IndooGrid module for DNN

User Guide

Module version: 2.00.22

Document version: 2.1

Last updated: 22. 2. 2008

In d.o.o.

components.init.si

Information in this document 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.

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 In d.o.o. In d.o.o. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. You don't have any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2008 In d.o.o. All Rights Reserved.

Abstracts

This document describes IndooGrid (or Indoo SQL Grid) module, its features and their usage. It describes how to install, setup and use the module within DotNetNuke environment.

Content

Introduction

Thank you for purchasing IndooGrid data management module. IndooGrid module for DNN is powerful tool to display data and data manipulation. It supports very large database tables and views (over million records – as many as the database can handle it), using native client (MS SQL and Oracle) or ODBC client (MS SQL, Oracle, MySql, MS Access etc.). It also supports work with both local and remote databases and it is optimized to work as fast as possible.

IndooGrid module is truly an interface between database and DotNetNuke Framework. Although there are plenty of wizards included in the module, it offers full customization (SQL commands, JavaScript code etc.) to best suits your needs.

If you don't find solution in this document please post your question on our support forum:

http://components.init.si/OnlineHelp/Forum/tabid/165/Default.aspx

PLEASE READ THIS MANUAL CAREFULLY AND USE BUILT-IN APPLICATION HELP (by clicking on help icons) BEFORE YOU POST YOUR QUESTION.

Icons names used in this document:

🪄	Wizard icon
❓	Help icon
✏	Edit icon
✕	Delete icon
🔍	Detail view icon
➕	Insert icon
⬆	Up icon
⬇	Down icon
💾	Save icon

Requirements

Module requires DotNetNuke Framework versions 4.0.3 to 4.8.0. All other versions are not tested.

NOTE: Some hosting provider (for example goDaddy) offers only Medium trust only. Module includes features that cannot be implemented without Full trust so ask your hosting provider (if you have one) about his policy before you install the module otherwise some features may not work. For further information please read the »Using IndooGrid on Medium Trust Environment« section of this document.

Licensing information

30-day trial license

You are allowed to use module for testing purposes for trial period of 30 days after activation for each DNN installation. Module is fully functional and has enabled all the features available in full licensed product. If you decided to buy product, license can be applied any time. For information on activation please read the »Before the first use« section of this document.

Normal scenario is to use trial version on development machine and registered module on production server.

However, you can also register module on your development machine and copy complete DNN installation to the server if both (host) domain names are the same.

Module will be disabled after the trial period is over.

Full license

Buying the product you are licensed to use the module on up to 5 (five) parent or child portals within 1 (one) DotNetNuke installation. Please read the »Before the first use« section of this document for information on register program.

For detail license information please read the License.txt file which is included in your copy of the module (zip package).

Module is based on IndooGrid ASP:NET component. Component is included in module for use with this module only.

Using module for development purposes

If you're using module for development purposes, you don't have to activate or register. Module skips activation (license) check for all requests from IP address 127.0.0.1 (localhost). Although there's no time limitation for developers, license is still required for testing environment when other users access test server (from other network locations than local server).

Getting started

Overview

Before you start you should understand some basic module's principles.

All settings that apply to all modules on all pages within your DotNetNuke installation are managed on »Module Settings« page of any of the IndooGrid module on your pages. Changes made on this page will affect all instances of the module within your portal.

All settings that apply to specific module (instance of the module) are managed through »Table settings« page of each module. Changes made on this page will affect only current module.

Each module is bounded to (at least one) table in the database. By default module have no associated table (columns to edit or display). Only after you'll connect the module to appropriate table and you'll add columns to edit or display (using either »Automatically add all columns« wizard, »Add new view column« or »Add new edit column«) you'll see the results.

Table depending settings are configured on »Table settings« page while column settings can be configured two ways: some can be set on »Table settings« page using fields in each column rows while all column properties can be configured on each columns setting by clicking on pencil icon near each individual column.

Installing the module on DNN

If you're not familiar with installing the modules on DotNetNuke framework, please follow these steps:

1. Go to your portal and log in as host user.
2. Go to Host menu and choose »Module definition«.
3. Select »Install new module« or »Upload new module« in some of the older versions.
4. Browse to the module files (IndooGridDNN.zip) and click on »Install new module«. In some older version you have to »Add« file first and then click »Upload new module«.
5. Check the log and continue if no error occurs. Errors are red colored.

NOTE: You should also always make backup copy of complete DotNetNuke installation before installing the IndooGrid module.

Upgrading the module

You can always override previous or demo (trial) version of the module with newer version. You simply install new version as described above.

You can upgrade module within current major release without new registration. For example, if you bought module version 2.0.1 and registered it on your DNN installation, you can download any 2.0.x module (2.0.99 if exists) and install it without registration.

If you want to upgrade module to next major release (for example if you bought version 2.0.1 and you want to install module version 2.1.1) you will be asked to register module again after installation. If you bought module with included subscription (for example with 6-months subscription after the date of purchase) and subscription period isn’t over yet, you simply install new version and register it using same data as the first time. If you want to upgrade module and your subscription period expired, you can buy upgrade. After buying upgrade you should install new module version on current installation and enter new data (you’ve got when purchasing upgrade) in registration form. Once module is upgraded, old license key (you’ve got when you first time bought module) will not be valid anymore.

NOTE: Never uninstall old module before upgrading to new version. If you'll uninstall old or demo version before installing new module, you WILL LOOSE all your settings of all existing instances of IndooGrid module on your portal.

You should also always make backup copy of module settings before installing new version if you already have any instances of the IndooGrid module installed on your DotNetNuke portal.

Copying (moving) registered module

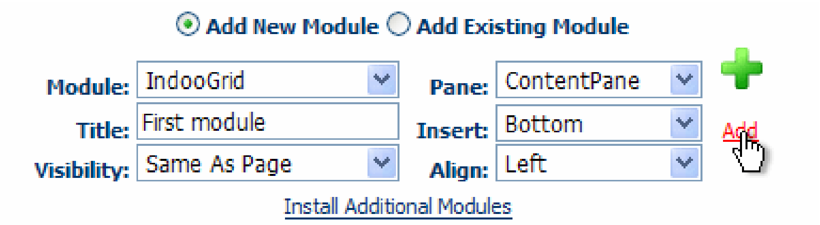
By default module license cannot be moved or copied after registration process is complete (first DotNetNuke URL - master portal URL also cannot be changed). Therefore it is most important that you register module only on DNN installation that module was bought for. Normally this is your production server – for testing and developing purposes you can use module on your local computer without registration as long as you access the module using localhost (127.0.0.1) address.

Adding module to the page

Now that we have installed module on the DNN portal, we can add instances of the module to the pages. Follow these steps:

1. Choose the IndooGrid module from list of modules in Control Panel.
2. Enter title of the module and other properties (visibility, position etc).

3. Click »Add« to add the module to the page.



● Add New Module ○ Add Existing Module

Module: IndooGrid

Title: First module

Visibility: Same As Page

Pane: ContentPane

Insert: Bottom

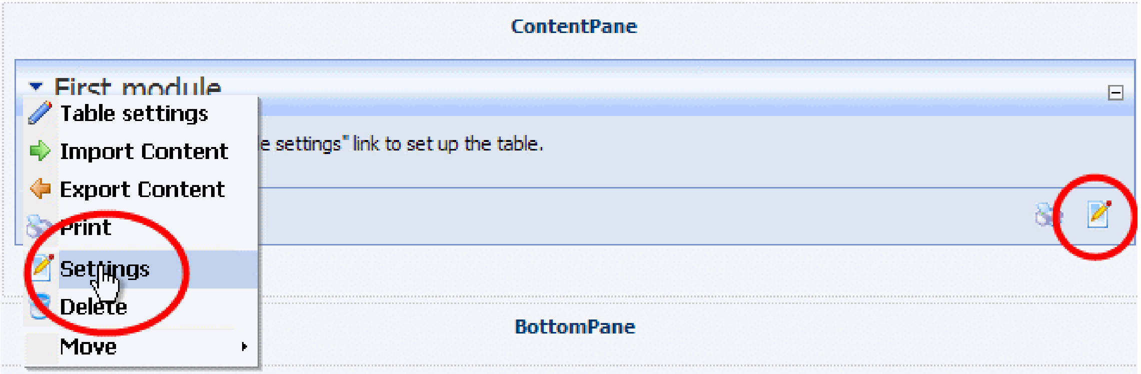
Align: Left

Picture 1: Adding module to the page

Module Settings (before the first use)

New instance of the module is now on the page. As the note says, you have either to register module or to activate trial period.

1. Click on the module »Settings«. Depending on your skin, the link can be in the right bottom or in action menu of the module in left upper corner.

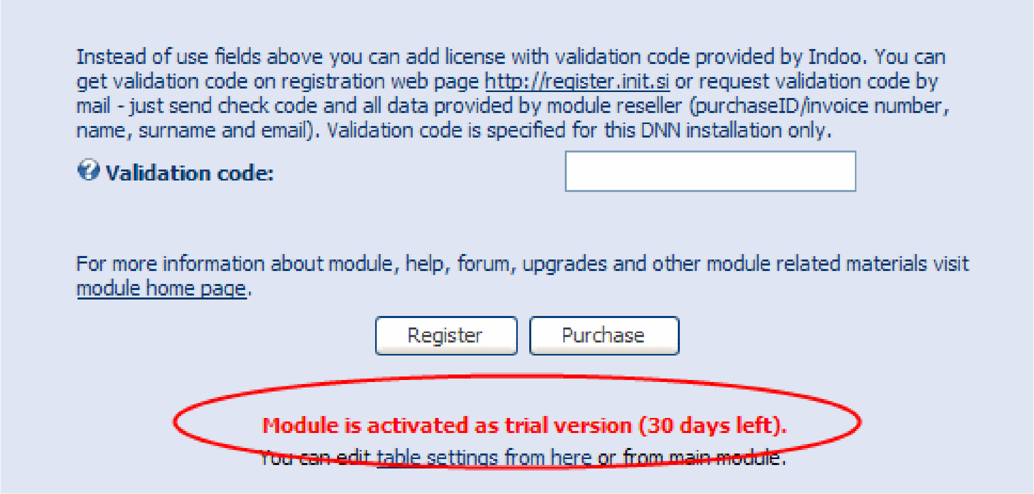


e settings" link to set up the table.

Picture 2: Module settings link can appear in action menu or at the bottom of the module (depending on your skin)

2. Scroll down and expand »IndooGrid settings« if they are collapsed.

If you want to activate trial click the »Activate trial license« button. Module will connect to registration server and trial license will be activated. You can use (full-featured) module for testing and demo purposes for 30 day. Expiration date is colored red.



Instead of use fields above you can add license with validation code provided by Indoo. You can get validation code on registration web page http://register.init.si or request validation code by mail - just send check code and all data provided by module reseller (purchaseID/invoice number, name, surname and email). Validation code is specified for this DNN installation only.

Validation code:

For more information about module, help, forum, upgrades and other module related materials visit module home page.

Module is activated as trial version (30 days left).

You can edit table settings from here or from main module.

Picture 3: Message displayed after successful trial activation

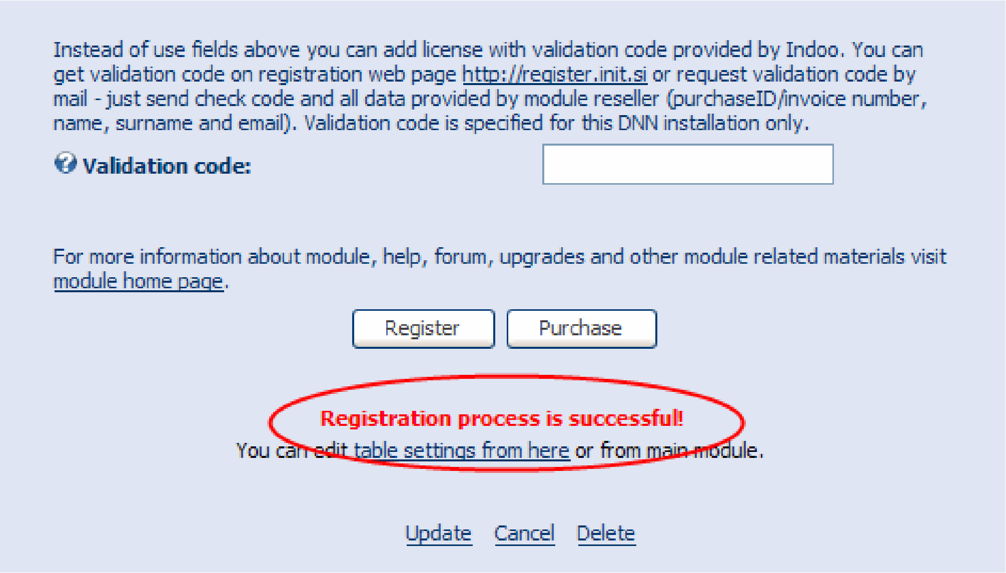
If you want to register module, follow this steps:

3. enter Purchase ID (or invoice ID) you got when you purchased the module on reseller site
4. enter first name you have entered to the reseller purchase form
5. enter last name you have entered to the reseller purchase form
6. enter email address you have entered to the reseller purchase form

Note that data entered should be exactly the same as entered when purchasing the module.

7. Click »Register« button.

The registration process will begin and the module will contact registration server at address http://register.init.si. When registration process ends, red colored message at the bottom will be displayed (either »success« or »failed« message). At the top of the module are now displayed number of portals with installed IndooGrid module and number of valid licenses.



Instead of use fields above you can add license with validation code provided by Indoo. You can get validation code on registration web page http://register.init.si or request validation code by mail - just send check code and all data provided by module reseller (purchaseID/invoice number, name, surname and email). Validation code is specified for this DNN installation only.

Validation code:

For more information about module, help, forum, upgrades and other module related materials visit module home page.

Registration process is successful!

You can edit table settings from here or from main module.

Picture 4: Message displayed after successful registration

If number of portal with installed modules exceeded number of licenses, first created portals within the quota will have registered modules and others will have trial licenses. You can buy additional license(s) any time using the module

After registration process copy all purchase data and check code from settings page to text file and print it on the paper for backup purposes.

NOTE: if registration process failed, you have to register module manually. Open http://register.init.si registration page in your browser and enter data as you have on registration form within module. You also have to enter check code which is specific for your DNN installation. Click Register and copy validation code to the field »Validation code« in module »Settings«. Before continuing, copy the validation code to text file and print it on the paper for backup purposes!

Click »Register button« (in the module »Settings«) and the process will finish offline.

Above the registration data you have three additional checkboxes. All these options will enable/disable some actions for administrator but some of the changes will also reflect on users. For further information about this option please read the »Managing access to the module (module features) » section in chapter »Security« of this document

Before continue click »Update« to save changes.

NOTE: Module is registered for use with specific DotNetNuke installation. For transferring license from one DotNetNuke installation to another contact us before transfer.

Table settings

All the options above have to be entered only first time you install the module. Now we have to set up the module by clicking the »Table settings« link on the bottom of the module.

There are four main sections on the »Table settings« page (from top to bottom).

In **basic settings** are settings to connect to appropriate database and select table or view. All options can be selected using wizards (icons). You can also add new table, modify table or delete table using wizards. Every table should also have unique identifier (unique key column) to identify records.

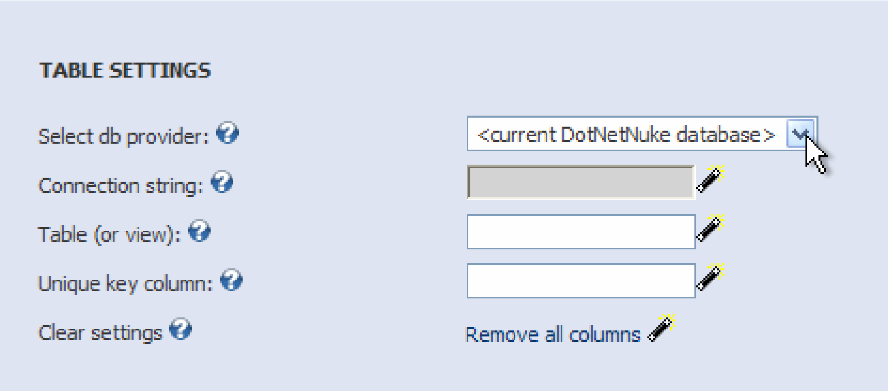
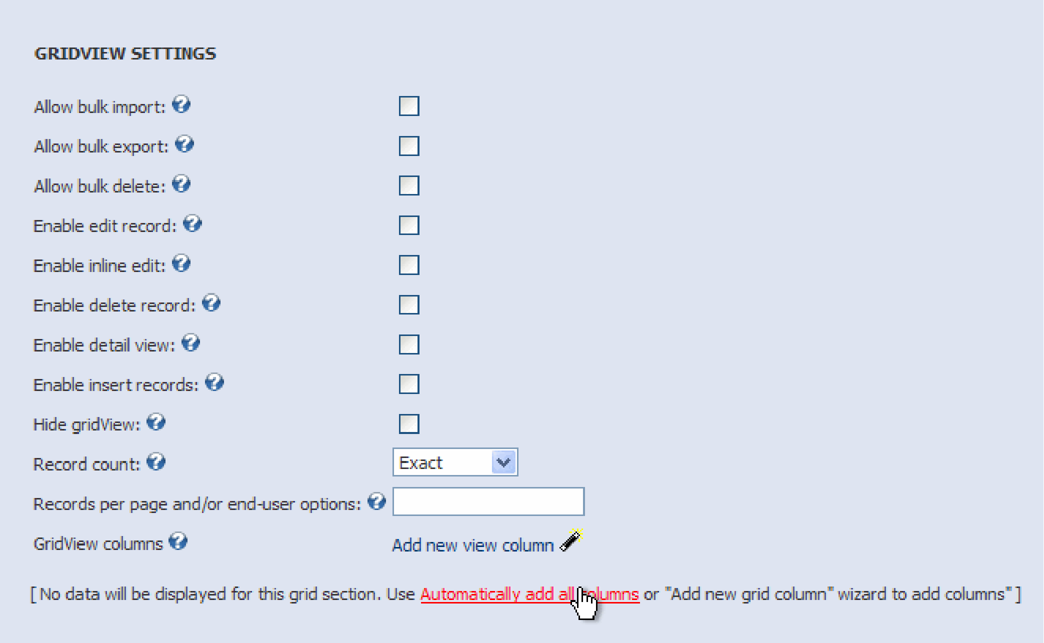


TABLE SETTINGS

Select db provider:	<current DotNetNuke database>
Connection string:	
Table (or view):	
Unique key column:	
Clear settings	Remove all columns

Picture 5: In basic settings we can define connection, select (create or edit) tables and define identifier

Using **gridView** settings we choose how to display data to the users. This section is used when working with listings, reports, galleries or any other displaying module's mode. At this stage there are no columns added yet.



GRIDVIEW SETTINGS

Allow bulk import: □

Allow bulk export: □

Allow bulk delete: □

Enable edit record: □

Enable inline edit: □

Enable delete record: □

Enable detail view: □

Enable insert records: □

Hide gridView: □

Record count: Exact

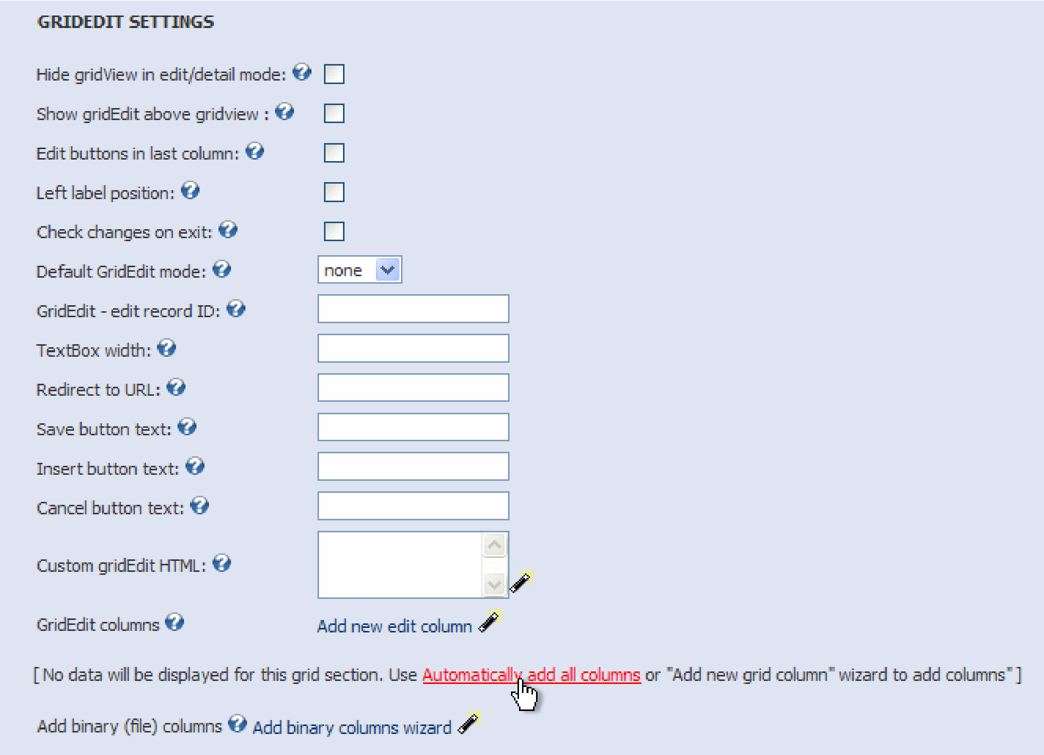
Records per page and/or end-user options:

GridView columns Add new view column

[No data will be displayed for this grid section. Use Automatically add all columns or "Add new grid column" wizard to add columns"]

Picture 6: We can allow user to edit, delete or add records in gridView settings (table-like view mode is enabled by default)

Using **gridEdit** settings we define how the users can edit or insert data.



GRIDEDIT SETTINGS

Hide gridView in edit/detail mode: □

Show gridEdit above gridview : □

Edit buttons in last column: □

Left label position: □

Check changes on exit: □

Default GridEdit mode: none

GridEdit - edit record ID:

TextBox width:

Redirect to URL:

Save button text:

Insert button text:

Cancel button text:

Custom gridEdit HTML:

GridEdit columns Add new edit column

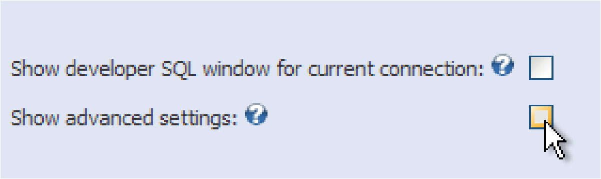
[No data will be displayed for this grid section. Use Automatically add all columns or "Add new grid column" wizard to add columns"]

Add binary (file) columns Add binary columns wizard

Picture 7: In gridEdit mode we can set options that will affect module in edit, detail view or insert mode

.

Clicking on checkbox »Advanced settings« new section **advanced option** will appear. In this section you can add (JavaScript or SQL) scripts or tokens to execute. You can also import/export table settings or use the SQL command window (by clicking on »Show developer SQL window for current connection«).



Show developer SQL window for current connection: □

Show advanced settings: □

Picture 8: Click on »Show advanced settings« checkbox to expand advanced options

Working with columns

As mentioned in previous sections, you can use »Automatically add all column« wizard to add all data table columns to the grid. Columns can be also added manually to both gridEdit and gridView section. You can also add same column more than once.

All grid columns are listed in two sections. In gridView section list are displayed columns that will appear in view mode of the data grid and in gridEdit section list are displayed columns that will appear when modifying or adding records.

Displaying data (gridView)

Using these settings you can specify how the user will see the grid (records displayed) and what actions can be executed by the user. The settings are divided in two sections: in first section you can set some global options (for example records displayed per page); in second section is list of columns added to grid. In the list you can set local settings for single column – for further information please read the »Working with columns« section of this document.

NOTE: All customization can be done during the setup steps above or any other time in the future.

Setting up the module for the first time

In this example we will display all DotNetNuke users from table Users (which already exists in default DotNetNuke database) using default options.

1. Click on »Table Settings« link.
2. Select »current DotNetNuke database« from Select db provider field (default option).
3. Either enter the database name »Users« in »Table (or view)« field or click on wizard icon near the field.

If you have clicked wizard, new page will appear.

4. Select »Select existing table« radio button and click »Next«.
5. On next screen select »Users« in the »Select table« list box.
6. Click »Next«. You will be redirected to the »Table Settings« page.
7. Either enter »UserID« in »Unique key column« or click on wizard icon near the field.
8. Click on »Automatically add all columns link« or on the wizard icon near the link. All the columns (except the unique identifier which was chosen in previous step and binary columns) from the selected data table will be automatically added to both gridView and gridEdit sections.

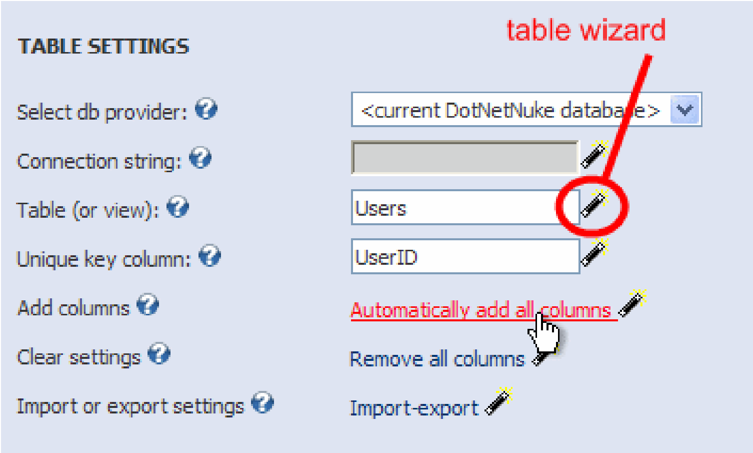


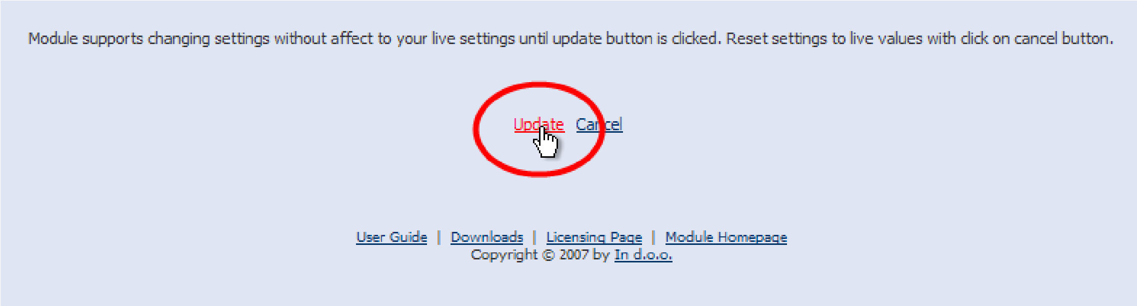
TABLE SETTINGS

table wizard

Select db provider:	<current DotNetNuke database>
Connection string:	
Table (or view):	Users
Unique key column:	UserID
Add columns	Automatically add all columns
Clear settings	Remove all columns
Import or export settings	Import-export

Picture 9: Built-in wizards allow us to go trough step-by-step procedures (in this example we can create, modify and select tables or views)

9. Scroll down and click »Update« button.

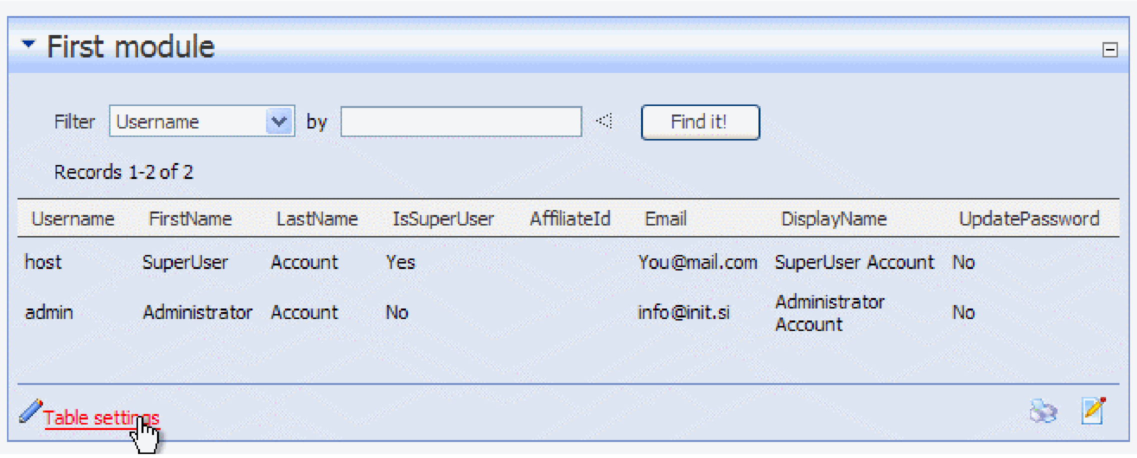


Module supports changing settings without affect to your live settings until update button is clicked. Reset settings to live values with click on cancel button.

Update Cancel
User Guide | Downloads | Licensing Page | Module Homepage
Copyright © 2007 by In d.o.o.

Picture 10: All changes will not be applied until changes are confirmed using »Update« command.

Now all the records from table Users with all the columns (except UserID, which is unique identifier of the table) are displayed on the screen.



Filter Username by

Records 1-2 of 2

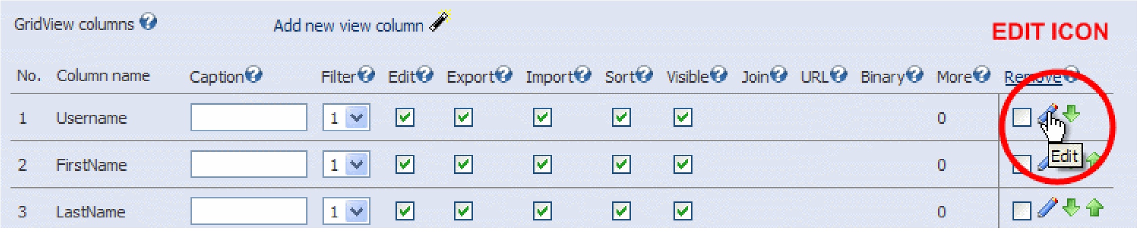
Username	FirstName	LastName	IsSuperUser	AffiliateId	Email	DisplayName	UpdatePassword
host	SuperUser	Account	Yes		You@mail.com	SuperUser Account	No
admin	Administrator	Account	No		info@init.si	Administrator Account	No

Picture 11: Simple example of Users table displayed in view mode (all settings were configured using gridView options)

Working with gridView columns

NOTE: All changes made in these sections will not change database table. Using columns and column properties as described in this section only refers on how the data will be show to the user within gridView object. For modifying data table columns and data table column properties in the database, please read the »Table (or view)« chapter.

Some gridView column properties can be modified directly on »Table settings« page – these changes are confirmed by clicking »Update” at the bottom of the »Table settings« page. All this properties and all other properties not listed in this main list can be reached by clicking on edit icon of each gridView column.



GridView columns   Add new view column

No.	Column name	Caption	Filter	Edit	Export	Import	Sort	Visible	Join	URL	Binary	More	Remove
1	Username		1	☑	☑	☑	☑	☑				0	□
2	FirstName		1	☑	☑	☑	☑	☑				0	□
3	LastName		1	☑	☑	☑	☑	☑				0	□

Picture 12: Properties of every column can be configured by clicking on edit icon (this is edit icon of first list – gridView section). The module will guide us through columns settings wizard

To delete specified column check the »Delete« checkbox near the column you want to delete and click »Delete« link in the header of the gridView list.

Use up icon and down icon to change the position of the column in the data grid (columns will be displayed from left to right). Click twice on up or down icon to define exactly position of the column (enter the position in window that it will open and click »Ok«).

Clicking »Add new view column« or the wizard icon near the link you'll go through same steps (for the new column) as if you click edit icon (for the existing column).

In first step you can set basic settings of the column. For example, if you enter the »Column caption«, the text will be displayed as column header (after you'll confirm changes made by clicking on »Update« button at the bottom of the grid). If you leave this field blank, column name will be displayed as column header.

Adding, editing and deleting data (gridEdit)

In this example we will show how to create simple form to modify data. By modifying data it is meant edit, insert or delete records.

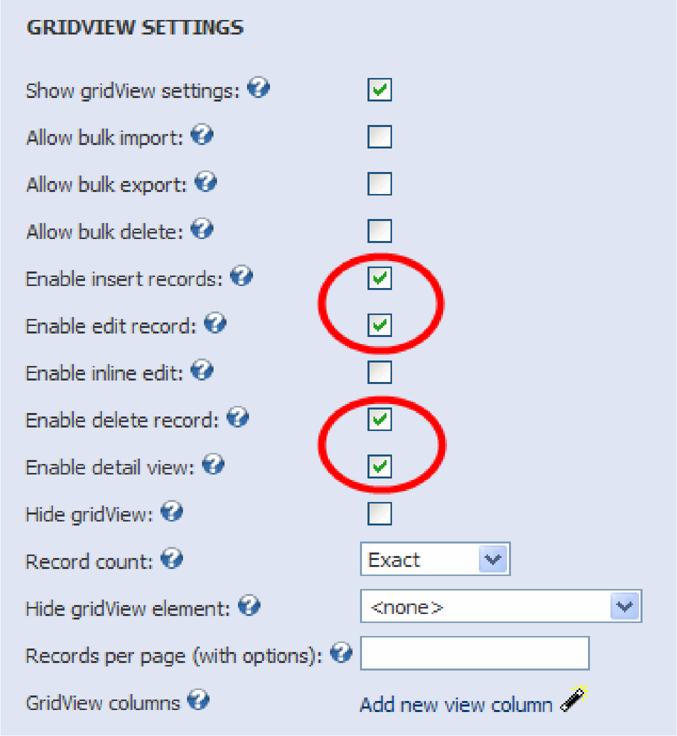
As mentioned above, IndooGrid »Table settings« are divided in four main sections: »Table options« section, gridView section, gridEdit section and »Advanced options« section. GridView defines displaying data but also allows adding built-in icons to manipulate data and custom action buttons. These options are in gridView section simply because of enabling to display action link (icon or button) for each single record. Insert icon (button) is displayed at the bottom of the grid while edit, delete and detail view icons and custom links (buttons) are displayed in each row.

All other options considering modifying data tables are in gridEdit section of the »Table options«.

To enable modifying data to end user, use following steps (you can use example from »Displaying data« using Users table):

Click »Table settings«.

1. In gridView section check the checkboxes:
2. »Allow insert« to enable inserts of new records. The plus icon will be displayed at the end of the grid.
3. »Allow edit« to enable editing records. Edit icon will be displayed in the beginning of each row.
4. »Allow delete« to enable deleting records. Red (cross) icon will be displayed in the beginning of each row.
5. »Allow detail view« to enable detail view of records. Magnifying glass icon will be displayed in the beginning of each row.



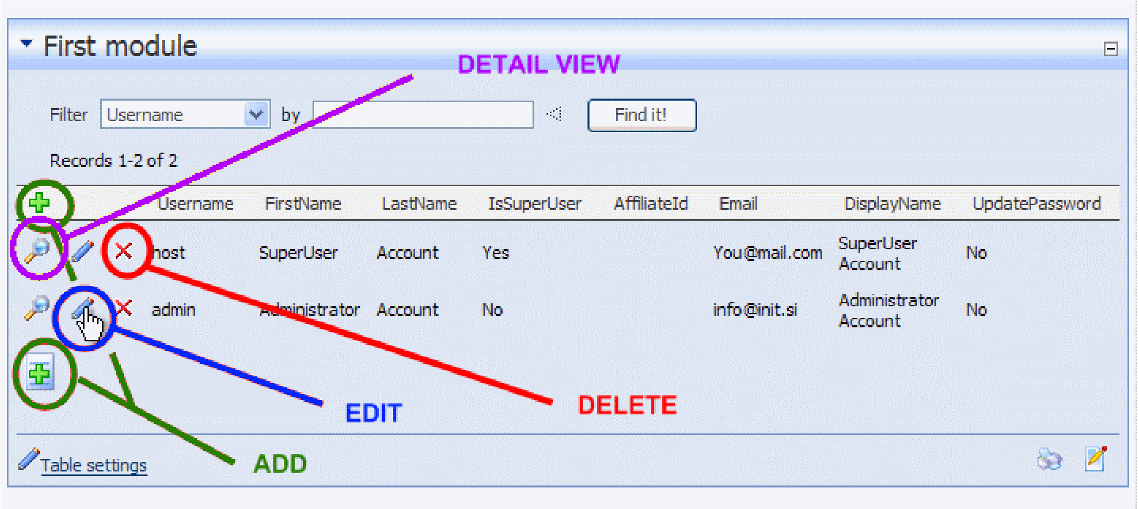
GRIDVIEW SETTINGS

Show gridView settings:	☑
Allow bulk import:	□
Allow bulk export:	□
Allow bulk delete:	□
Enable insert records:	☑
Enable edit record:	☑
Enable inline edit:	□
Enable delete record:	☑
Enable detail view:	☑
Hide gridView:	□
Record count:	Exact
Hide gridView element:	<none>
Records per page (with options):	
GridView columns	Add new view column

Picture 13: Enable different modes (insert, edit, detail view) and allow users to delete records by checking the appropriate checkboxes

6. Check »Left label position« to show field labels on the left instead on the top (you do not have to enable this option – this is just an example).
7. Click »Update«.

As described, we can now see icons in each record row and insert record icon at the bottom (click on icon to enter edit / detail / insert mode). It is important to notice that fields displayed in edit (or detail or insert mode) can differs from fields displayed in view mode, That's why there are two column lists (gridView and gridEdit) in »Table settings«.

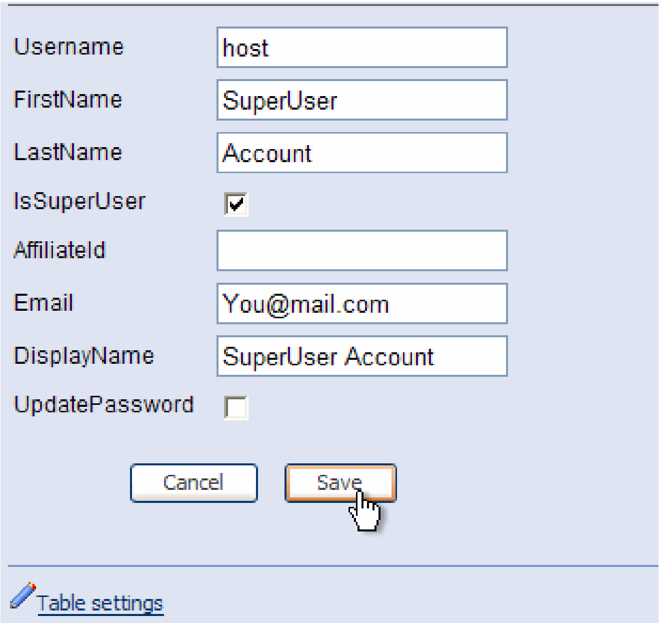


Records 1-2 of 2

Username	FirstName	LastName	IsSuperUser	AffiliateId	Email	DisplayName	UpdatePassword
host	SuperUser	Account	Yes		You@mail.com	SuperUser Account	No
admin	Administrator	Account	No		info@init.si	Administrator Account	No

Picture 14: Use icons to enter detail view, insert or edit mode for current table

Clicking on edit icon we can enter edit mode.



Username	host
FirstName	SuperUser
LastName	Account
IsSuperUser	☑
AffiliateId	
Email	You@mail.com
DisplayName	SuperUser Account
UpdatePassword	□

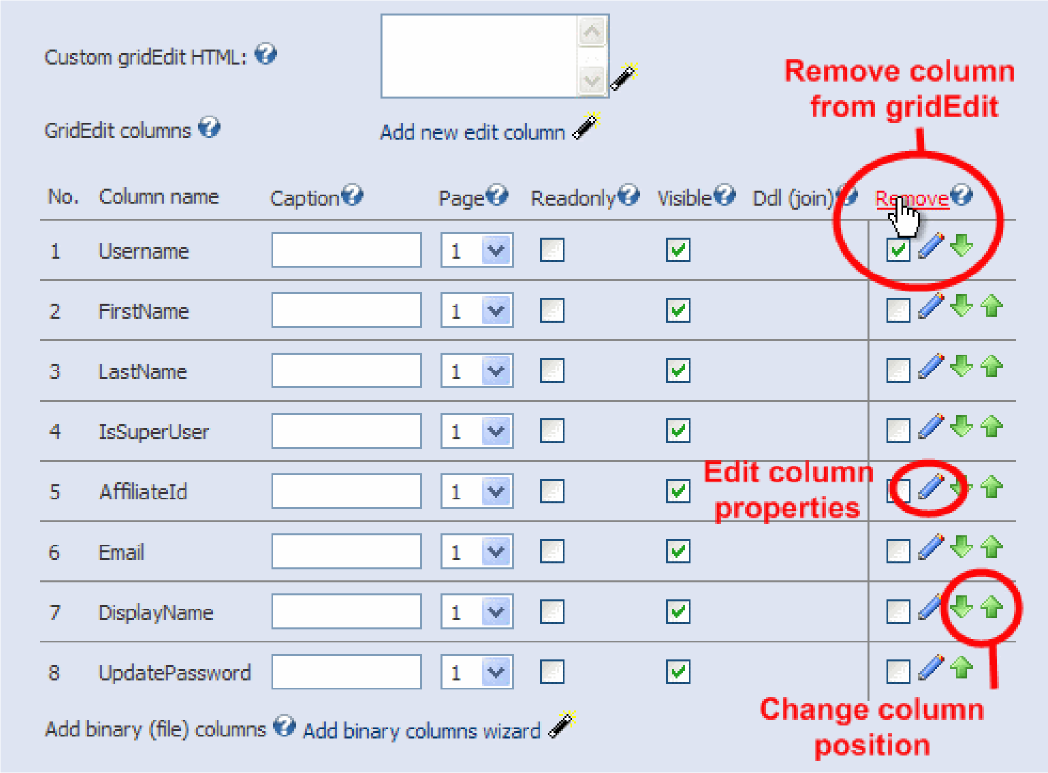
Picture 15: Standard (left labeled) edit form – you can enable edit mode by checking the »Enable edit record« checkbox.

Working with gridEdit columns

Some gridEdit column properties can be modified directly on »Table settings« page – these changes are confirmed by clicking »Update« command at the bottom of the »Table settings« page. All this properties and all other properties not listed in this main list can be reached by clicking on edit icon of each gridEdit column.

To delete specified column check the »Delete« checkbox near the column you want to delete and click »Delete« link in the header of the gridEdit list.

Use up icon and down icon to change the position of the column in the data grid (columns will be displayed from top to bottom). Click twice on up or down icon to define exactly position of the column (enter the position in window that it will open and click »Ok«). You can also use gridEdit Custom HTML to define how gridEdit will be displayed.



Custom gridEdit HTML:

GridEdit columns — Add new edit column

No.	Column name	Caption	Page	Readonly	Visible	Ddl (join)	Remove
1	Username		1	□	☑		☑
2	FirstName		1	□	☑		□
3	LastName		1	□	☑		□
4	IsSuperUser		1	□	☑		□
5	AffiliateId		1	□	☑		□
6	Email		1	□	☑		□
7	DisplayName		1	□	☑		□
8	UpdatePassword		1	□	☑		□

Add binary (file) columns — Add binary columns wizard

Remove column from gridEdit

Edit column properties

Change column position

Picture 16: Select columns to remove (Username in this example) and »click Remove« link to remove columns from gridEdit. Click on »Edit« icon to open column settings wizard. Click on »Arrow« icons to change column position within the grid.

Clicking »Add new edit column« or the wizard icon near the link you'll go through same steps (for the new column) as if you click edit icon (for the existing column).

Samples

Templates

You can try any of the samples bellow by downloading templates from our home page (http://components.init.si). Before downloading templates you should read the »Templates (Import-export)« section.

Reports

One of basic module feature is the ability to build powerful reports. Built-in filtering and sorting also offer end user power to customize their reports on-the-fly.

For purposes of this example we will use standard DotNetNuke SiteLog table. Built-in DotNetNuke report for this table is available on your DNN installation on Admin => SiteLog page. If you don't see any data on that page, you should first enable SiteLog on page Host => HostSettings => OtherSettings => SiteLogHistory(Days) and on page Admin => SiteSettings => AdvancedSettings => HostSettings => SiteLogHistory(days). Enter 999 in both textboxes and click »Update« to confirm changes.

SiteLog table stores requests to all pages within your DotNetNuke portal. It allows us to view page traffic and traffic sources.

After adding new IndooGrid to the page follow this steps:

1. Click on »Table Settings« link.
2. Use wizard to select »SiteLog« table.
3. Use wizard to select »id« field of SiteLog table.
4. Click on »Automatically add all columns« link.
5. Check »Allow bulk export« checkbox.
6. Check » Hide gridView in edit/detail mode« checkbox.
7. Scroll down and click »Update«.

We made very simple report which we'll modify to show results transparent and useful. First we'll remove the time part from the DateTime field and make some predefined filters.

8. Click on »Table settings«.
9. Click on edit icon of »DateTime« column in gridView column list.
10. Enter »Date« in »Column caption« textbox.
11. Enter »d;en-us« in »Format data« textbox.
12. Scroll down and click »Next«.
13. Select »date span« in »Date picker« dropdown list.

14. Click »Next«.

You were returned to »Table settings« page. In this step we will disable fields we will not use in this example:

15. Uncheck the »Visible« checkbox of »PortalId« column.
16. Uncheck the »Visible« checkbox of »UserAgent« column.
17. Uncheck the »Visible« checkbox of »URL« column.
18. Uncheck the »Visible« checkbox of »UserHostName« column.
19. Uncheck the »Visible« checkbox of »AffiliateId« column.

Next step is joining the UserId field on Users table. UserID field of SiteLog is integer value which doesn't tell us user's name but just integer value stored in SiteLog table. Therefore we will connect this field to the Users table and retrieve the Username from the Users table instead of UserID.

20. Click on edit icon of »UserID« column in gridView column list.
21. Enter »Username« in »Column caption«.
22. Check »Join another table data on this column« checkbox.
23. Click »Next«.
24. Select »Users« table from »Join table« dropdown list.
25. Click »Next«.
26. Select »UserID« from »Join table on column« dropdown list.
27. Select »Username« from »Select column contents« dropdown list.
28. Click »Next«.
29. Scroll down and click »Update« to confirm changes.

Now we will add advance filters with predefined values based on manually entered list (values can be pulled from existing table too, please take a look at »Filter« feature of »Module reference« section of this document). Note that instead of adding filters you can use existing gridView columns to set filters – we are adding filter (and not modifying existing columns) just to show the possibilities of the module.

30. Click on »Table settings« link.
31. Click »Add new view column« link.
32. Select »DateTime« from »Column name« dropdown list.
33. Enter »Date« in »Column caption« textbox.
34. Enter »d;en-us« in »Format data« textbox.
35. Select »5 – always on as dropdown list (excluded from general filter list)«
36. Uncheck »Is column visible« checkbox.
37. Click »Next«.
38. Select »Values from list« in »Select list source« dropdown list.
39. Check »Auto execute on change« checkbox.
40. Click »Next«.
41. Copy following code to »List values and text« textarea:

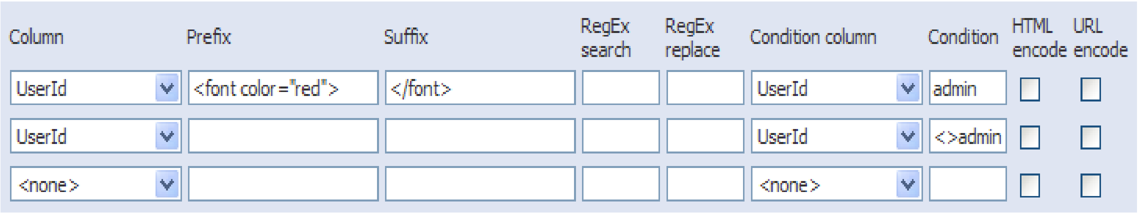
```
>=[indooGrid:currentDate];Today
>=1/1/2007 <1.1.2008;Year 2007
<>00;All time
```

42. Uncheck the »Add ALL option« checkbox.
43. Click »Next«.
44. Scroll down and click »Update«.

Try to select different values from »Date« dropdown list. You'll note that selecting the value automatically applies filter to the grid. You'll also notice that every time you'll selet the »Today« or »All time« value, record count is increased for 1. That's because every time you select the value, new log entry is created.

The next step is to make some design changes (please take a look at »CSS styling« section of this document to find out more about designing the grid). We will color »Username« column in red if username is »admin«.

45. Click on »Table settings«.
46. Click on edit icon of »UserID« column.
47. Select »1 - configurable« in »Show data from more columns (custom HTML)« dropdown list.
48. Click »Next«.
49. Click »Next«.
50. Click »Next«.
51. Enter the code as shown bellow:



Column	Prefix	Suffix	RegEx search	RegEx replace	Condition column	Condition	HTML encode	URL encode
UserId	<font color="red">	</font>			UserId	admin	□	□
UserId					UserId	<>admin	□	□
<none>					<none>		□	□

Picture 17: Custom HTML (option 1 – configurable) enables design modifications, link building and more using conditions

52. Click »Next«.
53. Scroll down and click »Update«.

We will now show the use of WYSIWYG editor for gridView columns. In this example we will build link for »TabId« column in way that by clicking on the link we will be redirected to the appropriate page. We will also add another view column to display the page name instead of page ID.
Note that editor's drag and drop is available in Internet explorer only. Click twice to add fields if you're using different browser.

54. Click on »Table settings«.
55. Click on »Add new view column« link.
56. Select »TabId« from »Column name« dropdown list.

57. Enter »PageName« in »Column caption« textbox.
58. Check »Join another table data on this column« checkbox.
59. Click »Next«.
60. Select »Tabs« from »Join table« dropdown list.
61. Click »Next«.
62. Select »TabId« from »Join table on column« dropdown list.
63. Select »TabName« from »Select column contents« dropdown list.
64. Click Next.

After adding the new column we will build a link. It doesn’t matter which column will be base for our link (Page or PageName) so we will use the first one.

65. Uncheck »Visible« checkbox of »TabId« (PageName) column.
66. Click on edit icon of »TabId« (the one without »PageName« caption) column.
67. Enter »Page« in »Column caption« field.
68. Select »2 - editor« in »Show data from more columns (custom HTML)« dropdown list.
69. Click »Next«.
70. Click on »PageName« button in the field list on the right of the screen and drag it to the editor. Release mouse button.
71. Click on Hyperlink icon.
72. Select »http« and enter »./Default.aspx?tabid=[columnField:TabId]« in URL field.
73. Click »Ok«.
74. Click »Next«.
75. Scroll down and click »Update«.

In this part we will add some additional text and calculated values to the grid.

76. Click on »Table settings«.
77. Click on »Show advanced settings« checkbox.
78. Enter following code in »Module header text« textarea:

```
<font face=»Georgia, Serif«> This is custom text using custom
font</font>
```

79. Enter following code in »Module footer text« textarea.

```
UserID with max. entries:<span style=»color: red«> <command
sql=»SELECT UserId, COUNT(*) as Entries FROM SiteLog GROUP BY UserID
ORDER BY Entries DESC«></command></span><BR>

Number of entries by this user (in the database):<span style=»color:
blue«> <command sql=»SELECT COUNT(*) as Entries, UserId FROM SiteLog
GROUP BY UserID ORDER BY Entries DESC«></command></span><BR>

Number of entries by this user (in the grid):<span style=»color:
green«> <command sql=»SELECT COUNT(*) as Entries FROM SiteLog
[indoogrid:gridViewWhereClause] GROUP BY UserID ORDER BY Entries
DESC«></command></span><BR>
```

80. Scroll down and click »Update«.

I first row we show the UserID of user who has maximum number of entries in the database. In second row we are displaying the number of entries of this user. In last row we are also showing number of entries made by this user but only those who are contained in the grid (with all grid filters applied).

In module footer we have used two special tokens to retrieve values: <command> token and `[indoogrid:gridViewWhereClause]` token. For more information on how to use tokens please read the »Working with tokens« section of this document.

In last part we will just define default sort order. Normally default sort order can be defined by checking »Default sort order« checkbox of the column we want to be sorted by default. In this case we will manually define sort order. The grid will be sorted descending by Date.

81. Click on »Table settings« link.
82. Enter »DateTime DESC« in »GridView ORDER BY part of the sql statement« textbox.
83. Scroll down and click »Update«.

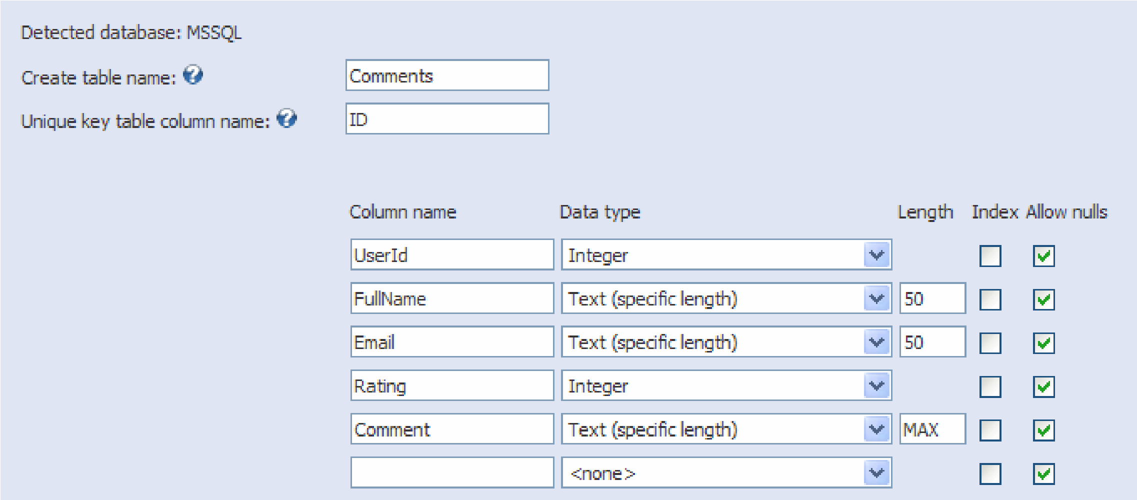
Forms

In this example we will show how to create simple custom form. It will allow users to comment our site. First, we will create new table named Comments (if you want to use one of existing tables, enter table name in »Table (or view)« field and go directly to step 11).

1. Click on »Table Settings« link.
2. Click on »Table (or view)« wizard icon.
3. Select »Create table« radio button and click »Next«.
4. Enter »Comments« in »Create table name« field.
5. Enter »ID« in »Unique key column« field.
6. In first row of the column list enter »userID« in »Column name« field and select »Integer« in list box.
7. In second row of the column list enter »fullName« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
8. In third row of the column list enter »email« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
9. In fourth row of the column list enter »rating« in »Column name« field and select »Integer« in list box.

10. In fifth row of the column list enter »comment« in »Column name« field and select »Text (specific length)« in list box.

You should see picture similar to this one:



Detected database: MSSQL

Create table name: Comments

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
UserId	Integer		□	☑
FullName	Text (specific length)	50	□	☑
Email	Text (specific length)	50	□	☑
Rating	Integer		□	☑
Comment	Text (specific length)	MAX	□	☑
	<none>		□	☑

Picture 18: Create table wizard offer simple interface for creating new tables

11. Click »Next«. You will be returned to »Table settings« page.
12. Click wizard icon near »Unique key column« field. ID will appear in the field.
13. Click »Automatically add all columns« link.
14. Check »Enable insert records« checkbox.
15. Select »insert« »Default GridEdit mode« dropdown list.
16. Scroll down and click »Update«.

This is very simple example to show how to build form. You'll see that form already has built-in validators depending on the database type – if you enter some text in »user« or »rating« field, error will appear when clicking on »Insert« button.

In real world such kind of form isn't something. Now we will configure the form to meet real-life requirements (there are two columns lists - gridView and gridEdit - on »Table Settings« page, in this example we will set gridEdit which is second columns list from the top).

17. Click on »Table Settings« link.
18. Click edit icon near »UserID« in gridEdit section field.
19. Uncheck »Visible« checkbox.
20. Click »Next«.
21. Enter »[User:UserID]« in »Field value« field.
22. Click »Next«.

We defined that this field will not be visible to the end users and that it will have automated value (DotNetNuke userID if user is logged in) inserted when posting new comment.

23. Enter »Your name« in »Caption« field of the »Fullname« gridEdit column.

Here we set caption displayed to the user to Your full name.

24. Click edit icon near »email« field in gridEdit settings.
25. Enter »Your email address« in »Caption« field.
26. Click »Next« twice.
27. Check »Custom validator« checkbox.
28. Click »Next«.
29. Select »Required« from first »Validator« dropdownlist.
30. Select »mail (check syntax)« from second »Validator« dropdownlist.
31. Click »Next«.

What have we done is prevent user to enter invalid mail address in »Your email address« textbox and we do not allow user to not enter mail at all.

32. Click edit icon near »rating« field in gridEdit section.
33. Enter »Rating« in »Caption« field.
34. Click »Next«.
35. Check »Drop down list or radio button list (join for view mode):« checkbox.
36. Click »Next«.
37. Select »Values from list« from »Select list source« dropdown list.
38. Click »Next«.
39. Enter following lines in »List values and text« textarea box.

```
1;Your page is crap (1/5)
2;Could be worse (2/5)
3;Average (3/5)
4;I like it (4/5)
5;You are my hero (5/5)
```

40. Check »Sort by value« checkbox.
41. Click »Next«.

In this part we will specify rating values user can choose from. For example if he will choose »You are my hero (5/5)«, value 5 will be saved in this field when posting new comment. We could also use another table and link values from our form to that table (you can try to create new table ratings with those five records).

42. Click edit icon near »comment« field in gridEdit section.
43. Enter »Comment« in »Caption« field.
44. Click »Next«.
45. Enter »400« in »Width« textbox.
46. Click »Next«.
47. Enter »6« in »Multiline« textbox.
48. Click »Next«.

49. Check »Hide gridView in edit/detail mode« checkbox (you can also leave this option to monitor results).
50. Enter »Submit« in »Insert button text« textbox.
51. Scroll down and click »Update«.

In last step we will add »Thank you for your comment.« text after valid data input. First we will create new page using DotNetNuke. On this page we will add text »Thank you for your comment«. Then we will click on that page and copy link from browser address bar.

52. Click on »Table Settings« link.
53. Paste the link (of »Thank you« page) you've copied from browser address bar in »Redirect to URL« textbox. Instead of entering whole link you can put following code (and replace myID with the ID of the »Thank you« page) into this field: »./Default.aspx?tabid=myID«.
54. Scroll down and click »Update«.

Try some options, for example enter invalid mail address or post comment as different user.

Questionnaires

The form above is simple one-step form to enter data. However, it is often required to guide user through multiple steps and allow him enter only data that are needed. Such kind of form (multiple page form for surveys, questionnaires or wizards) has at least two positive effects:

- User complete the data entry quicker and easier (user experience is much better).
- User cannot enter data that are not allowed.

Wherever you need survey, questionnaire or wizard you should build multiple page form. To show how to create survey we'll use the Comments table from previous example. First we will add additional fields to the table: subscriber (true or false) and phone (text). First page of our survey will be the same as it is made in previous example. On second page we will have two different options:

- If the user will choose ratings 3, 4 or 5 in first step, we will offer him checkbox to receive News (because he like our page, doesn't he).
- If the user will choose ratings 1 or 2 in first step, we will ask him for his phone number (and we will try to talk to him later and ask him what is wrong with him).

We already show how to create table, now we will show how to modify (alter) it.

1. Click on »Table settings« link.
2. Click on »Table (or view)« wizard icon.
3. Select »ALTER table« radio button and click »Next«.
4. Select »Comments« from »Select table« dropdown list.
5. Click »Next«.
6. Select »Add column« from »Select action« dropdown list.
7. Enter »Subscriber« in »Column name« field.
8. Select »Boolean« from »Column type« dropdown list.
9. Click »Alter table« button.
10. Enter »Subscriber« in »Column name« field.
11. Select »Add column« from »Select action« dropdown list.
12. Enter »Phone« in »Column name« field.
13. Select »Text« from »Column type« dropdown list. Enter »50« instead of MAX.
14. Click »Alter table« button.
15. Click »Back« until you'll be returned to the »Table settings« page.

The new columns were added to the data table but not yet to the grid.

16. Click on »Add new edit column« link (or wizard icon near the link).
17. Select »Subscriber« from »Column name« dropdown list.
18. Enter »Would you like to receive monthly Newsletter?«
19. Select »2« from »Multiple page« dropdown list.

Multiple page condition field will become enabled. Here we will enter condition to show this second page. Our condition is very simple, for more information on how to build conditions please read the »Multiple page condition« section in »Module Reference« chapter.

20. Enter »Rating=3 OR Rating=4 OR Rating=5« in »Multiple page condition« textbox.
21. Click »Next«.
22. Enter »true« in »Field value« textbox.
23. Check the »Use field value as default value only« checkbox.
24. Click »Next«.

In steps 22 and 23 we defined the default value for the field. If we don't check the checkbox, the value will be always applied (user cannot change value). That's what we have done in previous example (simple form).

Using same logic we will now add the third page. Altough we desay third, it will open as second or it will not open at all (so basically it is acting in same way as the page we already configured – the one with the »Send Newsletter« checkbox).

25. Click on »Add new edit column« link (or wizard icon near the link).
26. Select »Phone« from »Column name« dropdown list.
27. Enter »Please enter your phone number«
28. Select »2« from »Multiple page« dropdown list.
29. Enter »Rating=1 OR Rating=2« in »Multiple page condition« textbox.
30. Click »Next«.

31. Click »Next«.
32. Click »Next«.
33. Scroll down and click »Update«.

Now try to select random Rating values – if the value is lower than 3, second page will not be the same as if the value is greater than 2.

Data management

In previous examples (Forms and Surveys) we allow users to insert data. We are administrators so now we would like to manage this data. We will create module that will allow us to manage data.

After adding new IndooGrid to the page follow this steps:

1. Click on »Table settings« link.
2. Use wizard to select »Comments« table.
3. Use wizard to select »ID« field of the »Comments« table.
4. Click on »Automatically add all columns« link.
5. Check »Allow bulk export« checkbox.
6. Check »Enable insert« checkbox.
7. Check » Enable edit« checkbox.
8. Check » Enable delete« checkbox.
9. Check » Enable detail view« checkbox.
10. Check » Hide gridView in edit/detail mode » checkbox.
11. Check » Edit buttons in last column » checkbox.
12. Check » Left label position » checkbox.
13. Enter »10, 20, 50« in Records per page »(with options)« textbox.
14. Scroll down and click »Update«.

All records are now listed in the grid. By selecting records per page value from »page« dropdown list we can define how many records will be displayed on the page. By clicking on detail view icon details for current page will be displayed. By clicking on the insert icon similar form will appear as in »Forms« sample.

By clicking on edit icon of record we can edit record. When editing data (auto) validators are enabled by default.

Inline data management (Contacts)

Beside standard data management module enables additional option to edit data. This option enables you to edit multiple records on the same page (you can think about it as of AJAX feature although it is not AJAX because of performance capabilities).

For purposes of this example we will create two new tables: Contacts(id, text first_name, text_lastname, date birth_date, text email_address, text phone, integer company, text job_title) and Companies (id, text name, text address, integer zip_code, text city).

1. Click on »Table (or view)« wizard icon.
2. Select »Create table« radio button and click »Next«.
3. Enter »Contacts« in »Create table name« field.
4. Enter »id« in »Unique key table column name«.
5. In first row of the column list enter »first_name« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
6. In second row of the column list enter »last_name« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
7. In third row of the column list enter »birth_date » in »Column name« field and select »Date/Time« in list box.
8. In second row of the column list enter »email_address« in »Column name« field and select »Text (specific length)«.
9. In second row of the column list enter »phone« in »Column name« field and select »Text (specific length)«.
10. In second row of the column list enter »company« in »Column name« field and select »Integer« in list box.
11. In second row of the column list enter »job_title« in »Column name« field and select »Text (specific length)«.
12. Click »Next«. You will be returned to »Table settings« page.

We made table Contacts - following same procedure we can make table Company. After creating second table, create new page in DotNetNuke and add new instance of IndooGrid module to the page.

13. Click »Table settings«.
14. Enter »Contacts« in »Table (or view)« field.
15. Click on wizard icon near »Unique key column«.
16. Click on »Automatically add all columns link«.
17. Check following checkboxes: »Allow bulk import«, »Allow bulk export«, »Allow bulk delete«, »Enable edit record«, »Enable delete record«, »Enable Detail view«, »Enable insert record« and »Enable inline edit« in gridView section.
18. Check following checkboxes: »Hide gridView in edit/detail mode«, »Edit buttons in last columns« and »Left label position« in gridEdit section.
19. Scroll down and click »Update«.

Now add new instance of IndooGrid module to the page and repeat the above procedure for the Company table. Try to insert and edit some records - you can add new record by clicking insert (plus) icon.

What we have done here is enabling all common options for the grid – we done this for purposes of this example only: on your production server you should only options you'll need.

Go back to the first grid (Contacts) and click »Table settings«:

1. Click on edit icon near »company« column under gridView section.
2. Enter »Company« in »Caption« field.
3. Check »Join table on column« checkbox.
4. Click »Next«.
5. Select »Company« from dropdown list.
6. Click »Next«.
7. Select »id« in »Join table on column« dropdown list.
8. Select »name« in »Select column content« dropdown list.
9. Click »Next«.
10. Click on edit icon near »company« column under gridEdit section.
11. Enter »Company« in »Caption« field.
12. Click »Next«.
13. Check the »Dropdown list or radio button list (join for view mode)« checkbox.
14. Click »Next«.
15. Select »Values from database table« in »Select list source« dropdown list.
16. Select »Dropdown list« in »Show list as« dropdown list.
17. Click »Next«.
18. Enter »Company« in »Source table or view« field.
19. Click »Next«.
20. Select »id« in »Value column« dropdown list.
21. Select »name« in »Text column« dropdown list.
22. Click »Next«.
23. Click »Update«.

You may also enter captions for other columns and set some advanced properties (for example you can enter »D;en-us« in »Format data« field of »Date of birth« column in both gridView and gridEdit sections to display only date but not time).

All fields can be now edited without refreshing page. These fields can be ordinary text fields, dropdown lists, radio buttons or multiline text fields – as long as they are checked for inline editing. There are two options to check field for inline edit: either in main setting section (Allow inline edit) for all fields or within single column settings in gridView section for specific column. There's also indicator in list of gridView columns (Edit) showing which column is checked for inline editing.

After modifying fields records must be saved using one of SaveAll icons. One SaveAll icon is in module header (above command columns) and the second is on the bottom of the module. If you check »Check changes on exit«, you will be always asked to save changes before leaving the page.

It is important to know that inline edit depends on gridView settings. Sort, header caption and all other properties are the specified within grid view section. Only content of each grid cell is defined by grid Edit section. Therefore, if you want (for example) to enable joined values to the user, you should configure this (and others) features in gridView section of the module. If you want to enable joined values in (normal, not inline) edit or insert mode, you should configure this option in gridEdit section.

Catalog

When preparing catalogs it is often required to display only items from selected category. Items are normally organized in hierarchical groups, for example:

Home (root category)
.Electronics (main category)
.Audio (sub category)
.Video (sub category)
.TV (sub category)
.LCD TV (sub category)
.Plasma TV (sub category)
Sony 32KV (item)
Panasonic Viera (item)
.Special offer
Sony 32KV (item)

We can see that any category have its own parent category and are organized hierarchically.

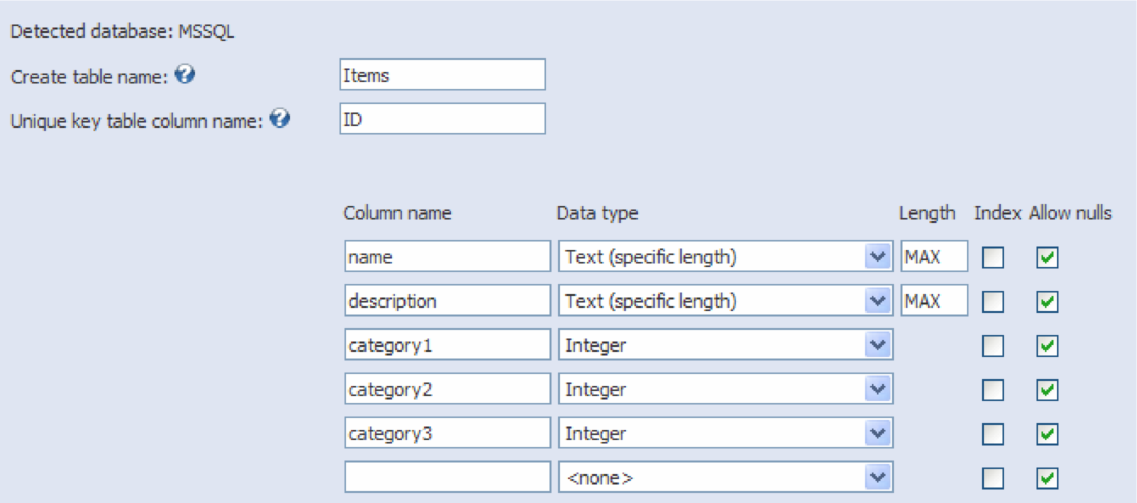
In such cases we would like to enable at least following options:

- Display all records from single main or sub category (either using link or category dropdown list) including all items in all sub categories of selected category

- Display category title bar
- Filter category
- Search in multiple categories simultaneously
- Single Item can be in one or more categories (let’s say in up to 3 categories)

For purposes of this example we will create two new tables (if you don't know how to create table, read about »Create table (or view) » option):

Items (int id, text name, text description, int category1, int category2, int category3) and



Detected database: MSSQL

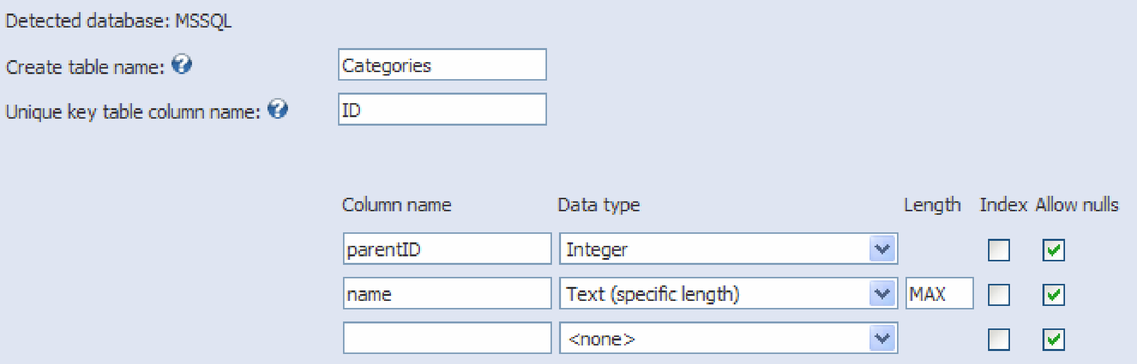
Create table name: Items

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
name	Text (specific length)	MAX	□	☑
description	Text (specific length)	MAX	□	☑
category1	Integer		□	☑
category2	Integer		□	☑
category3	Integer		□	☑
	<none>		□	☑

Picture 19: Creating Items table using Table (or view) wizard

Categories (int id, int parentId, text name).



Detected database: MSSQL

Create table name: Categories

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
parentID	Integer		□	☑
name	Text (specific length)	MAX	□	☑
	<none>		□	☑

Picture 20: Creating Categories table using Table (or view) wizard

After creating the tables we will add new instance of IndooGrid to the new page. This grid will be used to manage categories.

1. Click »Table Settings« link.
2. Use wizard to select »Categories« table.
3. Use wizard to select »id« field of Categories table.
4. Click on »Automatically add all columns« link.
5. Check »Allow edit« checkbox.
6. Check »Allow insert« checkbox.
7. Check »Allow delete« checkbox.
8. Click on »Add new view column« link.
9. Select »parentId« from »Column name« listbox .
10. Enter »Parent category name« in »Column caption« field.
11. Check »Join another table data on this column« checkbox.
12. Click Next«.
13. Select »Categories« in dropdown list and click »Next«.
14. Select »id« in »Join table on column« dropdown list.
15. Select »name« in »Select column content« dropdown list.
16. Click »Next«.
17. Click on edit icon near »ParentID« column under gridView section (first row of the list).
18. Enter »Parent ID« in »Column name« field.
19. Click »Next«.
20. Click on edit icon near »ParentID« column under gridEdit section (first row of the list).
21. Click »Next«.
22. Check »Drop down list or radio button list (join for view mode)« checkbox.
23. Click »Next«.
24. Select »Values from database table« in »Select list source« dropdown list.
25. Select »Dropdown list« in »Show list as« dropdown list.
26. Click »Next«.
27. Enter »Categories« in »Source table or view« field.
28. Click »Next«.
29. Select »id« in »Value column« dropdown list.
30. Select »name« in »Text column« dropdown list.
31. Click »Next«.
32. Click »Update«.

Now insert new records for categories as described above (enter Home category first because Home doesn't have parent category).

Second step will be creating new grid to manage items using same logic as for the categories (except that category1, category2 and category3 columns are related to Categories table instead of parented column). After creating grid insert some new items (Sony 32KV, Panasonic Viera etc.). Be sure to enter the appropriate categories for each item: id of category Plasma TV in category1 field of Sony record, id of category Special offer in category2 field of Sony record, id of category Plasma TV in category1 field of Panasonic etc.

Now we have enough data entered to create main grid to display items. Add new grid to the page and give it »Product catalog« title.

1. Click »Table Settings« link.

2. Use wizard to select »Items« table.
3. Use wizard to select »id« field of Items table.
4. Check »Allow detail view« checkbox.
5. Click »Add new view column« link.
6. Select »name« from »Column name« dropdown list.
7. Click »Next«.
8. Click »Add new edit column« link.
9. Select »name« from »Column name« dropdown list.
10. Click »Next« until you will be returned to »Table settings« page.
11. Click »Add new view column« link.
12. Select »description« from »Column name« dropdown list.
13. Click »Next« until you will be returned to »Table settings« page.
14. Check »Advanced options« checkbox.
15. Enter condition in »GridView WHERE part of the sql statement« field to show only items in current categories:

```
Category1 = [numericQueryString:catid] OR Category2 = [numericQueryString:catid] OR Category3 = [numericQueryString:catid] OR ([numericQueryString:catid] < 0)
```

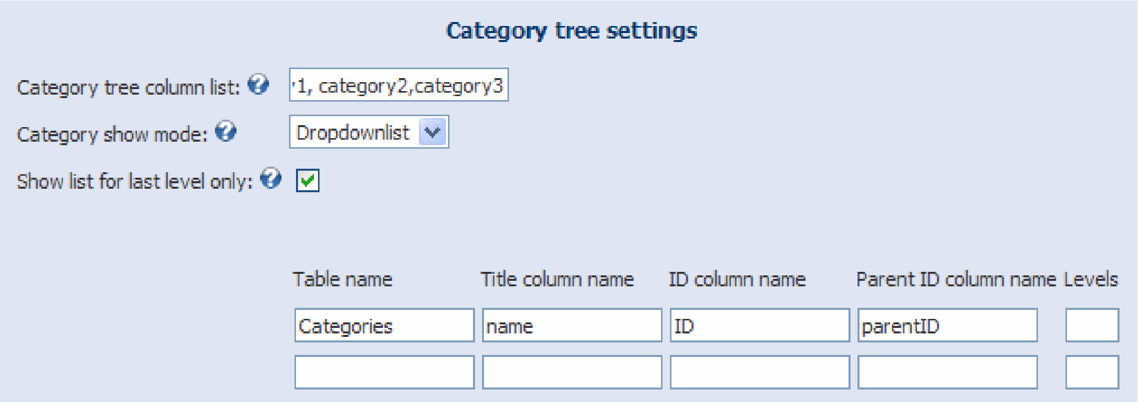
16. Scroll down and click »Update«.

This is our main product page. All items in single category are available using link or built-it category list.

First we will look at built-in category list. This feature enables displaying categories as dropdown list, list box or tree view. By selecting the category only items in selected category will be displayed (in »Product catalog« grid).

17. Click on »Table settings«.
18. Click on »<none>» or wizard icon of »Category tree column list«.
19. Enter »category1, category2,category3« in »Category tree column list«.
20. Enter »Categories« in first column (»Table name«) of first row of »Category table(s)« table.
21. Enter »name« in second column (»Title column name«) of first row of »Category table(s)« table.
22. Enter »ID« in third column (»ID column name«) of first row of »Category table(s)« table.
23. Enter »parentID« in fourth column (»Parent ID column name«) of first row of »Category table(s)« table.

Your screen should look like this:



Category tree settings

Category tree column list: '1, category2,category3

Category show mode: Dropdownlist

Show list for last level only: ☑

Table name	Title column name	ID column name	Parent ID column name	Levels
Categories	name	ID	parentID	

Picture 21: adding Category tree to the Catalog

24. Click »Finish«

There's new dropdown list available on first page. By selecting the category only items from selected category will be displayed.

.

Second option to show only items from selected category is using link. For example, if we want to build link to show all items in category Plasma TV, we should use this structure:

http://....Default.aspx?&catid=7

where 7 is id of category Plasma TV in Categories table. You can copy first par of the link (before Default.aspx) from address bar of you browser.

We can than use this link to open the grid with selected category. For example, we can make menu items in DotNetNuke which will link to predefined categories.

We can use other grids to refer to selected category. Here we will build links in first grid we made (Categories).

1. Click on »Table settings« link.
2. Click »Add new view column« link.
3. Select »id« from »Column name« dropdown list.
4. Uncheck the »Visible« checkbox.
5. Click »Next«.
6. Click on edit icon near »name« column under gridView section.
7. Check the »Column contents are URL links« checkbox.
8. Click »Next«.
9. Enter »catid« in »Parameter name« textbox.
10. Select »id« from »Select column for parameter value« dropdown list.
11. Click »Next«.

12. Scroll down and click »Update«.

Click on category name and watch the »Product catalog« grid. Only items in selected category will be displayed. That's because when clicking on category link, parameter »catid« (you can see it in the address bar of your browser) is passed to Products grid and filter the data to selected category only.

We normally also want to show where we are in category hierarchy (current category chosen) in some kind of title bar on the top. Enter (copy) following text to »Module header« textarea under Advanced options:

```
<font style=»color:blue«><script language=»javascript«>if ([numericQueryString:catid] >
-1) {document.write('You are here: ')}</script>
<command sql=»Select (ISNULL(categories_4.name + '.','') +
ISNULL(categories_3.name + '.','') + ISNULL(categories_2.name + '.','') +
ISNULL(categories_1.name + '.','') + categories.name) as name
FROM        categories categories_4 RIGHT OUTER JOIN
            categories categories_3 ON categories_4.ID = categories_3.ParentID RIGHT
OUTER JOIN
            categories categories_2 ON categories_3.ID = categories_2.ParentID RIGHT
OUTER JOIN
             categories categories_1 ON categories_2.ID = categories_1.ParentID
RIGHT OUTER JOIN
             categories ON categories_1.ID = categories.ParentID
WHERE categories.ID =
[numericQueryString:catid]«></command><font>
```

Using this code we have displayed text using javascript and SQL command. When parameter »catid« is zero or greater, we make database request to get all parent categories of selected category (defined by »catid«). Database request will work on up to five levels.

Next step is to enable user to search for product in all categories. As every item can appear in up to 3 categories, we have to enable multiple search for fields' category1, category2, category3. For instructions on how to do that, please read the »Allow filter« section of this document.

In next step we will allow item records to include pictures (»Product catalog« grid).

1. Click »Table settings« link.
2. Click on »Add binary (file) columns« link (or wizard icon).
3. Select »File (binary) content is stored in database column« from »File (binary) location« dropdown list.
4. Click »Next«.
5. Select »Pictures« from »Select the type of uploaded files« dropdown list.
6. Click »Next«.
7. Select »Image thumbnail and link« from »Show binary as (see help icon)«.
8. Enter »30« in »Thumbnail width:« textbox.

9. Enter »30« in »Thumbnail height:« textbox.
10. Click »Next«.
11. Click »Finish«.
12. Uncheck »Visible« checkboxes of gridView columns »file1_name«, »file1_size« and »file1_modify«.
13. Scroll down and click »Update«.

Ratings and Comments

When creating catalogs and listings it is often required to rate and comment items. Normally only logged users can rate or comment products (to prevent commercial bots to post adds).

We will continue from previous example (Catalog). First we will add new IndooGrid to the page and give it title »Comments«. This grid should be on the same page as »Product catalog« grid from the previous example.

In first step we will create new table called »Ratings«. Table structure is as follows:

- ID (integer) – unique identifier
- ItemID (integer) – relation to Items table
- UserID (integer) – relation to Users table
- Rating (integer) – rating form 1 to 5
- Comment (text) – additional comment

If you don't know how to create tables using grid please read the »Table (or view)« section. When our table is ready, we will configure grid:

1. Click »Table settings« link.
2. Use wizard to select »Ratings« table.
3. Use wizard to select »ID« field of Ratings table.
4. Click on »Automatically add all columns« link or wizard icon.
5. Uncheck »Visible« checkbox of »ItemID« column in gridView column list.
6. Click on edit icon near »UserID« column in gridView column list.
7. Enter »User« in »Caption« field.
8. Check »Join another table data on this column« checkbox.
9. Click »Next«.
10. Enter »Users« in » Join table« field.
11. Click »Next«.
12. Select »UserID« in »Join table on column« dropdown list.

13. Select »Username« in » Select column contents« dropdown list.
14. Click »Next«.
15. Check »Hide gridView in edit/detail mode« checkbox.
16. Check »Advanced options« checkbox.
17. Enter »gvRating« in »gridView ID« textbox.
18. Enter »ItemID=[getNumericObject:viewpage.gridByID(»geItem«).idvalue]« in »GridView WHERE part of the SQL statement« textbox.
19. Enter following code in »Module header text«:

```
<script language=»javascript«>
[setObject:viewPage.gridView.txt_noRecordsFound=<br>No ratings
yet. Be first to rate this item.<br>]
</script>
```

20. Enter following code in »Startup script block« textarea:

```
»setModuleVisibilityByGridId:gvRating=[compute:iif([idValueByGridE
ditId:geItem]<1,0,1)]«
```

21. Scroll down and click »Update«.

In steps 6-13 we defined that Username will be displayed for the specific column instead of author's userID.

In step 18 we defined filter for comments – only comments (ratings) for current Item will be displayed. Current item's ID is returned using token

[getNumericObject:viewpage.gridByID(»geItem«).idvalue]

This token will return us idvalue (that's ID of current record) of grid with ID »geItem«. At this time we do not have such grid yet but will mark our »Product catalog« to be this grid.

1. Click »Table settings« of »Product catalog« grid.
2. Check »Advanced options« checkbox.
3. Enter »geItem« in »gridEdit ID« textbox.
4. Scroll down and click »Update«.

In step 19 we defined text to display if there are no comments for selected item yet. In fact we replaced default »No records« text by using token

[setObject:viewPage.gridView.txt_noRecordsFound=something].

In step 20 we defined that Comments grid will not be displayed if current »Product catalog« mode is not view or edit – only in these two cases will idvalue of the »Product catalog« be greater than zero.

Now we have grid (Comments) that will be displayed only when detail or edit view of our second grid (Product catalog) will be selected.

Next step it will be to add form that will allow users to rate (comment) the item. Let's say only registered (and logged users) will have the ability to rate or comment items. Therefore we will first add normal Text/HTML module to the page (title: »Rate it!«, Pane: Right pane):

1. Enter text »You must be logged in to rate item or post comment« instead of content.
2. Click on »Module settings«.
3. Uncheck »Inherit View permissions from Page« checkbox
4. Check the »Unauthenticated users« checkbox.
5. Click »Update«.

This text will be displayed only to the users which are not currently logged in. Now we will add IndooGrid (title: »Rate it!«, Pane: Right pane) to the page and allow only registered users to see it:

1. Click on »Module settings«.
2. Uncheck »Inherit View permissions from Page« checkbox
3. Check the »Registered users« checkbox.
4. Click »Update«.

After playing with built-in DotNetNuke permissions feature we will create form to rate or comment items. Follow these steps:

1. Click »Table settings« link of IndooGrid (with title Rate it!).
2. Use wizard to select »Ratings« table.
3. Use wizard to select »ID« field of Ratings table.
4. Click on »Automatically add all columns« link or wizard icon.
5. Check »Enable edit record« checkbox.
6. Check »Hide gridView in edit/detail mode« checkbox.
7. Select »insert« from »Default GridEdit mode« dropdown list.
8. Enter »Rate it!« in »Insert button text« textbox.
9. Uncheck »Visible« checkbox of »ItemID« and »UserID« columns in gridEdit column list.
10. Click on edit icon near »ItemID« column in gridEdit column list.
11. Click »Next«.
12. Enter »[idValueByGridEditId:geItem]« in »Field value« textbox.
13. Click »Next«.
14. Click on edit icon near »UserID« column in gridEdit column list.
15. Click »Next«.
16. Enter »[User:UserID]« in »Field value« textbox.
17. Click »Next«.
18. Click on edit icon near »Rating« column in gridEdit column list.
19. Enter »<br><b>Your opinion counts! </b><br>» in »Caption« textbox.
20. Click »Next«.
21. Check »Drop down list or radio button list (join for view mode)« checkbox.
22. Click »Next«.
23. Select »Values from list« in »Select list source« dropdown list.
24. Select »Dropdown list« in »Show list as« dropdown list.
25. Click »Next«.

26. Enter following code in »List values and text« textarea:

```
5;<span a></span>Excellent
4;<span b></span>Very good
3;<span c></span>Average
2;<span d></span>Below average
1;<span e></span>Bad
```

27. Enter »3« in »Default value« textbox.
28. Click »Next«.
29. Click on edit icon near »Comment« column in gridEdit column list.
30. Enter »<br><br>Your comment (500 chars max)<br>» in »Caption« textbox.
31. Click »Next«.
32. Enter »200« in »Width« textbox.
33. Click »Next«.
34. Enter »10« in »Multiline« textbox.
35. Click »Next«.
36. Check the »Advanced option« checkbox.
37. Enter »geRating« in »gridEdit ID« textbox.
38. Enter following code in »Module header text« textarea:

```
<script language=»javascript«>
var mode = Number('
[getNumericObject:viewpage.gridByID(»geItem«).idvalue]');
var rating = Number('<command sql=»SELECT ID FROM Ratings WHERE
(UserID = [User:UserID]) AND (ItemID =
[getNumericObject:viewpage.gridByID(»geItem«).idvalue])«></command>'
);
if ((rating > 0) && (mode > 0)) {
document.write('You already did comment/rate this item but you can
do it again.<br> ');
}
</script>
```

39. Enter following code in »Startup script block« textarea:

```
»setModuleVisibilityByGridId:geRating=[compute:iif([idValueByGridEdi
tId:geItem]<1,0,1)]«
```

40. Enter following code to »Execute SQL script(s) on insert record« textarea:

```
delete from Ratings WHERE (Ratings.UserID = [user:UserID]) AND
Ratings.ItemID =
[getNumericObject:viewpage.gridByID(»geItem«).idvalue])
AND (Ratings.ID <> [indoogrid:idValue])
```

41. Scroll down and click »Update«.

In step 38 we defined note if user had already commented selected item.

In step 39 we defined that Rate it! grid will not be displayed if current »Product catalog« mode is not view or edit – only in these two cases will idvalue of the »Product catalog« be greater than zero.

In step 40 we delete current comment of current user for current item if exists (that is if user had already commented current item) to avoid duplicate entries.

NOTE: In steps 19 in 39 we disabled 4 parents elements of the grid. This examples assumes that your skin is DotNetNuke default. If you use different skin, these values can be different than 4. Start with 1 and increase value by 1 to find value that suits to your skin.

Photo Gallery

Media galleries are most common website functionalities used. IndooGrid provides powerful tool to work with media files – yet very simple but as extensible as you want.

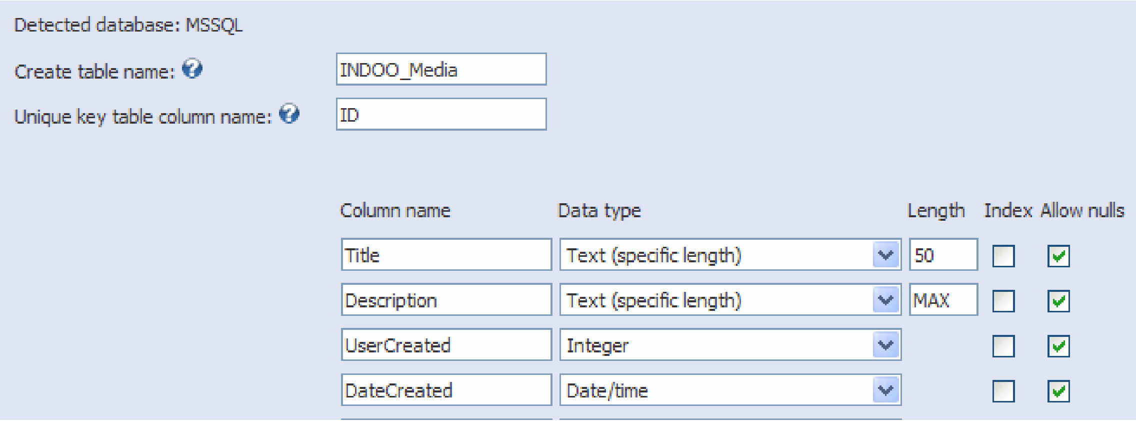
Before we begin it is necessary to explain how IndooGrid works with (media) files. Media (photos and videos) can be stored either in database or on the file system. IndooGrid stores files by default on file system because of performance reasons. Most DotNetNuke users use MS SQL Server Express as database – therefore storing files into database would quickly fill SQL Server beyond its limitations. For example, if average picture is 1MB, you could store about 4000 pictures to the database (MS SQL Server Express limit is 4 GB).

In this example we'll create Photo Gallery with 5 pictures in 4 rows. This example can be viewed on our Demo sites (Photo Gallery and Video Gallery samples) where you can also download it and try it on your own DotNetNuke portal.

In first step we will create new table called »INDOO_Media«. Table structure is as follows:

- ID (integer) – unique identifier
- Title (text 50) – picture title
- Description (text MAX) – picture description
- UserCreated (integer) – ID of user who uploaded the picture
- DateCreated (datetime) – date and time of uploading

Or as it shown on this picture:



Detected database: MSSQL

Create table name: INDOO_Media

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
Title	Text (specific length)	50	□	☑
Description	Text (specific length)	MAX	□	☑
UserCreated	Integer		□	☑
DateCreated	Date/time		□	☑

Picture 22: Creating Media table

If you don't know how to create tables using grid please read the »Table (or view)« section. After creating the table, add new IndooGrid module (title »Media management«) to the page this page should be visible to registered users only) and follow this steps:

1. Click on »Table settings«.
2. Enter »INDOO_Media« in »Table (or view)« textbox.
3. Use wizard to select »ID« field for »Unique key column«.
4. Click on »Automatically add all columns« link or wizard icon.
5. Check »Enable insert records« checkbox.
6. Check »Enable edit record« checkbox.
7. Check »Enable delete record« checkbox.
8. Check »Hide gridView in edit/detail mode« checkbox.
9. Check »Left label position« checkbox.
10. Check » Edit buttons in last column« checkbox.
11. Uncheck »Visible« checkboxes of gridEdit column list (second column list) for following fields: DateCreated, UserCreated.
12. Click on edit icon near »UserCreated« column under gridEdit section (second row of the second list).
13. Click »Next«.
14. Enter »[User:UserID]« in »Field value« textbox.
15. Click »Next«.
16. Click on edit icon near »DateCreated« column under gridEdit section (second row of the second list).
17. Click »Next«.
18. Enter »[indoogrid:currentDate]« in »Field value« textbox.
19. Check »Use field value as default value only« checkbox.
20. Click »Next«.
21. Click »Update«.

In this first step we created grid that allows us managing records. At this time records don't have associated files (photos or videos). We will add binary column (that can store photos, videos or other files) in next step.

1. Click on »Table settings«.
2. Click »Add binary columns wizard« link.
3. Select »File (binary) content is stored into database column« from »File (binary) location« dropdown list.
4. Click »Next«.

We defined how will files be stored on the server. Quick tip would be to choose database for smaller libraries and file system for large libraries. Please read about »Add binary (file) columns« section for more on this option (And options configured in following steps). If only database option is available in dropdown box, this indicates that current DotNetNuke installation doesn't have permissions to store files to file system or you didn't allowed users to store binary data to file system. Second option can be enabled in »Module settings« by checking »Allow save binary data to file system:« checkbox.

5. Select »Pictures« from »Select the type of uploaded files« dropdown list.
6. Click »Next«.

We defined what kind of files will be stored on the server. Here you can decide between media files and files of other types.

7. Select »Link« from »Show binary as (see help icon)« dropdown list.
8. Click »Next«.
9. Click »Next«.
10. Click »Finish« button.
11. Click twice on up (arrow) icon of »file1_title« column (fifth row of first list) in gridView column list.
12. Enter »1« in JavaScript dialog box.
13. Click »Ok« button of JavaScript dialog box.
14. Click on edit icon near »file1_title« column under gridView section (first row of the first list).
15. Select »ID« from »Column name«.
16. Enter »Picture« in »Caption« textbox.
17. Select »2 - editor« in »Show data from more columns (custom HTML« dropdown list.
18. Click »Next«.
19. Click »Next«.
20. Click »Next«.
21. Enter following code in editor (**be sure that »Raw« radio button is selected**):

```
<center> <div><img style=»border-right: white 2px solid; padding-
right: 5px; background-position: -120px -80px; border-top: medium
none; display: none; padding-left: 5px; font-size: 10pt; padding-
bottom: 5px; border-left: white 2px solid; width: 300px; padding-
top: 5px; border-bottom: white 2px solid; font-family: Verdana,
Helvetica; position: absolute; height: 200px; background-color:
white« alt=»» onmouseout=»this.style.display ='none'»
onclick=»this.style.display ='none'» /> <img alt=»»
onmouseover=»img2=this.parentNode.childNodes[0];if
(img2.src==''){img2.src='[indooGrid:binaryLink]'};img2.style.display
```

```
='block';« src=»[indooGrid:binaryLinkThumbnail]« /></div>
<strong>[columnField:Title]</strong> </center>
```

22. Click »Finish«.
23. Enter »Picture title« in »Caption« textbox of column »file1_title » under gridEdit section.
24. Enter »Picture« in »Caption« textbox of column »file1_content » under gridEdit section.
25. Click »Update«.

Click on add icon (plus icon) to add new pictures.

1. Enter »First picture« in »file1_title1« field.
2. Click »Browse« button in »file1_content« row and select any picture from your computer.
3. Click »Save«.

The picture you have added is now displayed as thumbnail with text called »First picture«. Move mouse over the thumbnail to view bigger picture.

The grid we created above will serve as management interface to add and edit picture. However, this is not the form we want to present to the users. Therefore we will now create gallery with 5 pictures in each row as it is usually required by the users.

To achieve that effect we must create views that will pull 5 records from database and merge them into one record we can display in our grid (grid do not support pivot tables at this time). We will create grid using SQL developer window which allows us to work with database using standard SQL. Add new IndooGrid to the page and click on »Table settings«. Scroll down and click on »Show developer SQL window for current connection« checkbox. New window will appear. Enter (copy and paste) following code to developer window:

```
CREATE view vw_INDOO_sample_Photos1 AS SELECT id, ROW_NUMBER() OVER
(ORDER BY id) AS rownum FROM INDOO_sample_Photos
GO

CREATE view vw_INDOO_sample_Photos2 as SELECT id, rownum, (rownum –
1) % 5 AS rg, FLOOR((rownum – 1) / 5) AS cg FROM
vw_INDOO_sample_Photos1
GO

CREATE view vw_INDOO_sample_Photos3 as SELECT  cg, max([0]) as m0,
max([1]) as m1, max([2]) as m2, max([3]) as m3, max([4]) as m4 FROM
vw_INDOO_sample_Photos2 PIVOT (max(id) FOR rg IN ([0], [1], [2] ,
[3] , [4])) AS pvt group by cg
GO
```

```
CREATE view vw_INDOO_sample_Photos4  as
SELECT vw_INDOO_sample_Photos3.cg AS newID,
INDOO_sample_Photos_1.Title AS Title_1, INDOO_sample_Photos_1.ID AS
ID_1, INDOO_sample_Photos_1.File1_Content AS FileContent_1,
INDOO_sample_Photos_1.File1_MimeType AS FileMimeType_1,
INDOO_sample_Photos_2.Title AS Title_2, INDOO_sample_Photos_2.ID AS
ID_2, INDOO_sample_Photos_2.File1_Content AS FileContent_2,
INDOO_sample_Photos_2.File1_MimeType AS FileMimeType_2,
INDOO_sample_Photos_3.Title AS Title_3, INDOO_sample_Photos_3.ID AS
ID_3, INDOO_sample_Photos_3.File1_Content AS FileContent_3,
INDOO_sample_Photos_3.File1_MimeType AS FileMimeType_3,
INDOO_sample_Photos_4.Title AS Title_4, INDOO_sample_Photos_4.ID AS
ID_4, INDOO_sample_Photos_4.File1_Content AS FileContent_4,
INDOO_sample_Photos_4.File1_MimeType AS FileMimeType_4,
INDOO_sample_Photos_5.Title AS Title_5, INDOO_sample_Photos_5.ID AS
ID_5, INDOO_sample_Photos_5.File1_Content AS FileContent_5,
INDOO_sample_Photos_5.File1_MimeType AS FileMimeType_5
FROM vw_INDOO_sample_Photos3
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_1 ON
vw_INDOO_sample_Photos3.m0 = INDOO_sample_Photos_1.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_2 ON
vw_INDOO_sample_Photos3.m1 = INDOO_sample_Photos_2.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_3 ON
vw_INDOO_sample_Photos3.m2 = INDOO_sample_Photos_3.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_4 ON
vw_INDOO_sample_Photos3.m3 = INDOO_sample_Photos_4.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_5 ON
vw_INDOO_sample_Photos4.m3 = INDOO_sample_Photos_5.ID
```

This part of code needs some explanation. We created view that will return five records from INDOO_sample_Photos as one record in vw_INDOO_sample_Photos4. For example, if we want to show 4 pictures in a row, we would need 4 records from main table in our view. In that case we would use following code (you can use DROP VIEW view_name option to delete views from previous step if you have executed SQL for five rows – in that case drop all four vw_INDOO_sample_PhotosN views we made in previous example, starting with »DROP VIEW vw_INDOO_sample_Photos1«):

```
CREATE view vw_INDOO_sample_Photos1 AS SELECT id, ROW_NUMBER() OVER
(ORDER BY id) AS rownum FROM INDOO_sample_Photos
GO

CREATE view vw_INDOO_sample_Photos2 as SELECT id, rownum, (rownum -
1) % 4 AS rg, FLOOR((rownum - 1) / 4) AS cg FROM
vw_INDOO_sample_Photos1
GO

CREATE view vw_INDOO_sample_Photos3 as SELECT  cg, max([0]) as m0,
max([1]) as m1, max([2]) as m2, max([3]) as m3 FROM
vw_INDOO_sample_Photos2 PIVOT (max(id) FOR rg IN ([0], [1], [2],
[3])) AS pvt group by cg
GO
```

```
CREATE view vw_INDOO_sample_Photos4  as
SELECT vw_INDOO_sample_Photos3.cg AS newID,
INDOO_sample_Photos_1.Title AS Title_1, INDOO_sample_Photos_1.ID AS
ID_1, INDOO_sample_Photos_1.File1_Content AS FileContent_1,
INDOO_sample_Photos_1.File1_MimeType AS FileMimeType_1,
INDOO_sample_Photos_2.Title AS Title_2, INDOO_sample_Photos_2.ID AS
ID_2, INDOO_sample_Photos_2.File1_Content AS FileContent_2,
INDOO_sample_Photos_2.File1_MimeType AS FileMimeType_2,
INDOO_sample_Photos_3.Title AS Title_3, INDOO_sample_Photos_3.ID AS
ID_3, INDOO_sample_Photos_3.File1_Content AS FileContent_3,
INDOO_sample_Photos_3.File1_MimeType AS FileMimeType_3,
INDOO_sample_Photos_4.Title AS Title_4, INDOO_sample_Photos_4.ID AS
ID_4, INDOO_sample_Photos_4.File1_Content AS FileContent_4,
INDOO_sample_Photos_4.File1_MimeType AS FileMimeType_4
FROM vw_INDOO_sample_Photos3
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_1 ON
vw_INDOO_sample_Photos3.m0 = INDOO_sample_Photos_1.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_2 ON
vw_INDOO_sample_Photos3.m1 = INDOO_sample_Photos_2.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_3 ON
vw_INDOO_sample_Photos3.m2 = INDOO_sample_Photos_3.ID
LEFT OUTER JOIN INDOO_sample_Photos AS INDOO_sample_Photos_4 ON
vw_INDOO_sample_Photos3.m3 = INDOO_sample_Photos_4.ID
```

After we have created view, we can configure the grid. In this example we will work with 5 pictures in each row.

1. Enter »vw_INDOO_sample_Photos4« in »Table (or view)« textbox.
2. Enter »newID« field in »Unique key column« textbox.
3. Click on »Automatically add all columns« link or wizard icon.
4. Uncheck »Visible« checkboxes of gridView column list (first column list) for all fields except the following: Title_1, Title_2, Title_3, Title_4, Title_5.
5. Select »0« in »Filter« dropdown list for all fields (except for Visible fields) in gridView.
6. Click on edit icon near »Title_1« column under gridView section (first list).
7. Check »Add binary data from another column« checkbox.
8. Click »Next«.
9. Select »FileContent_1« from »Add binary data from column« dropdown list (Title_1 will be related to column containing image 1).
10. Select »Image thumbnail« from »Show binary as (see help icon)« dropdown list.
11. Click »Next«.
12. Enter »90« in »Max thumbnail width« textbox.
13. Enter »1000« in »Max thumbnail height« textbox.
14. Click »Next«.
15. Repeat steps 6-14 for all four remaining fields marked as Visible (select FileContent_2 for Title_2 etc.)
16. Click on »Show advanced settings« checkbox.

17. Enter »[setObject:viewPage.gridView.isColumnsHeaderVisible=False]« in »Module footer text.
18. Click »Update«.

Media Library (photos, videos and other files)

In this example we will create media library that will allow users to upload photos and videos, store them on server, organize them in albums (categories) and allow other users to view them.

First we will create three new tables called INDOO_MediaCategories, INDOO_MediaAlbums and INDOO_MediaLib. Please read about »Table (or view)« property to learn how to create tables using module's built-in wizard (you can use any grid already added to any page to create tables).

First table (INDOO_MediaCategories) will have following structure:

- ID (integer) – unique identifier
- ParentID (integer) – ID of parent category
- Title (text 50) – title of the category
- Description (text MAX) – category description

This table will store category provided by administrator (in this example we are administrator offering users Media library service). For example, our categories will be Animals (with subcategories Pets and Wild animals), Holidays, Work etc.

Second table (INDOO_MediaAlbums) will have following structure:

- ID (integer) – unique identifier parent
- ParentID (integer) – ID of parent album
- UserID (integer) – ID of owner of the album
- Title (text 50) – title of the album
- Description (text MAX) – album description

This table will store albums provided by the users. Users will have ability to organize their photos and videos in way they'll specified. For example, user albums would be Arizona Trip, My House, My pets etc.

Third table (INDOO_MediaLib) will have following structure:

- ID (integer) – unique identifier
- Title (text 50) – media title
- Description (text MAX) – media description
- MediaCategoriesID (integer) – ID of media category
- MediaAlbumsID (integer) – ID of media album
- UserID (integer) – ID of user who uploaded the picture
- SortOrder (integer) – custom sort order
- DateCreated (datetime) – date and time of uploading
- DateModified (datetime) – date and time of updating (if any)

This table will store all media files (photos and videos) provided by all users. This table is related to Categories table with field MediaCategoriesID and to Albums with field MediaAlbumsID.

If you want to allow media files to exists in more than one category or album at the same time, add additional fields to the table: for example add MediaCategoriesID2 to enable to store media files in two categories at the same time or add MediaAlbumsID2 to allow two albums containing same media file. One example on how to work with items in more than category at the same time is described in Catalog sample of this document.

After we made tables we'll need later, we will first create grid that will allow us to manage categories that will be the same for all users (only we can change these categories).

1. Create new page with title Library Settings. This page should be visible to Administrators only.
2. Add new IndooGrid to the page (title Categories).
3. Click on »Table settings«.
4. Enter »INDOO_MediaCategories« in »Table (or view)« textbox.
5. Use wizard to select »ID« field for »Unique key column«.
6. Click on »Automatically add all columns« link or wizard icon.
7. Check »Enable insert records« checkbox.
8. Check »Enable edit record« checkbox.
9. Check »Enable delete record« checkbox.
10. Check »Hide gridView in edit/detail mode« checkbox.
11. Check »Left label position« checkbox.
12. Check » Edit buttons in last column« checkbox.
13. Enter »Title« in »Caption« of Title column in gridView column list.
14. Click on edit icon near »ParentID« column in gridView column list (first row of the first list).
15. Enter »Parent category« in »Column caption« field.
16. Check »Join another table data on this column« checkbox.
17. Click Next«.

18. Select »INDOO_MediaCategories« in dropdown list and click »Next«.
19. Select »ID« in »Join table on column« dropdown list.
20. Select »Title« in »Select column content« dropdown list.
21. Click »Next«.
22. Click on edit icon near »ParentID« column under gridEdit section (first row of the second list).
23. Enter »Parent category« in »Caption« field.
24. Click »Next«.
25. Check »Drop down list or radio button list (join for view mode)« checkbox.
26. Click »Next«.
27. Select »Values from database table« in »Select list source« dropdown list.
28. Select »Dropdown list« in »Show list as« dropdown list.
29. Click »Next«.
30. Enter » INDOO_MediaCategories » in »Source table or view« field.
31. Click »Next«.
32. Select »ID« in »Value column« dropdown list.
33. Select »Title« in »Text column« dropdown list.
34. Check »Sort by value« checkbox.
35. Click »Next«.
36. Click »Update«.

We have now list of all categories but as you can see, no category is yet in the table. Click on insert icon to add new category. You'll notice that there's no value in parent category- that's why we have to add top category first. You can call this top category Home or Root or All (we'll call it Home). After that add two new categories called Animals and Peoples. Both categories should have Home as parent category.

In next step we will create grid that will allow users to manage their albums.

1. Create new page with title My Library. This page should be visible to registered users only.
2. Add new IndooGrid to the page (title »My Albums)«.
3. Click on »Table settings«.
4. Enter »INDOO_MediaAlbums« in »Table (or view)« textbox.
5. Use wizard to select »ID« field for »Unique key column«.
6. Click on »Automatically add all columns« link or wizard icon.
7. Check »Enable insert records« checkbox.
8. Check »Enable edit record« checkbox.
9. Check »Enable delete record« checkbox.
10. Check »Hide gridView in edit/detail mode« checkbox.
11. Check »Left label position« checkbox.
12. Check » Edit buttons in last column« checkbox.
13. Enter »Title« in »Caption« of Title column in gridView column list.
14. Click on edit icon near »ParentID« column in gridView column list (first row of the first list).
15. Enter »Parent album« in »Column caption« field.
16. Check »Join another table data on this column« checkbox.
17. Click Next«.

18. Select »INDOO_MediaAlbums« in dropdown list and click »Next«.
19. Select »ID« in »Join table on column« dropdown list.
20. Select »Title« in »Select column content« dropdown list.
21. Click »Next«.
22. Uncheck »Visible« checkbox of UserID column in gridView (second row of first list).
23. Click on edit icon near »ParentID« column under gridEdit section (first row of the second list).
24. Enter »Parent album« in »Caption« field.
25. Click »Next«.
26. Check »Drop down list or radio button list (join for view mode)« checkbox.
27. Click »Next«.
28. Select »Values from database table« in »Select list source« dropdown list.
29. Select »Dropdown list« in »Show list as« dropdown list.
30. Click »Next«.
31. Enter » INDOO_MediaAlbums » in »Source table or view« field.
32. Click »Next«.
33. Select »ID« in »Value column« dropdown list.
34. Select »Title« in »Text column« dropdown list.
35. Check »Sort by value« checkbox.
36. Click »Next«.
37. Click on edit icon near »UserID« column under gridEdit section (second row of the second list).
38. Uncheck the »Visible« checkbox.
39. Click »Next«.
40. Enter »[User:UserID]« in »Field value« textbox.
41. Click »Next«.
42. Click »Show advanced settings« checkbox.
43. Enter »INDOO_MediaAlbums.UserID=[User:UserID]« in »GridView WHERE part of the sql statement« textbox.
44. Click »Update«.

Now we have grid that will allow users to manage their albums. In step 37-41 we defined that ID of current user will be inserted in the record in the background. In step 43 we filtered grid to show only Albums that were created by current user.

Enter some Album names in the database using same logic as for Categories – there must be one root album and than albums in subcategories. For example, create Home album as top album and My house and My pets albums with parent album Home.

In next step we'll create main grid that will allow users to upload media files. This grid (title My Media) should be added to the same page as previous grid and should be visible only to registered users.

1. Click on »Table settings«.
2. Enter »INDOO_MediaLib« in »Table (or view)« textbox.

3. Use wizard to select »ID« field for »Unique key column«.
4. Click on »Automatically add all columns« link or wizard icon.
5. Check »Enable insert records« checkbox.
6. Check »Enable edit record« checkbox.
7. Check »Enable delete record« checkbox.
8. Check »Hide gridView in edit/detail mode« checkbox.
9. Check »Left label position« checkbox.
10. Check » Edit buttons in last column« checkbox.
11. Click on edit icon near »MediaCategoriesID« column in gridView column list (third row of the first list).
12. Enter »Category« in »Column caption« field.
13. Check »Join another table data on this column« checkbox.
14. Click Next«.
15. Select »INDOO_MediaCategories« in dropdown list and click »Next«.
16. Select »ID« in »Join table on column« dropdown list.
17. Select »Title« in »Select column content« dropdown list.
18. Click »Next«.
19. Click on edit icon near » MediaAlbumsID » column in gridView column list (fourth row of the first list).
20. Enter »Album« in »Column caption« field.
21. Check »Join another table data on this column« checkbox.
22. Click Next«.
23. Select »INDOO_MediaAlbums« in dropdown list and click »Next«.
24. Select »ID« in »Join table on column« dropdown list.
25. Select »Title« in »Select column content« dropdown list.
26. Click »Next«.
27. Uncheck »Visible« checkbox of UserID column in gridView (fifth row of first list).
28. Uncheck »Visible« checkbox of DateCreated column in gridView.
29. Uncheck »Visible« checkbox of DateModified column in gridView.
30. Click on edit icon near »MediaCategoriesID« column under gridEdit section (third row of the second list).
31. Enter »Category« in »Caption« field.
32. Click »Next«.
33. Check »Drop down list or radio button list (join for view mode)« checkbox.
34. Click »Next«.
35. Select »Values from database table« in »Select list source« dropdown list.
36. Select »Dropdown list« in »Show list as« dropdown list.
37. Click »Next«.
38. Enter » INDOO_MediaCategories » in »Source table or view« field.
39. Click »Next«.
40. Select »ID« in »Value column« dropdown list.
41. Select »Title« in »Text column« dropdown list.
42. Check »Sort by value« checkbox.
43. Click »Next«.
44. Click on edit icon near »MediaAlbumsID« column under gridEdit section.
45. Enter »Album« in »Caption« field.
46. Click »Next«.
47. Check »Drop down list or radio button list (join for view mode)« checkbox.

48. Click »Next«.
49. Select »Values from database table« in »Select list source« dropdown list.
50. Select »Dropdown list« in »Show list as« dropdown list.
51. Click »Next«.
52. Enter » INDOO_MediaAlbums » in »Source table or view« field.
53. Enter »INDOO_MediaAlbums.UserID = [User:UserID]« in »Limit source records with Sql WHERE clause« textbox.
54. Click »Next«.
55. Select »ID« in »Value column« dropdown list.
56. Select »Title« in »Text column« dropdown list.
57. Check »Sort by value« checkbox.
58. Click »Next«.
59. Click on edit icon near »UserID« column under gridEdit section.
60. Uncheck the »Visible« checkbox.
61. Click »Next«.
62. Enter »[User:UserID]« in »Field value« textbox.
63. Click »Next«.
64. Click on edit icon near »DateCreated« column under gridEdit section.
65. Uncheck the »Visible« checkbox.
66. Click »Next«.
67. Enter »[indoogrid:CurrentDate]« in »Field value« textbox.
68. Click »Next«.
69. Click on edit icon near »DateModified« column under gridEdit section.
70. Uncheck the »Visible« checkbox.
71. Click »Next«.
72. Enter »[indoogrid:CurrentDate]« in »Field value« textbox.
73. Click »Next«.
74. Click »Show advanced settings« checkbox.
75. Enter »INDOO_MediaLib.UserID=[User:UserID]« in »GridView WHERE part of the sql statement« textbox.
76. Click »Update«.

We created main grid that will allow users to add, edit and delete records. In step 44-58 we limited albums user can see to those created by user only (every user can see only its own albums). In steps 59-63 we created automated value (userID) which will be stored by every user's insert. In steps 64-73 we created automated values for date of creation and date of modifying.

At this time each records doesn't have associated files (photos, videos or documents) yet. IndooGrid stores files using binary columns (for all binary data). In next steps we will add binary column so each record will have one file associated.

From this point on we will create Photo Library. However, grids that are already made can also be used to create Video library, Document Library or library of other types of files.

1. Click on »Table settings«.
2. Click »Add binary columns wizard« link.

3. Select »File (binary) content is stored into database column« from »File (binary) location« dropdown list.
4. Click »Next«.

In this step we defined how will files be stored on the server. Quick tip would be to choose database for smaller libraries and file system for large libraries. Please read about »Add binary (file) columns« section for more on this option (And options configured in following steps).

5. Select »Pictures« from »Select the type of uploaded files« dropdown list.
6. Click »Next«.

In this step we defined what kind of files will be stored on the server. As mentioned before, this procedure can also be used to different libraries than for photos only.

7. Select »Image thumbnail« from »Show binary as (see help icon)« dropdown list.
8. Click »Next«.
9. Enter »60« in »Max thumbnail width« textbox.
10. Enter »60« in »Max thumbnail height« textbox.
11. Enter »gif, jpg, tiff, png« in »Allowed extensions list« textbox.
12. Click »Next«.
13. Click »Finish« button.
14. Click twice on up (arrow) icon of ID column (ninth row of first list) in gridView column list.
15. Enter »1« in JavaScript dialog box.
16. Click »Ok« button of JavaScript dialog box.
17. Click »Update«.

Add some records (using insert icon) and play around with them to see possibilities user have when uploading and manage photos. NOTE: If you don't enter allowed extensions list, files with ALL extensions will be allowed to upload. For security purposes you should avoid that and always enter allowed extensions to upload.

It is obviously that this grid needs some cosmetic changes. Follow these steps:

1. Click on »Table settings«.
2. Select value »0« of Filter dropdown list of gridView for following fields: ID, UserID, SortOrder, DateCreated, DateModified.
3. Click on edit icon near »DateModified« column under gridView section.
4. Enter »d;en-us« in »Format data« textbox.
5. Click »Next«.
6. Click »Next«.
7. Enter »Picture« in »Caption« of first row of gridView column list (ID column).
8. Enter »Title« in »Caption« of Title column in gridView column list.
9. Enter »Sort« in »Caption« of SortOrder column in gridView column list.
10. Enter »Date modified« in »Caption« of DateModified column in gridView column list.
11. Enter »Sort« in »Caption« of SortOrder column in gridEdit column list.

12. Enter »Picture« in »Caption« of file_content1column in gridEdit column list.
13. Click »Update«.

From here now on should create Photo Gallery to show pictures uploaded by users (we decided that only registered users can upload pictures) to all users (including those who aren't logged in and registered within DotNetNuke).

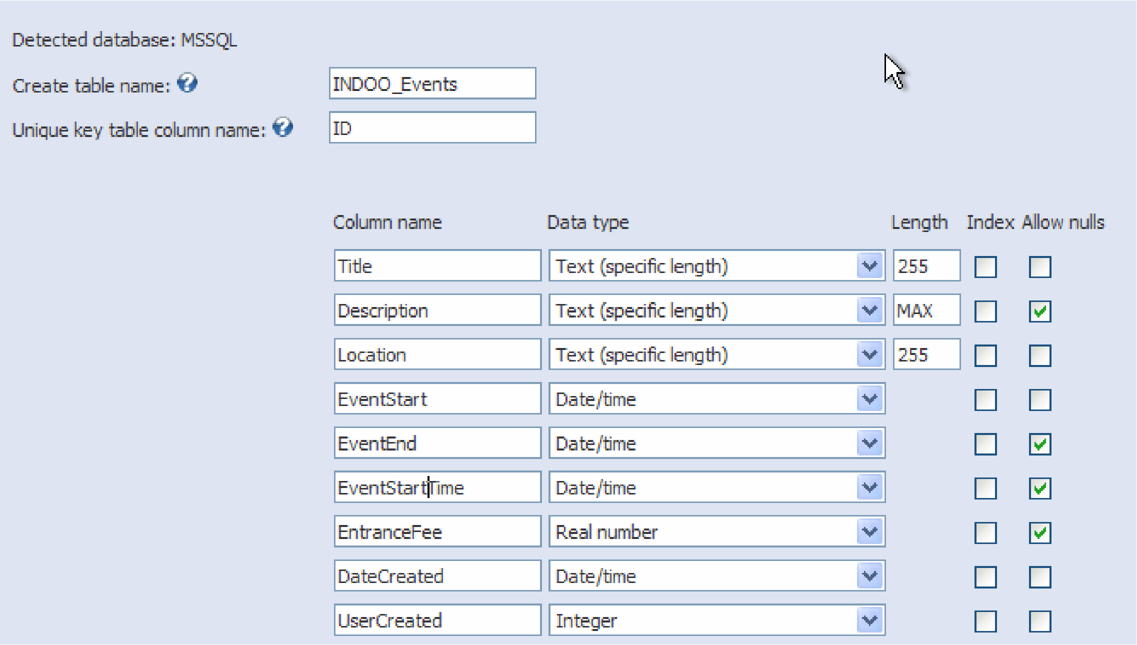
This gallery can look similar to the gallery we made in previous chapter (»Media Gallery«) but it will use tables we created in this chapter.

Events

In this example we will create Event module. Events module will display events from the database to all users allowing users to browse events and search for specific event, Event management module will allow us (as administrators) to manage events entered by users and Add new event module will allow users to enter events. Of course you can use different logic (for example to allow only administrator users to view events or to allow all users to post events) when working with events but in this sample we will use common approach.

In first part we'll create Manage events module that will allow us to manage all events. Create new page and add IndooGrid module (title »Event management«) to the page. Allow view (permissions under Module Settings) to administrators only.

1. Click on »Table settings« link.
2. Click on »Table (or view)« link or wizard icon.
3. Select »CREATE table« radio button and click »Next«.
4. Enter following data in Create table form:



Detected database: MSSQL

Create table name: INDOO_Events

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
Title	Text (specific length)	255	□	□
Description	Text (specific length)	MAX	□	☑
Location	Text (specific length)	255	□	□
EventStart	Date/time		□	□
EventEnd	Date/time		□	☑
EventStartTime	Date/time		□	☑
EntranceFee	Real number		□	☑
DateCreated	Date/time		□	□
UserCreated	Integer		□	□

Picture 23: Creating Events table

5. Click »Next« when finished.
6. Click on wizard icon near »Unique key column«.
7. Click »Automatically add all columns« link.
8. Check following checkboxes: Allow bulk export, Enable insert records, Enable edit record, Enable delete record, Hide gridView in edit/detail mode, Edit buttons in last column, Left label position.
9. Click »Update«.

This grid allows us to manage events entered by users. Note that you don't have to create table (steps 3-5) if you already have Events table in your database (if you made table once or you have other module including events).

Next step is creating a form which will allow users to enter events. Add IndooGrid module to the page (title »Add new event«) and set its view permission to »Registered users« within Module settings.

1. Click on »Table settings« link.
2. Enter »INDOO_Events« in »Table (or view)« textbox.
3. Click on wizard icon near »Unique key column«.
4. Click »Automatically add all columns« link.
5. Check following checkboxes: Enable insert records, Hide gridView in edit/detail mode, Left label position.
6. Select »insert« from »Default GridEdit mode« dropdown list.
7. Uncheck »Visible« checkboxes of gridEdit column list (second column list) for following fields: DateCreated, UserCreated.
8. Click on edit icon near »Description« column under gridEdit section.
9. Click »Next«.

10. Enter »300« in »Width« textbox.
11. Click »Next«.
12. Enter »10« in »Multiline« textbox.
13. Click »Next«.
14. Click on edit icon near »EventStart« column under gridEdit section.
15. Enter »Start of event« in »Caption« textbox.
16. Click »Next«.
17. Check »Add calendar« checkbox.
18. Enter »d;en-us« in »Format data« textbox.
19. Click »Next«.
20. Check »Custom validator« checkbox.
21. Click »Next«.
22. Select »Required« from first »Validator« dropdown list.
23. Click »Next«.
24. Click on edit icon near »End« column under gridEdit section.
25. Enter »End of event« in »Caption« textbox.
26. Click »Next«.
27. Check »Add calendar« checkbox.
28. Enter »d;en-us« in »Format data« textbox.
29. Click »Next«.
30. Click »Next«.
31. Click on edit icon near »EventStartTime« column under gridEdit section.
32. Enter »Starting time« in »Caption« textbox.
33. Click »Next«.
34. Enter »t;en-us« in »Format data« textbox.
35. Check »Drop down list or radio button list (join for view mode)« checkbox.
36. Click »Next«.
37. Select »Values from list« from »Select list source« dropdown list.
38. Select »Dropdown list« from »Show list as« dropdown list.
39. Click »Next«.
40. Enter following code to »List values and text« textarea

```
0:30 AM;0:30 AM
1:00 AM;1:00 AM
1:30 AM;1:30 AM
2:00 AM;2:00 AM
2:30 AM;2:30 AM
3:00 AM;3:00 AM
3:30 AM;3:30 AM
4:00 AM;4:00 AM
4:30 AM;4:30 AM
5:00 AM;5:00 AM
5:30 AM;5:30 AM
6:00 AM;6:00 AM
6:30 AM;6:30 AM
7:00 AM;7:00 AM
7:30 AM;7:30 AM
8:00 AM;8:00 AM
8:30 AM;8:30 AM
9:00 AM;9:00 AM
9:30 AM;9:30 AM
```

```
10:00 AM;10:00 AM
10:30 AM;10:30 AM
11:00 AM;11:00 AM
11:30 AM;11:30 AM
12:00 AM;12:00 AM
0:30 PM;0:30 PM
1:00 PM;1:00 PM
1:30 PM;1:30 PM
2:00 PM;2:00 PM
2:30 PM;2:30 PM
3:00 PM;3:00 PM
3:30 PM;3:30 PM
4:00 PM;4:00 PM
4:30 PM;4:30 PM
5:00 PM;5:00 PM
5:30 PM;5:30 PM
6:00 PM;6:00 PM
6:30 PM;6:30 PM
7:00 PM;7:00 PM
7:30 PM;7:30 PM
8:00 PM;8:00 PM
8:30 PM;8:30 PM
9:00 PM;9:00 PM
9:30 PM;9:30 PM
10:00 PM;10:00 PM
10:30 PM;10:30 PM
11:00 PM;11:00 PM
11:30 PM;11:30 PM
12:00 PM;12:00 PM
```

41. Enter »8:00 PM« in »Default value« textbox.
42. Check »Sort by value« checkbox.
43. Click »Next«.
44. Add Required validator - repeat steps 8-17 (except the input for »Format data« and »Add calendar« fields) for »Title« and »Location« columns (in gridEdit).
45. Click on edit icon near »EntranceFee« column under gridEdit section.
46. Enter »Entrance fee« in »Caption« textbox.
47. Click »Next«.
48. Enter »c;en-us« in »Format data« textbox.
49. Click »Next«.
50. Check »Custom validator« checkbox.
51. Select »Required« from first »Validator« dropdown list.
52. Click »Next«.
53. Click on edit icon near »DateCreated« column under gridEdit section.
54. Click »Next«.
55. Enter »[indoogrid:CurrentDate]« in »Field value« textbox.
56. Click »Next«.
57. Click on edit icon near »UserCreated« column under gridEdit section.
58. Click »Next«.
59. Enter »[User:UserID]« in »Field value« textbox.
60. Click »Next«.

61. Enter »inok=»indoogrid:idvalue«» in »Add parameters to gridEdit redirect URL« textbox.
62. Enter »inok« in »Remove parameters from gridEdit redirect URL« textbox.
63. Enter following code in »Module header text« textarea:

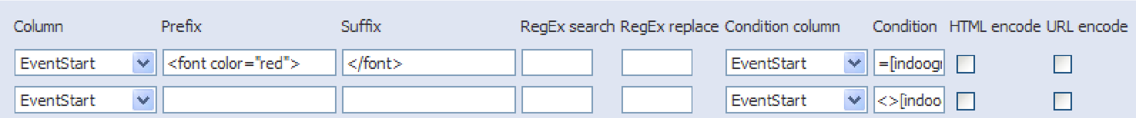
```
<script language=»javascript«>
var insertOk = Number('[numericQueryString:inok]');
if (insertOk > 0) {
   document.write('<font color=»red«>New event was succesfully
   added. Thank you for participating.</font>');
}
</script>
```

64. Enter »yourmail@mail.com« in »Send gridEdit summary to mail« textbox (replace your_mai@mail.com with your mail where you want the notification be sent to.
65. Click »Update«.

In last part we will create grid that will display events entered by (registered) user to all users. Add new IndooGrid module (title »Events«) to the page.

1. Click on »Table settings« link.
2. Enter »INDOO_Events« in »Table (or view)« textbox.
3. Click on wizard icon near »Unique key column«.
4. Click »Automatically add all columns« link.
5. Check following checkboxes: Allow bulk export, Enable detail view, Hide gridView in edit/detail mode, Edit buttons in last column, Left label position.
6. Uncheck »Visible« checkboxes of gridView column list (first column list) for following fields: Description, EventEnd, EntranceFee, DateCreated, UserCreated.
7. Uncheck »Export« checkboxes of gridView column list (first column list) for following fields: DateCreated, UserCreated.
8. Select »0« in »Filter« dropdown list of gridView for following fields: DateCreated, UserCreated.
9. Click on edit icon near »Title« column under gridView section.
10. Enter »100;« in »Format data« textbox.
11. Check »Column contents are URL links« checkbox.
12. Click »Next«.
13. Select »view« from »Add command parameters for gridEdit« dropdown list.
14. Click »Next«.
15. Click on edit icon near »Location« column under gridView section.
16. Check »Column contents are URL links« checkbox.
17. Click »Next«.
18. Enter following code in »URL« textbox: »http://maps.google.com/maps?f=q&hl=sl&geocode=&q=&ie=UTF8&z=12&iwloc=addr&om=0«
19. Enter »q« in »Parameter name« textbox.
20. Select »Location« from »Select column for parameter value« dropdown list.
21. Enter »_blank« in »Target« textbox.
22. Click »Next«.

23. Click on edit icon near »EventStart« column under gridView section.
24. Enter »Start of event« in »Caption« textbox.
25. Enter »d;en-us« in »Format data« textbox.
26. Click »Next«.
27. Click »Next«.
28. Click on edit icon near »EventEnd« column under gridView section.
29. Enter »End of event« in »Caption« textbox.
30. Enter »d;en-us« in »Format data« textbox.
31. Click »Next«.
32. Click »Next«.
33. Click on edit icon near »EventStartTime« column under gridView section.
34. Enter »Starting time« in »Caption« textbox.
35. Enter »t;en-us« in »Format data« textbox.
36. Click »Next«.
37. Click »Next«.
38. Click on edit icon near »EntranceFee« column under gridView section.
39. Enter »Entrance fee« in »Caption« textbox.
40. Enter »c;en-us« in »Format data« textbox.
41. Click »Next«.
42. Uncheck »Visible« checkboxes of gridEdit column list (first column list) for following fields: DateCreated, UserCreated.
43. Click on edit icon near »EventStart« column under gridEdit section.
44. Enter »Start of event« in »Caption« textbox.
45. Enter »d;en-us« in »Format data« textbox.
46. Select »3 – Always on as dropdown list« from »Allow filter« dropdown list.
47. Select »1 - configurable« from »Show data from more columns (custom HTML)« dropdown list.
48. Click »Next«.
49. Enter following data (as shown on the picture) to table:



Column	Prefix	Suffix	RegEx search	RegEx replace	Condition column	Condition	HTML encode	URL encode
EventStart	<font color="red">	</font>			EventStart	=[indoogı	□	□
EventStart					EventStart	<>[indoo	□	□

Picture 24: Using conditions to format gridView

Note that in column »Condition« values are »=[indoogrid:currentDate]« (in first row) and »<>[indoogrid:currentDate]« in second row.

50. Click »Next«.
51. Select »Values from list« from »Select list source« dropdown list.
52. Check »Auto execute on change« checkbox.
53. Click »Next«.
54. Enter following code in »List values and text« textarea:

```
>=[indoogrid:currentDate];Upcoming events
=[indoogrid:currentDate];Today events only
<>00;All events
```

55. Uncheck »Add all« checkbox.
56. Enter »>=[indoogrid:currentDate]« in »Default value« textbox.
57. Click »Next«.
58. Click »Next«.
59. Click on edit icon near »EventEnd« column under gridEdit section.
60. Enter »End of event« in »Caption« textbox.
61. Click »Next«.
62. Enter »d;en-us« in »Format data« textbox.
63. Click »Next«.
64. Click »Next«.
65. Click on edit icon near »EventStartTime« column under gridEdit section.
66. Enter »Starting time« in »Caption« textbox.
67. Click »Next«.
68. Enter »t;en-us« in »Format data« textbox.
69. Click »Next«.
70. Click »Next«.
71. Click on edit icon near »EntranceFee« column under gridEdit section.
72. Enter »Entrance fee« in »Caption« textbox.
73. Click »Next«.
74. Enter »c;en-us« in »Format data« textbox.
75. Click »Next«.
76. Click »Next«.
77. Click on »Show advanced settings« checkbox.
78. Enter »EventStart ASC, EventStartTime ASC« in »GridView ORDER BY part of the sql statement« textbox.
79. Click »Update«.

Core Reference

Basic settings

Selecting db provider

Use this option to use appropriate client. Default provider is MS SQL native client for current DotNetNuke database (used by your current DNN installation). You can use MS SQL and Oracle native clients or ODBC for all databases (MS SQL, Oracle, MySQL, MS Access, etc.). The module is designed to provide almost same performances for ODBC clients as for the native clients.

Access to current DNN database can be limited by administrator within the module settings.

Connection string

If you leave this field blank, default connection string for current DotNetNuke database will be used. For other providers you can use wizard icon to test connection to the database used. You can enter either local or remote database server address.

If you don't know how to use connection string, try this tutorial:

A Beginners Guide to the Connection String [1].

Example of Microsoft SQL Server string using ODBC provider

If you want to connect to remote MS SqlServer, use following connection string:

```
Server=your_server_address;Database=database_name;uid=username;pwd=p
assword;
```

If you are trying to connect to SqlServer 2000, use »pre-2005« db provider.

Please note that if you do not connect to remote server, you should use »Current DotNetNuke database« provider option.

Example of MySQL connection string using ODBC provider

```
DRIVER={MySQL ODBC 3.51 Driver};SERVER=server_address
;DATABASE=database_name;UID=username;PASSWORD=password;Port=port;OPT
ION=3
```

MySql ODBC driver should be installed within server environment.

Example of Oracle connection string using ODBC provider:

```
Driver={Oracle in
XE};;Server=server_address;Uid=username;Pwd=password;QTO=F
```

where »Oracle in XE« is the name of ODBC driver. Check if Oracle client is installed on same machine. You also use Oracle native provider with connection string like this:

Data Source=MyOracleDB;User Id=myUsername;Password=myPassword;Integrated Security=no;

You can find more about Oracle connection strings here How to connect to Oracle [2].

Example of Access connection string using ODBC provider:

```
Driver={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ=C:\your_path\database_name.accdb;DriverId=25;FIL=MS
Access;
```

Your server should have installed appropriate ODBC driver, which can be found on Microsoft download page [3]. For more information on how to work with Access please take a look at »Crate MDB file« feature.

Table (or view)

Select source table or view to use with the grid.

IndooGrid is always related to the table stored in database. Therefore first step when configuring grid is always to select appropriate table we will work with. Besides selecting the table as we have done in previous example IndooGrid also offers some additional options to work with tables using grid interface only.

Click on »Table (or view)« wizard on the »Table settings« page to select, create, delete or modify table. Select one of the radio button options and click »Next« to continue. There are 5 radio buttons options.

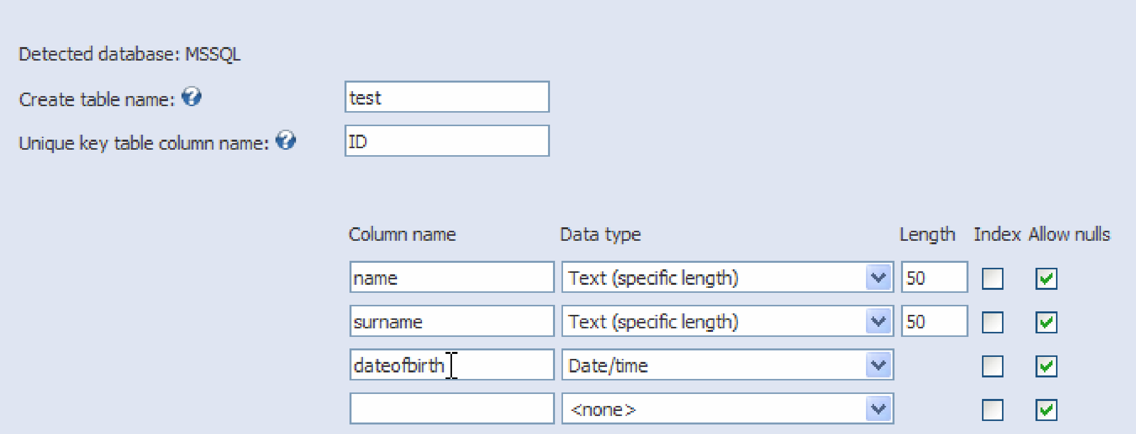
Using »Select existing table« you can select database table which already exists in the database. Simply choose from list box where all existing tables are displayed and click »Next«. When list of tables is not available due to database or permission limitation a regular textbox is available (in such case you have to enter table name manually).

Using »Create table« you can add new table. The wizard will display settings page with properties for field you should enter.

In first row wizard will show detected database, which you have selected in previous steps. Enter name of new table in »Create table name«. Name should be unique as there cannot be two tables with same name in the database.

Enter unique key column (identifier) in »Unique key table column name« filed. This column is used as identifier and will be unique for each record in the table. Common name for such column is simply »ID« or »tablename_ID« (where tablename should be replaced with the name of newly created table).

Bellow is set of fields to add basic columns to the table where each row represents data table column. »Column name« and »datatype« are required while »length«, »index« and »null« properties are optional. Data types are common representation of basic database column type, but in the background depend on database you're using (MS SQL Server, Oracle etc.) so for further information you should examine the database documentation.



Detected database: MSSQL

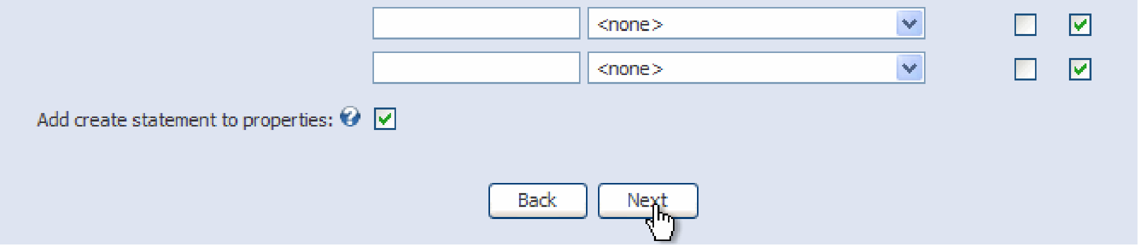
Create table name: test

Unique key table column name: ID

Column name	Data type	Length	Index	Allow nulls
name	Text (specific length)	50	□	☑
surname	Text (specific length)	50	□	☑
dateofbirth	Date/time		□	☑
	<none>		□	☑

Picture 25: Creating table

If you check »Add create statement to properties«, script with data table structure will be saved and included when you'll export/import the module properties.



Picture 26: Module can store CREATE statements and export them for copying

Click »Next« and new database table will be created.

Following example will show how to create new table in default DotNetNuke database using step-by-step procedure:

1. Click on »Table (or view) » wizard icon.
2. Select »Create table« radio button and click »Next«.
3. Enter »test« in »Create table name« field.
4. Enter »ID« in »Unique key table column name«.
5. In first row of the column list enter »name« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
6. In second row of the column list enter »surname« in »Column name« field, select »Text (specific length)« in list box and enter »50« in »Length« field (default value is MAX).
7. In third row of the column list enter »dateofbirth« in »Column name« field and select »Date/Time« in list box.
8. Click »Next«. You will be returned to »Table settings« page.

Using »Alter table« you can change data table properties. In first step you select the table you want to change from list box and click »Next«. The table properties page will appear. In first row data table name is displayed. In second row (»Select action«) you can choose type of command (action) from the list box. There are three possible actions: ADD (column), ALTER (column) and DROP (column). In »Column name« field enter the name of the column you want to modify. In »column type« select new data type for newly added or modified column.

You can also change »Allow null« and »Create index« checkboxes if needed, for specific columns you can also enter »length«.

Check »Primary key and auto increment« if you want to set the column as identifier (unique column key).

Using »Add alter statement to properties«, alter script with data table structure will be saved and included when you'll export/import the module properties.

In the table on the bottom you have list of all existing columns in selected table. Some properties in the table cannot be displayed for specific databases. For example, in Oracle auto increment properties is implemented as trigger so it will not appear in the table. Click »Alter table« to confirm changes.

Using »Drop table« you can delete database table. Select table from the list box and click »Next«. Confirmation dialog will appear. Confirm and table will be deleted. NOTE: Be careful. This is irreversible operation and there's no way to retrieve dropped table.

»Show column info« simply displays all column information about selected table. This is especially useful when you want to view table structure for developing purposes. Select table from the list box and click »Next«. Table with all .NET column metadata will be displayed (some properties cannot be always correct displayed as described above). Using list box and clicking »Show« you can simply select another data table.

When you have finished working with tables, always click »Update« at the bottom of the page to confirm and save changes made.

Unique key column

Module requires database column that contains unique integer values as »Unique key column«. Some tables does not have appropriate column to use as unique identifier for use with module. In such cases the module will offer creating new such column or selecting one of existing columns (for which you can be sure that it is unique). Unique key column should be auto increment type for inserting records. Module tries to detect such column or can create new one. When using Oracle database module also create adequate sequence.

You can either use wizard to select appropriate unique key column or enter it manually. If you have selected wizard icon, module will try to find unique identifier for the selected table itself. In this case, the appropriate column name will appear in the field. If this field is not empty, new page will appear where you should manually select the identifier column from the list box. Recommended column will have » (recommended) » text displayed near the column name.

Automatically add all columns

When clicking this link all data table columns will be added to gridView and gridEdit lists. If columns are already in the grid, only missing columns (for current table) will be added. Binary columns are ignored and will not be added to the list (unique key column will also not be added to the list). Duplicated and/or binary columns add manually.

When configuring the grid it is normal procedure to use this link to add all existing columns to the grid and the remove (or uncheck the »Visible« property) the columns you do not need.

Remove all columns

This option will remove all existing columns from both gridView and gridEdit lists. It will also clear all other settings made to the current module.

Templates (Import – export)

All settings of the module can be imported or exported. You can save them to *.txt file or *.xml document and store them for test or backup purposes. This feature is also very useful when copying modules.

You can use your own templates or download templates from our pages (go to Demo pages on module's homepage). If you want to try our templates (examples on our Demo pages), follow these steps:

1. Download settings file of example you have selected (on our Demo pages).
2. Install IndooGrid on your DNN portal.
3. Add new page to your DNN portal.
4. Add new IndooGrid to your page.
5. Go to Module settings page and click »Activate trial« if you didn't done that before.
6. Click on »Table settings« link of IndooGrid module.
7. Click on »Import-export« or wizard icon near link.
8. Click on »Browse« button.
9. Select the file you have downloaded (in step 1).
10. Check the »Execute sql(s)« on import checkbox.
11. Click »Import« button.
12. Click »Finish«.
13. Click »Update«.

Please also read Execute sql(s) on import section of this chapter on details of creating appropriate tables.

By clicking on Import-export link (or wizard icon near-by) import-export page will be displayed.

Download settings (file)

Use this link to download current module's settings. Download file (data are in .txt format) and save it to the file system.

Upload module settings (file)

Use this option to select file with module settings and import it to the current module.

1. Click on »Browse« button.
2. Select the file (in .txt format) with module settings.
3. Click on »Import« button.
4. Click »Finish«.

After importing notification will be displayed (in red at the bottom of the module).

Import module settings (text)

Use this option to manually copy settings to or from the module.

To import settings follow this procedure (let's say we have settings stored in file):

1. Open script file (txt).
2. Select ALL content (using CTRL+A or using SHIFT + mouse/keys) and copy it to clipboard (using CTRL+C or mouse).
3. Paste the content from the clipboard to the »Import module settings (text)« text area field (using CTRL+V or mouse).
4. Click »Import«.
5. Click »Finish«.

To import settings follow this procedure (let's say we want store settings to the file):

1. Click »Export«.
2. Select ALL content in »Import module settings (text)« text area (using CTRL+A or using SHIFT + mouse/keys).
3. Copy content (using CTRL+C or mouse).
4. Open new text document.
5. Paste the content to the text file (using CTRL+V or mouse).
6. Save document as »yourfilename.txt« or »yourfilename.xml« file.

Execute sql(s) on import

Use this option to allow execution of SQL procedures if they are included in the settings. Leave this option disabled if you're not sure.

When you're creating tables using module, you have additional option when creating table called »Add SQL statement to the properties«. This option creates SQL procedures that are stored within module settings.

When you want to copy the module to other DNN installation or if you download the samples from our template pages, enable this option and database table (for which SQL procedure is stored within settings) will be created on import. Rule should be: if table that module settings are based on already exists in your database, disable this option. If you want table to be created on import (table based on these settings doesn't exists yet), enable this option.

Note: Disable this option when settings file or text does not come from trusted source and you did not check it for potential malicious SQL code. Unfortunately this also can cause that db structure is not prepared for module settings. You can verify statements manually before import – find out »sqls« xml node in settings file or plain text. SQL statement(s) that are executed on module import can be later edited using »Edit sql(s)« option within »Advanced options« section.

GridView settings (gridView properties)

Allow bulk import

Checking option »Allow bulk import« the end user will have ability to import data using advanced options, for example importing data from clipboard or MS Excel application. This feature is very useful when user has to execute mass update or insert of data. You can mark individual column as (non)importable by (un)checking column property »Import«.

User can import data using »Import« link at the top of the grid. Data can be imported either in tab delimited format (values separated by spaces, for example when using copy/paste from Microsoft Excel) or in XML format (for example when importing data from file). When importing tab delimited data first row must contain column headers. Order of data is not important as import is implemented using column headers to match columns.

NOTE: Updating or inserting data requires »Unique key column«.

Default action when importing data is »Insert«. If the user selects »Update«, the selection pasted in the text area field must contain unique key column (identifier) to match the records already existing in the data table.

Insert (update) is relatively slow process and it is not intend to insert more than couple of thousand rows at once. In case of failure red error message will be displayed on the screen.

NOTE: When importing data using bulk import all fields' validators will be ignored.

Import feature is disabled by default. Current version of module does not supports import/export binary (files: pictures, documents, etc.) data yet.

After completing the operation the message will appear showing how many records were successfully updated. If operation fails, rollback will be executed automatically (if database supports transactions).

If you're not sure how to import data, export the existing data first, examine the datable structure, change data and import data back to the data table. This is example of structure for Users table (superuser account only):

UserID	Username	FirstName	LastName	IsSuperUser	AffiliateId	Email	DisplayName	UpdatePassword	Username
1	host	SuperUser	Account	True		You@mail.com	SuperUser Account	False	host

Allow bulk export

Checking option »Allow bulk export« the end user will have ability to export data using advanced options, for example export data to clipboard or file or to MS Excel application. You can mark individual column as (non) exportable using column property »Export«.

User can export data using »Export« link at the top of the grid. Selection of data to export can be managed using filters as described in »Filtering data« sections. By clicking the export link, user can export data in following formats:

- Tab delimited in clipboard (for Office etc) – data will be separated by tabs and will include the first row with column headers. This is most common option for exporting data to Microsoft Excel or Word.
- Tab delimited into browser window
- Tab delimited into file
- HTML in clipboard – data will be exported as HTML table including column headers. This option is commonly used for quick preview of data.
- HTML into browser window
- HTML into file
- XML in clipboard– data will be exported as XML formed document. This option is often used to export to another application
- XML into browser window
- XML into file
- Excel application
- Word application

Users can define the target of exportation – data can be exported to new window, file (this option is usually used when we want save export file, for example XML file) or on the clipboard (this option is usually used to export to other applications, for example in Microsoft Word or Excel).

Export feature is disabled by default. It can be enabled (disabled) by checking checkbox »»Allow bulk export«» on »Table settings« page.

Each individual column can be marked as exportable by checking column property »Export« to true (by default all columns are marked as exportable) – either in gridView section or in each columns individual settings.

Allow bulk delete

By enabling option »Allow bulk delete« end user will have the ability to delete multiple records at once. At he top of the grid new link »Delete« will appear – by clicking that link all records in the grid will be deleted (including records on other grids pages if the grid has more than one page).

NOTE: Although this option is listed under gridView settings, you should read the »Adding, editing and deleting data« chapter before enabling it – as you should do before enabling »Enable edit record«, »Enable inline edit« or »Enable insert records« options.

Enable edit record

By checking option »Enable edit record« command column containing edit icon will be added to the grid. This will allow user to edit each individual record. The appearance and behavior of each individual column depends on gridEdit settings (gridEdit properties will be applied).

Enable inline edit

By checking option »Enable inline edit« users will have ability to edit multiple records without submitting the page. The columns that are listed in both gridView and gridEdit list and have the same column names will be replaced by editable fields. Although columns can look the same as in view mode, they can be edited. The appearance and behavior of each individual column depends on gridEdit settings (gridEdit properties will be applied instead of gridView properties).

Enable delete record

By checking this option command column containing delete icon will be added to the grid. This will allow user to delete each individual record. The appearance and behavior of each individual column depends on gridEdit settings (gridEdit properties will be applied).

Enable detail view

By default users see the list of records. Normally we do not show all the data for each individual record in such list because of transparency and lack of space on the screen. By checking this option command column containing detail view icon will be added to the grid. By clicking this icon user will see only the data for the selected record (normally we would list all the records fields on such page). User cannot modify but just view record.

Enable insert records

Allows user to add (insert) new records. After enabling this option two insert icons will be added to the grid both providing same functionality. First icon will appear in the header of the command column and second one will appear at the bottom of the grid. When user clicks on the insert icon, new page (form) for inserting record will open.

Hide gridView

This option will hide complete gridView. However, it will not hide the additional content defined in »Advanced options« or gridEdit. For example, use this option when you want to manually display data from the database on the page or when you want to show gridEdit or nothing (depending on users action).

Record count

By default at the top of the grid number of records (stored in the table) is displayed. There are three options you can choose from:

- Exact will show the number of all records in the data table. If any filters applied to the grid it will show the number of all filtered records. Additional (single) database query is executed when this option is enabled.
- Description
- Hide will not show number of records

Records per page

You can specify number of records displayed on each page of the grid. You have following options available:

- Fixed page size - enter single value to define fixed page size. Example: »10« shows ten records per page with no custom options for enduser.
- Custom page size – enter single value inside brackets to allow users to enter custom value. This is not recommended when in-line edit option is enabled because this can hit servers performance when more hundreds records is shown on same page. Example: »(10)« shows ten records per page with textbox for custom page size for end-user
- Dropdownlist for page size – enter separated values. Dropdownlist with auto execution (when value is selected it is automatically applied) will be displayed. Example: »10,50,100,250« shows dropdown list with optional page sizes for 10, 50, 100 and 250 records per page, default value is 10. »10,(50),100,250« give the same results as previous example but with default value 50 records per page.

When custom option (textbox or dropdownlist) is available for end-user then value is stored in cookie – in this case is default last chosen value (when is available).

When single value is positive with + (plus character) at beginning then paging is in previous-next mode. Example: »+12« is paging in previous-next mode. When single value is negative then no paging is available. Example: »-10« is page size 10 but without options for next pages. When you want to show all records (without paging) then you can set very large page size - like »999999« for example.

Add new view column

Use this wizard to manually add new gridView column. You can add unlimited number of columns – this is especially useful when you need column values in for advanced options.

GridView settings (gridView column list properties)

These settings are displayed in the gridView column list under gridView settings. They can be applied by entering the value (caption) or by (un) checking the checkboxes for each column. Changes are saved by clicking on »Update« button at the bottom of the »Table settings« page.

Caption

Enter caption for each column. If no caption is entered (default), database column name will be used. You can use tokens and HTML formatting in this field, for example enter

```
<font color=red>MyCaption</font>
```

to display column header in red. For more information on tokens please take a look at »Working with tokens« section of this document.

Filter

By default end users can filter all data in the grid. Users can also filter data using multiple filters and value expressions.

We will show example simulating end users action (use the previous example based on Users table – as user will see the grid):

1. Select »Firstname« in »Filter« list box

2. Enter »SuperUser« in »Filter« textbox.
3. Click on »Filter« link.
4. Select »Lastname« in »Filter« list box
5. Enter »Account« in »Filter« textbox.
6. Click »Find it«.

As you can see data are now filtered. Clicking again on »Filter« link in first row user can add another filter while clicking on the »Filter« link in last row (in our example this is second row), last filter (second) is removed.

Except of specific value we can also use expressions for filtering data. Dates and numbers can be filtered in different ways using special characters (<,>, <=, >=,=). Here are some examples:

- >1.1.2007
- >1.1.2007 < 1.2.2007
- >=256

Note that in databases dates are usually stored as Datetime fields which means that date is stores together with time.

When filtering text we can use only part of the text. For example, filtering data using »port« will return all records containing »port« in specified field: port, portal, portals will all be included in the selection.

Filter value can be removed by removing content from the filter value field and clicking »Find it«.

TIP: By clicking on the arrow icon near »Find it« button quick help will be displayed. Help also contains examples.

Data can be also filtered simultaneously on more than one field. For this example we will use User table from previous example and show how to define search on three fields at the time.

1. Click »Table settings« link.
2. Click on edit icon of the »Username« field in gridView section.
3. Enter »Firstname, Lastname« in »Filter additional column list« field.
4. Click »Next«.
5. Scroll down and click »Update«.

When end user filters data using »Username« field, grid will also use data in fields »FirstName« and »Lastname«. For example, if user enters »ABC«, all records containing »ABC« in »Username« column or »Firstname« column or »Lastname« column will be displayed. This is useful for composite search like article search for example; when end-user filter (searches the values) title and description at same time.

Advanced filtering options

Although end user has built-in options to use filters, we should use advanced filtering options to limit records and to help user with shortcuts.

Every gridView column has six options on how it will be used for filtering. You can set these options in gridView column settings (if you click edit icon near the gridView column you want to configure) or on »Table settings« page – both ways will lead to the same results. Indicator of filter type used is displayed in gridView columns list as Filter. If this indicator is followed by asterisk (*) you'll know that composite search is used for this filter field (additional filter columns).

Please note that selecting options 3 or 5 on »Table settings« page requires additional configuration on column property page. If you'll not define these additional settings, such filter will act as option 2 (instead of 3) and 4 (instead of 5).

Field »Filter« (or field »Allow filter« if you're setting column properties using edit column icon) offers following options to select:

0 – Disable

Using this option column name will not be displayed in general filter dropdown list where users can select which field they want to filter.

1 – Enable

Using this option column name will be displayed in general filter dropdown list. Users can select this column to filter and enter search string (phrase) in filter textbox. This is default value of »Filter« (»Allow filter«) dropdown list (by default all columns will be displayed and allowed to use for filtering)

2 – Always on as textbox

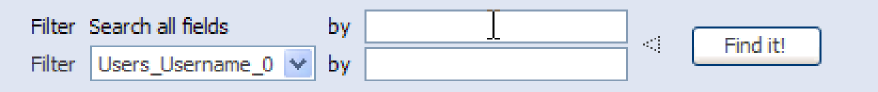
If you want to display filter textbox for specified column only, select this option. This option is also very useful to create general search field.

We'll now show how to display general search field - for purposes of this example we will use Users table from previous sample (»Setting up the module« chapter).

1. Click on »Table Settings«.
2. Click on »Add new view column« link.

3. Select »Username« in »Column name« dropdown list.
4. Enter »Search all fields« in »Column caption« field.
5. Enter »FirstName, LastName, Email, DisplayName« in »Filter additional column list«.
6. Select »Always on as textbox« from »Allow filter« dropdown list.
7. Scroll down and click »Update«.

This is the picture you should see:



Picture 27: Always on as texbox option

If you enter »s« in »Search all fields« field and click »Find it! » button, all records that contain »s« in any of the fields we defined (in »Username« because base column selected is Username and in »FirstName«, »LastName«, »Email«, »DisplayName« because we entered them as additional filter fields) will be displayed. In this case we used only text columns but we could also use date or integer columns or mix them all together.

3 – Always on as dropdownlist

If you want to help users and/or limit their ability to filter records you should use this option. We will show two examples of how to use it – in first one we will use existing table for predefined values and in the second we will enter list of values manually.

In first example we will show how to allow users to filter specified field only by values that actually exist. We will use table UserRoles for purposes of this example.

1. Add new IndooGrid module to the page.
2. Click on »Table Settings«.
3. Either enter the »UserRoles« in »Table (or view) » field or click on wizard icon near the field to select this table.
4. Click on wizard icon near »Unique key column« field.
5. Click on »Automatically add all columns« link.
6. Click on edit icon near »UserID« gridView column.
7. Enter »Username« in »Column caption« field.
8. Select »Always on as dropdownlist« from »Allow filter« dropdown list.
9. Check »Join another table data on this column«.
10. Select »Users« from »Join data« dropdown list.
11. Click »Next«.
12. Select »UserID« in »Join table on column« dropdown list.
13. Select »Username« in »Select column content« dropdown list.

14. Click »Next«.
15. Select »Values from database table« in »Select list source« dropdown list.
16. Check »Auto execute on change« checkbox (if you check this checkbox, the filter will be applied onSelect, if not user will have to click on Find it! button to apply filter).
17. Click »Next«.
18. Enter »Users« in »Source table or view« field.
19. Click »Next«.
20. Select »UserID« in »Value column« dropdown list.
21. Select »Username« in »Text column« dropdown list.
22. Click »Next«.
23. Scroll down and click »Update«.

When selecting »admin« in dropdown filter all records that have value »userID=0« will be displayed. You can see that only records with »admin« username are displayed.

It is important to explain difference between »View column wizard« (steps 9-14) and »View column wizard - select dropdownlist source for filter value« (steps 15-22). They are independent: first one is triggered by »Join another table data on this column« checkbox and defines how data will be displayed in grid (admin instead 0, if we wouldn't use this wizard in first column would be values 0,1,2 etc.). Second one is triggered by »Always on as dropdownlist« from »Allow filter« dropdown list and it defines the »Username« filter dropdown list on the top of the grid.

Try to do same steps for second column (RoleId, table should be Roles) to figure out the logic.

In second example we will show how to use this option to help users with their report. We will use EventLog table because it contains dates. You have probably seen this type of dropdown before (for example Google use it for almost all of its reports like Google Adwords reports).

1. Add new IndooGrid module to the page.
2. Click on »Table Settings«.
3. Enter »EventLog« in »Table (or view)« textbox.
4. Click on wizard icon near »Unique key column« field.
5. In »Set key column wizard« select »Create new column for me« and click »Next«.
6. Click on »Automatically add all columns« link.
7. Click on edit icon near »LogCreateDate« gridView column.
8. Enter »Date Created » in »Column caption« field.
9. Select »Always on as dropdownlist« from »Allow filter« dropdown list.
10. Click »Next«.
11. Select »Values from list« from »Select list source« dropdown list.
12. Check »Auto execute on change« checkbox.
13. Click »Next«.

14. Enter following lines in

```
>=[indooGrid:currentDate];Today
>=1/1/2007 <1.1.2008;Year 2007
<>00;All time
```

15. Uncheck the »Add ALL option« checkbox.
16. Enter »>[indooGrid:currentDate]« in »Default value« field.
17. Click »Next«.
18. Uncheck »Visible« checkbox for »LogProperties« gridView column (we will not show this field because normally it would extend wide of our screen too much)
19. Scroll down and click »Update«.

What have we done is enabling user to use predefined filters: if user selects Today, filter »more or equal currentDate« is applied (token [indooGrid:currentDate] returns current date in short date format). String »<>00« stands for resetting filter to none (like <>nothing_or_empty).

You can play with dates using token in format [indoogrid:currentdate:=Y,M,D,H,M]. For example, token [indoogrid:currentdate:=-1,0,1,0,0] will return current date minus 1 year + 1 day.

If you don't know how filter will behave (in source list) you can first try values in standard filter, in this case select LogCreateDate from filter dropdown list, enter value in textbox and click on »Find it!« button. Play with different values to get records you wants and than use same values in source list.

For more information on how to use tokens please read the »Working with tokens« section of this document.

4 – Always on as textbox (excluded from general filter list)

Using this option same result will be displayed as when using option 2 – the only difference is that this field (column) will not be available in general filter dropdown list by default.

5 – Always on as dropdown list (excluded from general filter list)

Using this option same result will be displayed as when using option 3 – the only difference is that this field (column) will not be available in general filter dropdown list by default.

Edit

If there's appropriate column (with same database column name) within gridEdit list then this column is editable. Although this property is listed in gridView column list, all fields properties are actually taken from gridEdit section and properties from this section are ignored (except width property). Column cannot be edited in-line when content of this column is command, URL or image. When column is editable use custom validators for more user friendly end-user experience.

This property will take effect only if »Allow inline edit« property of gridView settings is also checked (on). For more information read about the »Allow inline edit« and »GridEdit settings (gridEdit properties)« section

Export

Marks the column as exportable. This property will take effect only if »Allow bulk export« property of gridView settings is also checked (on).

Import

Marks the column as importable. This property will take effect only if »Allow bulk import« property of gridView settings is also checked (on).

Sort

By clicking on the header of the column user can sort data in the table by default. First click will sort data ascending; second click on the header will sort data descending. Current column sorted and the direction of sort is displayed near column header using arrow icon: up icon for ascending and down icon for descending.

By default all columns displayed to the user are sort able (if sort is supported by database) but we can disable this option for every single column. You can also define default sort order.

1. Click »Table settings«.
2. Scroll to gridView section and (un)check the »Sort« checkbox at column you want disable/enable sort feature for.
3. Click »Update«.

If you want to set default sort order (that will be displayed for every new visit of the grid by existing or new users), check the »Default sort order« checkbox. Checked column will be sorted ascending by default.

1. In gridView section click on edit icon near the column you want to set as default sort column.
2. Click »Next« (depending on your settings you'll maybe have to click »Next« more than once).
3. Click »Update«.

Advanced default sort order can be also set by defining sort order in »GridView ORDER BY part of the SQL statement« textbox under »Advanced section« (please read the »Advance options« chapter of this document for details).

Visible

By disabling this option column will not be visible in grid (gridView). However, even with this option unchecked column may appear in filters, custom HTML or when exporting data. These three options depend on setting of these three properties and are not affected by »Visible« property.

Join

Read only property. Indicates whenever each column is related to other data table in your database (JOIN SQL statement is build in the background). Please read about »Join another table data on this column« property for more information.

URL

Read only property. Indicates whenever each column is URL link (to another web page). Please read about »Column contents are URL links« property for more information.

Binary

Read only property. Indicates whenever each column shows binary picture or binary data from another column. Please read about »Add binary (file) columns« option for more information.

More

Read only property. Indicates whenever each column is composed:

- 0 – No custom HTML is used

- 1 - Column is composed (Custom HTML) using configurable data
- 2 - Column is composed (Custom HTML) using WYSIWYG editor

For more information please read about the »Show data from more columns (custom HTML)« option where all these options can be set.

Delete, Edit icon, Move icon

Please read the »Working with gridView columns« section of this document for more information. The »Delete« checkbox, »Edit« and »Move« icons help when administrating (configuring) the module.

GridView settings (gridView column properties)

These settings can be configured by clicking on edit icon in each gridView column row. As mentioned before, some of these settings can also be set on »Table settings« page (and are explained above).

Column name

Select the column to be used for current column. Column names are pulled from database and can be used more than once. This means that you can add two or more gridView columns for each database column. This is particularly useful when you want to display custom links within grid or when displaying calculated columns.

If there's more than one column in gridView using same database column name, the module will rename gridView columns in form `tablename_columnname_x` where x is replaced with number (0, 1, 2, 3 etc.). In such cases (when two or more gridView columns are sharing database column as source), you should enter captions for those fields.

Column caption

Enter caption for current column. This property is the same as »Caption« property listed on main »Table settings« page (in gridView columns list). If you'll leave this field empty, column name will be used as column caption. HTML formatting and tokens are allowed. Here are some examples:

- `<font color=»red«>Columnname</font>` will display column header in red
- `Columnname in<br>tworows` will display column header in two rows
- `[User:UserID]` will display ID of current user in column header

Column width

Use this property to define fixed width (in pixels, for example enter 300) of the column. If no value is entered, the column will be resized dynamic.

Content

Use this property to replace current column content. This is useful when link or image need to be displayed instead of column values. Examples:

- Enter »http://localhost/480/DesktopModules/IndooGrid/images/gridViewModule/help.gif« to display help icon instead of column content
- Enter »./Default.aspx« to display link to first page instead of column content

Format data

»Format data« field represents the format to display column content. For example, if column content is date, you can use »D;en-us« to show long date (1. January 2007) using English culture. Using this field you can format date, time, number and text data.

Use .NET format style:

- http://msdn2.microsoft.com/en-us/library/system.globalization.datetimeformatinfo.aspx [4] for date and time formats (for example 'dd.MMMM.yyyy')
- http://msdn2.microsoft.com/en-us/library/system.globalization.numberformatinfo.aspx [5] for numeric formats (for example 'E' or 'F;fr-fr').
- http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes [6] for ISO 639-1 (two-letter) codes list

For text data input max characters (text can be shortened only). If you want to use column content us link semicolon and optionally word as link – for example: '100' shows only first 100 characters of text, '100;' shows first 100 characters which are also link, '100;more' shows 100 characters from text and the word »more« is link. If you want to use link you must also check »Column contents are URL links« checkbox - see instructions below for details.

Date example for all cultures (long date and short date format:

- `D`
- `d`

Date example for specified culture (form of format;TwoLetterISOLanguageName). Example: long date for English-American culture

- `D;en-us`

When culture is not specified data is shown according to culture defined by current DotNetNuke user's profile.

Import default value

When importing record using bulk import value of this field will be imported as default but only if it is not defined by the user. If import is disabled for this column (that is the user cannot import values for this column manually), this value will always be imported.

When user updates records, this value will use as indicator – only records where this value is the same as existing value of this column will be updated.

Filter additional column list

Enter coma separated list of columns for composite search. By default this textbox is empty: when user will enter filter value in filter field, only records which contain search string in current column will be displayed. When list of columns is entered in this field, records which contain search string in any of the fields listed will be displayed (when filtering records). This is useful when you want to allow users to search (filter records) on more columns simultaneously, for example (if your table is Products) for columns Title and Description.

Example:

4. Enter »FirstName, LastName, Email, and DisplayName« in this field for UserID column of Users table (look at from previous examples)

When end user enters search string, all records containing search string in any of those columns (and in column UserID) will be displayed.

Instead of column names captions can be used to define additional filter fields.

Allow filter

This property is the same as »Filter« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Allow inline edit

This property is the same as »Edit« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Allow export

This property is the same as »Export« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Allow import

This property is the same as »Import« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Allow sort

This property is the same as »Sort« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Is column visible

This property is the same as »Visible« property listed on main »Table settings« page (in gridView columns list). Please read the »GridView settings (gridView column properties on Table settings page)« section of this document for more information.

Default sort order

Enable this option to set current column as default when grid is displayed for the first time. Ascending sort will be applied. If you want to sort the data descending or by two or more columns, use the »ORDER BY option« under »Advanced options« section.

Column contents are URL links

Use this option to link columns to another web page (column content will be also links). If you want to leave column content in its original state and use additional text as link, you should also read the »Format data« option.

After clicking »Next« new page will appear (»Set view column wizard - URL«). Here you have two options – you can either enter HTML link only (for example http://www.yourdomain.com/report.aspx in »URL« textbox) or you can enter HTML link and parameters separated. Parameters are useful when you need to parse data from the grid to another page (for example to partner page with »username« as parameter) or to another grid. If you leave this field blank, current URL will be used (to the current grid).

Parameters (»Parameter name« textbox) can be either custom (for using on another page as is your partner's page) or build in. Built in parameters are described below.

In »Select column for parameter value« you can choose from list of existing columns. For example, if you're passing parameter username (let's say that parameter name is »userid«), you should select »userid« as parameter value. The link clicked by the user will now look like this »http://www.yourdomain.com/report.aspx?&userid=value« where value will be replaced with the user's ID (userid).

You can (optional) also set target for URL link (»target« textbox; see HTML reference for details) and display button instead of URL link (check the »Show link as textbox« checkbox).

»Add command parameters for gridEdit« allows you to define grid Edit mode of targeted grid. This can be useful if you want to lead user s to the detail or edit pages using custom links. Options (modes) available are:

- View – opens grid in Detail view mode
- Edit – opens grid in edit mode
- Delete – deletes record

In this example we will start with standard grid based on Users table (as shown in previous examples):

1. Click on »Table settings« link.
2. Click on edit icon of »Username« column in gridView columns list.
3. Check »Column contents are URL links« checkbox.
4. Click »Next«.
5. Enter »uid« in »Parameter name« textbox.
6. Select »UserID« from »Select column for parameter value« dropdown list.
7. Select »1 - view« from »Add command parameters for gridEdit« dropdown list.
8. Click »Next«.

We are now back to the main »Table settings« page. In previous steps we defined how the parameter will be passed, in following steps we will define how the parameter will be accepted.

9. Check »Enable detail view« checkbox.
10. Check »Hide gridView in edit/detail mode« checkbox.
11. Check »Left label position« checkbox.
12. Enter »[numericQueryString:uid]« in »GridEdit - edit record ID« textbox.
13. Click on »Show advanced settings« checkbox.
14. Check »Allow GridEdit to use URL parameters« checkbox.
15. Click »Update«.

If you'll click on any value of Username column, detail view page for this record will open. If you look at the http address in the address bar of your browser you'll notice our parameter uid. In detail view mode record with this parameter as id will be displayed.

To retrieve parameter we used request token. For more information on how to use tokens please read the »Working with tokens« section. You should also read about properties »Allow GridEdit to use URL parameters« and »GridEdit - edit record ID« when using columns as links.

As mentioned above you can use link to parse data from one grid to another. Here is another example of such usage (parsing data from one grid to another): we have two related tables (Users and UserRoles from original DotNetNuke database). We will display Users in the first grid, where username will be link. Clicking this link the user will go to another grid on another page where all user roles for this user will be displayed.

1. Add IndooGrid to the first page.
2. Click »Table settings«.
3. Click on wizard icon near »Table (or view)«.
4. Choose table »Users«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Click »Add new view column« (we are adding this column because the unique key column is not added to the gridView list by default).
9. Select »UserID« from the dropdown list.
10. Uncheck the »Is column visible« field.
11. Click »Next«.
12. Scroll down and click »Update«.

We have completed the first grid (we'll define some options later) and have to move to second grid now:

13. Add IndooGrid to the second page.
14. Click »Table settings«.
15. Click on wizard icon near »Table (or view)«.
16. Choose table »UserRoles«.
17. Click »Next«.
18. Click wizard icon near »Unique key column« field.
19. Click »Automatically add all columns« link.
20. Click »Update«.

Now the data from UserRoles are displayed on the screen. Choose »UserID« in filter field and enter the value 2 in the filter value field. Click Find it!

Look in address line of your browser. There should be address like this:

http://www.yourdomain.com/yourpagename.aspx?&gv378__gvac=2&tabid=54&gv378__gvff0=1&gv378__gvfl0=0

Of course the numbers will be different in your case. Look at the first number near gv. In this case this is 378 (gv378). That's the generated ID of the gridview. Tabid is DotNetNuke page ID.

Custom ID of the gridView can be also set within »Advanced settings« section under »Table settings« (for example when custom ID would be x1, instead of »gv378« we would use »x1«).

Parameter gvff0 is value for first filter field and gvfl0 is index of the first filtered column. That's why value of first parameter is 2 and value of second parameter is 0 - 0 is for zero-based index of the column. Zero based stands for starting index with zero – index of first column is 0, index of second column is 1 etc.

The zero in gvff0 stands for index of filter – we can have more than one filter on the page: gvff0 is first filter, gvff1 is second filter value etc.

Same pattern is for gvfl0 – we can have more than one filter on the page. First filter is determined with gvfl0=column_index and gvff0=column_value, second filter is determined with gvfl0=column_index and gvff0=column_value.

Remember this values and return to the first page with first grid (table Users).

21. Click »Table settings«.
22. Scroll down to the gridView column listing.
23. Click on edit icon in the first (»Username«) row of the gridView list.
24. Check the »Column contents are URL links« checkbox.
25. Click »Next«.
26. Enter address (link) of the second grid (that you have remembered) in the »URL« field. In this case the link is

http://www.yourdomain.com/yourpagename.aspx?&gv378__gvac=2&tabid=54&gv378__gvfl0=0

You can see that &gv378__gvff0=2 from the link we remembered is omitted. Instead of using index of the column (in this case column index is 0), you can replace index value (gv378__gvfl0=0) with caption value of the column(gv378__gvfl0=column_caption). Note that this will works only if caption for the column is entered manually (it will not work if the caption is inherited from data table column name).

27. Enter »gv378__gvff0« in the »Parameter name« field.
28. Select »UserID« form the »Select column for parameter value« dropdown list.
29. Click »Next«.
30. Scroll down and click »Update«.

Now click on »admin« username and all user roles for admin will be displayed.

Add binary data from another column

Use this option to show binary image or link for binary data from another column. If there's no binary column in current data table, this option will be disabled. Please read the »Add binary (file) columns« section of this document for details.

Join another table data on this column

This option allows you to connect related tables using current column as key for related table. When this option is enabled JOIN SQL statement will be created in the background to the selected table.

When using this option you'll be redirected to the first page of Join table wizard (step1). On this page you can select the table to join on current column (current column is key for that table).

On next page (Join table wizard – step 2) you can map the appropriate table columns together. In first dropdown (»Join table on column«) the key column of the joined table should be selected (that's the column related to current column in the grid); in second dropdown list (»Select column contents«) column to display (from related table) should be selected.

In this example we'll show how to display usernames and roles for specific user (user is always member of at least one real in DotNetNuke environment). We'll use three tables:

- UserRoles (we'll use fields UserID and RoleID fields in that table)
- Users (we'll use UserID and Username fields)
- Roles (we'll use RoleID and RoleName fields)

Follow these steps:

1. Add new IndooGrid module on the page and choose »Table Settings«.
2. Choose Database table (»UserRoles«).
3. Use wizard to choose unique key (»UserRoleID«).
4. Click on »Add new view column«. New page will appear.
5. Choose »UserID« from dropdown list. That's UserID from UsersRoles datatable.
6. Choose »Column caption« (»UserName«).
7. Check »Join another table data on this column«.
8. Click »Next«.

Now we have to choose the table we want to join.

9. Choose table »Users« and click »Next«.

We also have to connect appropriate fields:

10. Choose »UserID« in the first row – that's UserID from Users datatable.
11. This field will be used for connection (UserID FROM UsersRoles = UserID FROM Users).
12. Choose »Username« in the second row – that's the field that it will be displayed in grid.
13. Click »Next«.

We have now completed first column and have to add another.

14. Click on »Add new view column«.
15. Choose »RoleID« from dropdown list. That's RoleID from UsersRoles datatable.
16. Choose »Column caption« (»RoleName«).
17. Check »Join another table data on this column«.
18. Click »Next«.

Now we have to choose the table we want to join on.

19. Choose table »Roles« and click »Next«.

We also have to connect appropriate fields.

20. Choose »RoleID« in the first row – that's RoleID from Roles datatable.

This field will be used for connection (RoleID FROM UsersRoles = RoleID FROM Roles).

21. Choose »RoleName« in the second row – that's the field that it will be displayed in grid.
22. Click »Next«.
23. Click »Update«.

Now we can see that user with username »admin« is member of roles Registered users, Subscribers and Administrators.

HTML attributes (JavaScript etc.)

Beside the built-in column features you can also use custom HTML attributes. Supported are all attributes that are supported by user browser. Please read about »Show data from more columns (custom HTML)« property to use advanced HTML options.

For demonstration purposes we will use the example from the previous section (»Join another table data on this column«).

1. Click »Table settings«.
2. Click edit icon near column »Username« under gridView section.
3. Check last checkbox »Custom HTML attributes (JavaScript, etc.)« and click Next.

Input table is displayed. In first column enter name of the attribute and in the second value for that attribute. In this example we will add JavaScript attribute which will raise dialog box if user clicks on username.

4. Enter »onClick« in first attribute name field.
5. Enter »alert('hello');« in first attribute value field.
6. Enter »style« in second attribute name field.
7. Enter »cursor: hand« in second attribute value field.
8. Click »Next« to continue.
9. Click »Update« to save changes.

Although nothing changed on the first sight click on one of the username records (for example click on admin); JavaScript dialog with »hello« text will appear. You will also notice hand cursor when you move cursor over the username. For use of complex JavaScript functions please refer to »Advanced options« settings.

Show data from more columns (custom HTML)

Columns can be also customized in the way to join data from different columns together. The user will see only one column but in fact the column will be composite (from more columns). This option is also very useful when building links from columns because it offers more possibilities than »Column contents are URL links« option.

Following options are available:

- 0 – No custom HTML is used
- 1 - Column is composed (Custom HTML) using configurable data
- 2 - Column is composed (Custom HTML) using WYSIWYG editor

Configurable data (option 1) offers more options to play with but it is more technical and less intuitive. It is also very useful when you need data to include but you want to prevent users to see such data in its own column. In such case you should add new view column, mark it as not visible and include it in Custom HTML.

After enabling this option new page »View column wizard - Show data from more columns« will be displayed. Each row in the table represents data from single database column. Here is row walkthrough:

Select »Column« to define column value you want to include in your build. Columns that are not marked as visible are also in the list. If column is joined to another table (»Join another table data on this column« option is used), value from related (joined) table will be displayed.

Textboxes »Prefix« and »Suffix« accepts HTML code. The output code will be in following form:

```
PrefixColumn_valuSuffix
```

Example 1:

Select »your_column« from Column dropdown list and enter

```
<img src=»http://mydomain.com/
```

in Prefix and

```
=.gif« />
```

In Suffix texboxes. The output will be: <img src=http://mydomain.com/column_valu.gif« />

Example 2:

Enter <br> in Suffix and next column will be shown in next row.

RegEx and RegEx replace textboxes will perform RegEx expressions on the column output. This option is useful when you want to prevent scripts to be executed (when content is entered by users) or when you want to replace special characters.

Condition column will shows data (Column value, Prefix and Suffix) only if condition column value match condition (criteria). Condition field accept DNN tokens and [AND] and [OR] tokens. As condition value parameter you can use: <,>,>=,<=,<>,!=,= or just value. Sign = is for null or empty value and <> or != is not null/not empty value.

Examples:

>15 [AND] <40 [OR] 10.

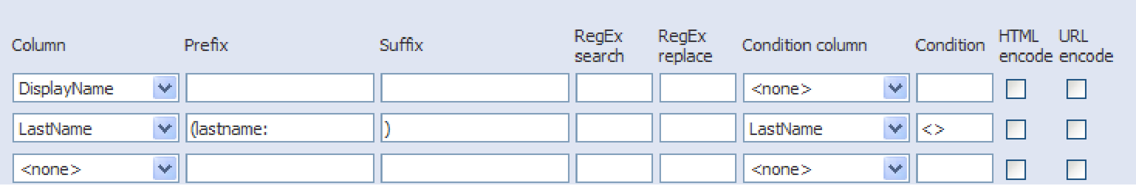
The output will be shown only if the column value is less than 40 and greater than 15 or when it is exactly 10.

Output can be also URL encoded or HTML encoded.

Step-by-step example: we will use standard Users table (as in previous examples).

1. Add IndooGrid to the first page.
2. Click »Table settings«.
3. Click on wizard icon near »Table (or view) ».
4. Choose table »Users«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Click »Add new view column« under gridView section.
9. Select »Username« from »Column name« list box.
10. Enter »Composite« in »Column caption« field.
11. Selesct »1 - configurable« from »Show data from more columns« dropdown list.
12. Click »Next«. A table will be displayed.
13. Select »DisplayName« in first column (»Column«) of first row.
14. Select »Lastname« in first column (»Column«) of second row.
15. Enter » (lastname:« in second column(»Prefix«) of second row.
16. Enter ») » in third column (»Suffix«) of second row.
17. Select »lastname« in fourth column (»Condition column«) of second row.
18. Enter »<>"» in fifth column (»Condition«) of second row.

Your screen should look like this:



Column	Prefix	Suffix	RegEx search	RegEx replace	Condition column	Condition	HTML encode	URL encode
DisplayName					<none>		□	□
LastName	(lastname:	)			LastName	<>	□	□
<none>					<none>		□	□

Picture 28: Using conditions to display data

19. Click »Next«.
20. Click »Update«.

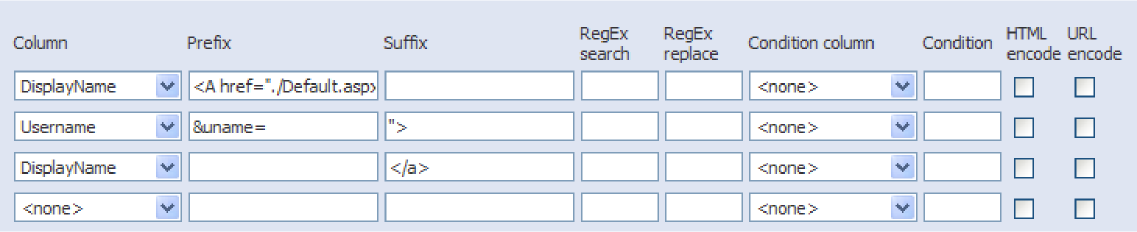
We can see that the »Composite« column have two parts, username and Lastname. Lastname and additional text is displayed only if the value of Lastname column is different than empty string (»»).

In second example we will show hove to use Custom HTML to build links with parameters (although this can be done by using »Column contents are URL links« too). This is often used when we want to redirect user to another page or module which requires parameter.

1. Add IndooGrid to the page.

2. Click »Table settings«.
3. Click on wizard icon near »Table (or view)«.
4. Choose table »Users«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Click edit icon of »Username« column in gridView column list.
9. Select »1 - configurable« from »Show data from more columns« dropdown list.
10. Click »Next«. A table will be displayed.
11. Select »DisplayName« in first column (»Column«) of first row.
12. Enter »<A href=»./Default.aspx?dname=» in first column (»Prefix«) of first row.
13. Select »Username« in first column (»Column«) of second row.
14. Enter »&uname=» in second column (»Prefix«) of second row.
15. Enter »">» in third column (»Suffix«) of second row.
16. Select »DisplayName« in first column (»Column«) of the third row.
17. Enter »</a>» in third column (»Suffix«) of the third row.

Your screen should look like this:



Column	Prefix	Suffix	RegEx search	RegEx replace	Condition column	Condition	HTML encode	URL encode
DisplayName	<A href="./Default.asp				<none>		□	□
Username	&uname=	">			<none>		□	□
DisplayName		</a>			<none>		□	□
<none>					<none>		□	□

Picture 29: Building links using Configurable option

18. Click »Next«.
19. Click »Update«.

You can see that output link (when you click on DisplayName link) contains two parameters: uname with Username value and dname with DisplayName value.

There’s another example of how to use Custom HTML (configurable) in Reports chapter of this document.

Editor (option 2) offers more intuitive interface but offers less options than configurable data. Standard editor included in the DotNetNuke is used to customize design. On the right side of the screen is list of all gridView fields added to the grid. Click twice on column you want to add to the page or drag (only in IE) the field to the editor area.

Note that drag and drop option is available in Internet Explorer only. If you use different browser click twice on the field you want to add to the editor.

Code in following form will appear:

```
[columnField:columnName] or [columnField:columnCaption]
```

Tokens are displayed only in design mode – in running mode the user will see values of the columns (tokens will be replaced with column values).

If column name is used and the column is joined (»Join another table on this data« option is used for the column), value from current table will be displayed: for example if the [columnField:UserID] is related to UserName column of the Users table, UserID will be displayed (1 for host).

If column caption is used and the column is joined value from current table will be replaced with the value from related table: for example if the [columnField:UserID] is related to UserName column of the Users table, UserName from Users table will be displayed will be displayed (»host« for host).

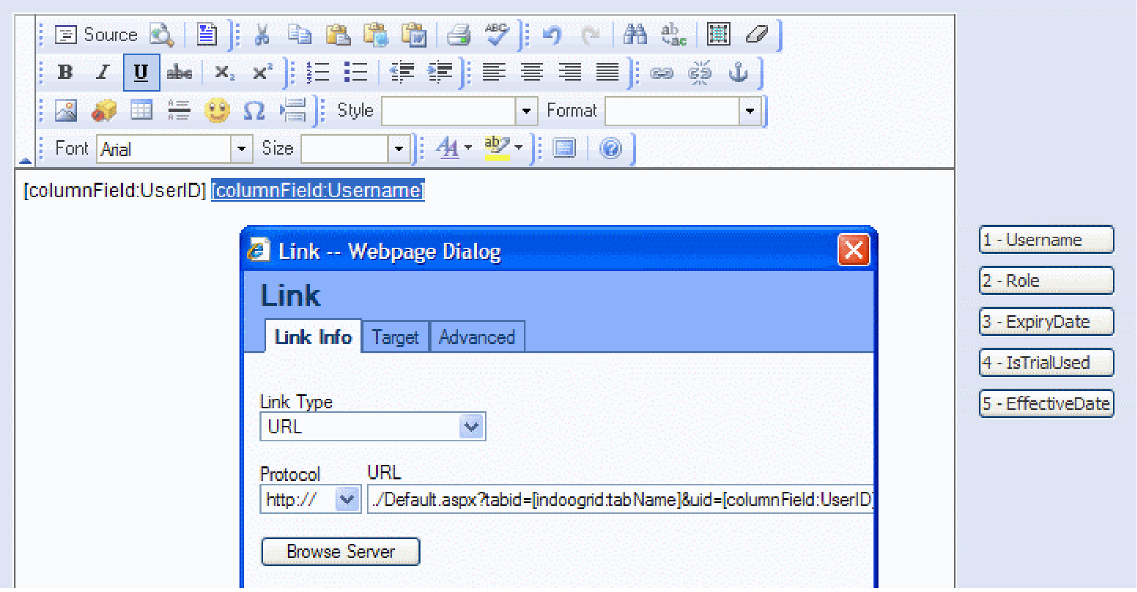
In editor you can also use standard tokens.

In this example we will show how to build links using editor option. We will use standard DNN UserRoles table for purposes of this example.

1. Add IndooGrid to the page.
2. Click »Table settings«.
3. Click on wizard icon near »Table (or view)«.
4. Choose table »UserRoles«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Click edit icon of »UserID« column in gridView column list.
9. Check the »Join another table data on this column« checkbox.
10. Select »2 - editor« in »Show data from more columns (custom HTML)« dropdown list.
11. Click »Next«.
12. Select »Users« from »Join table« dropdown list.
13. Select »UserID« from »Join table on column« dropdown list.
14. Select »Username« from »Select column contents« dropdown list.
15. Click »Next«.
16. Click on »Username« button in the field list on the right of the screen and drag it to the editor. Release mouse button.
17. Copy the result and change the first »[columnField:UserID]« to »[columnField:Username]«.
18. Select »[columnField:Username]« using mouse.

19. Click on Hyperlink icon.
20. Select »<http>» from Protocol dropdown list.
21. Enter »./Default.aspx?tabid=[indoogrid:tabName]&uid=[columnField:UserID]« in URL field.

This is the picture you should see now:



[columnField:UserID] [columnField:Username]

Link Type: URL

Protocol: http://

URL: ./Default.aspx?tabid=[indoogrid:tabName]&uid=[columnField:UserID

Picture 30: Links can be build using WYSIWYG editor

22. Click »Ok«.
23. Click »Next«.
24. Scroll down and click »Update«.

There's also an example on how to use the editor in »Reports« section of this document.

GridEdit settings (gridEdit properties)

Hide gridView in edit/detail mode

Use this option to hide gridView list in edit mode. If this option is disabled, user will see both objects (gridView and gridEdit, for example listing and form) in edit or insert mode.

Show gridEdit above gridView

Use this option to switch gridView and gridEdit position. Default position (when this property is disabled) is gridView at the top and gridEdit at the bottom. By checking this option gridEdit (for example form) will be displayed at the top and gridView (for example listing) will be displayed at the bottom of the grid.

Edit buttons in last column

Use this option to display command columns as last column (from left to right) in the gridView. By default (when this option is disabled), command columns (edit, delete and view icons) will be in first column of the grid.

Left label position

Use this option to display gridEdit labels on the left instead on the top. By default (when this property is disabled), labels for the fields (textboxes, dropdowns etc.) are in one row and the fields itself are in next row. You can also define how to show labels by using »Custom gridEdit HTML« option.

Check changes on exit

In gridEdit inline mode user has to click on »Save« icon to apply changes made. Use this option to notify (JavaScript dialog box will appear) user that data are not saved if he made changes on the page (but didn't save them) and is trying to leave the page.

Default GridEdit mode

Use this option to specify gridEdit mode on first user's visit. Following options are available:

- None

- Edit
- Insert
- View

When Edit or View mode is selected, »gridEdit –edit record ID« is required to specify the record to edit (or view). Insert mode is disabled if inline edit option is enabled.

When using this option it is also recommended to use »Redirect to URL« (with token [indoogrid:currentURL]) option to refresh page with changes made.

In this example we will show how to use this option to show detail view of record in related (joined) table. We will use two standard DNN tables: Users and UserRoles. In first step we will define target grid and parameter that should be accepted:

1. Add first IndooGrid to the page (with title »User profile«).
2. Click »Table settings«.
3. Click on wizard icon near »Table (or view)«.
4. Choose table »Users«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Check the »Hide gridView« checkbox.
9. Select »view« from »Default GridEdit mode« dropdown list.
10. Enter »[numericQueryString:uid]« in »gridEdit – edit record ID« textbox.
11. Click »Update«.

In second step we will define base grid and parameter to send:

1. Add second IndooGrid to the page (with title »User Roles«).
2. Click »Table settings«.
3. Click on wizard icon near »Table (or view)«.
4. Choose table »UserRoles«.
5. Click »Next«.
6. Click wizard icon near »Unique key column« field.
7. Click »Automatically add all columns« link.
8. Click on edit icon of gridView column »UserID«.
9. Check »Column contents are URL links« checkbox.
10. Click »Next«.
11. Enter »uid« in »Parameter name« textbox.
12. Select »UserID« from »Select column for parameter value« dropdown list.
13. Click »Next«.
14. Click »Update«.

Now if you click on UserID column of the second grid, page will reload and details for user with that ID will be displayed.

There's another example on how to use Default gridEdit mode in section Forms.

GridEdit - edit record ID

Use this property to define which record will be edit (or viewed) when »Default gridEdit mode« is enabled. Field accepts tokens - for example if you want to request parameter from the URL use token [numericQueryString:parameter_name] or if you want to retrieve database information like [sqlQuery: Select UserID from Users where UserID = [getObject:UserID]].

Altough DNN tokens are supported in this field, they are not supported inside the indoogrid tokens: for example we cannot use [sqlQuery: select max(myID) from myTable where UserID = [User:UserID]] but should use the example above. For more information about tokens please read the Working with tokens section.

Textbox width

Use this property to define default width for input fields (textboxes) of gridEdit. Width should be entered in pixels (for example »300«).

Redirect to URL

Use this option to redirect user after valid data input (insert or edit). This is almost nessecary when you have both gridView and gridEdit on the same page. When user enters values and clicks »Submit« (or »Insert)« in gridEdit, changes will not be seen in gridView before refreshing the page. If you enter [indoogrid:currentURL] token in this field, the page will refresh automatically after changes are saved (inserted).

This field can also be used to redirect user to another web page (for example »Thank you for your cooperation…«) or to display text after valid input.

Please read the Forms chapter of this document to see the example of using of this option.

Save button text

Use this option to replace default text of Save button with your custom text. Default text is retrieved from DotNetNuke language pack.

Insert button text

Use this option to replace default text of Insert button with your custom text. Default text is retrieved from DotNetNuke language pack.

Cancel button text

Use this option to replace default text of Cancel button with your custom text. Default text is retrieved from DotNetNuke language pack.

Custom gridEdit HTML

This option allows you to customize gridEdit appearance – that's how will gridEdit look in detail view mode, insert mode or in edit mode. Module design can also be customized using CSS styles (read the Styling chapter for more information).

By clicking on link or wizard icon gridEdit editor will be displayed on the left and list of all gridEdit fields on the right. Editor used is standard editor included in the DotNetNuke.

Click twice on column on the right to add column to the page. You can also drag the field to the editor area.

Note that drag and drop option is available in Internet Explorer only. If you use different browser click twice on the field you want to add to the editor.

Add new edit column

Use this wizard to manually add new gridEdit column. There can be only one gridEdit column for each column in the database added to the gridEdit.

GridEdit settings (gridEdit column list properties)

These settings are displayed in the gridEdit column list under gridEdit settings. They can be applied by entering the value (caption), selecting value from dropdown list or by (un) checking the checkbox for each column. Changes are saved by clicking on »Update« button at the bottom of the »Table settings« page.

Caption

Enter caption for each column. If no caption is entered (default), database column name will be used. You can use tokens and HTML formatting in this field, for example enter

```
<font color=red>MyCaption</font>
```

to display column header in red. For more information on tokens please take a look at »Working with tokens« section of this document.

Page

This option allows us to show columns on multiple pages. By selecting the value for specific column we defined on which page (column) field will appear. Pages that do not contain fields (for example if we selected some fields to be on page 1 and others on page 3) will be ignored.

Advanced settings (conditions) for multiple pages can be set on gridEdit column property pages (click on edit icon for column you want to configure). Please read about »Multiple page condition« property for details.

You should also look at Surveys & Questionnaires example when using this option.

Readonly

If this option is enabled, user will see the field value but he will not be able to change it (enter value). For example, we want to show the user current date but don't want to allow him to change it.

Visible

By disabling this option column will not be visible in grid (gridView). However, we can still use this field to insert data using »Field value« property (see description for example). Common use of this option disabled is inserting UserID of current user.

DDL (join)

This property indicates whenever current column is related to another table that's if dropdown list or radio button is enabled for current column (in insert and edit mode) or join is created on current column (for view mode).

GridEdit settings (gridEdit column properties)

These settings can be configured by clicking on edit icon in each gridEdit column row. As mentioned in previous chapter, some of these settings can also be set on »Table settings« page (and are explained above).

Column name

Select the column to be used for current column. Column names are pulled from database. Note that only one gridEdit column per one database column is allowed.

Column caption

Enter caption for current column. This property is the same as »Caption« property listed on main »Table settings« page (in gridEdit columns list). If you'll leave this field empty, column name will be used as column caption. HTML formatting and tokens are allowed. Here are some examples:

- `<font color=»red«>Columnname</font>` will display column header in red
- `Columnname in<br>tworows` will display column header in two rows
- `[User:UserID]` will display ID of current user in column header

Caption (label) design can be also set using editor; read the »Custom gridEdit HTML« description for details.

Separator

Use this option to enter text that will be displayed between caption and field (textbox, dropdown etc.). For example, enter »<font color=red>*</font>» to tell the user that this field is required.

CSS TR class name

Use this option to define row style. This is useful when you want to tell the user what is important when entering data or if you have many fields on the page and want to organize them (or show them in alternate rows).

You can either use built-in styles that came with the module or define your own style (CSS class). Custom CSS style should be entered in »Html head block« textarea of the »Advanced options«.

For example enter »myEditClass« in »Css TR class name« and put following code in »HTML head block« of »Advanced options«:

```
<style>

</style>
```

Note that you can also use editor (»Custom gridEdit HTML« option on Table settings page) to design gridEdit.

Help text

If this option is not empty, help icon will be displayed near the field caption. By clicking on that help icon text entered in this field will be displayed to the user. This options uses built-in DotNetNuke help.

Multiple page

This property is the same as »Page« property of column list settings of gridEdit on »Table settings« page. It allows us to show columns on multiple pages. By selecting the value for specific column we defined on which page (column) field will appear. Pages that do not contain fields (for example if we selected some fields to be on page 1 and others on page 3) will be ignored. First page is always shown (default for this option).

In this example we will use Comment table we made in one of previous examples (read Forms chapter of this document).

1. Click on »Table settings« link.
2. Click on »Table (or view)« wizard icon.
3. Select »Select existing table« radio button and click »Next«.
4. Select »Comments« from »Select table« dropdown list.
5. Click on »Unique key column« wizard icon. »ID« will appear in the field.
6. Click on »Automatically add all columns« wizard (or link).
7. Check the »Enable insert record« checkbox in gridView settings.
8. Check the »Hide gridView in edit/detail mode« checkbox in gridEdit settings.
9. Select »insert« from »Default gridEdit mode« dropdown list in gridEdit settings.
10. Select »2« in »Page« dropdown list of both »Rating« and »Comment« columns in gridEdit column list.
11. Select »3« in »Page« dropdown list of »Subscriber« and »Phone« column in gridEdit column list.
12. Select »4« in »Page« dropdown list of »Phone« column in gridEdit column list.
13. Click »Update«.

We can see that our form is now on four different pages and that user moves through this pages in wizard style.

Advanced settings (conditions) for multiple pages can be set on gridEdit column property pages (click on edit icon for column you want to configure). Please read about »Multiple page condition« property for details.

You should also look at Surveys & Questionnaires example when using this option.

Multiple page condition

Use this field to enter condition for current page to display. This property is disabled by default and can be enabled only if »Multiple page« field is not 1 (first page is always shown).

If page is not first, it is displayed if condition entered is true or if there's no condition (field is empty). If condition is false or if error occurred due wrong syntax or other causes, the page will not be displayed. Therefore it is necessary that each page contains at least one condition (or that this field is empty) that can be true otherwise page is ignored and will not be shown at all.

When more conditions are used all should be true. More conditions can be defined using reserved words »AND« and »OR« or by separating conditions using semicolon.

Condition syntax is standard Microsoft expression syntax. It is available at http://msdn2.microsoft.com/en-us/library/system.data.datacolumn.expression(VS.71).aspx.

Use double square brackets when using MS expressions.

Example 1:

Enter »Rating=3 OR Rating=4 OR Rating=5« where Rating is value that is entered by user in previous step – in other words you should enter this condition on second page where Rating is field on first page.

Example 2:

Enter following code (and than replace dbColumnName with your column name):

```
dbColumnName >1/1/2000 AND dbColumnName <1/1/2005
```

Condition will be true if dbColumnName is greater than 1/1/2000 and less then 1/1/2005.

Example 3:

If your column names contain spaces, use double brackets. Use single brackets to use tokens. Enter following code (and than replace dbColumnName with your column name):

```
[[db Column Name]] > [sqlQuery: select count(*) from users]
```

Condition will be true if value is greater than number of records in database table users.

Example 4:

When using nested tokens you should separated square brackets with spaces. Enter following code (and than replace dbColumnName with your column name):

```
 [[user mail]] = [sqlQuery: select email from users where
userid=[User:UserID] ]
```

Condition is true if column named »user mail« contains is the same as current user's email address.

Read only

This property is the same as »Readonly« property listed on main »Table settings« page (in gridEdit columns list). Please take a look at »Readonly« property description in gridEdit settings (column list properties) for details.

Visible

This property is the same as »Visible« property listed on main »Table settings« page (in gridEdit columns list).

HTML attributes (JavaScript etc.)

Beside the built-in column features you can also use custom HTML attributes. Supported are all attributes that are supported by user browser. For advanced options you can use built-in editor; please read the »Custom gridEdit HTML« property for details.

For demonstration purposes we'll use Comment table we made in previous examples (see Forms sample for details).

1. Click »Table settings«.
2. Click edit icon near column »Rating« under gridEdit section.
3. Check last checkbox »Custom HTML attributes (JavaScript, etc.)« and click Next.
4. Click »Next«.

Oldest human being

Input table is displayed. In first column enter name of the attribute and in the second value for that attribute. In this example we will add JavaScript attribute which will raise dialog box if user clicks on username.

5. Enter »onClick« in first attribute name field.
6. Enter »alert('hello');« in first attribute value field.
7. Enter »style« in second attribute name field.
8. Enter »cursor: hand« in second attribute value field.
9. Click »Next« to continue.
10. Click »Update« to save changes.

Although nothing changed on the first sight click on one of the username records (for example click on admin); JavaScript dialog with »hello« text will appear. You will also notice hand cursor when you move cursor over the username. For use of complex JavaScript functions please refer to »Advanced options« settings.

Field value

Use this field to enter default or predefined value for current column. Tokens are allowed.

If you want to use it as predefined value that cannot be modified as user it is recommended that you also enable the »Readonly« option although the user will not have ability to change the value. Example of such use is tracking where we want to know who made the changes to the record and when were these changes made. For example, in UserModified field we can enter »[User:UserID]« token to track who is the owner or who modified record. In DateModified we can enter token [Date:Current] to track when the changes were made.

If you want to use this option as default value that can be modified by the user, also check the »Use field value as default value only«. In that case the default value will appear in the field and will be stored into database when saved only if user will not change it.

There is one special token for this field [DNN:userinput]. This token returns the value that was entered by the user.

Width

This property defines the default width of current column textbox. Value should be entered in pixels, for example »200«.

Allow URL input

If this option is enabled, the value for this column can be accepted using parameters (parameter name should be the same as name of the column). This allows automation of input, for example (if we have link exchange) we can send to our partners link with their unique id, like somelink.aspx?partnerID=1 where partnered is id of that partner in our database. In such case we will always know who referrer is.

Input values are parameterized to prevent SQL injections.

Use field value as default value only

Use this option when you using »Field value« (reod more about this property) option and want to use value as default. Default value will be shown on insert form (always) or on edit form (if the field in current record is empty) .

Drop down list or radio button list (join for view mode):

Use this option when you want to make user's life easier (help him with values he can choose from) or when you want to prevent user from entering values that are not allowed. When displaying detail view, use this option to show data from another table (join table on that field).

There are two ways of enabling dropdown list or radio buttons. First one is to relate current column to another data table (current column will act as key for that table); the second one requires custom values (manually created list of values).

In this example we will use our Comment table that we have created in previous steps (Managing table chapter). We will display the list of entries and insert form.

1. Click »Table settings« link.
2. Enter »Comments« in »Table (or view)« textbox.
3. Click on wizard icon of »Unique key column« textbox.
4. Click on »Automatically add all columns« link.
5. Check »Enable edit records« checkbox.
6. Check »Enable insert record« checkbox.
7. Check »Enable delete record« checkbox.
8. Check »Hide gridView in edit/detail mode« checkbox.
9. Check »Left label position« checkbox.
10. Click edit icon of column »UserID« in gridEdit section.
11. Click »Next«
12. Check last checkbox » Drop down list or radio button list (join for view mode)« and click Next.
13. Click »Next«.
14. Select »Values from database table« of »Select list source« dropdown list.
15. Select »Drop down list« from »Show list as« dropdown list.
16. Enter »Users« in »Source table or view« textbox or select User table using wizard.
17. Click »Next«.
18. Select »UserID« from »Value column« dropdown list.
19. Select »Usernam« from »Text column« dropdown list.
20. Click »Next«.
21. Scroll down and click »Update«.

If you try to edit or insert record you'll notice that instead of textbox UserID field is displayed as dropdown list containing all users of your DotNetNuke portal.

In second example we will show how to use predefined list (manually entered). For demonstration purposes we will use test data table from »Validator« section (please take a lok at »Validator« property).

1. Click »Table Settings« link.
2. Click on edit icon of first column (»num«) under gridEdit section.
3. Click »Next«.
4. Check » Drop down list or radio button list« checkbox.
5. Click »Next«.

Select element (dropdown list or radio buttons) to use from second dropdown list (»Show list as« field). In this example we'll use dropdown list (list box element).

6. Select »Values from list« in first field.
7. Select »Drop down list« in second field.
8. Click »Next«.
9. Enter following values in text area field:

```
1;Start
2;In progress
3;End
```

Note that first you have to enter value, then semicolon and then text to display in dropdown list or radio buttons.

10. Click »Next«.
11. Click »Update« button.

Instead of »Numeric value« textbox now user will see dropdown list with predefined values to select.

All above procedures are also valid for editing data as the wizards and settings are identical. Only difference is the way to the insert/edit form – when inserting records, insert icon is displayed at the bottom of the grid and when edit the record, end user should click on Edit icon in the row with appropriate record.

Enable autovalidators

Use this field to enable auto validation of data entered by the user. Validation is executed depending on data type of the column (defined in the database). For example, if data type of current column in MS SQL Server 2005 is datetime, input will be validated against date and/or time.

Validation is executed twice on the server side (ASP.NET validation and database validation for insert/update) and on client side if this option is enabled.

Default value is on. Error messages are available within resource file (for ASP.NET validation) or they're provided by the database (database validation).

Custom validator

Use this option to define custom validators to be performed on current field. After clicking Next new page will open to add validators. Please read about Validator property for details.

Enable client script validation

Use this option to enable client side validation. Validation will be executed using JavaScript without submitting the page to the server. If you have more grids on the same page it is recommended not to use client side validation.

Password

Enabling this option will mark field (textbox only) as password field. When entering data in such field user does not see the value but asterisks. Use this option for password (or other sensitive data) fields.

Multiline

Use this option to define the height of the field. If this field is empty the field will have standard height. When this field contains number, this number will define the height of the editor (when »Textbox is editor« option is enabled) in pixels. If field is not editor, the field will be multiline field with as many rows as defined.

Textbox is editor

Use this field to enable users to enter data using standard DotNetNuke editor. When enabling this option make sure that anonymous users cannot enter scripts (for security reasons).

Validator

This option is available when »Custom validator« checkbox is checked. You can use up to two predefined (built-in) custom validators to use with the field. You can choose from following validators: Currency, Date, Integer, Mail, Number, Required. Validation will be executed on server side and (if »Client validation« option is enabled) on client side using JavaScript.

In this example we will create new table called test with five different fields: ID (unique key column), num (integer), bool (Boolean), dat (datetime), txt (long text, MAX). If you don't know how to create table, examine the »Table (or view)« section. After you'll be returned to the »Table settings« page, following steps will be required:

1. Click wizard icon near »Unique key column field«.
2. Click »Automatically add all columns« link.
3. Check »Allow insert« checkbox in gridView section.
4. Check »Enable insert by default« checkbox in gridEdit section. This option will always display »insert record« form and is not required.
5. Click on edit icon for first column (num) in gridEdit section. New page will appear.
6. Enter »Numeric value« in »Caption« textbox.
7. Click »Next«.
8. Click »Next« again.

Now we can see to options. First option, »Enable autovalidators«, is checked by default while second option, »Custom validators«, is unchecked by default.

Auto validators are built-in validators depending on data type, data size and null value check for the column. Values are validated on server twice: first with ASP validation (messages are available within resource file); second validation is executed on database insert/update (optional error messages are produced by database provider). When this field is on size and optional null values are also validated on client and both with dbtype validation are checked twice on server side otherwise only within insert/update validation is executed (with optional db provider's error messages which are not localized).

Custom validators are executed on both server-side and client-side (browser) and should be pre-defined.

1. Check »Custom validators« checkbox and click »Next«. Page to define custom validators is displayed. We can add two custom validators per field and also use Regular Expressions to validate values.
2. Select »Integer« from first »Validator« list box.
3. Select »Required« from second »Validator« list box.
4. Check »Enable client validation« checkbox.
5. Click »Next«.
6. Scroll down and click »Update«.

Insert form is now displayed as it will be to end users. Try some options, for example enter »x« in first (Numeric value) field. If you move focus to one of the other textboxes, red error message will appear saying »Value is not integer«. That's client validation. Now delete »x« from first field and click on Insert. Red error message will appear near the textbox »Numeric value«. This is client autovalidator. Red error message with same text will also appear at the bottom of the grid – that's server side autovalidation. Our custom validator will appear as JavaScript dialog.

Server side regular expression

This option is available when »Custom validator« checkbox is checked. Use this option when built-in validators are not enough. Regular expression will be checked on the server side. Use .NET syntax available on http://msdn2.microsoft.com/en-us/library/ms998267.aspx.

Message when RegEx check failed

This option is available when »Custom validator« checkbox is checked. Use this field to define message displayed to the user when regular expression check fails.

Add binary (file) columns

Using this option you can add binary columns to your grid. Binary columns are columns that can store files (pictures, videos or other types of files).

Binary columns (marked as fileN_content where N is replaced with number of the column) can exist in database but cannot be added to the gridView. They can only be bound to existing gridView column using »Add binary data from another column« option. However, binary column can be added to gridEdit.

By clicking on the link (or wizard icon near-by) new page will appear.

File (binary) location

Media (photos and videos) and other files (for example documents) can be stored either in database or on the file system. Each approach has its own (dis) advantages.

When contents is stored to file system then this columns hold information about stored file location and mime type information (there is no need for additional columns for this data). Storing files on file system is good because of performance reasons. Most DotNetNuke users use MS SQL Server Express as database – therefore storing files into database would quickly fill SQL Server beyond its limitations. For example, if average picture is 1MB, you could store about 4000 pictures to the database (MS SQL Server Express limit is 4 GB). So if you intend to have a lot of files on your system, you should choose this option.

When file content is stored to database then mime type should be specified or taken from another column. Filename for download can be specified or taken from another db column also. Storing files to database offers availability of files stored to all clients that have access to the database. Also, it is usually better to have all data within database for a lot of reasons (data consistence, maintenance etc.). This option is default option used.

Select the type of uploaded files

Use this option to select type of files that will be uploaded to the server. This helps this wizard to offer additional options in further steps. Options available:

- Files – use this option if all type of files will be uploaded
- Pictures – use this option if only pictures or videos will be uploaded. Wizard will offer additional options to work with media files in further steps.
- Mixed – use this option if all types of files will be uploaded. Wizard will offer additional options to work with media files .

Show binary as (see help icon)

Use this option to select how column contents will be presented. Following options are available:

- Link – use this option to display only link. Image title will be used for link text and by clicking on the link full image will be displayed.
- Image thumbnail – use this option to display image thumbnail. By clicking on the thumbnail full image will be displayed.
- Image thumbnail and link – use this option to display image thumbnail and link at the bottom of the thumbnail. By clicking on the thumbnail or on the link bellow full image will be displayed.
- Image thumbnail and text - use this option to display image thumbnail and text at the bottom of the thumbnail. By clicking on the thumbnail full image will be displayed.
- Image full size – use this option to show full image in the grid.
- Image full size and link – use this option to show full image and link at the bottom of the image. By clicking on the link bellow full image will be displayed.
- Image full size and text - use this option to display full image and text at the bottom of the thumbnail.
- Image thumbnail only – use this option to show thumbnail only. No further action will be allowed (user cannot click on the thumbnail to open image)

Note that thumbnail option is recommended for small files only! Thumbnails are created automatically and on-the-fly - this operation can consume a lot of server's resources when processing very large files. Avoid this option when expecting very large files in this column.

Max thumbnail width

Use this option to define width of the thumbnail displayed. Width should be entered in pixels. If you don't enter thumbnail height, all thumbnails will be as wide as this value suggest. If you also enter thumbnail height, maximum value will be applied when creating thumbnails. For example if you enter 100 for width and 100 for height and the picture's size is 1000*800, thumbnail size will be 100*80 pixels.

Max thumbnail height

Use this option to define height of the thumbnail displayed. Width should be entered in pixels. Please read about thumbnail width for details.

Select mode of browser window

Use this option to select click action. When user clicks on image thumbnail or link, picture can open in:

- New window – image will open in new browser window (popup).
- Parent window – image will open in parent window (parent window is window where you came from using »Open in new window« option or, if current window is popup window, parent window of current popup window).
- Same window – image will open in current window. User can go back to the list (gallery) using browser Back button
- Top window – image will open in floating window (unframed window in current browser). This option is recommended.

Max. upload length (bytes)

Use this option to enter maximum allowed length in bytes. By using this option you can limit users to upload only limited files. For images this value can also be negative - in this case module try to convert image to JPG and reduce it to specified size (minus=reduce).

Enter integer value (for example 5000) for bytes length and comma separated values (for example 500, 500) for »height, width«. Use minus size to reduce size (for example -5000 or -500, 500). If negative values are used, module will discard images that cannot be converted (when file is not GIF, JPG, BMP, EMF, EXIF, PNG, TIFF or WMF image).

Allowed extensions list

Comma separated list of extensions that are allowed for upload. When this field is empty, all extensions will be allowed to upload. File without extension is also allowed. For example, enter »jpg, gif« in this field to allow only GIF and JPG files to be uploaded.

Denied extensions list

Comma separated list of extensions that are denied for upload. When this field is empty, all extensions will be allowed to upload. File without extension is also allowed. For example enter »exe, com« in this field to prevent uploading of EXE or COM files.

Add db table column for filename

If this option is enabled, database column for original filename of uploaded file will be created. By default four columns are added to data table for each file (N is replaced with No. of binary column, starts with 1):

- fileN_title stores title (caption) of the file (used for link as default)
- fileN_content is binary column itself (contains picture)

- fileN_name stores name of original file uploaded (used when downloading file from server)
- fileN_mimetype stores MIME type of original file uploaded (used when downloading file from server or viewing file on server)

Add db table column for size

If this option is enabled, database column for size of uploaded file will be created.

Add db table column for date

If this option is enabled, database column for date of uploading will be created.

[Add sql statement(s) to settings]

Enabling (disabling) this option will not affect settings related to presenting or editing files. By enabling this option SQL statement(s) that will be created during this wizard process will be added to settings of current module. When (if) these settings will be later exported and imported in another module (to another DNN installation with different database), appropriate database structure will be created during import process. SQL statement(s) can be edited using »Edit SQL« wizard under »Advanced options«.

Category tree

Use this feature if records in the table belong to categories from another table(s). It displays categories as dropdown list, list box or tree view. By selecting the category only records in selected category will be displayed (in »Product catalog« grid).

When creating catalogs, albums or other listing, we usually organize items (category Items) in categories. Normally categories have their own table (we will call it Categories) with ID, Name and Description for each category. We have to options to relate items to appropriate categories:

- We can store categoryID in each record or have multiple category fields in each record if item can appear in more than one category: for example for three categories we add CategoryID1, CategoryID2, CategoryID3 to each record.
- We can make another table, call it ItemsCategory and store one record for one relation in that table. For example: Item with ID=1 belongs to Category with ID=33 so we will create one record in intersection table with keys 1 and 3

You should use first option unless your (single) item can exists in large number of categories at the same time. In this case you can use built-in category tree. After creating Items listing choose »Category tree columns list« from »Table settings«. New page will appear.

LIMITATIONS: module builds appropriate recursive SQL queries in the background for fast data presentation. Due to this you should keep your category tree structure as flat as possible. There is no problem with one (hierarchal) table as source or with more tables with single level but when more tables with multiple levels are in use then this can cause some performance hit. There can be also problems with more hierarchal tables together when database cannot handle more sub-queries or table names. In this case test your configuration first. Consider these limitations for LARGE and COMPLEX category tree only.

Category tree columns list

Use this field to enter name of the column from main table (for example from Items) that contains category ID. If you have more than one table for categories (for example CatType and CatSpecial), enter the field that contains categoryID from last category table (category ID that is related to CatSpecial table).

If your item can exists in more than one category at the same time (for example you have categoryID1, categoryID2 and categoryID3 columns in your Items table) enter comma separated list of columns (»categoryID1, categoryID2, categoryID3«).

Category show mode

Use this option to select the way the categories will be displayed: Following options are available:

- Custom – use this option if you want to customize appearance of the categories using advanced techniques.
- Dropdown list – this is default option. All categories will available in dropdown list: when user will select the category, page will refresh and only items in select category will be displayed.
- List box – categories will be displayed from left to right (hierarchically).
- List tree - categories will be displayed from top to bottom (hierarchically).

Show list for last level only

Use this option to show only current category level and first parent category (enabled by default). When this option is disabled, whole category tree (all parents to the top of the tree) will be displayed.

Category table(s)

Use this table to enter category table(s). If you have single category table, enter data only in first row. Each category table should contain at least ID column and one text column (name, description etc.).

If categories are hierarchical, each category record should also contain parentID (ID of parent category).

When first top level is clicked then all category filters are removed. When this page is set then you can use same settings for input data with category table source on dropdownlist page for gridEdit column.

Table name

Name of Category table. If more than category table exists, enter one category table name per row.

Title column name

Name of text column from category table. In fact, this should be category name column from category table– value of this field will be displayed to the user.

ID column name

All ID columns should be integers - if you already have categories in table that not contains id column as integer then you can create view with hash function and get appropriate ID column.

Parent ID column name

Name of the column that stores ID of parent category (parentID).

Levels

Levels to display in categories structure:

Example No. 1: You are creating Shop webpage. Your main table is called Products. Each product can have category associated (each record in Products table has also categoryID field which is related to Category table).

Normally we want to have categories organized in following in following way:

- First level, for example furniture, electronics, garden…
- Second level, for example for electronics category: TV, Audio, Video etc.
- Third level, for example for TV category: plasma, LCD, CRT etc.

If we have such categories, our Category (call it Cats) table structure should look like this:

- ID (auto number/identity)
- parentID (integer)
- catTitle (nvarchar).

Here are steps needed to enable category tree:

1. Click on »Table settings« link of your Products grid.
2. Click on »Category tree« wizard icon.
3. Enter »CategoryID« in »Category tree column(s)« textbox.
4. Enter »Cats« in first column (»Table name«) of first row of »Category table(s)« table.
5. Enter »catTitle« in second column (»Title column name«) of first row of »Category table(s)« table.
6. Enter »ID« in third column (»ID column name«) of first row of »Category table(s)« table.
7. Enter »parentID« in fourth column (»Parent ID column name«) of first row of »Category table(s)« table.
8. Enter »3« in last column (»Levels«) of first row of »Category table(s)« table.
9. Click »Finish«.
10. Click »Update«.

Example No. 2: Your main table contains parts for automobiles (each part belong to single model therefore each record also contains column modelID).

You also have following tables:

- Countries - columns: id, countryName
- Make - columns: id, countryId, makeName
- Model - columns: id, makeId, modelName

Here are the steps needed to enable category tree:

1. Click on »Table settings« link of your Products grid.
2. Click on »Category tree« wizard icon.
3. Enter »modelID« in »Category tree column(s)« textbox.
4. Enter »Countries« in first column (»Table name«) of first row of »Category table(s)« table.
5. Enter »countryName« in second column (»Title column name«) of first row of »Category table(s)« table.
6. Enter »ID« in third column (»ID column name«) of first row of »Category table(s)« table.
7. Enter »Make« in first column (»Table name«) of second row of »Category table(s)« table.
8. Enter »makeName« in second column (»Title column name«) of second row of »Category table(s)« table.
9. Enter »ID« in third column (»ID column name«) of second row of »Category table(s)« table.
10. Enter »Model« in first column (»Table name«) of third row of »Category table(s)« table.
11. Enter »modelName« in second column (»Title column name«) of third row of »Category table(s)« table.
12. Enter »ID« in third column (»ID column name«) of third row of »Category table(s)« table.
13. Enter »makeID« in fourth column (»parent ID column name«) of third row of »Category table(s)« table.
14. Click »Finish«.
15. Click »Update«.

Show advanced settings

Use this option to set additional options – click on help icon for details on selected option.

SQL development window

By enabling this option (»Show developer SQL window for current connection« checkbox) SQL window will appear. Especially in development phase it is useful to have direct database access. SQL Development window is interface to work with the database. It supports all the features that are supported by the database.

Use »Run as script« when SQL doesn't return any result or when SQL statements are separated using word GO and next-line character (CRLF - enter key). Window is not available when connection if connection is not configured.

Advanced options

By default this settings are hidden and are available by checking the »Advanced options« checkbox. In this section you can set some advanced features:

Allow GridEdit to use URL parameters

You can allow gridEdit to insert or update records by using link. Link should contain two parameters.

First parameter defines record to edit/update and is composed from id of gridEdit and keyword __gevi (for example gvNNN__gevi=1 for the first record with ID=1) with value of record ID. (__geiv = grid edit value identifier)

Second parameter is composed from id of gridEdit and keyword __gemo (for example gvNNN__gemo=0 for viewing the record; __gemo=grid edit mode) with possible values for each mode:

- 0 for view
- 1 for edit
- 2 for insert
- 3 for cancel.

There is example of how to use this option in »Column contents are URL links« section.

Refresh page after in-line edit save

When inserting (editing) data we often have two grids on the page: one for editing data and second to display data. When user saves data they are stored in database but they are not displayed yet. Use this option to refresh current view and display changes. Enabled option cause page refresh (extra loop).

Custom gridView page number

This option allows users to use Go to page feature. If this option is enabled page number will show in editable textbox, so user can input custom page number. This is not recommended for very large tables or for remote database (when database engine is separated from web server). It depends on database paging capability. If you are not sure leave this field unchecked or test performance when very large page number is used. Use this only if you know what you are doing. Enabled value is normally not problematic for lest then 100K records and if database in on same server as web server. When this value is not enabled then module use limitation 10K records but end-user usually cannot notice this at all because it is difficult to reach end of limit using paging links - otherwise user get message with suggestion to use sort or filter. Limitations and paging setting can be changed using direct access to underlying component.

Replace empty column captions

This option is enabled by default. It replaces empty columns (when we do not enter column captions) with column names pulled from database.

Show gridView SQL and connection

Use this field while developing the module to display current SQL statement executed and SQL connection. DO NOT use this option in production environment due security reasons.

Show gridView as dropdown list

Use this option if you want to allow the user select records from dropdown list instead of having them displayed in the grid. For example, you have 100 records to choose from and display details for selected record. Instead of having all 100 records on one or more pages, you can display gridView as dropdown list and allow users to edit (delete or view details) record using only one action button.

1. Click »Table settings«.
2. Check »Show advanced settings« checkbox to expand additional settings.
3. Check »Show gridView as dropdown list« checkbox to expand additional settings.
4. Click »Update«.

Of course you have to allow user to do something with selected record. You should enable one of the following options for gridView:

- enable edit record
- enable delete record
- enable detail view.

Instead of enabling above options you can define your own action link (or button) using »Column content are URL links« option. You can enable this option for any of the gridView columns or – even better – add new column for action only.

Using URL option you can also parse parameters from one grid to another (read the example »Parsing parameters from one grid to another«). You can use both grids (Users on the top and UserRoles on the bottom) on the same page and enable dropdown option for first grid (Users). User will have record selector on the top and when selecting the user, all roles for selected users will be displayed in the bottom grid.

GridView ID

If you leave this field blanks, ID will be generated automatically in form gvNNN where NNN will be replaced with auto generated numeric value.

This option is useful when you need custom ID to use in JavaScript (when referring the objects) or when you intent to use parameters.

GridEdit ID

If you leave this field blanks, ID will be generated automatically in form geNNN where NNN will be replaced with auto generated numeric value.

This option is useful when you need custom ID to use in JavaScript (when referring the objects) or when you intent to use parameters.

Add parameters to gridEdit redirect URL

Use this option to add parameters to gridEdit when redirecting. When gridEdit is redirected (usually after click on insert or edit button), custom parameters can be added using following form »parameterName=parameterValue« for single parameter. If more parameter are used they should be separated using commas: »parameterName=parameterValue, parameterName2=parameterValue2«.

For example if we have insert form and need newly added ID (ID of the record that was inserted in previous step) on redirected page, we can use special token parameterName=[indoogrid:idvalue].

If we need any columns value, we can use special token [indoogrid:column:<column name, caption or index>] .

Remove parameters from gridEdit redirect URL

Use this option to remove parameters that were added by you or by third party. For example, in first step we added parameter (myPar) using »Add parameters to gridEdit redirect URL« option. In second step we want to start again with insert form and want to clear the settings so we should enter myPar in this field.

Send gridEdit summary to mail

Use this option to define email address to receive notification on successful insert or update. If more addresses are used, they should be separated using semicolon. This is very useful option when we want to have immediately notification of users input, for example on »Contact us« form.

GridView import errors

Using this option you can set number of errors before import process stops. Special value is zero - when value is set to zero then module first scans all records and if no errors found then imports records to database. If any record occurs during the scan process rollback is executed (if database supports transactions).

GridView ORDER BY part of the SQL statement

Use this option to set default sort order for the grid. Although there's also »Default sort order« option for each column you should use this option when you want to sort the grid on more columns or when you want the data to be displayed in descending order. Use SQL reserved words ASC for ascending order and DESC for descending order.

For example, on »Users« table you can enter following code:

Username ASC, Lastname DESC

When user will come to the page, data will be sorted ascending by Username and (within first sort) by Lastname descending. If user clicks on any column header and sort data on specific column, for example on Displayname once, data will be sorted by DisplayName ascending, then within this sort by Username ascending and within this sort by Lastname descending.

GridView WHERE part of the SQL statement

By default users can filter data (if we didn't have disabled this option) but sometimes we want to filter data before they're available to users. For example, we can prevent users seeing other user's data or prevent users to see obsolete data. Condition can be of any type using DNN tokens, IndooGrid tokens and objects or just using SQL. This is part of SQL statement without WHERE word. This SQL clause is used as plain text and is not parameterized so use of object/values that return only numeric values is strongly recommended. SQL dialect is depend on underlying database.

First example will show use of DNN tokens. DNN tokens are part of DotNetNuke framework and are global available (read the »Working with tokens« chapter for more information). For demonstration purposes we will use same setting that we made in »Join another table data on this column« example .

1. Click »Table settings«.
2. Check »Show advanced settings« to expand additional settings.
3. Enter »UserID = '[User:UserID]'» in »GridView WHERE part of the SQL statement« field.
4. Click »Update«.

It is obviously that now only our record is displayed in the table. That's because we filtered the selection using condition:

»userID« is fieldname in our table(table Users)
[User: UserID] is DNN token which returns username of current user.

For complete list of tokens read the »Working with tokens« chapter.

Module header text

You can use HTML, text, scripts or custom commands in header (at the beginning) and footer (at the bottom) of the module. Custom commands can be SQL statements, tokens or combination of both.

Form of SQL statement:

```
<command sql=»SQLStatement« format=»format« textblock=»»
type=»»>text</command>
```

where attributes are:

- SQLStatement (required) – standard SQL statement
- format (optional) – for format options read the »Format data« section above
- textblock (optional) - can be html encoded text (or JavaScript) which will be rendered when link or button is clicked

- type (optional) – type of element – default type is link, other options are button, button1, smallbutton or button2
- text (optional) - button's or link's inner text

In this example we will retrieve last registered user date and display it in module header using French culture (enter this example in »Module header text« under »Advanced settings« section of »Table settings«):

```
<command sql=»SELECT MAX(CreateDate) FROM aspnet_Membership«
format=D;fr-fr></command>.
```

Tokens can be used in similar way as anywhere else in the grid (read the »Displaying data using WHERE clause and tokens« section of this document). Here are some examples (enter these examples in »Module header text« under »Advanced settings« section of »Table settings«).

- `<b>[getobject:Request.PhysicalApplicationPath]</b>` will return physical path of application in bold text.
- `[setObject:viewpage.gridView.id=myID]` will set ID property of gridView object to myID.
- `[getobject:PortalSettings.PortalID]` will return current portalID, same value as original DNN token `[Portal:PortalID]`.

Instead of entering commands in header you can use »Module footer text«.

For example, you can put following code into the field if users come from partners site using parameter pID as partner ID:

```
<font style=»color:blue«>

<script language=»javascript«>

if ([numericQueryString:pID] > -1) {

    document.write('Your Partner ID is: [numericQueryString:pID]')

}</script>

</font>
```

Beside commands you can use HTML as in first example when we used bold HTML tag.

Module footer text

This option is similar to the »Module header text« but the content displays at the bottom of the grid. Here are some examples:

We often need to display totals in view mode (sometimes in both edit and view modes). We can add totals to the grid using »Page header« or »Page footer« option and SQL command.

In this example we'll show display total of all records, total of records contained in the grid, total of values in all records and total of values contained in the grid. We'll use the grid showing table Users (look at one of previous examples, for example Displaying Data (gridView settings).

1. Click »Table settings«.
2. Click on »Show advanced settings« to expand advanced options (if they're collapsed).
3. Enter following text in »Page footer«:

```
<span style=»color: red«>Number of records (all): <command
SQL=»SELECT COUNT(*) FROM Users » ></command></span>
```

4. Scroll down and click »Update«.

At the bottom of the gridView we can see number of all records in Users table. This total will be always the same, no matter how user filters data or we filtered data using WHERE clause. If we want to use same set of data as it already is in gridView, we should enter following text in »Page Footer« (step 3):

```
<span style=»color: red«>Number of records (filtered by user or
selected using WHERE clause): <command sql=»SELECT COUNT(*) FROM
Users [indoogrid:gridViewWhereClause]«></command></span>
```

At the bottom is now displayed record count using same filter parameters that users has defined by using »Filter fields« or we have defined by using WHERE clause. In fact, token indoogrid:gridViewWhereClause] returns whole WHERE part of SQL command currently use in grid (including word WHERE)

In both examples above we counted records. If we want to sum values from the fields, we have to use SUM function. Enter following text in »Page footer« (step 3):

```
<span style=»color: red«>Total of values in field UserID (all
records): <command sql=»SELECT SUM(UserId) FROM Users »
></command></span>
```

Again we didn't take care about how user have selected records (or we have by using WHERE clause) and we have displayed sum of all records in table »Users«. If we want to display data that are actually in the grid, we have to enter this text in »Page footer« (step 3):

```
<span style=»color: red«>Total of values in field UserID (filtered
by user or selected using WHERE clause): <command sql=»SELECT
SUM(UserId) FROM Users [indoogrid:gridViewWhereClause]«
></command></span>
```

Try to filter data and compare the results. You can see that displayed total is sum of all data in the grid from all pages – even from the pages that aren't displayed within page limit.

You should also read the »Working with tokens« chapter and »Reports« sample when using these options (Module header or Module footer text).

Create MDB file

Use this option to create Microsoft Access database. Use of MDB databases is sometimes better for smaller projects or for other special needs (testing and developing) otherwise is recommended to use SQL server or other enterprise database – this is much better solution in many ways.

Clicking on the link (or wizard icon) new page will appear. Here you can set option for the new database.

Database name

Enter database name to create, upload or delete MDB file. Module adds mdb extension automatically.

Upload MDB file.

Browse the file with mdb extension to upload it to the server.

Delete

Delete the MDB file from the server.

Change module settings

Change module connection string and provider for access to specified database. When using this option all current gridEdit and gridView settings, table name, and key column will be deleted from the grid.

Instead of this option you can also manually configure ODBC provider and connection string to connect to the database. In this case use connection string like:

```
Driver={Microsoft Access Driver
(*.mdb)};DBQ=[DataDirectory]myDB.mdb;DriverId=25;FIL=MS Access
```

And replace string »myDB« with your file name.

Html head block

Use this option to change design and appearance of the grid or to define scripts that will be used within the grid.

Use style sheets to define appearance of the grid. For example:

```
<style>
 .ge_label{
 font-weight:bold;
 text-align:right;
}
</style>
```

Now all the elements with class ge_label (those are labels in gridEdit) will be aligned right and bold.

You can also use direct styles for specified module:

```
<style>
 gridViewIDgridEditID .ge_label{
 font-weight:bold;
 text-align:right;
}
</style>
```

where gridViewID and gridEditID should be replaced with actual gridView Id and with actual Id of the gridEdit.

Here is another example for designing reports. Copy the code below to the »HTML head block«. There are comments for every style to explain what we done with specific style.

```
<style>
/*this is filter field (where user enters search string) style*/
input.gv_ff {
background-color:#FFE89E;
color:  black;
}
/*this is filter field (where user selects field to filter) style*/
select.gv_fl, select.gv_vl{
background-color:#FFE89E;
color:  black;
}
/*this is filter button (the one with Find it!) style*/
.gridView TD.gv_cd2 {
color:black;
padding: 5px 1px 0px 0px;
}
/*this is header row (column headers) style*/
TABLE TD.gv_hcc
{
font-weight: bold;
background: #FFD000;
background-image: none;
background-repeat: no-repeat;
}
/*this is basic row (single row in the gridView) style*/
TD.gv_cc {
padding: 7px 3px 3px 5px;
background: #FFF8E0;
}
TD.gv_cc DIV {
font-size: 12px;
font-family: Georgia, Serif;
}
/*this is alternate row (every second row in the grid) style*/
TD.gv_cc2 {
padding: 7px 3px 3px 5px;
background: #FFE89E;
}
TD.gv_cc2 DIV {
font-size: 12px;
font-family: Georgia, Serif;
}
/*this is header link (column headers text) style*/
TD.gv_hcc A, TD.gv_hcc A:hover, TD.gv_hcc A:active, TD.gv_hcc
A:link, TD.gv_hcc A:visited
{
font-weight:bold;
}
/*this is style of surrounding space of the module */
TD#dnn_ctr640_ContentPane {
background: #FFF8E0;
}
</style>
```

Another example on how to design forms. Copy the code below to the »HTML head block«.

```
<style>
/*standard textbox, textarea and dropdown list*/
TD.ge_label {
}
input.ge_textBox {
background-color:#FFF8E0;
}
textarea.ge_textBox {
background-color:#FFF8E0;
}
select.ge_dropDownList {
background-color:#FFF8E0;
}
/*filter textbox*/
input.gv_ff {
background-color:#FFF8E0;
color:  black;
}
/*filter button*/
.gridView TD.gv_cd2 {
color:black;
padding: 5px 1px 0px 0px;
}
/*header row*/
TABLE TD.gv_hcc
{
font-weight: bold;
background: #FFD000;
background-image: none;
background-repeat: no-repeat;
}
/*basic row*/
TD.gv_cc {
padding: 7px 3px 3px 5px;
background: #FFF8E0;
}
TD.gv_cc DIV {
font-size: 12px;
font-family: Georgia, Serif;
}
/*alternate row*/
TD.gv_cc2 {
padding: 7px 3px 3px 5px;
background: #FFE89E;
}
TD.gv_cc2 DIV {
font-size: 12px;
font-family: Georgia, Serif;
}
/*header link*/
TD.gv_hcc A, TD.gv_hcc A:hover, TD.gv_hcc A:active, TD.gv_hcc
A:link, TD.gv_hcc A:visited
{
font-weight:bold;
}
</style>
```

As mentioned you here you can also define scripts that will be used across the pages. In one of the examples (in »Using custom HTML attributes« section) we added custom HTML attributes »onClick« with value »alert('hello');« to the grid element. Sometimes we need complex functions which we will use on many locations of website. In such cases it is simpler to write script function and add it in »HTML head block« than entering it in value fields.

For example, instead of »alert('hello');« from previous sample we can enter »myFunction(this.value);« in value field of custom HTML attribute and put following code in »HTML head block«:

```
<script type=»text/javascript« >
Function myFunction(v) {
alert('value of this element is ' + v + '.');
}
</script>
```

Now we can call this function anywhere within the grid. We have also expanded the function functionality with included element value.

Instead of scripts we can also insert styles or other HTML head information.

We have two other options to execute scripts:

- on page load
- on page submit

Note that when entering script for those options, you should omit the <script></script> tags.

Startup script block

Use this option to define scripts that will be executed on pageLoad (that's after all values on the page are already available). Enter script without <script>...</script> tags.

OnSubmit script block

Use this option to define scripts that will be executed onSubmit (that's when page is submitted). Enter script without <script>...</script> tags.

SQL triggers

Using triggers you execute commands even if the database does not support triggers. Standard SQL syntax should be used to create triggers (with optional use of tokens).

There are four application triggers available

On insert

This trigger will be executed on successful insert of data. For example, let's say that we want to track all inserts and want to add new record in MyLog table when new record is inserted in current table. We should use following trigger:

```
Update MyLog set date = getDate(), recID = [indoogrid:idValue]
```

Every time new record will be inserted in current table additional record will be inserted in MyLog table containing current date and current record id (of inserted record in current record).

On update

This trigger will be executed on successful update of data.

On module load

This trigger will be executed on module load (before any possible action of the user).

On view record

This trigger will be executed on every view of the record. For example, we would use this trigger to count page visits. We add new column called viewCount to our MyTable and put following code into the field:

```
Update MyTable set viewCount = coalesce(clickCount,0)+1 where
id=[indoogrid:idValue]
```

Every time the record will be viewed, viewCount value will increase for 1.

How to…

Use IndooGrid on Medium Trust Environment (goDaddy and some other hosting providers)

Medium trust environment is environment that offers limited access to system resources. Some host provider offers only this type of hosting. Module should works fine within this environment but only some module functionalities are disabled like some database providers etc. Each medium trust environment can be set individually depends on hosting policy so there is no specific rule which of functionalities are disabled. Functionalities are disabled on table settings page and possible description error messages may appear within table setting process so end-user cannot notice real background of environment.

Connect to remote database (MS SQL Server, Oracle, MySql, MS Access)

You have two options to connect to remote database:

- Use Connection string option if you want to work with remote databse tables and views.
- Use SQLQuery token if you to get (set) values in remote database within the module

First option is global for all grid while second affects only expression it is used with.

Use GUID or string instead of numeric value for »Unique key column«

It is not possible at this time to use column that is not numeric for unique key column. However, you can create view and use it instead of table. For example, execute following SQL statement using SQL Developer window:

```
CREATE view MyView AS SELECT mynewid, columnname1, columnname2 FROM
MyTable
```

Where My table is your table without numeric key column and mynewid is numeric key column.

Display data from multiple databases

There's no direct way to display data from two different databases in the same grid. However, there is walk around to achieve this. If your database support linking tables, you simply link table from second database in your first database. Now the table acts like all other tables in the first database.

Second option is to use special token SQLQuery and define connection string to second database. Please read “Working with tokens section” for further information.

Show list of items (and details for selected item)

This feature allows us to display limited set of columns in view mode and full set of fields in detail mode. End user will have transparent list of items and all the data of specified item, including pictures and links. In this example we will show use of detail view features for Module definition page.

1. Click »Table settings« link.
2. Click on wizard icon near »Table (or view)«.
3. Choose table »Module definitions«.
4. Click »Next«.
5. Click wizard icon near »Unique key column« field.
6. Click »Automatically add all columns« link.
7. Check »Enable detail view« checkbox in gridView section to allow detail view.
8. Check »Hide gridView in edit/detail mode« checkbox in gridEdit section. Using this option in detail mode only details of single record (selected record) will be displayed but not the whole gridView list.
9. Check »Edit buttons in last columns« checkbox in gridEdit section to move detail view icon to the end of record in gridView. Such form (action icons, buttons and links in last column) is commonly used.
10. Check »Left label position« checkbox in gridEdit section to display labels on the left of the field value. Default position of the label is on the top of the field value.
11. Scroll down and click »Update«.

Display records from two related tables in two grids

It is sometimes useful to show data in two grids, for example in first grid records from one table and in second grid records from second table using selected record in first grid as parameter. In this example we will show how to display users in first grid and by clicking detail icon we will display user roles for selected user.

First we will add two IndooGrid modules to the page and configure first grid:

1. Click »Table settings« link.
2. Click on wizard icon near »Table (or view)«.
3. Choose table »Users«.
4. Click »Next«.
5. Click wizard icon near »Unique key column« field.
6. Click »Automatically add all columns« link.
7. Check »Enable detail view« checkbox in gridView section to allow detail view.
8. Check »Hide gridView in edit/detail mode« checkbox in gridEdit section. Using this option in detail mode only details of single record (selected record) will be displayed but not the whole gridView list.
9. Check »Edit buttons in last columns« checkbox in gridEdit section to move detail view icon to the end of record in gridView. Such form (action icons, buttons and links in last column) is commonly used.
10. Check »Left label position« checkbox in gridEdit section to display labels on the left of the field value. Default position of the label is on the top of the field value.
11. Check »Advanced option« checkbox.
12. Enter »ABC« in »GridEdit ID« field.
13. Scroll down and click »Update«.

In second grid we will use UserRoles table.

1. Click »Table settings« link.
2. Click on wizard icon near »Table (or view) ».
3. Choose table »UserRoles«.
4. Click »Next«.
5. Click wizard icon near »Unique key column« field.
6. Click »Automatically add all columns« link.
7. Check »Advanced option« checkbox.
8. Enter following text in »GridView WHERE part of the sql statement« field:

```
[getNumericObject:viewpage.gridByID(»ABC«).idvalue]=0 OR
[getNumericObject:viewpage.gridByID(»ABC«).idvalue]=UserRoles.UserID
```

We have defined that only records with userid from first grid (in detail mode) will be displayed when detail mode for first grid will be selected.

9. Scroll down and click »Update«.

After click on detail icon near record, detail view for selected user will be displayed in first grid and all user roles for selected user will be displayed in second grid.

Insert multiple records at once

You can insert multiple records at once using »Import« feature of the grid. For further information please refer to section »Allow import ».

Use two dependent dropdown lists in same grid

When user is entering data, it is often usefully to enable related dropdown list to choose from. For example, in left dropdown box are car manufacturers (Ford, Mercedes, Toyota etc.) and in right dropdown list are car models (Fiesta, Focus, Mondeo, C220, E280, S500 etc.). When user selects manufacturer on the left, right dropdown list is automatically filled with manufacturer's models (only Fiesta, Focus and Mondeo when we select Ford).

In this example we will make three new tables: »user_car«, »car_make« and car_model« (each record represent single model) with following fileds (if you don't know how to create table, examine the »Managing tables« section):

user_car (int id, text note, int makeId, int modelId)

car_make (int id, text name)

car_model (int id, int makeId, text name)

When we will select make from the left dropdown list, in right dropdown list only models of this manufacturer will be displayed. First fill the appropriate data in car_make and car_model tables and then add new instance of IndooGrid to the page.

1. Click »Table Settings« link.
2. Use wizard to select »user_car« table.
3. Use wizard to select »id« field of user_car table.
4. Click »Automatically add all columns« link.
5. Check »Enable insert records« checkbox.
6. Click on edit icon of »makeId column« in gridEdit section.

7. Enter »Make« in »Caption« field.
8. Click »Next«.
9. Check »Drop down list or radio button list« checkbox.
10. Click »Next«.
11. Select »Values from database table« and »Dropdown list«.
12. Click »Next«.
13. Enter »car_make« in »Source table or view« field.
14. Click »Next«.
15. Select »id« from »Value column« dropdown list
16. Select »name« from »Text column« dropdown list.
17. Click »Next«.
18. Click on edit icon of »modelId« column in gridEdit section.
19. Enter »Model« in »Caption« field.
20. Click »Next«.
21. Check »Drop down list or radio button list« checkbox.
22. Click »Next«.
23. Select »Values from database table« and »Dropdown list«.
24. Click »Next«.
25. Enter »car_model« in »Source table or view« field.
26. Enter »makeId, makeId« in »Limit source records- advanced options« field.

First makeId is the dbcolumn name of first dropdownlist (the one with caption Make). Second makeID is the dbcolumn name from source table of second dropdownlist, which is related to the source table of first dropdownlist.

27. Click »Next«.
28. Select »id« from »Value column« dropdown list
29. Select »name« from »Text column« dropdown list.
30. Click »Next«.
31. Click »Update«.

Now click on Insert icon at the bottom of the grid and select Make from first dropdown list. Values in second grid will change dynamically and only related data will be displayed.

Save the current state of the grid

Full URL support enables users to save current grid’s state. After end user have sorted and filtered data, they can add current page to browser favorites. Using link the user will always see the grid using same parameters as when it was saved. All data and sort information will be preserved.

Security

Built-in security options

Module has some built-in security features you should know. These options can be set in module »Settings« and are global for all the modules on all portals (pages).

Do not allow connection to current DNN database

This option will disable access to the local DotNetNuke database.

Allow save binary data to file system

Binary data are stored within DotNetNuke by default. Selecting this option you will allow users to save binary data (for example images) directly on the file system.

Do not allow access to all inner DNN objects

Selecting this option will prevent users to get or set any object within DNN .net component (including connection string). If you choose to enable this option, you'll allow use of experimental tokens for getting or setting objects. For more information about using tokens read the »Displaying data using where clause and tokens« above.

Security by design

Level of security always depends on level of usability (features that module provides). As IndooGrid provides full customization features it can not provide more security than the user (administrator) of the module set it. For example, when editing and inserting records all inputs are verified not to contain harmful code. However, when you use advanced options like parsing parameters from URL direct to the database, you must take some preventive actions:

Here are some tips for working with module to maximize security.

- Use only getNumeric object (if not original value of parameter is passed trough URL, the numeric function will return error instead passing not original value to the database) when parsing parameters from URL to the database
- Use different names for caption than original column name (somebody listening to the traffic could retrieve columns names).
- Prevent access to the module settings

- Make regular backups (always make backup before upgrading the module)

Preventing access to module settings

If you want to enable your users to use module as administrators, you can enable them access to the »Table settings« but not to the module »Settings« where are some security options and registration options.

To achieve this you have to grant users access to the module but not to the page. You can edit module permissions in module settings.

Working with tokens

Because of its smart interface IndooGrid does not provide programming option (except for SQL and JavaScript) by design but does provide tokens and commands.

Tokens are program variables that can be used in the fields and allow us to get or set values on-the-fly (in running mode).

Commands are special programming statements that allow us to execute custom SQL statements.

Basically there are three types of tokens available to use within the module:

- Indoo static tokens are standard module tokens
- Indoo parameterized tokens are module tokens that accept parameters
- DotNetNuke tokens (there are two types of DNN tokens but there's no difference between them except rendering)

All tokens use square brackets [] to differ from static values.

Here are some examples on using tokens. In first example we will show how to match data from the database to those requested from the query string from other source (let's say URL parameter is »uname« and our module displays Users table):

1. Click »Table settings«.
2. Check »Show advanced settings« to expand additional settings (if advanced settings are collapsed).
3. Enter »username = '[getNumericObject:page.Request.queryString(»uname«)]'» in »GridView WHERE part of the SQL statement« field.
4. Click »Update«.

It is strongly recommended that you use getNumericObject instead of getObject always when retrieving data from query string to prevent SQL injections (if you pass the parameter to the database of course) because this SQL clause is used as plain text.

In case described above we can also use special token [numericQueryString:parameterName], for example above it will be »username = '[numericQueryString::uname]'». In fact, this is recommended token (both tokens are IndooGrid tokens).

In second example we will show how to use tokens to log date of insertion and username of the author (of inserted record) where both values will be included in the record. For such purposes we can add automatic values (it is not necessary that end user see these values).

Our base table for this example will be table test from Validator example (because it contains various data types).

1. Click »Table settings«.
2. Click edit icon of »dat« column in gridEdit section.
3. Uncheck »Visible« checkbox. The end user will not see this column during inserting records.
4. Click »Next«.
5. Enter »[Date:Current]« in »Field value« textbox.
6. Click »Next«.
7. Scroll down and click »Update«.

Enter valid number in first field (»num« textbox), for example 1 and click »Insert« button. Now we can see date of insertion displayed in the gridView part (list) of the grid. Token we used ([Date:Current]) is DNN token.

IndooGrid tokens

These tokens are part of IndooGrid module and are available within the IndooGrid module. By default they could be used wherever tokens are supported (there are some exceptions, marked with SPECIAL).

Some tokens cannot be used on medium trust environment but on full trust only (marked as FTONLY).

You can prevent users that have »Edit« access to use IndooGrid tokens if you check »Do not allow access to all inner DNN objects« under the »Module Settings« page of the module (users with edit access have access to »Table settings« but do not have access to module »Module Settings«). For information on how to achieve this please read the »Preventing access to module settings« section of this document. Theses tokens (that will not be enabled as described) are marked as INNER.

If you're not sure what to do with selected token, enter it in »Module header« or »Module footer« and observe results.

[setObject:objectName=value] – (FTONLY, INNER) use this token to set values of any IndooGrid object. ObjectName is any .NET objects. If object name is not correct then token is ignored. Example (we'll use options to edit gridView column links but as mentioned, ObjectName can be any .NET object of the grid):

- [setObject:viewPage.gridView.columnList(1).navigateParameterListRemove=+] – removes all parameters from URL of second column (from left to right in the gridView, zero based index)
- [setObject:viewPage.gridView.columnList(0).navigateParameterListAdd=param1,value1] – adds parameters with name param1 and value value1 to URL of first column (from left to right in the gridView, zero based index)
- [setObject:viewPage.gridView.columnList(1).navigateParameterListAdd=dbColumnName1,param1; dbColumnName2, param2] – adds parameters with name param1 and value of dbColumnName1 and parameter with name param2 and value dbColumnName2 to URL of second column (from left to right in the gridView, zero based index)
- [setObject:viewPage.gridView.columnList(0).navigateParameterListRemove=param1, param2] – removes parameters param1 and param2 from URL of first column (from left to right in the gridView, zero based index)

[getObject:objectName] or [getObject:objectName:format] - (FTONLY, INNER) use this token to get values of any IndooGrid object. ObjectName is any .NET objects. If object name is not correct then token is ignored. WARNING: do not use this token to store values in database (or execute database scripts). USe getNumericObject instead to prevent SQL injections. Example:

To retrieve value of any column of the gridView use token

```
[getObject:viewPage.gridView.columnList(index).value]
```

where index is replaced with zero based index of the columns (note that all columns of gridView are in the grid, including columns marked as not visible).

To retrieve value of ID column of any grid in gridEdit mode on the page (before the current grid, that is grids that are higher on the page than current grid), use token

```
[getObject:viewpage.gridByID(»gridID«).idvalue]
```

where gridID should be replaced with appropriate value (this value can be set in grid we are getting values from under Advanced options).

[getNumericObject:objectName=value] - (FTONLY, INNER) use this token to get numeric values of any IndooGrid object (value is converted to numeric value . this is necessary when we store such values into database because it prevents malicious code – SQL injections). ObjectName is any .NET objects. If object name is not correct then token is ignored. Example: Use tokens

```
[getNumericObject:viewPage.gridView.columnList(index).value]
```

```
[getNumericObject:viewpage.gridByID(»gridID«).idvalue] and
```

instead of tokens used in previous examples.

To access all gridEdit and gridView objects you use setObject or getObject (getNumericObject) tokens and viewPage parent object. For example:

[getNumericObject:viewPage.gridView.isRecordInfoVisible]

Programmer's reference for gridView and GridEdit object can be viewed here (Namespace Indoo.Web.Controls): http://www.components.init.si/grid/webhelp/index.html.

[indoogrid:dnnUserinput] and [indoogrid:dnnUserinput] – (SPECIAL, only in gridEdit) use this token to retrieve user input (what user has entered). For example use following code:

```
[User:UserID]_[indoogrid:dnnUserinput]
```

in »Field value« of gridEdit. If user id je 25 and user input is »John« (user has entered John into textbox), output stored in database will be 25_John

[indoogrid:gridViewWhereClause] – use this token to retrieve current grid's filters. If no filter is applied to the grid, returned value is empty string. If any filter applied, returned value will include WHERE keyword. For example use following code

```
<command sql=»SELECT COUNT (ID) FROM Orders
[indoogrid:gridViewWhereClause]«></command>
```

in Module footer textarea to retrieve number of records in Orders table (ID is one of the columns in Order table). If user will filter selection using grid's built-in filters, only filtered results will be retrieved.

[indoogrid:idValue] – get ID value of current record. This token can be used when grid is in insert, edit or detail view mode and it will return Unique key (ID) of current record displayed (edited or inserted). For example use this token to build links which will include ID of current record as parameter (to use it in IndooGrid module or any other module). Enter this code in module footer:

```
<a
href=»http://www.mysite.com/anyApp?myid=[indoogrid:idValue]>Record
[indoogrid:idValue]</a>
```

where link should be replaced with yours value. For example, if a current record ID is 25, text of link will be »Record 25« and href will include parameter with name myid and value 25.

[indoogrid:currentURL] – use this token to retrieve current URL address (including http:// part if there's any). Example: enter following code to »Redirect to URL« field [indoogrid:currentURL]&step=1

and after valid data input user will be redirected to same page but with parameter step with value 1.

[indoogrid:moduleID] – use this token to retrieve ID of current DNN module (DNN ID of current IndooGrid). Example: we can hide current module using Javascript. Use following code:

```
var obj = getElementByIdEnd(»span«,«[indoogrid:moduleID]«);
parent = obj.parentNode;
var n = 2 //this value can depend on your skin, raise it by 1
for (i=0;i<n;i++) {
     parent = parent.parentNode;
}
parent.style.display = 'none';
```

in »Startup script block« textarea. Current grid will be hidden on page load.

[indoogrid:tabID] – use this token to retrieve ID of current DNN page (DNN ID of current page with current IndooGrid on). Example: put following code in module header:

```
<a href=»./Default.aspx?tabid=[indoogrid:tabID]«>Open this page without parameters</a>
```

By clicking on the link current page will open without any parameters in the URL.

[indoogrid:navigateURL]

[indoogrid:queryString] use this token to get query string from URL or form in plain text.

[indoogrid:numericQueryString] use this token to get query string from URL or form in plain text.

[indoogrid:currentURLEncrypted] – use this token to retrieve current URL.

[indoogrid:sessionID] – use this token to retrieve ID of current session.

[indoogrid:binaryLink] – use this token to retrieve binary link. Token is available only within custom or configurable HTML in gridView binary column. When this column shows link to binary content this link is converted to regular text when this token is used inside.

[indoogrid:linkImage] – this token does same as token above but it replace link to thumbnail (for pictures only) instead to whole picture.

[indooGrid:binaryLinkThumbnail] – this token does same as token above but it replace link to thumbnail (for pictures only) instead to whole picture.

[queryString:parameterName] – (INNER) use this token to retrieve parameters from query string. For example, if link contains parameter myparam (...Default.aspx?myparam=x), use this token

```
[queryString:myparam]
```

where returned value will be x. If you intend to use value returned to work with database you should use queryStringSafeSql because of security reasons.

[numericQueryString:parameterName] – (INNER) use this token to retrieve parameters from query string in numeric format. Default value is -1 (if parameter cannot be retrieved or cannot be converted to numeric value). For example, put this code in Module footer:

```
<a
href=»http://mysite.com/Default.aspx?tabid=[numericQueryString:tabid
]>
Current
</a>»
```

Link with text Current will be displayed and will open current site.

[queryStringSafeSql:parameterValue] (INNER) use this token to retrieve parameters from query string in safe format. For example, when using values in WHERE PART of SQL statement, use this token:

```
ID=[queryStringSafeSql:userid]>
```

To filter records in the table by userid parameter (retrieved from URL or form).

[indoogrid:moduleFolder] – (INNER) use this token to get current module folder. For example if you want to display detail view icon, you can use this token in following way

```
<IMG src=»[indoogrid:moduleFolder]images/gridview/view.gif«/>
```

[indoogrid:applicationFolder] – (INNER) use this token to get folder of current application (DNN). For example put following code in Module footer:

```
Application folder is [indoogrid:applicationFolder].
```

[indoogrid:dataFolder] – ((INNER) use this token to get physical folder for current module database folder. Usually is this same as string for web aplicationfolder + "\App_data" This is used only internally when filesystem path is required – when binary content is stored to file system, for example.

[indoogrid:currentDate] and [indoogrid:currentDate:format] – use this token to get dates in various formats. For example use this token ([indoogrid:currentdate]) in »Field value« texbox of your DateModified column to track last modification of the record

This token has two additional options: format option and compute option. Using format option you can select how date will be displayed.

Base syntax of format option is [indoogrid:currentdate:format], here are some examples:

[indoogrid:currentdate:G] – general format
[indoogrid:currentdate:d] – short date format
[indoogrid:currentdate:D] – long date format
[indoogrid:currentdate:en-us] – US culture format
[indoogrid:currentdate:D;en-us] – long date, US culture format
[indoogrid:currentdate:G;en-us] – general format, US culture
[indoogrid:currentdate:M/d/yyyy] – custom short date

Base syntax of compute option is [indoogrid:currentdate:=Y,M,D,H,M] where default value (current date without changes) would be [indoogrid:currentdate:=0,0,0,0,0]. Examples:

[indoogrid:currentdate=-1,0,1,0,0] – current date minus 1 year minus 1 day will be returned

Of course you can use both format and compute option in same token. Examples:

[indoogrid:currentdate:G;en-us=-0,0,1,0,0] – general format, today minus 1 day, US culture
[indoogrid:currentdate:d;en-us=-1,0,1,0,0] – short date format, today minus 1 year 1 day, US culture
[indoogrid:currentdate:D;en-us=-1,0,1,0,0] – long date format, today minus 1 year 1 day, US culture

NOTE: Some databases except only specified date format. When necessary use format/culture options. For information on how to use format please read the section »Format data« section.

[setObject:viewPage.gridView.isColumnsHeaderVisible=true_or_false] – use this token to hide column header (the one with headers of the columns). Put this token

[setObject:viewPage.gridView.isColumnsHeaderVisible=false]

to Module header when creating gallery for example (header will not be visible).

[setObject:viewPage.gridView.isHeaderVisible=true_or_false] – use this token to hide general header (the one with Records and Page texts). Put this token

[setObject:viewPage.gridView.isHeaderVisible =false]

to the Module header when creating gallery for example (columns will not have headers displayed).

[indoogrid:gridEditID] – use this token to retrieve ID of current grid when in insert, edit or detail mode. For example, use this token to find gridEdit element on the page using JavaScript:

```
<script language=»javascript«>
var objGE = getElementByIdEnd(»table«,«[indoogrid:gridEditID]«);
</script>
```

[indoogrid:gridViewID] - use this token to retrieve ID of current grid when in view mode. For example, use this token to find gridView element on the page using JavaScript:

```
<script language=»javascript«>
var objGV = getElementByIdEnd(»table«,«[indoogrid:gridViewID]«);
</script>
```

In both case you can than use retrieved object (objGE or objGV) to manipulate with its appearance.

[indoogrid:moduleID] – use this token to retrieve current module ID. For example, use this token to find module element on the page using JavaScript:

```
<script language=»javascript«>
var objModule = getElementByIdEnd(»DIV«,«[indoogrid:moduleID]«);
</script>
```

You can than change module properties on-the-fly.

[indoogrid:tabID] - use this token to retrieve current page ID. For example, when building advanced links, you can use this token in following way:

```
<a href=»http://mysite.com/Default.aspx?tabid=[indoogrid:tabID]>
Current
</a>»
```

When user will click on the link, current page in its initial state (without any parameters) will open.

[DNN:parameterValue] – use this token if original DNN token doesn't work. Replace parameterValue with original DNN token, for example

[DNN:User:UserID]

This token is especially important if you have backward compatibilities problems – tokens in DotNetNuke are sometimes changed with new version.

[cultureText:parameterValue] – use this token when different text for specific culture is required. Syntax for parameter value is cultureCode;text(|cultureCode;text(|…)). Code is standard two letter iso language and country code or just language code. Token

```
[indoogrid:en-us;text1|fr-fr;text2]
```

Returns text1 for any culture but fr-fr.

```
[indoogrid: text1]
```

Always returns text1 for any culture.

```
[indoogrid:fr;text1|en;text2|en-us;text3]
```

Returns text1 for all cultures but any English culture. In this case text2 is returned but when en-us is used then text3.

[idValueByGridEditId:parameterValue] – use this token to get idValue (value from fey column) from gridEdit element on page. On insert this value is created on the middle of the page life cycle so you can use same token to get this value after this – in this case use double pointed quotes »idValueByGridEditId:parameterValue«. Example:

```
[idValueByGridEditId:myGrid1]
```

Returns value from key column for currently open record using gridEdit with id myGrid1

[setModuleVisibilityByGridId:parameterValue] – (SPECIAL, only in Module header) use this token to set visibility of current grid. For example, when displaying Comments grid for catalog, we want to show comments only when detail view of selected item is displayed. We can use Javascript to detect the state of main grid and then hide grid with comments if main grid is in detail view. To show grid use 1 and to disable it use 0.

```
[setModuleVisibilityByGridId:yourGridID=0] - do not show grid

[setModuleVisibilityByGridId:yourGridID=1] - show grid
```

[sqlQuery:parameterValue] – use this token to execute custom SQL. This token can be used to set values in the database or to get values. For examples, if we have grid based on table1 and want to retrieve value from table2 using data from table1, we can use this token in following way:

```
[sqlQuery:select Company FROM Users WHERE userID=[User:UserID]]
```

You can notice that we used nested tokens (please read bellow for details).

If you have connection string stored somhere in specified database you can use this token to access different database:

```
[sqlQuery: Select statement ~x connection string]
```

where x is:

- 0 or omitted=current dnn database
- 1=native sql server 2005
- 2=native sql serer pre-2005
- 3=native oracle

- 4=odbc.

Connection string can be omitted also. Tokens for connection string will be replaced when option "allow connect to current DNN database" is not disabled (on regular module's settings page).

[compute:parameterValue] – – use this token to execute mathematical operations. Format of computing is standard Microsoft format for column expressions available http://msdn2.microsoft.com/en-us/library/system.data.datacolumn.expression(VS.71).aspx. Examples:

[compute:5*4] – multiplies 5 and 4
[compute:iif(a=b,1,0)] – returns 1 if a value is same as b otherwise returns 0.

Execution of tokens

Tokens with parameters are executed twice: first time at beginning and second time on prerender. On prerender square brackets are replaced with double quotes (for values that are retrieved on load).

Therefore you can nest tokens, for example you can first retrieve one value and than set the value using retrieved value from previous step. To define order of executing tokens add number to the end of token. For tokens that should be executed in second cycle, add 2 to the end of token. For example to use [getValue:viewPage.gridEdidID] token and than set value using token setValue, build token this way:

[setValue2:viewPage.gridViewid=[getValue:viewPage.gridEdidID]]

First (in first cycle) getValue token will be executed and only after set value (in second cycle) token will be executed.

DNN tokens

Syntax of DNN tokens:

[objectName:propertyName]

There is no known documentation for original DNN tokens. Token names in Appendix are taken directly from DotNetNuke running process from 4.0.1 version - newer versions probably have more properties.

Host
PerformanceSetting
UseCustomErrorMessages
HelpURL
HostFee
DemoPeriod
SiteLogBuffer
HostSpace
SMTPServer
SMTPAuthentication
HostURL
Copyright
FileExtensions
AutoAccountUnlockDuration
HostEmail
HostTitle
SiteLogStorage
ControlPanel
UsersOnlineTime
ProxyServer
HostPortalId
EnableModuleOnLineHelp
SkinUpload
EventLogBuffer
HostCurrency
SiteLogHistory
ProxyPort
PaymentProcessor
ModuleCaching
DisableUsersOnline
UseFriendlyUrls
DemoSignup
SchedulerMode

User
UserID
Username
FirstName
LastName
FullName
PortalID
IsSuperUser
AffiliateID
Membership
Profile

Membership
Password
Email
Username
LastLoginDate
LastLockoutDate
CreatedDate
Approved
LockedOut
ObjectHydrated

Portal
PortalId
PortalName
HomeDirectory
HomeDirectoryMapPath
LogoFile
FooterText
ExpiryDate
UserRegistration
BannerAdvertising
Currency
AdministratorId
Email
HostFee
HostSpace
AdministratorRoleId
AdministratorRoleName
RegisteredRoleId
RegisteredRoleName
Description
KeyWords
BackgroundFile
SiteLogHistory
AdminTabId
SuperTabId
SplashTabId
HomeTabId
LoginTabId
UserTabId
DefaultLanguage
TimeZoneOffset
Version
DesktopTabs
ActiveTab
HostSettings
PortalAlias

Profile
FirstName
LastName
FullName
Street
City
Region
PostalCode
Country
Unit
Telephone
Cell
Fax
IM
Website
TimeZone
PreferredLocale
ProfileProperties
ObjectHydrated

Custom
Undefined

Date
Current