Page Wizard - Step 8

Master/Detail link.

Select a field from each record source that links two sources together.



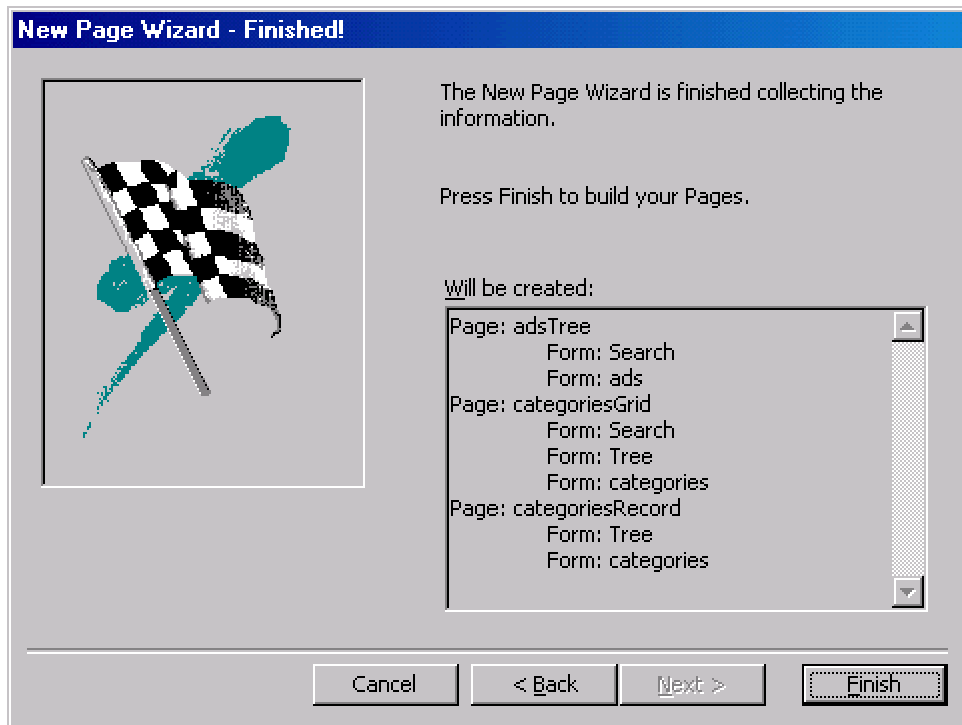
Master fields

- ad_description
- ad_id
- category_id**
- date_posted
- member_id
- name
- price

Detail fields

- category_id**
- name
- par_category_id

- Review the configuration details and click **Finish** to generate the page.



See also [Pages](#).

6. Master/Detail Page Wizard

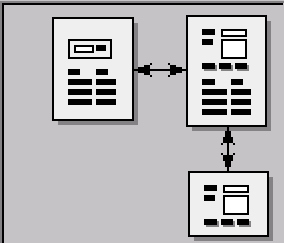
As the name suggests, the Master/Detail page wizard is used to create pages that display grouped data. To this end, there usually is a table that contains the Master categories and another that contains records that belong to one of the Master categories. As such, for each Master category, the Master/Detail pages allow the user to edit the category information as well as view the records that belong to the particular category.

To create Master/Detail pages using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Master/Detail** in the **Page Type** window and set the **Security Level**.

New Page Wizard - Step 1

Select the Set of Pages for your website.



Three pages: Master/Detail forms.

Page type

- Grid
- Record
- Grid and Edit
- Grid and Edit (one page)
- Tree View
- Master/Detail**
- Master/Detail Four Pages

Security Level

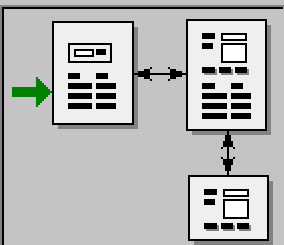
No authentication

Cancel < Back Next > Finish

2. Select the table and fields to be used in the Master Grid form

New Page Wizard - Step 2

Select parameters for the Master Grid form.



Record Source Refresh

ads

Primary Key ad_id

Available Fields

- ad_description
- ad_id
- date_posted
- member_id
- name
- price

Fields for Grid form

- category_id**

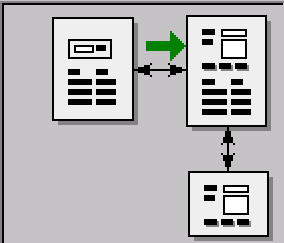
Order by (none)

Cancel < Back Next > Finish

3. Select the fields from the Master database table that should be available for editing in the Master Edit form.

New Page Wizard - Step 3

Select parameters for the Master Edit form.



Available Fields

price

Fields for Edit form

ad_description
ad_id
category_id
date_posted
member_id
name

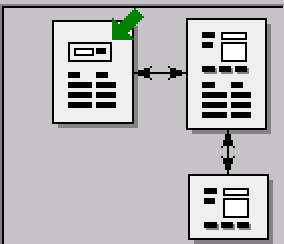
Order by

Cancel < Back Next > Finish

- Optionally add a Search form and select the fields to be used in the search. This search form will be used to search for records in the Master table.

New Page Wizard - Step 4

Select search fields for Master Grid form.



☒ Use search

Search type

☒ And
☐ Or

Available Fields

ad_id
category_id
date_posted
member_id
price

Selected fields

name
ad_description

Cancel < Back Next > Finish

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.
- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

5. Select the field in the **Master Grid Form** that will link to the **Master Edit Form**:

New Page Wizard - Step 5

Select the field, which will link to the Master Edit form.

Select the URL field

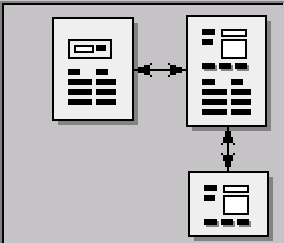
category_id

Cancel < Back Next > Finish

6. Specify the operations that you would like to be available in both the Master Grid form as well as the Master Edit form.

New Page Wizard - Step 6

Select Master Grid and Edit form actions.



Grid form

- ☒ Allow Sorting
- ☒ Use Page Scroller

Records per Page:

Edit form

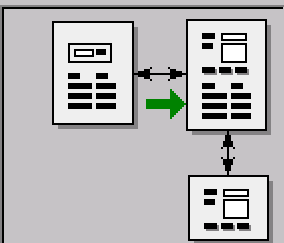
- ☒ Allow Insert
- ☒ Allow Update
- ☒ Allow Delete
- ☒ Allow Cancel
- ☒ Check Validation Rules

Cancel < Back Next > Finish

7. Moving on to the Detail forms, Select the table and fields to be used in the Detail Grid form.

New Page Wizard - Step 7

Select parameters for the Detail Grid form.



Record Source: Refresh

Primary Key:

Available Fields:

- ad_description
- category_id
- date_posted
- member_id
- price

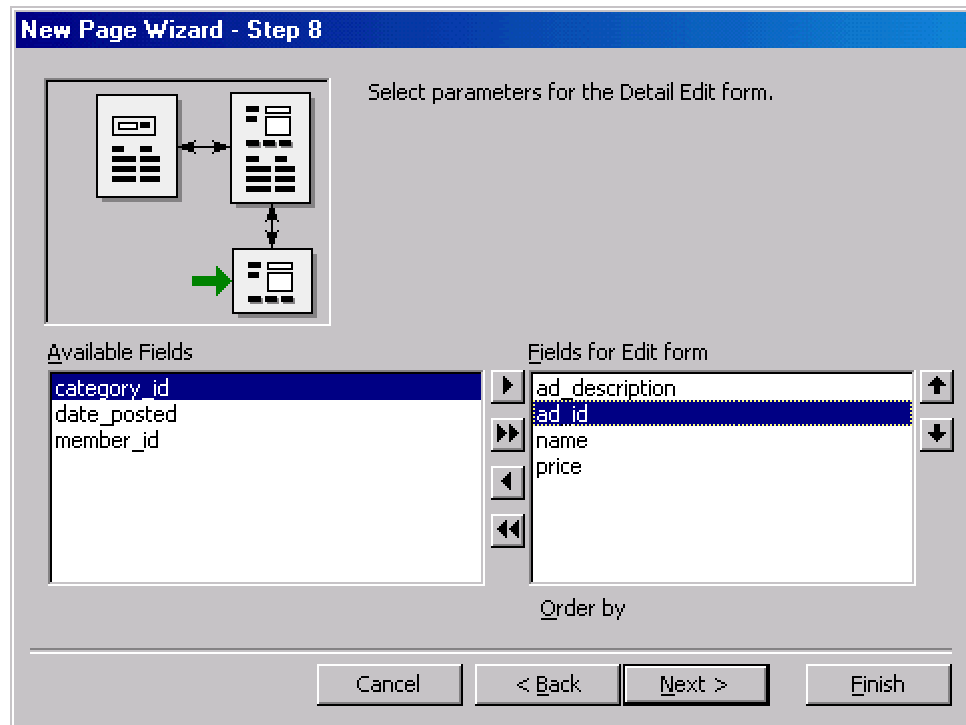
Fields for Grid form:

- ad_id
- name

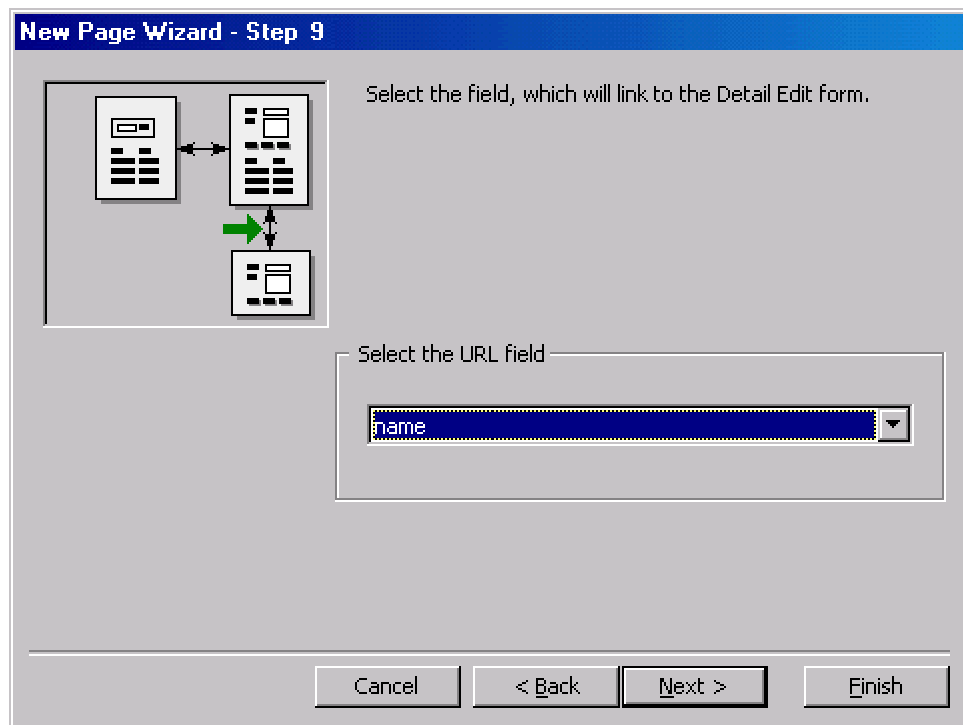
Order by:

Cancel < Back Next > Finish

8. Then select the fields to be used in the Detail Edit form.



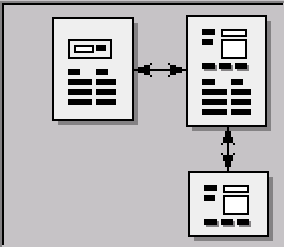
9. Select the field in the **Detail Grid Form** that will link to the **Detail Edit Form**:



10. Specify the operations that you would like to be available in both the Detail Grid form as well as the Detail Record form.

New Page Wizard - Step 10

Select Detail Grid and Edit form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page:

Edit form

☒ Allow Insert

☒ Allow Update

☒ Allow Delete

☒ Allow Cancel

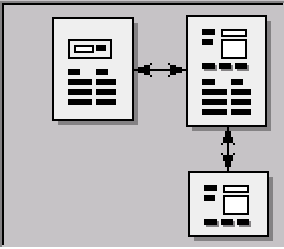
☒ Check Validation Rules

11. Specify the two fields that define the relationship between the Master records and the Detail records. This is essentially a one-to-many relationship whereby the primary field resides in the Master table and the secondary field resides in the Detail table. The field in the Detail table would link to a field in the Master table and hence determine the Master category to which the Detail record belongs to.

New Page Wizard - Step 11

Master/Detail link.

Select a field from each record source that links two sources together.



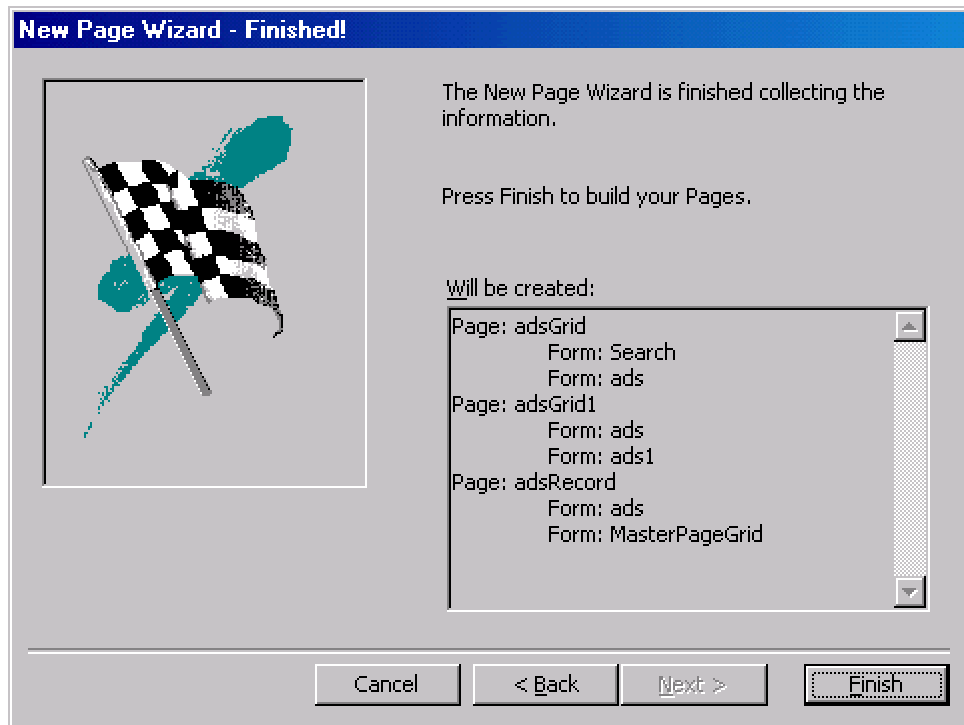
Master fields

- ad_description
- ad_id
- category_id**
- date_posted
- member_id
- name
- price

Detail fields

- ad_description
- ad_id
- category_id**
- date_posted
- member_id
- name
- price

12. Review the configuration details and click **Finish** to generate the page.



See also [Pages](#).

7. Master/Detail (four pages) Page Wizard

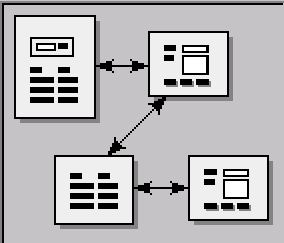
The Master/Detail (four pages) option is conceptually similar to the Master/Detail option with the exception that the Master/Detail (four pages) option produces four pages instead of three. This is because the Grid forms and the Edit forms all appear in separate pages.

To create Master/Detail (four pages) pages using the wizard, click on the  **New Page Wizard** button on the [Tool Bar](#) and follow the steps outlined below:

1. Select **Master/Detail Four Pages** in the **Page Type** window and set the **Security Level**.

New Page Wizard - Step 1

Select the Set of Pages for your website.



Four pages: Master/Detail forms.

Page type

- Grid
- Record
- Grid and Edit
- Grid and Edit (one page)
- Tree View
- Master/Detail
- Master/Detail Four Pages**

Security Level

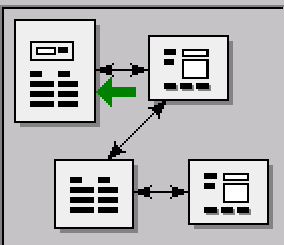
No authentication

Cancel < Back Next > Finish

2. Select the table and fields to be used in the Master Grid form.

New Page Wizard - Step 2

Select parameters for the Master Grid form.



Record Source Refresh

ads

Primary Key ad_id

Available Fields

- category_id
- date_posted**
- member_id

Fields for Grid form

- ad_description
- ad_id
- name
- price

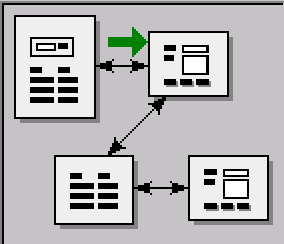
Order by (none)

Cancel < Back Next > Finish

3. Select the fields from the Master database table that should be available for editing in the Master Edit form.

New Page Wizard - Step 3

Select parameters for the Master Edit form.



Available Fields

Fields for Edit form

- ad_description
- ad_id
- category_id
- date_posted
- member_id
- name
- price

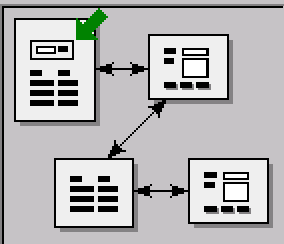
Order by

Cancel < Back Next > Finish

4. Optionally add a Search form and select the fields to be used in the search.

New Page Wizard - Step 4

Select search fields for Master Grid form.



☒ Use search

Search type

☒ And

☐ Or

Available Fields

Selected fields

- ad_description
- ad_id
- category_id
- date_posted
- member_id
- name
- price

Cancel < Back Next > Finish

Activate the **Use Search** checkbox to indicate your intention to include search functionality in the page.

You can also specify the desired **Search type**:

- Using the **And** option produces a search form with an input field for all the database fields selected for the search and the AND SQL operator is used to conduct the search.

- Using the **Or** option, only one field appears in the search form. The single field can be used to input search criteria corresponding to any of the database fields specified. For instance, you could have a single field from which you could conduct a search by Name, Author and ISBN Number. In this case, the OR SQL operator is used.

5. Select the field in the **Master Grid Form** that will link to the **Master Edit Form**:

New Page Wizard - Step 5

Select the field, which will link to the Master Edit form.

Select the URL field

name

Cancel < Back Next > Finish

6. Specify the operations that you would like to be available in both the Master Grid form as well as the Master Edit form.

New Page Wizard - Step 6

Select Master Grid and Edit form actions.

Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page:

Edit form

☒ Allow Insert

☒ Allow Update

☒ Allow Delete

☒ Allow Cancel

☒ Check Validation Rules

7. Moving on to the Detail forms, Select the table and fields to be used in the Detail Grid form.

New Page Wizard - Step 7

Select parameters for the Detail Grid form.

Record Source:

Primary Key:

Available Fields

Fields for Grid form

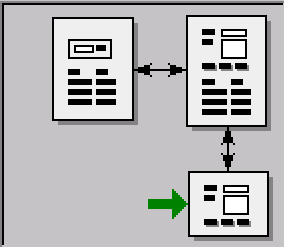
☐ ad_description
☐ ad_id
☐ category_id
☐ date_posted
☐ member_id
☐ name
☐ price

Order by:

8. Then select the fields to be used in the Detail Edit form.

New Page Wizard - Step 8

Select parameters for the Detail Edit form.



Available Fields

category_id
date_posted
member_id

Fields for Edit form

ad_description
ad_id
name
price

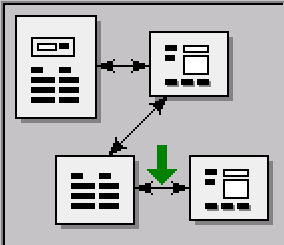
Order by

Cancel < Back Next > Finish

9. Select the field in the **Detail Grid Form** that will link to the **Detail Edit Form**:

New Page Wizard - Step 9

Select the field, which will link to the Detail Edit form.



Select the URL field

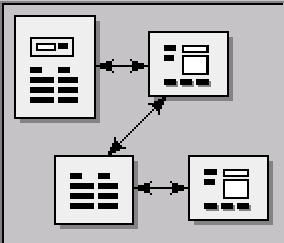
name

Cancel < Back Next > Finish

10. Specify the operations that you would like to be available in both the Detail Grid form as well as the Detail Record form.

New Page Wizard - Step 10

Select Detail Grid and Edit form actions.



Grid form

☒ Allow Sorting

☒ Use Page Scroller

Records per Page:

Edit form

☒ Allow Insert

☒ Allow Update

☒ Allow Delete

☒ Allow Cancel

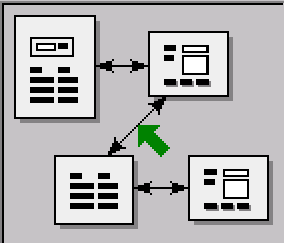
☒ Check Validation Rules

- Specify the two fields that define the relationship between the Master records and the Detail records. This is essentially a one-to-many relationship whereby the primary field resides in the Master table and the secondary field resides in the Detail table. The field in the Detail table would link to a field in the Master table and hence determine the Master category to which the Detail record belongs to.

New Page Wizard - Step 11

Master/Detail link.

Select a field from each record source that links two sources together.



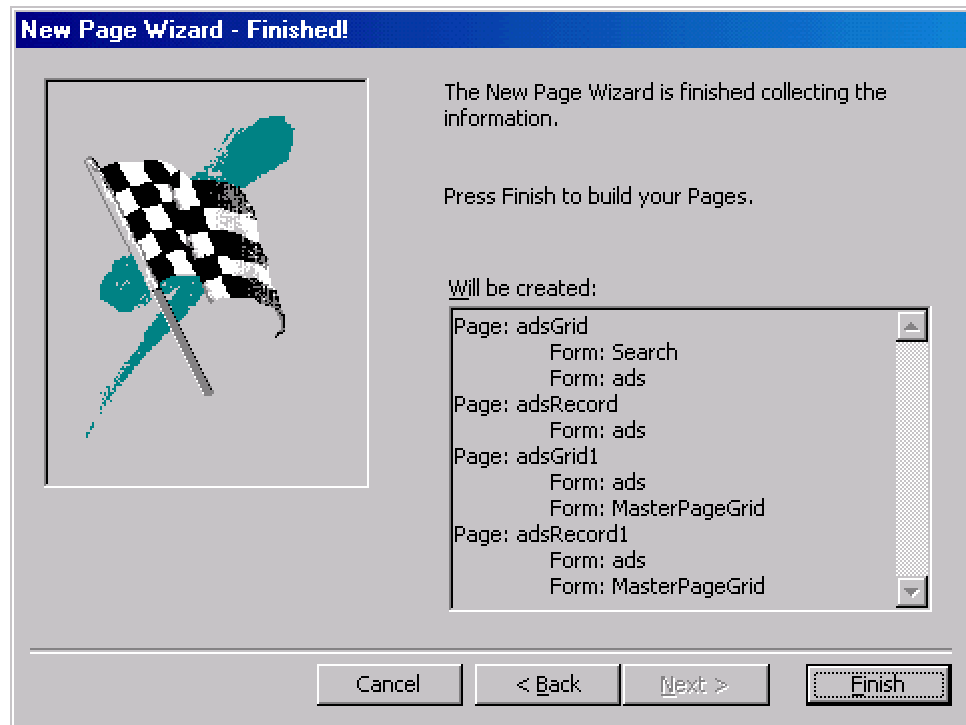
Master fields

- ad_description
- ad_id
- category_id**
- date_posted
- member_id
- name
- price

Detail fields

- ad_description
- ad_id
- category_id**
- date_posted
- member_id
- name
- price

- Review the configuration details and click **Finish** to generate the page.



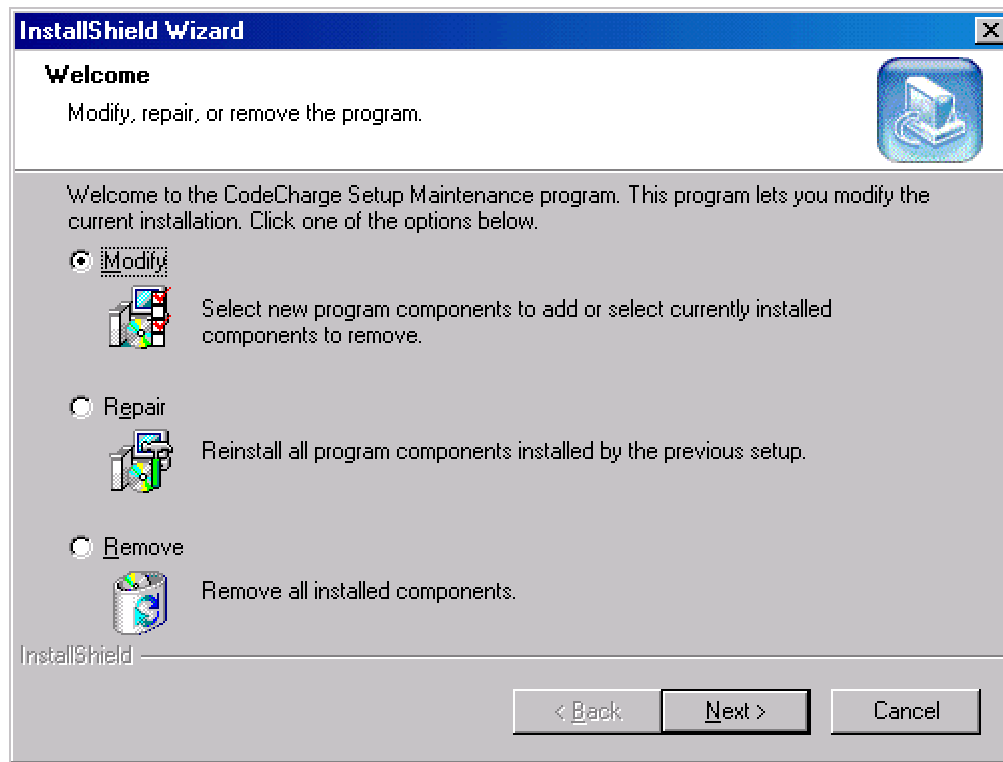
See also [Pages](#).

VII. Appendix A:

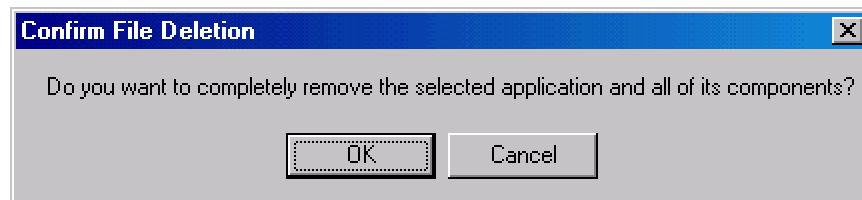
1. Uninstalling CodeCharge

To uninstall CodeCharge, use the following steps:

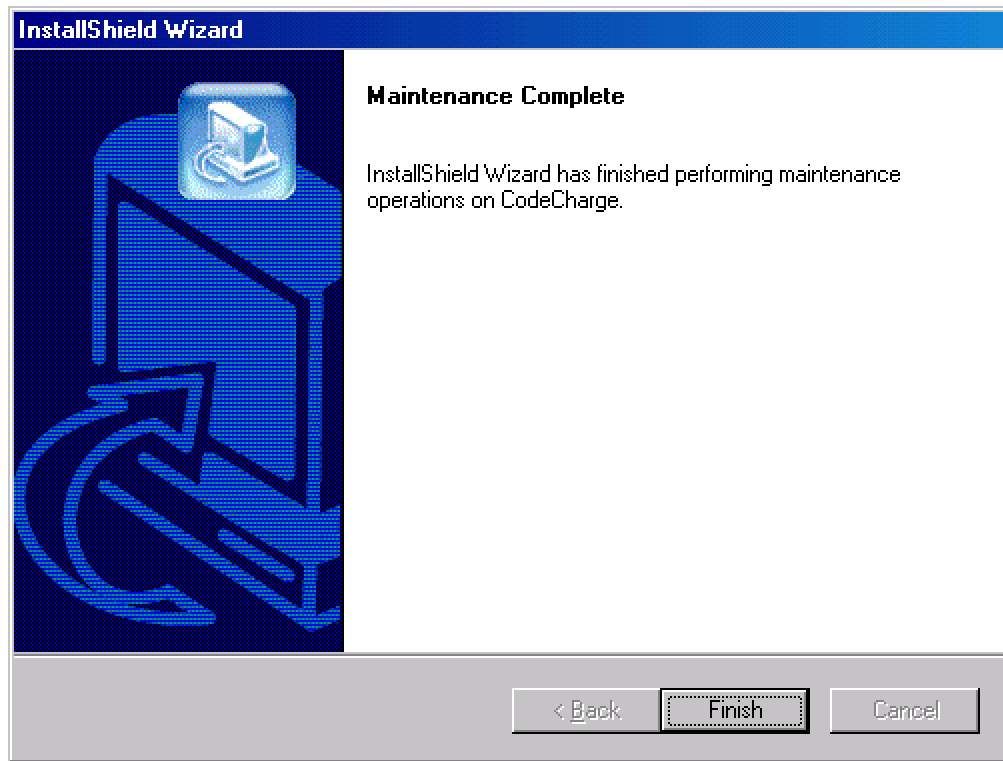
1. Open **Control Panel**
2. Double-click on **Add/Remove Programs**
3. Select **CodeCharge** from the list of software
4. Click **Add/Remove**
5. The Uninstall program displays the following dialog:



6. To remove all installed components of CodeCharge, click **Remove**, then click **Next**. The Uninstall program displays the following dialog:



Click **OK** to confirm your intent. The uninstall program will remove all CodeCharge components and display the following dialog:



To complete the Uninstall process, click **Finish**.

2. CodeCharge Extensions

Extension	Description
.ccs	CodeCharge Site
.ccp	CodeCharge Language Pattern
.cct	CodeCharge Style Sheets
.css	Cascading Style Sheets
.cci	CodeCharge information file – contains locking status for those files that are “locked”, because they were manually modified

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.