DotNetNuke™ Form and List

Using the DotNetNuke Form and List Module

Stefan Cullmann, Sebastian Leupold, Brett Conlan



Form and List v 5.1.2
Updated: February 19, 2010
Category: DotNetNuke 5.1.3+ Module

Information in this document, including URL and other Internet web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of DotNetNuke Corporation. DotNetNuke Corporation may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from DotNetNuke Corporation, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2010, DotNetNuke Corporation. All Rights Reserved.

DotNetNuke® and the DotNetNuke logo are either registered trademarks or trademarks of DotNetNuke Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Abstract

This user guide is an introduction to the DotNetNuke Form and List module.

This guide assumes that you have an operational DotNetNuke installation.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Contents

Chapter 1: About Form and List 1

Introduction 1

What is the Form and List Module? 1

Brief History 2

Form and List Resources 3

Chapter 2: Prerequisites 4

Using the DotNetNuke Application Framework 4

Module Installation 4

Chapter 3: Quick Start 5

Creating Your First List 5

Creating Your First Form 9

Chapter 4: Managing Form and List 12

Module Action Menu 12

Add New Column 14

Data Types 15

Hidden Columns 18

Chapter 5: Advanced Options 22

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Advanced Column Options 22

Token Use in Column Options 24

DotNetNuke System Tokens 25

Chapter 6: Column & Page Settings 27

Column Settings 27

Page Related Settings 29

List Settings 30

Default Grid Table Settings 32

Search Options 33

XSL using Self Made or Generated Stylesheets 34

XSL using Built-in Stylesheets 34

Chapter 7: Token Based XSLT Generator and Editor 36

What is the Token Based XSLT Generator/Editor? 36

Creating Your First XSLT Generator Style Sheet 37

Advanced XSLT 43

Chapter 8: Module Settings 45

Permissions 45

Chapter 9: Managing Records 49

View Records 49

View/Edit > Only Own Items (Records) 50

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Add a Record 51

Edit a Record 51

Delete a Record 52

Delete All Records 52

Show Data as XML 52

Additional Information 54

Errors and Omissions 55

Appendix A: Document History 56

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 1:
About Form and List

Introduction

This chapter provides a basic overview of the Form and List project. It includes a brief history of the User Defined Table module, the codebase predecessor to Form and List module. This section highlights the importance of community participation in expanding and improving the DotNetNuke (DNN) framework and its associated projects.

What is the Form and List Module?

Form and List is a DotNetNuke (DNN) content module that allows you to define a table (a group of columns or fields) that has the capability to store and display records.

In this context, a record is a set of column values where each field is named and has a specific data type definition. Take for example a list of friends, where for each friend you may store a first name, last name, date of birth, phone number, email address, home page and a photo. In this list you could input first name in column 1, last name in column 2, date of birth in column 3, etc. Only data of a predefined type is allowed in any one column. For this example, that is for first/last name columns 1 & 2 - text – and for date of birth, column 3 – date.

With Form and List, once you create a list, an input form containing all the column fields you defined is automatically created for you. After you input at least one record, a default report view is also automatically created for you. This list can be sorted by column and you can specify the number of records to display per page. In addition to this “Default Grid Table” view, the Form and List XSL rendering feature allows for even more flexible display options. Entering records, multiple users can do so simultaneously, and you can set Add, Edit and Delete permissions based on DotNetNuke security Roles.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Brief History

The Form and List module was originally named the User Defined Table (UDT). It was renamed Form and List in 2009 with the addition of many new form-specific and list enhancement features. This is a brief history of the User Defined Table module.

Initially created as a DotNetNuke core module, an independent company, Gamma Concept, then used that core module as the basis to create an enhanced version for DotNetNuke 3 known as gammacon.UserDefinedTable, popular with DNN 3 users.

In December 2004, the initial concept of utilizing and incorporating third party modules as well as spinning off the core modules into individual projects became a reality. This change brought further development into designing the core DotNetNuke framework to be modular by design. For each core module a DNN Project was created and each project is now regarded with equal importance in the development of future DNN releases.

The concept of DNN Projects further advanced the participation of more individuals in the DotNetNuke community. This allowed talented software developers to submit code to be included in the core DotNetNuke framework, while following development principles outlined by the DotNetNuke Core Team Architects.

Gamma Concept offered the enhancements of gammacon.UserDefinedTable module to the core framework. Sebastian Leupold, together with Markus Hamburger, created a new version of UDT, UserDefinedTable 3.2.0, which was an amalgamation of the features of the Gamma Concept module as well as other major enhancement such as additional data types, paging, search support and user interface (UI) improvements.

Stefan Cullmann entered the project and started a complete refactoring of the code, a necessary step to prepare for future UDT module development. Stefan also added an XSL rendering alternative that allows for more flexible display output. XSL rendering was the most significant new capability added, however, other important features were included, such as improved search support, import/export capabilities and calculated columns. All these options significantly extended the capabilities of UDT and the scenarios in which the new UDT 3.3.0 module could be used.

The User Defined Table team then drafted the first version of this documentation, to help you become familiar with UDT more quickly, so that you may get most out of the User Defined Table module. Brett Conlan then joined the UDT team in 2007 to provide comprehensive module testing, documentation and UI/Help Tooltip English editing with a focus on end-user experience continuity control.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

The enhancements for version 3.4.0 include extended field settings, record filtering and search support, and the first implementation of input validation using Regular Expressions. Also, the all new Token Based XSLT Generator and Editor significantly reduces the effort required to create XSL data rendering style sheets. 3.5.0 added data type User Profile Link and the Records Returned filter.

With the release of DotNetNuke version 5.x, a full set of form based features were added, list features were significantly enhanced, and the User Defined Table module was renamed **Form and List** module.

Form and List Resources

The Form and List is installed as part of the core installation of the DotNetNuke (DNN) framework. All resources such as the source code version, installable module, this document, and anything else associated with the official project are always attainable from the projects home page located at:

http://www.dotnetnuke.com/Development/Forge/ModuleFormandList/tabid/877/Default.aspx

From this page, you can also find other DotNetNuke projects listed in the lower left column. Resources that can assist you in using Form and List module include:

- **Form and List Downloads** where you can download the current version of the module and additions like templates, migration support and manuals http://www.dotnetnuke.com/tabid/125/default.aspx
- **Form and List Issue Tracker** where you have an overview about issues existing/solved and enhancements added http://dnnfnl.codeplex.com/
- **Form and List Forums** where you can post questions and suggestions about the usage and further development of this module http://www.dotnetnuke.com/Community/Forums/tabid/795/forumid/56/scope/threads/Default.aspx
- **Form and List Blog** where you can get the latest news about this project http://www.formandlist.com/Blogs.aspx

Use the DotNetNuke.com website as a resource to help accelerate learning, answer questions and to receive updates and other information about Form and List.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 2: Prerequisites

Using the DotNetNuke Application Framework

DotNetNuke (DNN) is a very powerful application framework for creating web sites using Microsoft ASP.Net technology. It is a modular system that allows for the easy extension of functions using modules, exchange base technologies, a provider model and user interface design utilizing skinning techniques. For further details, please refer to the DotNetNuke Core Framework documentation available from the DotNetNuke.com Download page.

DotNetNuke modules extend the functionality of the framework while utilizing framework services and settings specific to the DotNetNuke application framework. DotNetNuke modules cannot be used outside the DotNetNuke framework, and in many cases, DotNetNuke modules will require a specific version of the DNN framework.

Module Installation

Install the Form and List module as you would any other DotNetNuke module. Because there are multiple installation options that may vary between the framework versions, please refer to the DotNetNuke core manual for the version of DNN you are using.

Please Note	Version 5.1.2 of the Form and List module requires DNN5 framework version 5.1.3 or later.

After the Form and List module package is installed, the Form and List module can be used on any portal website page, adding as many Form and List instances as you require.

This user guide assumes that you have already installed DotNetNuke, configured your portal, installed the Form and List module and you are familiar with the process of creating pages. You are now ready to put the power of the DotNetNuke Form and List module to work for you – so let's go!

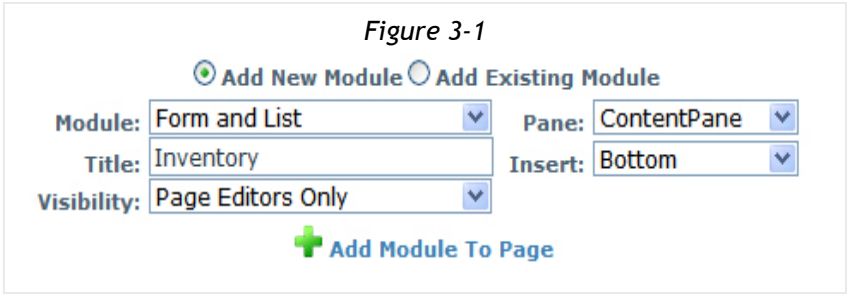
Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 3: Quick Start

Creating Your First List

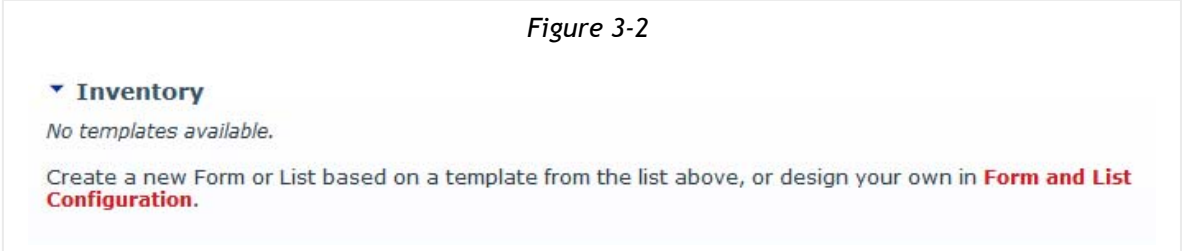
When working with the Form and List in a publicly viewable web site, we recommend that you add your first Form and List module to a new portal page that only Administrators can view and edit. By creating a new page viewable by Administrators only, your information will be hidden from public view during the time you are working to configure the Form and List module. Professional website designers and authors work in this manner to ensure that their site does not accidentally display incomplete or unapproved information. Once you have selected the module from the dropdown list, select Page Editors Only from the Visibility control, then click **Add Module to Page** to place the Form and List module to your page *(Figure 3-1)*. For this example, we will create a Form and List inventory of items that may be stored in a home, so we designate the Title 'Inventory' for this Form and List instance.

Figure 3-1



After the module has been placed on the page, click the **Form and List Configuration** link to begin defining columns for this Form and List instance. (*Figure 3-2*).

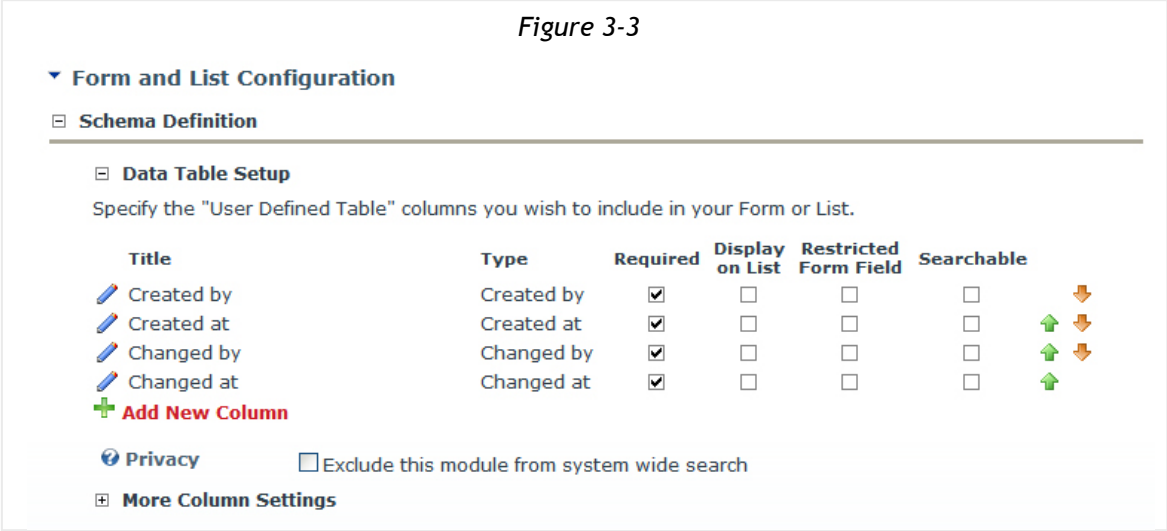
Figure 3-2



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

When you click **Form and List Configuration**, or select the Form and List Configuration command from the action menu, the Form and List Configuration, Schema Definition should display (*Figure 3-3*).

Figure 3-3



Four system columns are automatically created for every List and Form instance: Created by, Created at, Changed by and Changed at. All four columns are required by Form and List for tracking records. These columns can be renamed but they cannot be deleted. By default, these columns do not display to users unless 'Display on List' is enabled.

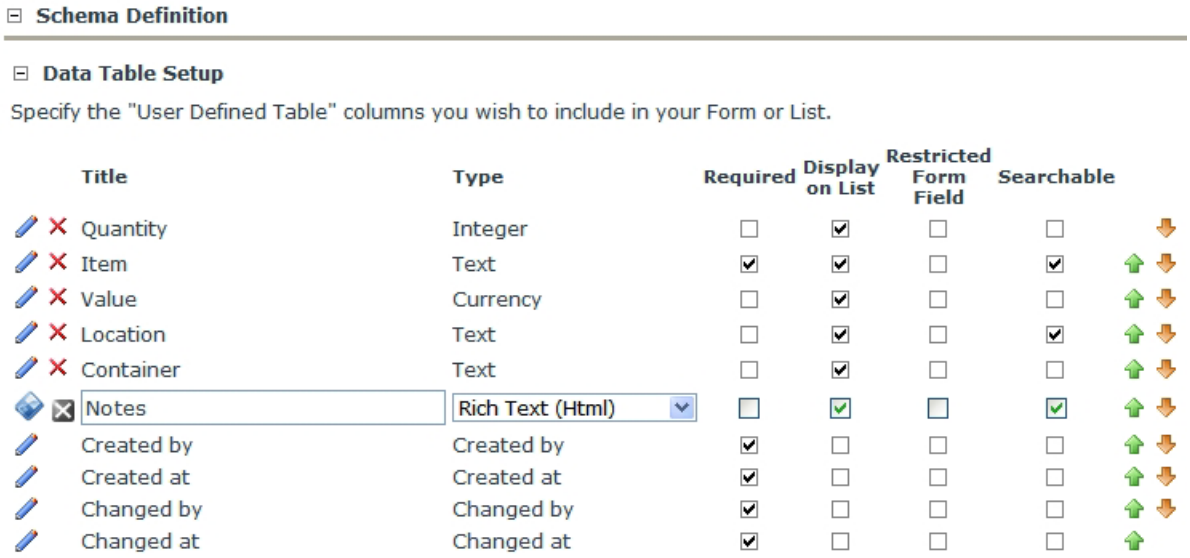
Listed just below the columns, the **Privacy** setting is enabled by default. If you would like to keep the data stored in this Form and List instance from appearing to users searching with DotNetNuke portal search function, keep this setting enabled. When Form and List is used in form mode to collect info from users that is not intended for display to any other users, enabling Privacy keeps the data secure.

For our example, we will create a list of personal items we have around the house, using the following data elements: Quantity, Item, Value, Location, Container and Notes.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

You can add a column by clicking **Add New Column**. (*Figure 3-4)*. After adding the columns for our inventory example, the Data Table Setup looks like this:

Figure 3-4



Note that we have assigned column Titles that indicate what data that will be stored in the columns. In this example, our custom column Titles are only single words, but a Title may contain two or more words with spaces, for example, 'First Name'.

When creating a column, you have the option to choose the Data Type, and set other properties such as 'Required', 'Display on List', 'Restricted Form Field', and 'Searchable', simply by enabling or disabling the property checkbox.

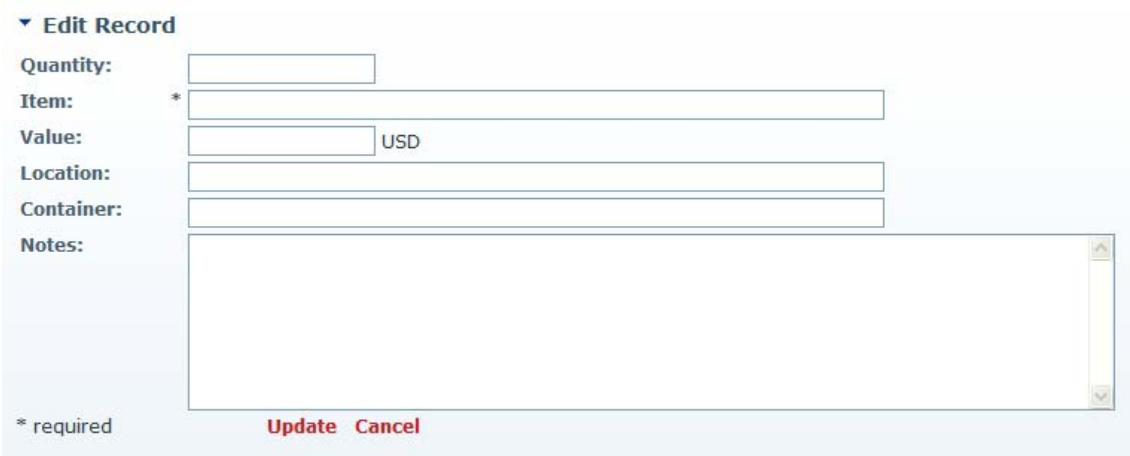
Column Property	State: Enabled
Required	When entering a new record, this column must contain a value or the record cannot be stored.
Display on List	Show this column when displaying a list of records.
Restricted Form Field	Special permissions set with module Settings will control which User or Roles have access to this column.
Searchable	This column will be searched for data by the Form and List search function.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

When you are finished defining your columns, click **Save Configuration and Return** to exit Form and List Configuration and return to the display view.

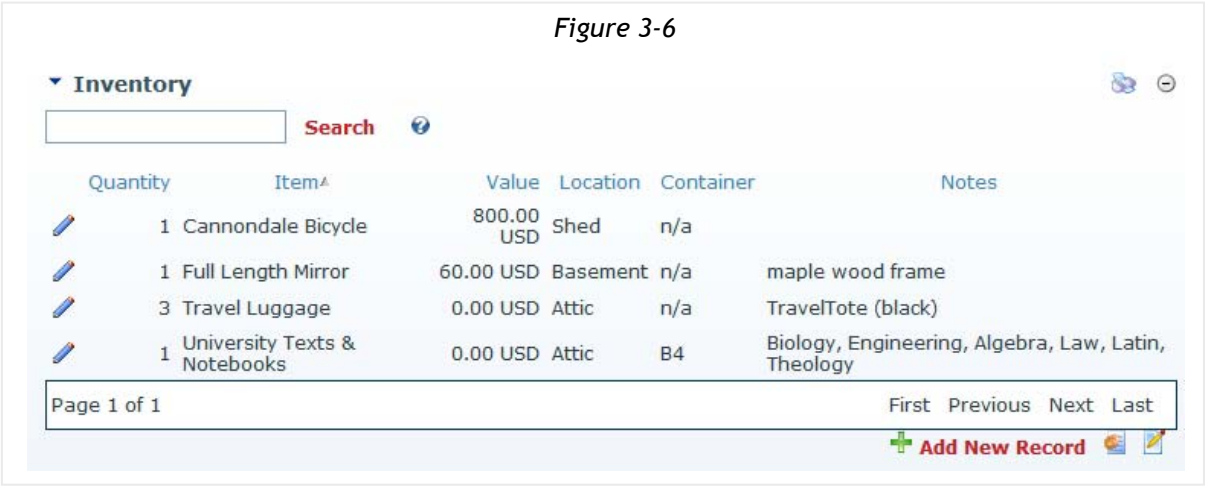
You can now add a record to your table by clicking **Add New Record**, and a blank entry form will display *(Figure 3-5)*.

Figure 3-5



We enter several items into our Form and List inventory data collection. After entering four records, our list looks like this *(Figure 3-6)*:

Figure 3-6



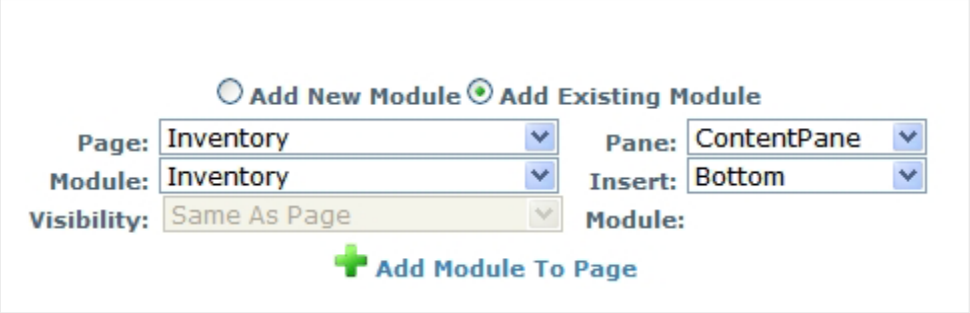
To sort the records by any column, simply click the associated column header. To edit a record, click the edit pencil located in the left column.

Congratulations! You have created your first list using Form and List. More detailed options and functionality are explained in the following chapters.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Creating Your First Form

In this example, we will use the existing inventory Form and List instance to create a Form view. We want to display this form on a different page than the inventory list, so we first create a new page. After creating the page, we choose **Add Existing Module** then click **Add Module to Page** to place Form and List on this page *(Figure 3-7)*.



The Form and List module from the Inventory page will display in List mode. (*Figure 3-8)*. Mouse over the down arrow next to the module Title 'Inventory' to activate the action menu, and choose **Form and List Configuration** from the menu.

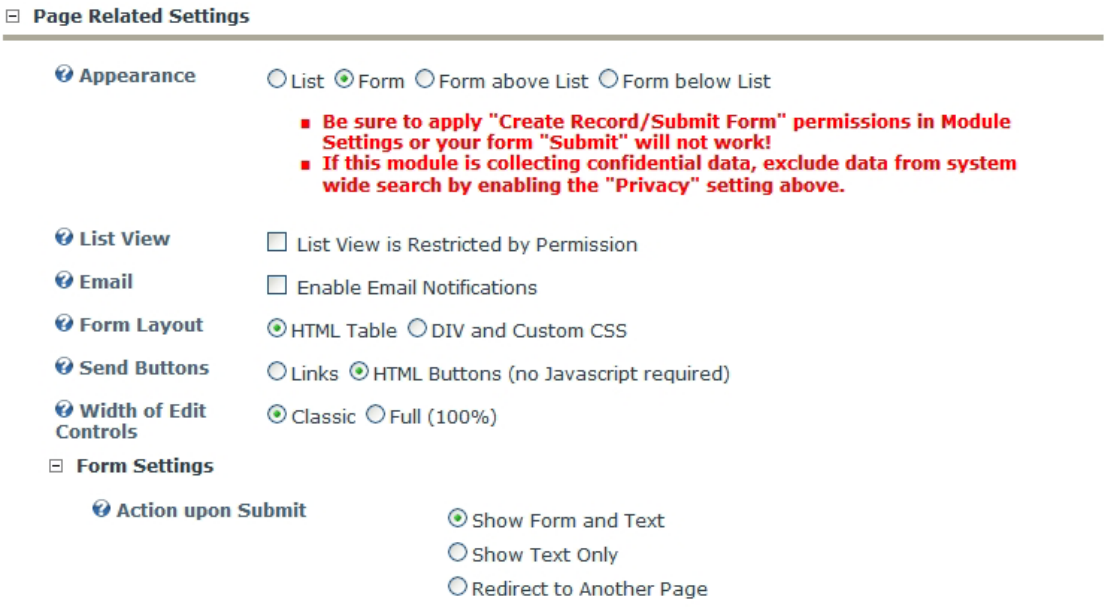
Figure 3-8



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Under the **Page Related Settings**, choose **Appearance: Form** – and for Send Buttons, choose HTML Buttons (no JavaScript required). (*Figure 3-9*).

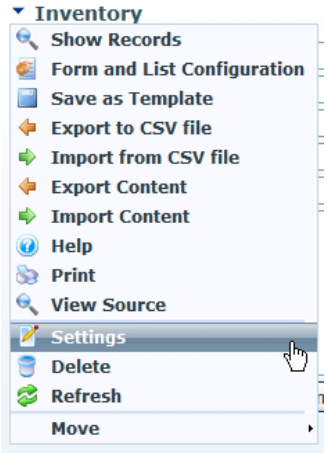
Figure 3-9



For this example, we will leave all other settings as default, and click **Save Configuration and Return**.

We need to apply Permissions to this Form and List module before the form will work properly. Mouse over the action menu, and choose 'Settings' (*Figure 3-10*).

Figure 3-10



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

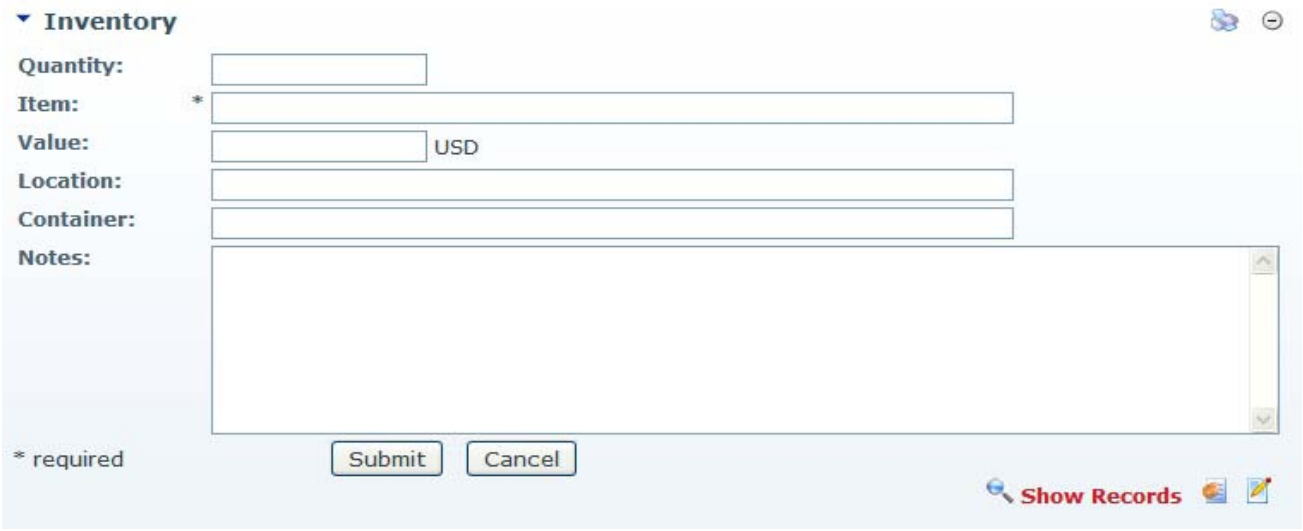
We want to allow all website visitors to use the form, so for All Users role, we enable the **View Module** and **Create Record/Submit Form** permissions. (*Figure 3-11*).

Figure 3-11

	View Module	Edit Module	Edit Record	Create Record/ Submit Form	Delete Record	Display All Columns	Edit Also Restricted Columns	View List while in Form mode
Administrators	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒
All Users	✓	☐	☐	✓	☐	☐	☐	☐
Registered Users	☐	☐	☐	☐	☐	☐	☐	☐
Subscribers	☐	☐	☐	☐	☐	☐	☐	☐
Unauthenticated Users	☐	☐	☐	☐	☐	☐	☐	☐

Congratulations! You have successfully created your first Form. (*Figure 3-12*).

Figure 3-12



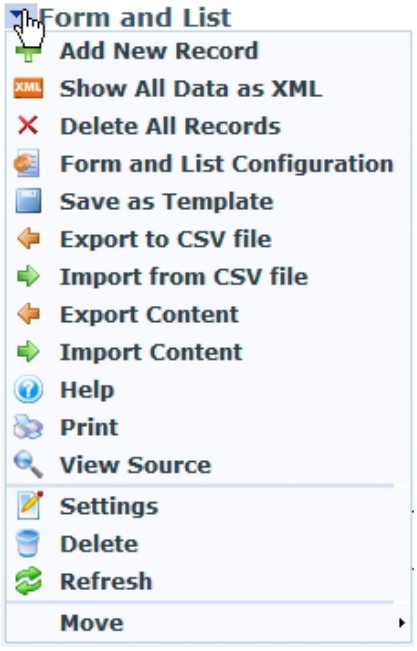
Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 4:
Managing Form and List

Module Action Menu

Once you have configured a Form and List with columns, you can return to edit the configuration, to modify the characteristics of a column, to change or add Advanced Column Options, or to modify display rendering options.

Figure 4-1



To edit the table, mouse-over the module action menu (*Figure 4-1*).

Select **Add New Record** to add a new record to your Form and List.

Select **Show All Data as XML** to open a browser window with all data in XML format.

Select **Delete All Records** to delete all records currently stored in this Form and List. "Delete All Records" shows only when at least one record exists.

Select **Form and List Configuration** to display the central Form and List Configuration settings where you specify all module definitions and properties (*Figure 4-2*).

Select **Save as Template** to create a Form and List template containing the configuration of this Form and List instance. Templates can be used to quickly configure new instances of Form and List based on existing instances of Form and List.

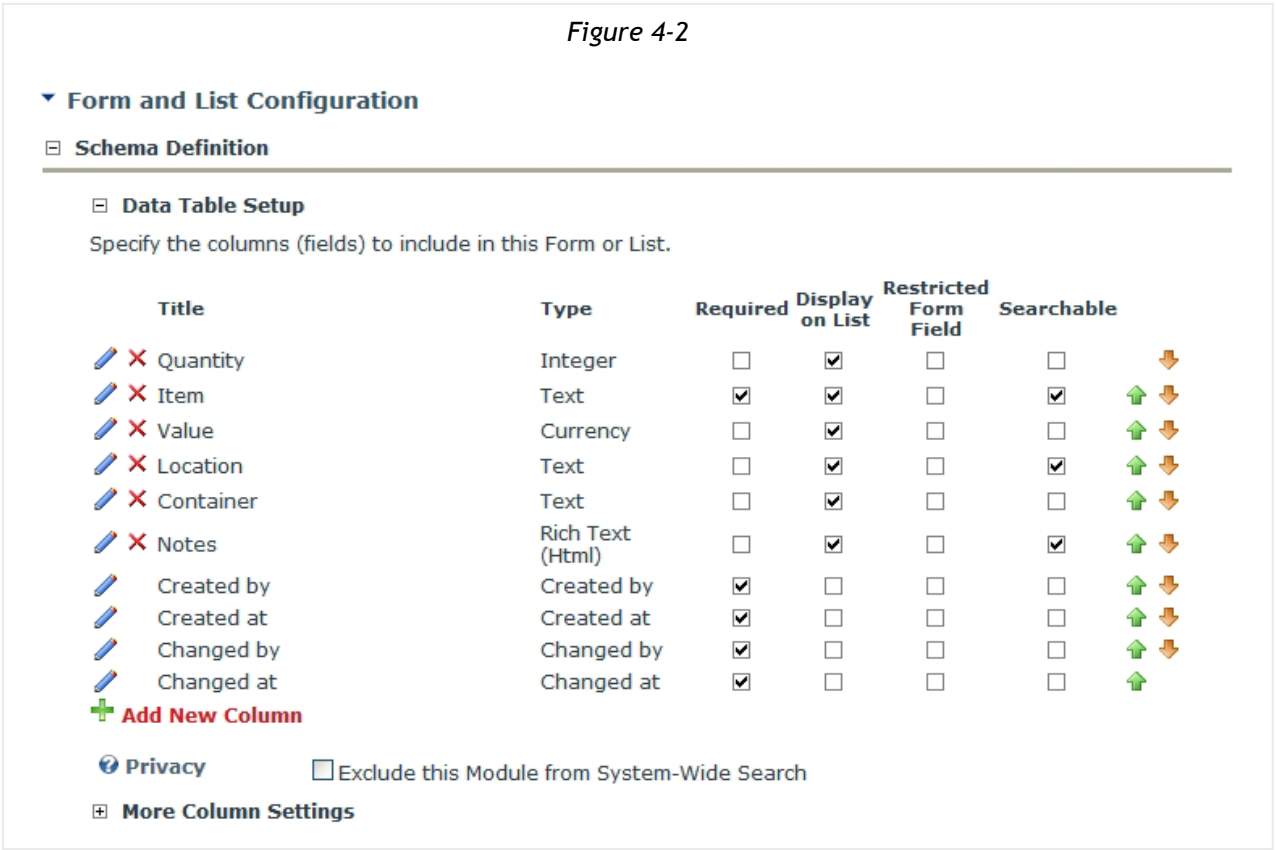
Select **Export/Import to CSV** to export or import a Comma Separated Value data file, useful for importing data in or exporting data out of Form and List.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Working with the Form and List Configuration, Schema Definition is easy and intuitive:

- Click the pencil icon to **Edit** a column and to modify Advanced Column Options (see Advanced Options chapter);
- Click [X] to **Delete** a column. All data stored in this column will be deleted;
- Click the Up Arrow or Down Arrow to **Move** the position of a column;
- Click **Add New Column** to add another field/column to this data collection.

Figure 4-2

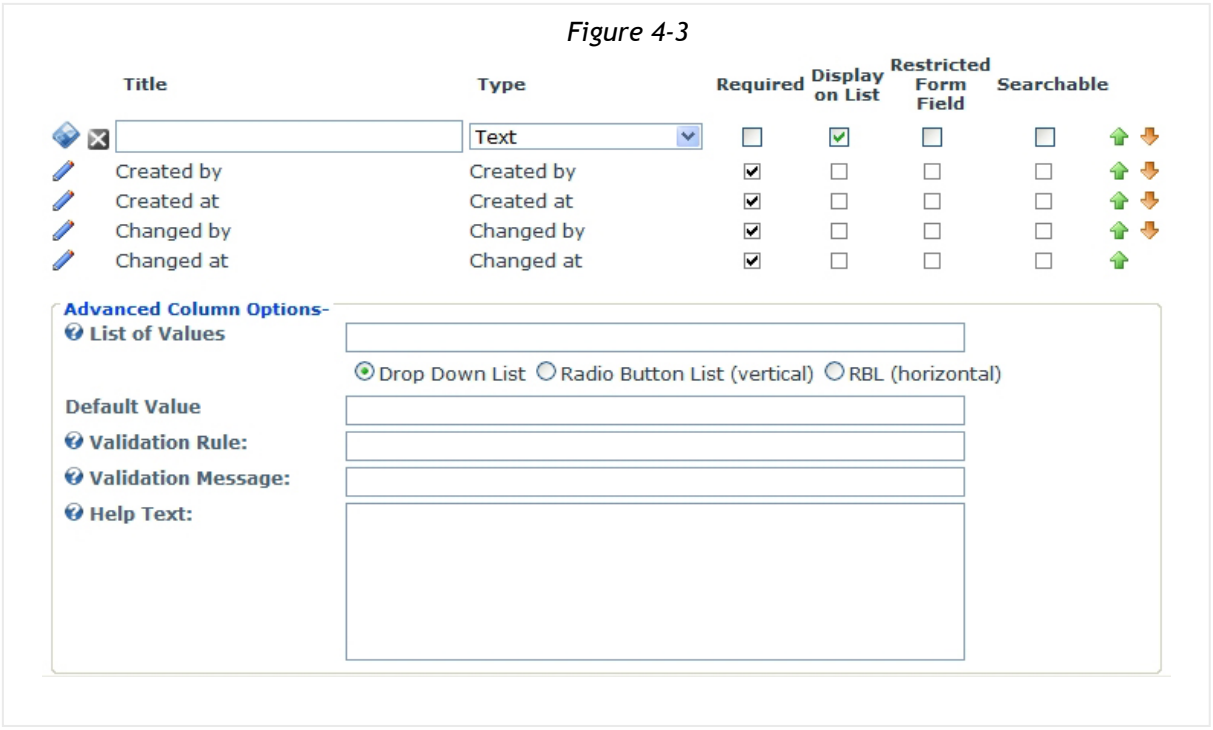


Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Add New Column

Click **Add New Column** (*Figure 4-3*).

Figure 4-3



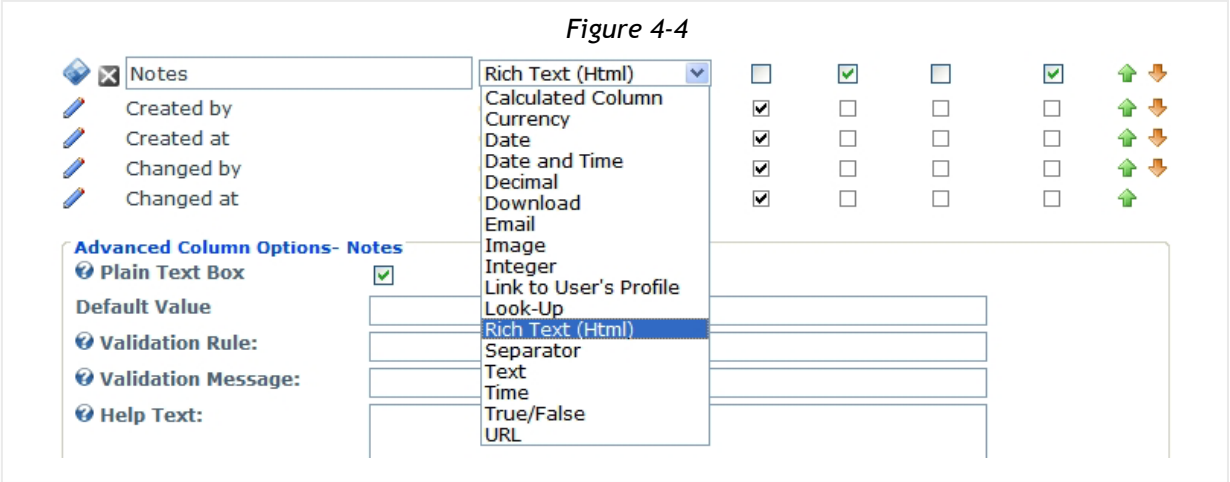
Title

The Title is the name of the column used internally by Form and List and also used as the name of the label that displays next to the input field. A Title may contain spaces, too.

Type

From the dropdown list box, choose the data Type for the data in a column. *(Figure 4-4)*.

Figure 4-4



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Required
Enable this option to force the user to input information for this field.

Display on List
Enable this option to display the contents of this column in the list view, and display an input field for this column on the input form.

Disable this option for a column of data that you do not wish to display in list view.

Restricted Form Field
Enable this option to display the contents of this column in the list view, and display an input field for this column on the input form.

Note: All Admin role members and all roles that have the Form and List "Show all User Defined Columns" permission enabled will still see disabled Visible columns in the List View.

Searchable
Enable this setting if you want the Form and List Search function to search this column, when Search is activated in Grid Table Settings.

Data Types

There are multiple data types available for defining the columns of your Form and List, each with distinguishing characteristics. These provide you with a wide range of input and display controls that will enhance the presentation of your data. To gain the most value from the Form and List, carefully select the appropriate data type for each column.

Calculated Column
To display the results of a calculated expression consisting of tokens, constants, functions, column values and mathematical operators, use the Calculated Column data type. Calculated Column expression examples are found in Chapter 5 section: "**Expression**". Some expressions are made possible by the use of Hidden Columns, described below.

Enable the **Show on Edit Form** setting to display the result of the expression when Form and List is in edit/form mode. This can be useful, for example, to display an image inline in a form. The value is NOT updated dynamically as fields are input or edited.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Currency

Currency contains monetary amounts, in the currency format defined by default in Portal Settings. Note: changing the DotNetNuke portal currency will change the display format, but it will not recalculate the entered values!

Date

Date stores the date specified. You can select the date from the popup calendar. Date is formatted based on the specified settings of the client browser.

Date and Time

Date and Time stores both the date and the time specified. You can select the date from the popup calendar. Date and Time are formatted based on the specified settings of the client browser.

Decimal

The Decimal data type stores numeric data with decimal fractions, e.g. 3.142 or 124.75. Values can be positive, negative or zero. Math operations may include rounding differences. The character used to represent the decimal point depends on the web browser's local settings.

Download

To present files for download, choose the Download data type. Download supports external and internal links to unsecured and secured file system areas as well as access to files and folders that are securely stored in the DotNetNuke database.

Email

Stores data in the format of an Email address. Data of this Type will by default be displayed as a clickable link. If the user has entered a valid "mailto:" address such as info@dotnetnuke.com, clicking the link opens the default Email client.

Note that email addresses are not stored literally but instead the address is cloaked to protect against spam spiders. Also, you cannot use this format to launch to an IM (instant messaging) client.

Image

Use the Image type to display an image in this column, a unique image for each record. The user will be prompted to provide a link to an external URL, or select an image stored in a non-secure folder of the portal where the user has access. The user will be presented with non-secure folder access regardless of the user's permissions.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Integer

Data type Integer displays data in numerical format. Integers are typically used to represent numbers and can be positive, negative or zero in the limits of $\pm 2^{31}$.

Link to User's Profile

Link to User's Profile data type allows the user entering a record to specify the Username of a valid member of the current site. When displayed in Form and List, the Username entered will appear as a hyperlink that, when clicked, will display the User Profile for that site member. Username entry is not case sensitive, but otherwise it must be entered exactly as a valid Username or else no link will appear, no warning or error is displayed, and the name entered is not saved.

Look-Up

Data type Look-Up doesn't store any data, but instead it looks up additional properties from another column, defined in the Advance Column Option - Source Column. Additional properties are available for column types that point to Files (Download, URL, Image) or Users (Created by, Changed by, Email, Link to User's Profile). See 'Available Token Help' for more detail.

Rich Text (HTML)

Stores and displays formatted HTML text, presenting the user with a Rich Text editor. Note, however, that HTML Rich Text editing may be suppressed by control settings or module security configured outside of the Form and List.

Separator

This column type displays a line or rule, allowing you to separate fields into sections. When you choose the option to 'Display on List', the Title or name for the column will also display; disable this setting and only the line will display. 'Acts as Fieldset' setting will create a line and wraps a line box around all the fields that follow, until the next Separator is found. 'Acts as Fieldset' works only when 'Appearance > Form Layout : DIV and Custom CSS' is selected.

Text

Text data type stores and displays a sequence of characters (a string), which may include letters, numbers, diacritics (accent marks) and other special characters. By default, text is displayed "as-is" with no formatting applied. Sorting is literal - as data is sorted in a phonebook. Numbers are sorted by order of their character values (not numerically), that is: 1... 10... 100... 11... 111... 12 ...

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Time

Time stores the time of day specified. You can select the date from the popup calendar. Time is formatted based on the specified settings of the client browser.

True/False

With True/False, the user is presented with a check box, which can be selected (enabled) or unselected (disabled). This allows for “yes or no”, “on or off” choices. An unchecked or disabled value equals “no or off”; there are no null values.

URL

To store a website address for a page that is either in or outside of your portal, use URL. This data type will display to the user the DotNetNuke URL control. The user selects the appropriate link type (URL, Page or File) and depending on the link type, has the option to either enter data or select it from a list.

Hidden Columns

In addition to the columns Types available from the Type dropdown list, depending on the column type you define, additional hidden columns are created. These hidden columns allow for data formatting, conditional testing, localization and XSL rendering.

Shown in the following table, columns beginning with “+” are the hidden column names you may use in Filter Statements, calculated formulas, XSL style sheets, etc. Substitute the “+” with the Title (name) of your user defined column, for example, “myColumnName_UDT_Original” or “myColumnName_UDT_Url”.

A hyphen “-” in this table means a hidden column is not available for this column Type.

Link values are either “TabID=n”, “FileID=n” or URL.

Type	Column Name (renders)	+_UDT_Original	+_UDT_Caption	+_UDT_Value	+_UDT_Ticks	+_UDT_Url
Changed at	Date	-	-	Localized Date	DateTime in Ticks	-
Changed by	Display Name/ Link to Profile	Username	Display Name	-	-	Profile URL
Created at	Date	-	-	Localized Date	DateTime in Ticks	-

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Type	Column Name (renders)	+_UDT_Original	+_UDT_Caption	+_UDT_Value	+_UDT_Ticks	+_UDT_Url
Created by	Display Name/ Link to Profile	Username	Display Name	-	-	Profile URL
Calculated Column	Result of Expression	-	-	-	-	-
Currency	Value	-	-	-	-	-
Date	Date	-	-	Localized Date	Value in Ticks	-
Date and Time	DateTime	-	-	Localized Date	Value in Ticks	-
Decimal	Value	-	-	-	-	-
Download	HTML Link	FileID=	File Name	-	-	File URL
Email	HTML Link	Value	Caption	-	-	-
Image	HTML Image	FileID= or URL	Alt Text	File Path	-	Image URL
Integer	Value	-	-	-	-	-
Link to User's Profile	HTML Link	UserID	Displayed Text	Username	-	Profile URL
Rich Text /(HTML)	Value	-	-	-	-	-
Text	Value	-	-	-	-	-
Time	Value	-	-	-	Value in Ticks	-
True/False	Image [(un) checked.gif]	['true'|'false']	['true'|'false']	[1|0]	-	-
URL	HTML Link	FileID=, TabID or URL	Caption	-	-	URL

Example 1:

When a column named "WebSite" of Type URL is added to Form and List, there are three additional hidden columns created:

- WebSite_UDT_Original
- WebSite_UDT_Caption
- WebSite_UDT_Url

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Entering www.example.net as "WebSite", the hidden fields yield these values:

Column Name	Content
WebSite	Web Site
WebSite_UDT_Original	www.example.net
WebSite_UDT_Caption	Web Site
WebSite_UDT_URL	http://www.example.net

Pointing URL to an internal portal page, WebSite_UDT_Original may be "32" (TabID), and WebSite_UDT_URL would be /home.aspx, for example.

Example 2:

When a column named "Available" of Type True/False is added to Form and List, there is also a hidden columns created with the following names:

- Available_UDT_Original
- Available_UDT_Caption
- Available_UDT_Value

Testing on Available_UDT_Original is useful for filtering purposes. For example, you can set a Filter Statement such as: [Available_UDT_Original] = 'true' to display only records where the 'Available' column is enabled.

Example 3:

By default, the built-in column named "Created by" will render the user's Display Name. However, if you reference the hidden column:

- [Created by_UDT_Caption]

... will display the user's Username.

Quick Tip	Form and List column Titles may contain spaces in the names. However, when you refer to these columns in style sheets, formulas and Filter Statements, you should place brackets around the name, for example: [First Name] or [Changed by]

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Hidden Column EditLink

Form and List automatically creates the hidden column *EditLink*. *EditLink* provides the URL to edit the current row. If the current user does not have Row Edit permission, the column ID is not set, and the Edit image is not displayed. *EditLink* can be used in XSL templates to display the edit image or a custom link to enable the user to edit the row.

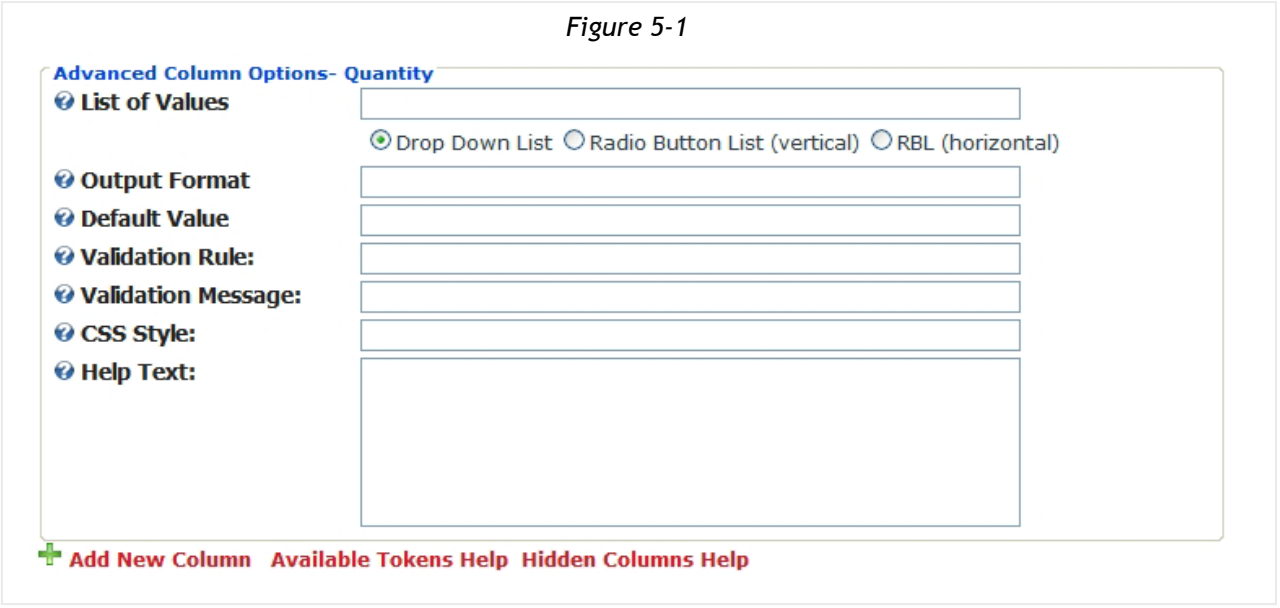
Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 5: Advanced Options

Advanced Column Options

When you click the pencil icon to Edit an existing column, Advanced Column Options will display below the columns list (*Figure 5-1*).

Figure 5-1



Advanced Column Options will vary based on the data type of the column you are currently editing. The following is an alphabetical list of all available Advanced Column Options. Only select Advanced Column Options are available for each data type.

Abbreviated URL

If enabled, only the file name or tab name is displayed as a clickable URL. If disabled (default), the full URL path is displayed.

ALT Tag

To display a fixed text label instead of the URL link, enter the static text to be displayed.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

CSS Style
Enter a CCS Style parameter to apply to this column , for example: width:300px – to set the displayed field width to 300 pixels.

Currency Symbol
Enter the preferred currency symbol or letters to be displayed aside monetary values.

Data Type
From the drop down list, select the data Type for the evaluated Expression property.

Default Value
Enter a preset default value to be displayed. The value can be edited by the user before the record is saved. The value must conform to a valid value for that data Type.

Expression
All valid Data.Column.Expression expressions are allowed. For more information, please refer to the MSDN help in the Additional Information section at the end of this guide.

Example: The following expressions assume that the names “Price”, “Quantity” and “Unit Price” are pre-existing columns already defined in a Form and List instance:

```
Price * Quantity
[Unit Price] * 3
ISNULL([Unit Price],0) * Quantity
Convert([Unit Price] * 3, 'System.String ') + '  Some text'
'<a href=” ' + LinkCol + '”>' + Description + ' </a>'
IIF(Quantity > 10, '10% Discount', 'Regular Price')
```

For more help with expressions, see Chapter 4, **Hidden Columns**.

Help Text
Enter customized tooltip help text for this column. When the user clicks the “?” icon displayed before the input field label, your custom help text will be displayed.

Hyperlink
To display this column as a clickable hyperlink, enable this setting.

Link Caption
To display a fixed text label instead of the link itself, enter the static text to display.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

List of Values
Create a drop down list containing preset choices by entering values separated by semicolons (";"). Leave this option blank to display the default input control.

Output Format
Specify a format mask that will be applied when the column is displayed, i.e. #,###

Date, Time, Date and Time format strings that will be applied to the displayed value:

d	Short Date	f	Long Date and Short Time
D	Long Date	F	Long Date and Long Time
t	Short Time	g	Short Date and Short Time
T	Long Time	G	Short Date and Long Time
M	Month and Day	Y	Year and Month

Plain Text Box
Use a Plain Text Box editor instead of the Rich Text (HTML enabled) editor.

UTC Time Zone
Enable to save Date and Time based on the user's DotNetNuke default Time Zone, otherwise the Date and Time will be saved according to the current host server time.

Validation Message
Text message to display to the user when the Validation Rule is not met.

Validation Rule
Enter a required input pattern using the "Regular Expressions" language. For more information about "Regular Expressions", see the "Additional Information" chapter.

Token Use in Column Options

Within the Advanced Column Options: Default Value, Link Caption and ALT Tag - you can specify strings, which might contain either Module or System Tokens. A complete list of available System Tokens are listed in the next section.

Module Tokens allow you to refer to column values of any defined or hidden column in the current module. Use the syntax [Row:ColumnName] to replace the associated column value of the current record. There is no option to refer to values of other rows.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

DotNetNuke System Tokens

Using tokenized options and expressions, the following system tokens will be replaced with the current value of the associated property:

[Portal:Currency]	Currency String
[Portal:Description]	Portal Description
[Portal:Email]	Portal Admin Email
[Portal:FooterText]	Portal Copyright Text
[Portal:HomeDirectory]	Portal Path (relative) of Home Directory
[Portal:LogoFile]	Portal Path to Logo File
[Portal:PortalName]	Portal Name
[Portal:PortalAlias]	Portal URL
[Portal:TimeZoneOffset]	Difference in Minutes between Portal Default Time and UTC
[User:DisplayName]	User's Display Name
[User:Email]	User's Email Address
[User:FirstName]	User's First Name
[User:FullName]	[deprecated]
[User:LastName]	User's Last Name
[User:UserName]	User's Login User Name
[Membership:Approved]	Is User Approved?
[Membership:CreatedDate]	User Registration Date
[Membership:IsOnline]	Is User Currently Online?
[Profile:<Property>]	Use any default or custom Profile Property as defined in Users > Manage Profile Properties. Use non-localized Property Name only.
[Tab:Description]	Page Description Text for Search Engine
[Tab:EndDate]	Page Display Until Date
[Tab:FullUrl]	Page Full URL
[Tab:IconFile]	Page Relative Path to Icon File
[Tab:KeyWords]	Page Keywords for Search Engine
[Tab:PageHeadText]	Page Header Text
[Tab:StartDate]	Page Display from Date
[Tab:TabName]	Page Name
[Tab:TabPath]	Page Relative Path
[Tab:Title]	Page Title (Window Title)
[Tab:URL]	Page URL

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

[Module:Description]	Module Definition Description
[Module:EndDate]	Module Display Until Date
[Module:Footer]	Module Footer Text
[Module:FriendlyName]	Module Definition Name
[Module:Header]	Module Header Text
[Module:HelpURL]	Module Help URL
[Module:IconFile]	Module Path to Icon File
[Module:ModuleTitle]	Module Title
[Module:PaneName]	Module Name of Pane (where the module resides)
[Module:StartDate]	Module Display from Date

[DateTime:Now]	Current Date and Time
[Ticks:Now]	CPU Tick Count for Current Second
[Ticks:Today]	CPU Tick Count since Midnight
[Ticks:TicksPerDay]	CPU Ticks per Day (for calculations)

[Server:*Name*]	Server *Name*
[Querystring:*Name*]	Value of Querystring *Name*
[Form:*Name*]	Form Posted Value of *Name*

Data Type "Look-Up" to Source Column Type "Download" or "URL" also supports:

[File:Name]	Name of File
[File:Folder]	Folder of File
[File:Path]	Complete Path to File
[File:Size]	File Size in Bytes
[File:SizeMB]	File Size in Megabytes
[File:Clicks]	Number of Downloads
[File:Extension]	File Extension

Data Type "Look-Up" to Source Column Type "Email", “Created by”, “Changed by” or “Link to User’s Profile” also supports:

[Gravatar:MD5Hash]	MD5 Hash of the Email Address (if available)
[Gravatar:URL]	URL to the Gravatar Image
[Gravatar:Image]	Gravatar Image

For Date/Time and numeric values, you can also append a <format> string defined by the .NET framework, for example:

[DateTime:Now|<format>]	current date/time formatted according to <format>, e. g. [DateTime:Now|f] displays current date in short format (does not apply to Calculated Column expressions)

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

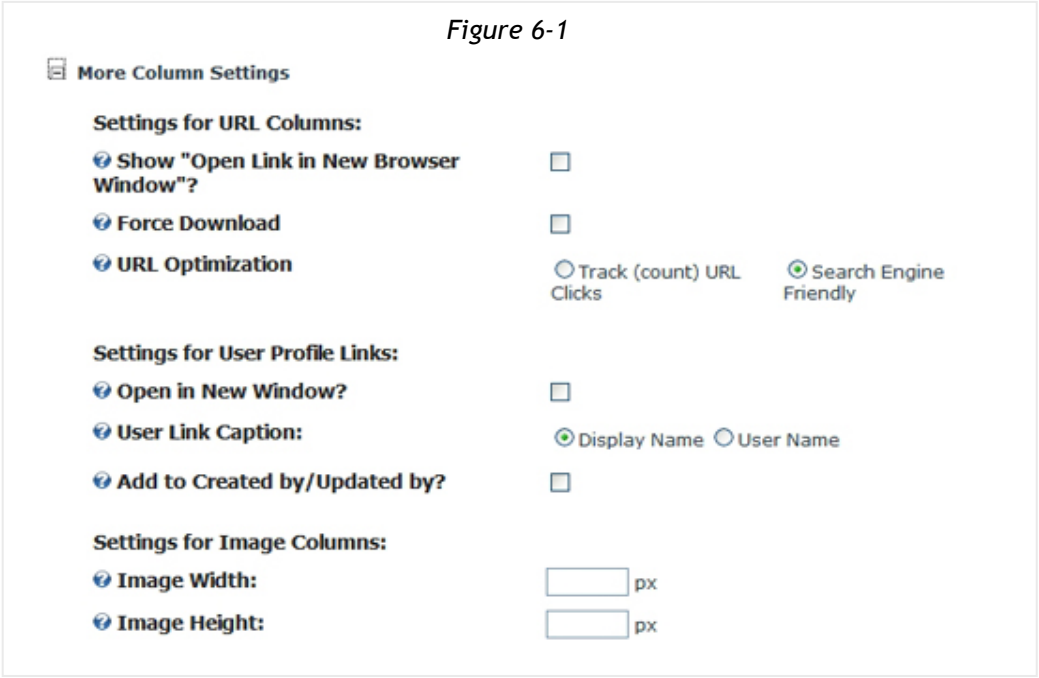
Chapter 6: Column & Page Settings

Located below the Data Table Setup in Form and List Configuration, there are many options that allow you to control the way Form and List operates and displays data.

Column Settings

Column Settings allows you to manage display options that are applied to all Type URL and Image columns in this Form and List module. (*Figure 6-1*)

Figure 6-1



Show "Open Link in New Browser Window?"

Enable this option to offer the user the option to open a new browser window whenever a Type URL column hyperlink is clicked. Disabled by default, all Type URL columns will open pages and local URL addresses in the currently opened window. All links to files and external URLs will open in a new browser window.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Force Download

Enable this option to configure all Form and List resource URLs as file download, forcing the browser to immediately download the resource.

URL Optimization

When you do not wish to count link clicks for page URLs contained in the Form and List module, choose the option Search Engine Friendly to instruct Form and List to use search engine friendly links. If you wish to utilize LinkClick.aspx and count clicks to page URLs contained in Form and List, choose Track (count) URL Clicks option instead. For Form and List links to files, LinkClick.aspx is always used.

Open in New Window

Enable this setting to set Form and List to open all links to a User Profile in a new browser window.

User Link Caption

Choose to display either the user's Display Name or Username when Form and List renders a link to the User's Profile.

Add to Created by / Updated by?

Enable this setting to have the Form and List columns Created by or Updated by display as links to the User's Profile.

Image Width / Image Height

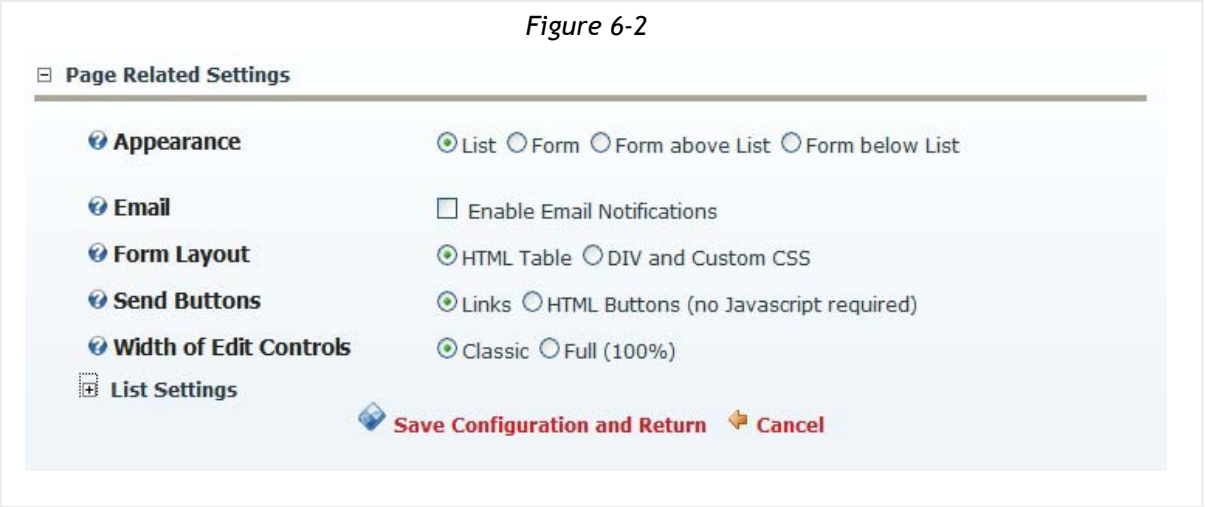
If you leave these fields blank, images are displayed in their original size. If you specify a width or height value (numeric, in pixel) all images are scaled proportionally to fit the sizes you specify.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Page Related Settings

Settings in this section apply only to the Form and List instance on the current page only. If you use the same Form and List instance on different pages, you may vary the way Form and List displays on various pages using these settings (*Figure 6-2*).

Figure 6-2



Appearance

Appearance is the primary setting that configures Form and List for either List or Form mode. Choose to display in either List or Form mode, or if you choose, display both form and list on the same page by selecting the appropriate Appearance setting.

Email

You can configure Form and List module to send a notification email when one or more trigger conditions are met, e.g. when a new record is added, or when an existing record is changed or deleted. You may specific one or more notification recipients, and even send to select recipients based on the contents of email addresses collected in a Form. Notification messages are completely customizable, and you may include variable [token] values based on data input in a Form.

Form Layout

Choose to render Forms using underlying code that utilizes HTML Table structure or DIV and Custom CSS structure.

Send Buttons

Select either Links or HTML Buttons to choose if the Send and Cancel links under input Forms should display as simple text links, or as HTML buttons.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Width of Edit Controls

Set the width of Form and List edit controls by choosing either Classic (fixed width) or Full (100%), which attempts to size fields based on the width of the browser display.

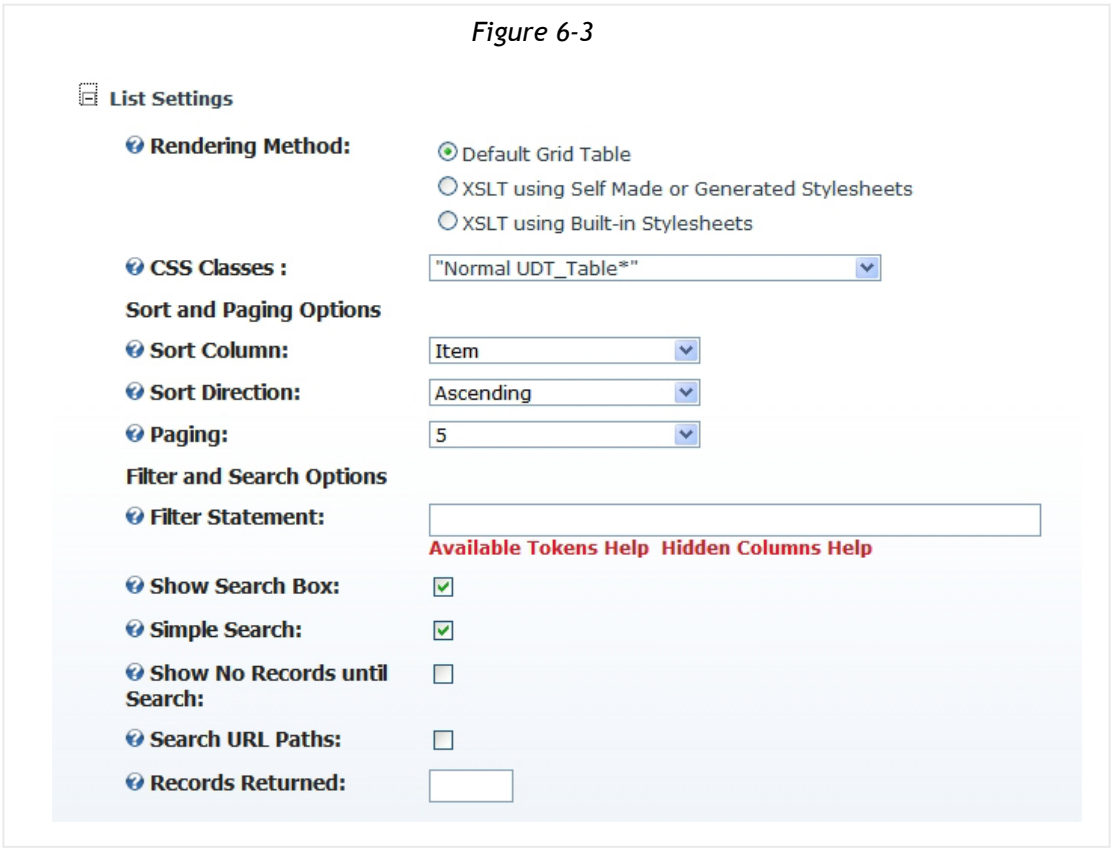
List Settings

Rendering Method

Choose between two rendering methods - Default or XSL (eXtensible Stylesheet Language) - (Figure 6-3):

- Default Grid Table: simple tabular column format, one column aside the next
- XSLT using Self Made or Generated Stylesheets: use your own XSL code, or use the Token Based XSLT Generator and Editor to create XSL scripts, wizard style
- XSLT using Built-In Stylesheets: pre-built XSL that come with Form and List

Figure 6-3



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

CSS Classes

This option determines which CSS classes are used to format the output of the Default Grid Table (*Figure 6-3*). You can switch between three CSS class sets:

- "Normal UDT_Table*"
- "DataGrid_*"
- YUI DataTable Styled

Each CSS class has a unique look and feel. Experience the various CSS classes by applying them to an existing Form and List instance containing at least a few records.

The following table lists the applied CSS classes for the corresponding <tr> tag.

	"Normal UDT_Table*"	"DataGrid*"
Table Head	NormalBold UDT_Table_Head	DataGrid_Header
Table Row	Normal UDT_Table_Item	DataGrid_Item
Table Row (Alternating)	Normal UDT_Table_AlternateItem	DataGrid_AlternatingItem

The CSS classes Normal and NormalBold are part of the default DotnetNuke CSS. These classes typically define the layout of your DNN portal/site. Careful not to disturb settings outside of Form and List, you should only override the "Normal UDT_Table*" classes.

Example:

```
.UDT_Table_Head
{
   background-color: yellow
}
.UDT Table AlternateItem
{
   background-color: whitesmoke
}
```

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Default Grid Table Settings

When you select Rendering Method: Default Grid Table, you can specify any of the following settings. When you choose XSLT Self Made or Built-In rendering, Sort and Paging options are disabled, but Filter, Search and Records Return are active:

Sort Column

Select the column on which to sort Form and List records in List view. When using the Default Grid Table, Form and List allows you to choose only one primary column on which to sort. If you choose no sort column, Form and List records are displayed in the order the records were entered.

Sort Direction

Choose the default sort direction to be applied to the Sort Column. Choose Ascending (lowest to highest) or Descending (highest to lowest).

Paging

If Paging is set to <no paging>, all records are displayed in one long list. By specifying a paging size, you set the maximum number of records to display per page, using the DotNetNuke paging control, with the built-in navigation control to move between pages.

Filter Statement

Enter an expression that must be 'True' for a Form and List record to be displayed. When you apply a Filter Statement, you cannot access all records. It is recommended that you use Filter Statement only on a copy of a Form and List module ('Add Existing Module' from the DotNetNuke toolbar).

Enter a Boolean expression that follows the Data.Column.Expression syntax. For more information on the Data.Column.Expression syntax, please refer to the MSDN link in the Additional Information section at the end of this user guide.

Click **Available Tokens Help** or **Hidden Columns Help** for information on which Tokens and Hidden Columns may be used in your Filter Statements.

Here are a few examples of valid Filter Statements (assuming the column names "Price", "Quantity", "City", "Available" and "Unit Price" are columns defined in Form and List):

```
Price * Quantity <> 0
(City LIKE 'London') AND (Country = 'UK')
Available_UDT_Original = 'true'
([Unit Price]) is not null
```

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Search Options

Show Search Box: Display a Search Box that allows users to search for data contained in the columns marked 'Searchable' in the column definition list.

Simple Search: Enable this setting to activate a single, simple Search Box. This Search Box will perform a LIKE (contains) search on all the columns currently specified as 'Searchable' in the column definition list.

Show No Records Until Search: Enable this option to show no records until the user performs a search. Only the Form and List Search Box will display until a search is performed. Disable this option to display records in the default sort order.

Search URL Paths: enable this option, in order to search URL strings as well, i.e. find portal in *http://www.mydomain.com/portals/0/my.gif* URL of an image, URL or download data type as well.

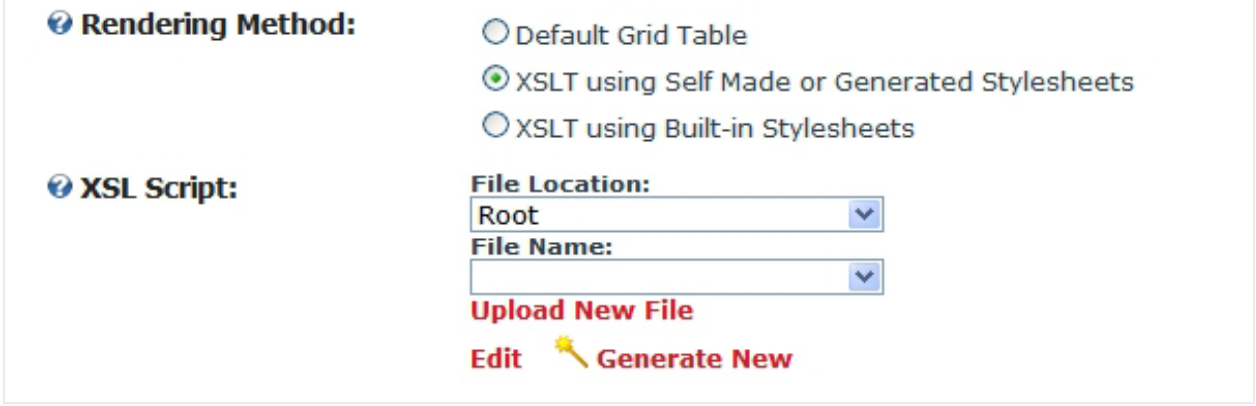
Records Returned

Enter an integer value between 1 and 1000 to set the maximum number of records returned from file for this Form and List instance. This setting is applied after all other settings which may limit the number of records returned, e.g. Filter Statement.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

XSL using Self Made or Generated Stylesheets

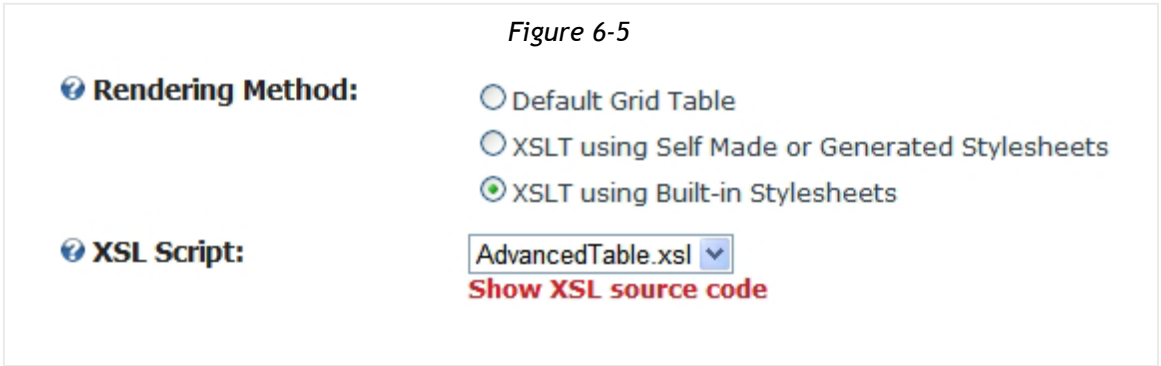
Figure 6-4



When you select Rendering Method: **XSL using Self Made or Generated Stylesheets**, select an XSL from the DotNetNuke file system, or you may **Upload New File** from your local system. (*Figure 6-4*). Choose **Edit** to modify an existing XSL script, or to create a brand new XSL transformation using the powerful Token Based XSLT Generator and Editor, choose **Generate New**.

XSL using Built-in Stylesheets

Figure 6-5



When you select Rendering Method: **XSLT using Built-in Stylesheets**, select from one of the predefined XSL Scripts included with Form and List (*Figure 6-5*):

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

- ***AdvancedTable*** appears as the Default Grid Table, and includes a clickable 'magnifier' to show more detail about a record.
- ***BasicTable*** appears as the Default Grid Table and is an excellent starting point for editing your own XSL transformations.
- **CardView** will show the first visible text column as a Title, the first Type Image column as the record image and the first HTML/Rich Text column as a description.

Quick Tip

You can examine any style sheet by clicking "Show XSL source code". Save the XSL locally, modify it, then upload your changes as *XSLT using Self Made or Generated Stylesheet*.

- ***OnlyOwnItems*** shows only the rows entered by the current user. You should also enable Form and List Module Settings > "*Users are only allowed to manipulate their own items*" to prevent users from guessing foreign edit URLs and viewing other user's data.
- ***Transpose*** renders a table with records displayed in a vertical column format, each record occupying one column.

Show XSL Source Code

Click this link to see the source code for the current XSL script. View the sheet to examine the logic of the stylesheet, or to save the script and modify it. You can upload your changed stylesheet as an **XSLT using Self Made or Generated Stylesheets**.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 7: Token Based XSLT Generator and Editor

What is the Token Based XSLT Generator/Editor?

XSLT style sheets allow you to customize or 'Transform' the display of data to your exact specification. However, for those unfamiliar with XSL (eXtensible Stylesheet Language) syntax, creating XSL transformation style sheets can be very difficult.

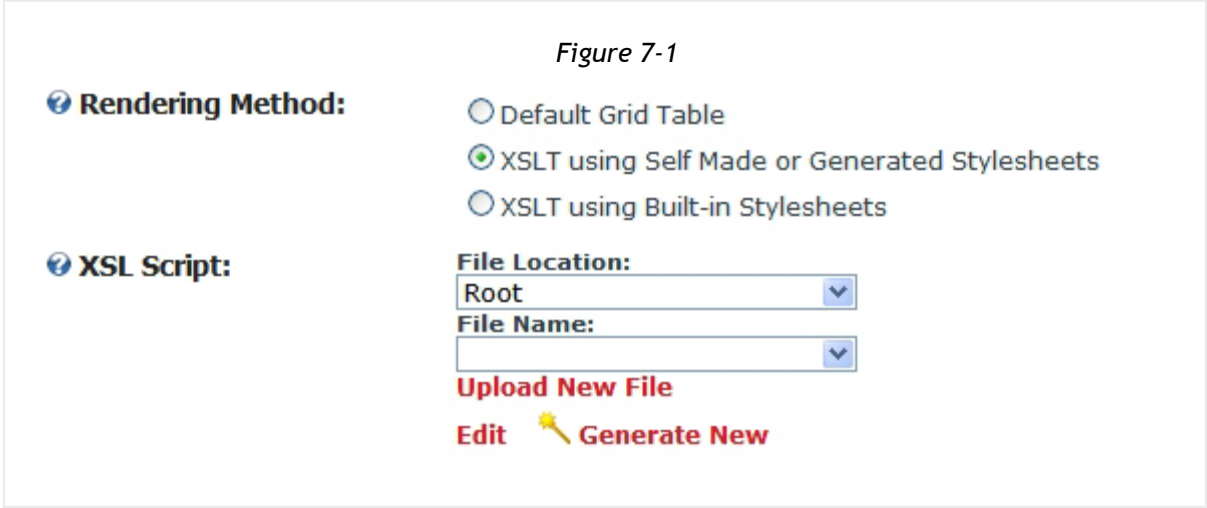
The Form and List **Token Based XSLT Generator/Editor** allows you to create complex XSLT scripts with very little XSL knowledge, simply by selecting fields/columns from a drop down list. Each content column or context [Token] you select is inserted into an HTML template. You can also select options to enabling Sorting, Paging, Detail View and Searching functionality.

After you set a few options and complete the layout of your HTML template, you create an XSL script by clicking the **Generate from HTML Template** button. The XSLT generated by Form and List is immediately displayed. You may modify the generated XSLT, allowing for further customization. Or if you prefer HTML, return to the HTML template, make modifications to the HTML code, then click the **Generate...** button again to re-generate your XSLT! Once you save your generated XSLT, you may reuse it in other Form and List instances, too.

The Form and List Token Based XSLT Generator and Editor is a great tool for building powerful XSLT solutions, and to learn more about XSL syntax and functionality.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Figure 7-1



XSLT using Self Made or Generated Stylesheets

If you already have an XSL file you would like to use with Form and List, select **Upload New File** (*Figure 7-1*) to upload an XSL style sheet file from your PC to the website.

Choose the appropriate folder where you wish to store XSL files. Select a folder where you have write permissions. **XslStyleSheets/** may be available on your website.

You may select an XSL file previously saved or uploaded to your site. If you have just begun working with XSL, no style sheets will be found, and the File Name field will be empty.

Creating Your First XSLT Generator Style Sheet

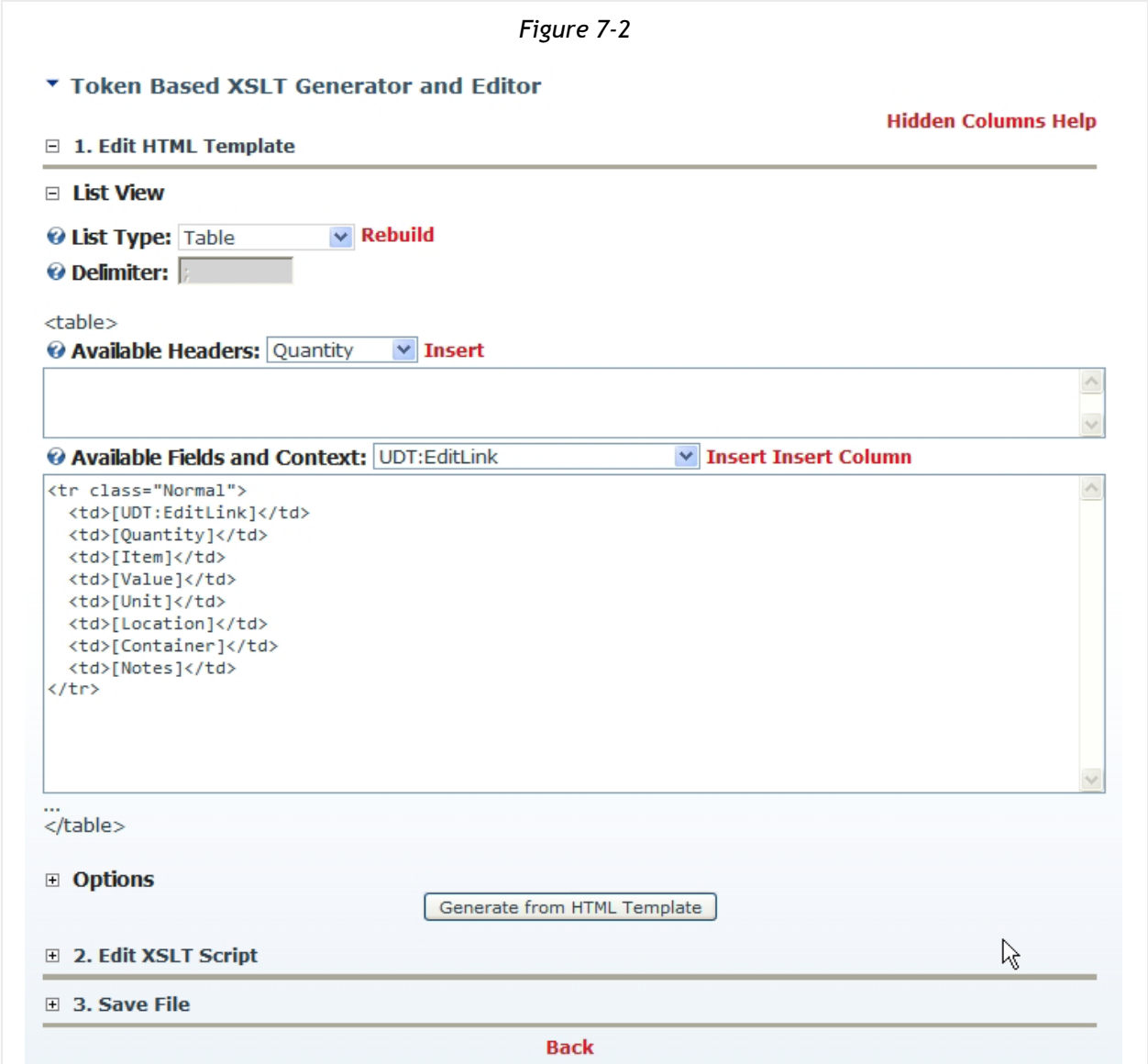
If you are new to Form and List, the following process assumes that you have first created the Form and List instance described in Chapter 3, Quick Start, and you have added a few records as described in Chapter 3. Before following these steps, first navigate to the Form and List instance created in Chapter 3.

If you have already created your own Form and List instance, the screens you see in this chapter will vary depending on the columns defined in your Form and List instance.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Click **Generate New** to create a new XSL transformation style sheet using the Token Based XSLT Generator and Editor. (*Figure 7-2*)

Figure 7-2



Based on the columns defined in the Form and List column definition list, XSLT Generator will display a default layout for your data. Depending on your current Form and List instance, the **Available Fields and Context** will vary. If you created the Chapter 3 Form and List example instance, the above columns would be displayed.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Each time you **Generate New** using the Token Based XSLT Generator/Editor, a simple table layout will first display in the HTML template window. You can manually change this HTML to your specification, or use **List Type** to change the basic template layout.

List Type

To change the basic layout of the HTML template, choose one of several HTML elements. To see how it works, select various List Types from the dropdown list, and watch the basic layout of the HTML template change with each selection.

Delimiter

Choose a delimiter to insert between columns. A delimiter can be one or more characters, or even a tag, for example, <br /> to place a break between columns.

Available Headers

Choose columns as headers from the dropdown list that includes all Form and List columns. If 'Add Sorting Support to the Style Sheet' is enabled, Header columns also allow for a user click to alter the list order (ascending or descending) based on a column.

Available Fields and Context

All available Form and List columns (including hidden/system columns) are listed in this dropdown list. Select a column, or other special Context content, to insert the [Token] in the current cursor position in the HTML text box.

Please Note	All code in the HTML template must be HTML in well-formed XML format, otherwise the Generator will fail to make an XSLT. At the very least, well-formed XML means that you must be sure that all open tags are closed (...), and use instead of .

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Options

Figure 7-3



Figure 7-3 lists several built-in options that are available to enhance your XSL template.

Please note that these options also increase the complexity of your XSLT style sheets. If you are new to XSL and using the Generator to create style sheets for learning, you might choose to leave all Options disabled. This will simplify the XSL created by the Generator.

Add Sorting Support to the Style Sheet: If you have set Sort Column and Sort Direction in your Form and List instance, this option will use those settings to sort data.

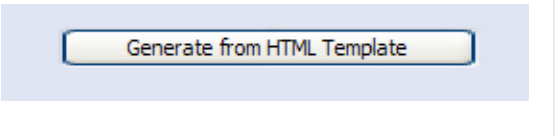
Add Paging Support to the Style Sheet: This option provides 100% XSLT paging control. This is not the same pager control used by the Default Grid Display.

Add Search Box to the Style Sheet: This option adds a case-sensitive Search Box that allows the user to search any columns marked as 'Searchable' in the column definition list. This setting is marked (Obsolete) to indicate that an improved case-insensitive search can be enabled in XSL Mode using a brief online tutorial that shows you how to customize your XSL for search case insensitivity:

http://www.formandlist.com/Documentation/CustomSearchwithRecordFilter.aspx

Add Detail View: This option will add a magnifier to the side of each record, allowing the user to click the magnifier for more details about a record.

Figure 7-4



Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Creating an XSLT script from an HTML template is quick and easy. Just click **Generate from HTML Template** (*Figure 7-4*) to make your XSLT style sheet (*Figure 7-5*).

Figure 7-5

2. Edit XSLT Script

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:udt="DotNetNuke/UserDefinedTable" exclude-result-prefixes="udt">
  <xsl:output method="xml" indent="yes" omit-xml-declaration="yes" />
  <!--
  This prefix is used to generate module specific query strings
  Each querystring or form value that starts with udt_{ModuleId}_param
  will be added as parameter starting with param
  -->
  <xsl:variable name="prefix_param">udt_<xsl:value-of
select="//udt:Context/udt:ModuleId" />_param</xsl:variable>

  <xsl:template match="udt:Data" mode="list">
    <tr class="Normal">
      <td>
        <xsl:call-template name="EditLink" />
      </td>
      <td>
        <xsl:value-of select="udt:Quantity" disable-output-escaping="yes" />
      </td>
      <td>
        <xsl:value-of select="udt:Item" disable-output-escaping="yes" />
      </td>
      <td>
        <xsl:value-of select="udt:Value" disable-output-escaping="yes" />
      </td>
      <td>
        <xsl:value-of select="udt:Unit" disable-output-escaping="yes" />
```

The HTML template is instantly transformed into an XSL script. The field/column and context [Tokens] are replaced by their XSL equivalents, normally *xsl:value-of* commands. If, however, you have placed a token inside a HTML attribute, it will be handled as an Attribute Value Template {token}.

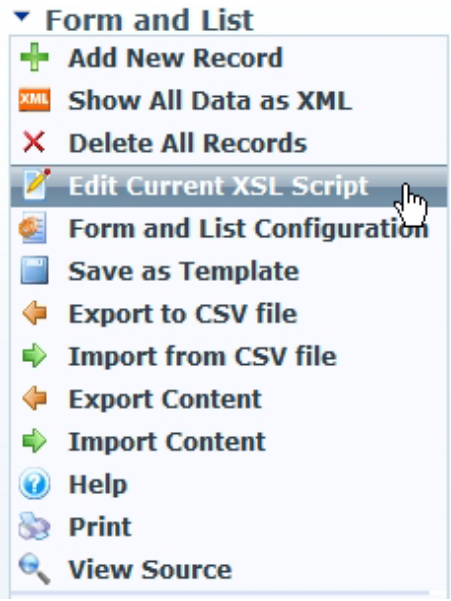
Enter a File Name for your style sheet, then click **Save File and Return**. You do not need to add the file extension; the .XSL extension will be added for you automatically.

To return to this XSLT style sheet, click **Edit** located to the left of Generate New from the Manage Form and List screen.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

You may also return directly to the XSLT Generator/Editor by using the Form and List action menu option, **Edit Current XSL Script** (*Figure 7-6*):

Figure 7-6



Not only can you return to edit the generated XSL script, but you can also return to the HTML template! You may edit the HTML template and regenerate a fresh XSLT to include changes you have made to the HTML template and associated options.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Advanced XSLT

Form and List maintains the memory representation of its data inside a dataset. The dataset contains 3 tables named Data, Fields and Context. This dataset gets converted as XML and is given to the XSL transformation for access to data and context.

Table	Content
Data	Form and List Table (including the hidden columns)
Fields	Field definitions and their settings
Context	(see Context Data table below)

The Context Data Table provides additional information for creating better style sheets:

Columns	Comments
ModuleId, TabId, PortalId	Ids to identify the current module
UserName, BestUserName	BestUserName is usually the display name of the current user. If the user is deleted it defaults to UserName.
Homepath, ApplicationPath	Path to the portal root and path of the application `<img src="[Homepath]/img/myimage.gif">`
UserRoles	Roles are concatenated and surrounded by |. Example: |role1|role2|role3| You can test whether a user is in a certain role using: `<xsl:if test="contains(/udt:Context/udt:UserRoles,'|role1|'">`
IsAdministratorRole	"True" if user is in the AdminstratorRole
CurrentCulture, LocalizedDate, Now, LocalizedSearchString	Useful for Localization
Parameter	No longer used; see 'Parameter Support' below
NowInTicks, TicksPerDay	XSL has no date functions, but every date column has also its value in ticks inside a hidden column. You can use this to compare date times or to calculate time spans. `<xsl:value-of select="(d_UDT_Ticks -NowInTicks) /TicksPerDay"/> (Pseudo Code!!)`

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Parameter Support

The new XSLT Generator engine integrates the ModuleID as part of the parameter name. Each querystring or form parameter should start with *udt_{moduleId}_param*

We recommend adding this definition to the top of every XSL script:

```
<!--
This prefix is used to generate module specific query strings
Each querystring or form value that starts with udt_{ModuleId}_param
will be added as parameter starting with param
-->
<xsl:variable name="prefix param">udt <xsl:value-of select="//udt:Context/udt:ModuleId"
/>_param</xsl:variable>
```

If you wish to 'pass-back' the query string or form value, use this 'how to' example:

```
<!—query string -->
<a href="?&amp;{$prefix param} demo1={$paramvalue}" >…</a>

<!-form based →
<input type="text" name="{$prefix_param}_demo2" />
```

All query string / form parameters that start with the correct string are returned to the XSL script as XSLT parameters:

```
<xsl:param name="param_demo1" />
<xsl:param name="param_demo2" />
```

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

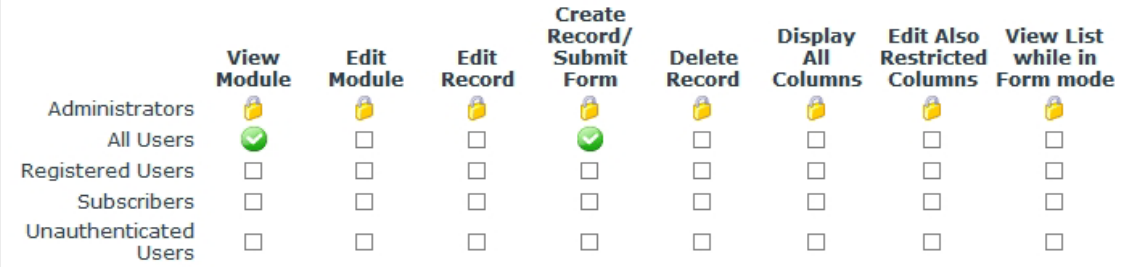
Chapter 8: Module Settings

In addition to the standard module settings available for all DotNetNuke modules, the Form and List module includes several settings unique to the Form and List module.

Permissions

The default DotNetNuke permission grid consists of "View Module" and "Edit Module" settings only. For Form and List, the permission grid has been extended to allow for more advanced control *(Figure 8-1):*

Figure 8-1



View Module

Designates which site roles are allowed to view the Form and List module.

Edit Module

Designates which site roles are allowed to edit module settings and "Manage Form and List" (to configure columns).

Edit Record

Designates which site roles are allowed to edit Form and List records.

Create Record / Submit Form

Designates which site roles are allowed to add Form and List records and submit forms.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Delete Record

Designates which site roles are allowed to delete Form and List records.

Display All Columns

Designates which site roles can view all columns, even those columns with “Restricted Form Field” setting enabled.

Edit Also Restricted Columns

Designates which site roles are allowed to edit columns with “Restricted Form Field” setting enabled.

View List While in Form Mode

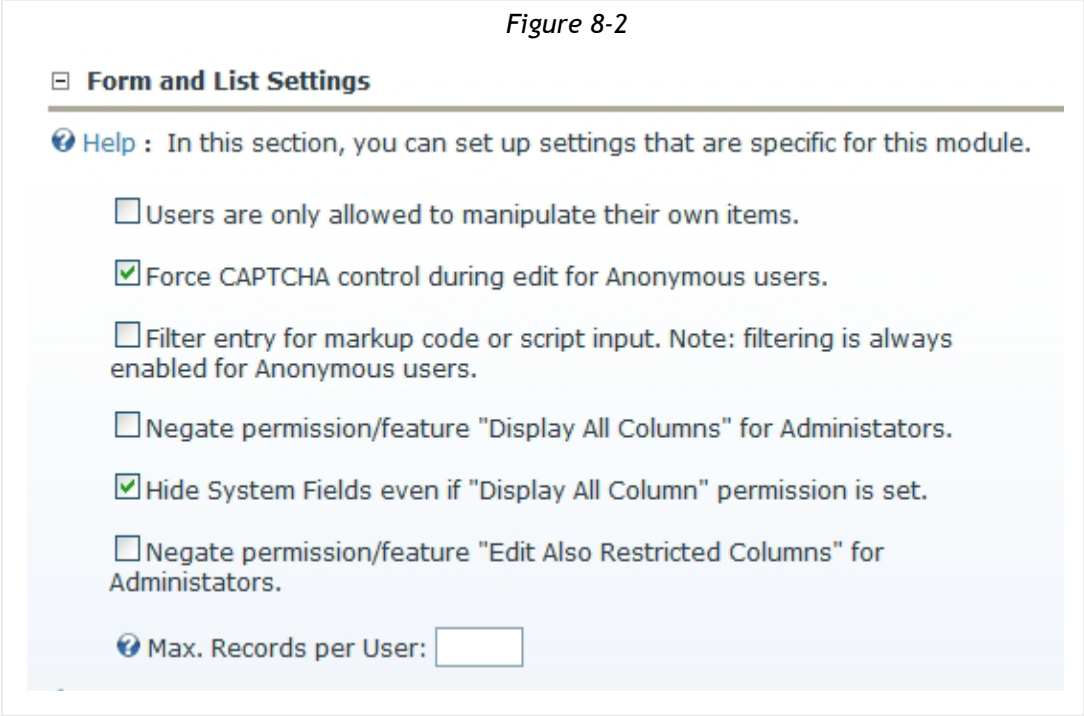
Designates which site roles are allowed to view list, to Show Records from the action menu, when the Form and List module is set to Form mode.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Form and List Settings

These settings offer further security and permission control over the Form and List instance *(Figure 8-2)*:

Figure 8-2



Users are only allowed to manipulate their own items

When you enable this option, each user is only allowed to view, edit and delete only the records they created.

Force CAPTCHA Control during edit for Anonymous users

If you allow anonymous users to enter data into your module, there is a risk that Internet BOTS will also find Form and List and possibly add unwanted SPAM to your tables. To prevent this, you can either allow only registered users to add records, or you can enable CAPTCHA so that anonymous users must enter a CAPTCHA verification code, displayed as an image inside Form and List, before adding a record.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Filter entry for markup code or script input

Enable this option to prohibit anonymous users from entering text with markup code (tags i.e. BOLD) and to prevent non-Admin users from entering scripts. Scripts may harm the user's PC or the web server by executing malicious code.

Note: For security reasons, filtering for script code is always activated for Anonymous Users, even if this option is not enabled.

Negate Permission/Feature "Display All Columns" for Administrator

Enable this option to disable or disallow the "Display All Columns" override that is normally granted to Administrators, always allowing Admin to display all columns. This setting allows the Administrators to experience Form and List as a user who does not have "Display All Columns" enabled.

Hide System Fields Even If "Display All Column" Permission is Set

Enable this option to hide the built-in system fields (Created by, Created at, Changed by, Changed at) even if the "Display All Column" permission is enabled.

Negate Permission/Feature "Edit Also Restricted Column" for Administrator

Enable this option to disable or disallow the "Edit Also Restricted Column" override that is normally granted to all Administrators, always allowing Admin to edit restricted columns. This setting allows the Administrators to experience Form and List as a user who does not have "Edit Also Restricted Column" enabled.

Max. Records per User

To set a limit on the number of records any one user can add, enter an integer value in Max. Records per User. Form and List "Created by" system column is used to maintain the maximum limit you set. Leave blank, for unlimited records per user.

Click **Update** to save your settings.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Chapter 9: Managing Records

View Records

Form and List allows you to view records in a variety of layouts, using various Rendering Methods. For the examples that follow, the **Rendering Method** is set as Default Grid Table, and the CSS Classes: YUI DataTable Styled *(Figure 9-1)*.

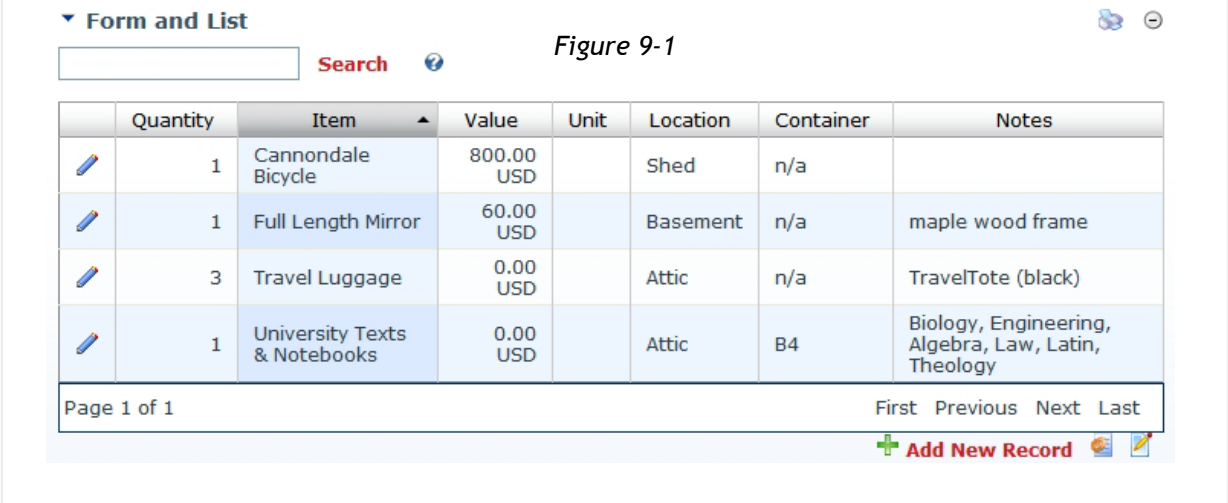


Figure 9-1

You can easily configure Form and List to allow users to Search for records. When you enable the **Simple Search** setting, Form and List performs a 'contains' only search on all columns marked 'Searchable' in the columns definition list *(Figure 9-1)*.

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

When you disable 'Simple Search', Form and List allows users to select a column to search (or all searchable columns), and choose a more specific condition *(Figure 9-2)*.

Figure 9-2



To enable Search, from the module Action Menu ( ), select Form and List Configuration > 'Show Search Box'. The 'Simple Search' option is located just below 'Show Search Box'. Before Form and List search will work, you must designate at least one column as 'Searchable' in the Schema Definition, Data Table Setup.

View/Edit > Only Own Items (Records)

The ability to configure Form and List to allow users to view and edit only their own records is achieved by one of two methods, Filter Statement or XSL style sheet.

Regardless of which method you choose, you should first enable the following security setting in Module Settings (see *Figure 8-2*):

Users are only allowed to manipulate their own items

Filter Statement

If you are using Rendering Method: Default Grid Display, a simple Filter Statement will enable users to View/Edit only their own records:

```
[Created by_UDT_Original] = '[User:UserName]'
```

This example assumes you have not renamed the default Form and List system column: **Created by**.

XSL Style Sheet

If you are using Rendering Method: Predefined XSL Transformation, select the predefined style sheet named "**OnlyOwnItems.xsl**".

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Add a Record

Whether a user is allowed to add, edit or delete rows depends on the module permissions defined and the DotNetNuke security Roles to which a user belongs. To add a new record, select **Add New Record** from the module action menu (*Figure 9-3)* or click the **Add New Record** command link (*Figure 9-4)*.

Figure 9-3

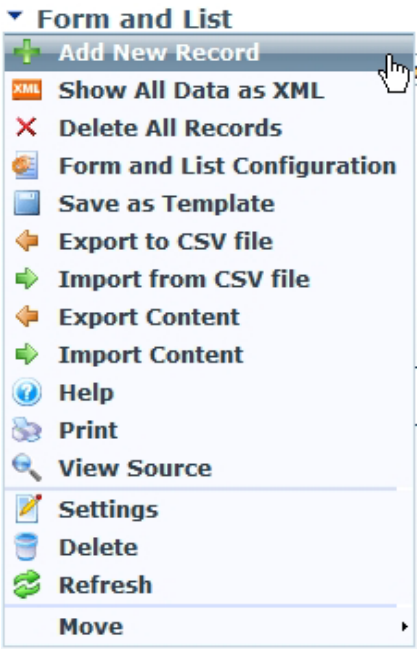


Figure 9-4



Quick Tip Add New Record is always available from the module action menu(▾) , but command link availability depends on the active skin.

Edit a Record

To Edit an existing record, **click the pencil** to the left of the desired row you wish to edit. The Edit form view is displayed.

Enter changes to the record. Depending on the column Type and Advanced Column Options set for a column, the user may be able to do one or more of the following:

- Select an option from a dropdown list;
- Edit HTML text using the WYSIWYG editor;
- Enter a Date using the popup calendar;

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

- Enter a URL (Uniform Resource Locator) to a page, image or download file. You may enter an external link, select a page from the current portal or select a file in an accessible Server folder. If you are allowed to, you can upload files onto the server.

- URLs and Email addresses must be formatted correctly.

You cannot edit the automatically provided values (author, date row created, last update), or any values generated via calculated columns. After you have finished editing, click **Update**. If you want to cancel without saving changes, click **Cancel**.

Delete a Record

To permanently delete the currently displayed record, click **Delete**. Once you click Delete and confirm OK, the record is immediately deleted! The Form and List module does not use the DotNetNuke or Windows Recycle Bin. You cannot restore a deleted Form and List record.

Delete All Records

To permanently delete all records of this module instance (even those, that are not displayed due to a filter applied), select **Delete All Records** from the action menu. Once you selected this item and confirm OK, all records are immediately deleted! The Form and List module does not use the recycle bin. You cannot restore deleted records.

Show Data as XML

This allows you to display the XML source of the table data, allowing you to view all available columns and associated content. (see example below)

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

```
<UserDefinedTable xmlns="DotNetNuke/UserDefinedTable">
  <Data>
    <UserDefinedRowId>25</UserDefinedRowId>
    <CreatedBy>XXXX</CreatedBy>
    <CreatedAt>2006-10-26T22:28:47.0000000+02:00</CreatedAt>
    <ChangedBy>XXXX</ChangedBy>
    <ChangedAt>2006-10-26T22:28:47.0000000+02:00</ChangedAt>
    <Caption>Konrad Zuse</Caption>
    <Short_x0020_Description>Konrad Zuse was a German engineer and
                        computer pioneer</Short x0020 Description>
    <EditLink>http://…/UserDefinedRowId/25/Default.aspx</EditLink>
    <CreatedAt UDT Value>10/26/2006 10:28 PM</CreatedAt UDT Value>
    <CreatedAt UDT Ticks>632974985270000000</CreatedAt UDT Ticks>
    <ChangedAt_UDT_Value>10/26/2006 10:28 PM</ChangedAt_UDT_Value>
    <ChangedAt_UDT_Ticks>632974985270000000</ChangedAt_UDT_Ticks>
  </Data>
  <Data>
    <UserDefinedRowId>26</UserDefinedRowId>
    <CreatedBy>XXXX</CreatedBy>
    <CreatedAt>2006-10-26T22:30:15.0000000+02:00</CreatedAt>
...
```

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Additional Information

The DotNetNuke Portal Application Framework is constantly being revised and improved. To ensure that you have the most recent version of the software and this document, please visit the DotNetNuke website at: http://www.dotnetnuke.com

The following websites provide helpful information about technologies and concepts related to DotNetNuke and referred to in this manual:

DotNetNuke
http://www.dotnetnuke.com

Microsoft® ASP.Net
http://www.asp.net

MSDN DataSet Expression (Calculated Columns)
http://msdn2.microsoft.com/en-us/library/system.data.datacolumn.expression(vs.71).aspx

Open Source
http://www.opensource.org/

Regular Expressions
http://en.wikipedia.org/wiki/Regular_expression

W3C Cascading Style Sheets, level 1
http://www.w3.org/TR/CSS1

XML and XSL Style Sheets
http://www.w3schools.com/xsl/

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Errors and Omissions

If you discover any errors or omissions in this user guide or in the module, if you wish to report functional issues about the module, or if you have a suggestion for improving the user guide or the module, please make an entry in the DotNetNuke Form and List – Issue Tracker – located at:

http://dnnfnl.codeplex.com/WorkItem/List.aspx

Support for Form and List module can be found in the DotNetNuke forum at:

http://www.dotnetnuke.com/Development/Forge/ModuleFormandList/tabid/877/Default.aspx

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.

Appendix A: Document History

Version	Date	Author(s)	Notes
5.1.2	February 19, 2010	Brett Conlan	UDT to Form and List; DNN 4.x to 5.x
2.0.8	July 19, 2008	Brett Conlan	3.5.1 Release (Final)
2.0.6	May 1, 2008	Sebastian Leupold	3.5.1 Release (Draft)
2.0.5	Apr 9, 2008	Brett Conlan	3.5.0 Release (Final)
2.0.3	Aug 30, 2007	Brett Conlan	3.4.0 Release (Final)
2.0.2	Aug 27, 2007	Stefan Cullmann	XSLT Generator/Editor (Draft)
1.0.5	Nov 26, 2006	Sebastian Leupold Stefan Cullmann	Added Changes for V 3.3.7
1.0.3	Oct 06, 2006	Sebastian Leupold	Added Changes for V 3.3.5 & 3.3.6
1.0.1	Jul. 16, 2006	Cathal Conolly	Revised for Release
1.0.0	Jul. 03, 2006	Sebastian Leupold Stefan Cullmann	Initial Version
0.0.1	Oct. 26, 2005	Nina Meiers	Template

Copyright © 2003-2010 DotNetNuke Inc. All Rights Reserved.