PHPRunner 10.0 Manual

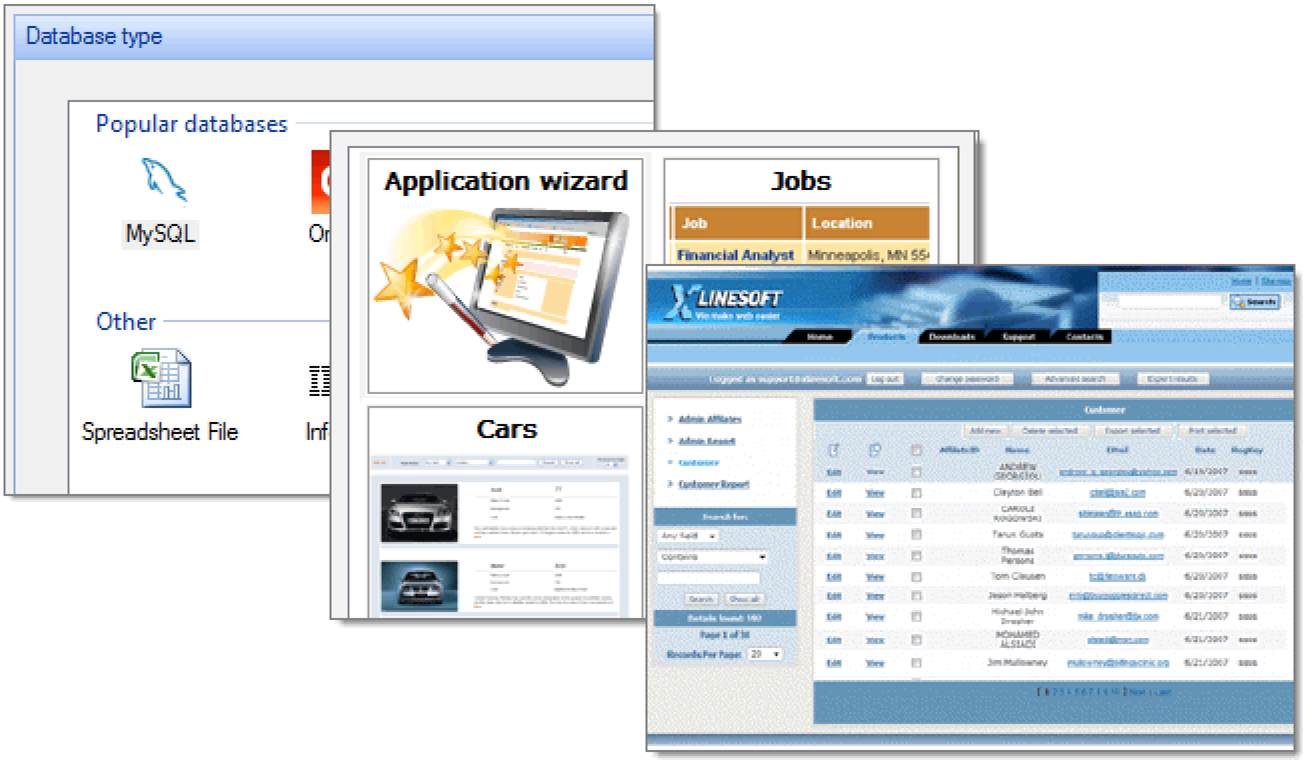


Table of Contents

Part I Introduction 1

1 Welcome ... 1
2 System requirements ... 14
3 Editions comparison ... 15
4 Licensing details ... 15
5 What is the registration ... 16
6 Quick start guide ... 17

Part II Using PHPRunner 24

1 Working with projects ... 24
2 Navigation bar ... 29
3 Templates ... 30
What are templates ... 30
Cars ... 34
Classified ads ... 34
Events ... 35
Jobs ... 35
Knowledge base ... 36
News ... 36
Paypal ... 37
Real estate ... 37
Sporting ... 38
Vacation houses ... 38
4 Connecting to the database ... 39
5 Datasource tables ... 46
6 Master-details relationship between tables ... 55
7 SQL query page ... 69
About SQL query designer ... 69
Query Designer ... 70
SQL ... 75
Results ... 78
Tabs and additional WHERE clause ... 79
8 Charts ... 81
Creating chart ... 81
Chart types ... 85
List of chart types ... 85
Accumulation chart ... 86
Area chart ... 87
Bubble chart ... 90
Column/Bar charts ... 93
Combined chart ... 96
Financial OHLC/Candlestick charts ... 99
Gauge chart ... 101
Line chart ... 104

Pie/Doughnut charts 107
Chart parameters 109
Chart appearance 111
Using SQL to shape chart data 114
9 Reports 118
Creating report and setting parameters 118
Report totals and layout 123
10 Dashboards 125
Creating dashboards 125
Master-details dashboard 140
Dashboard search 146
11 Choose pages 152
List page settings / Click actions 152
Add/Edit page settings 162
Update selected page 163
CAPTCHA on Add/Edit page 164
Key columns 167
Export/Import pages 169
Printer page/PDF view settings 173
Geocoding 174
12 Choose fields 175
13 Miscellaneous settings 184
14 Adding CAPTCHA 198
15 Security 202
Login page 202
Login form appearance 204
Registration and passwords 209
Advanced security settings 216
User group permissions 219
Dynamic Permissions 221
Audit and record locking 225
Active Directory 229
Encryption 233
Facebook connect 236
Two-factor authentication 239
16 Page Designer 240
About Page Designer 240
Working with Common pages 247
Working with Table pages 250
Working with cells 257
Working with page elements 260
Page layout and grid type 269
Tabs and sections 273
Adding PHP code snippet 277
Insert button 281
Insert Map 289
"View as" settings 297
"View as" settings 297
Audio 304
File 304
Image 305

QRCode ... 308
Video ... 309
Map ... 310
Custom ... 313
Conditional formatting ... 316
"Edit as" settings ... 319
"Edit as" settings ... 319
ColorPicker ... 327
Date ... 328
File/Image ... 330
Lookup w izard ... 334
SignaturePad ... 345
Text area ... 346
Text field ... 346
Time ... 347
Validation types ... 348
"Filter as" settings ... 353
"Filter as" settings ... 353
Boolean ... 357
Values list ... 358
Interval list ... 361
Interval slider ... 365
17 Editor ... 365
About Editor ... 365
Customizing CSS examples ... 370
Menu builder ... 383
18 Event Editor ... 392
19 Output directory settings ... 400
20 After you are done ... 402
21 FTP upload ... 404
22 Desktop Applications ... 406
23 FrontPage Publishing ... 409

Part III Advanced topics 411

1 Events ... 411
Predefined actions ... 411
Send simple email ... 411
Send email w ith new data ... 412
Send email w ith old data ... 413
Save new data in another table ... 414
Save old data in another table ... 416
Insert a record into another table ... 417
Check if specific record exists ... 418
Display a message on the Web page ... 419
Redirect to another page ... 420
Sample events ... 421
Appearance ... 421
Add custom field to form ... 421
Add dropdow n list box w ith values for search ... 422
Add link to user profile to the menu ... 422
Add new button to Add/Edit pages ... 423

Change cell background color ... 423
Change font size in text box ... 424
Change 'Logged on as' message ... 424
Change message after record w as added or saved ... 424
Change row background color ... 425
Change w idth of edit box w ith AJAX popup ... 426
Change w idth of text field on Quick Search panel ... 426
Disable record editing ... 426
Hide buttons in some row s of datagrid ... 426
Hide controls on Add/Edit pages, based on logged user name ... 427
Hide empty fields on View page ... 428
Hide repeating values on List page ... 428
Print search parameters on List page ... 429
Redirect to details page after master record w as added ... 430
Show data from master table on detail view /edit/add page ... 431
Show dropdow n list of US states if US w as selected in country list ... 431
Database ... 432
Before deleting a record check for related records ... 432
Dynamic SQL query ... 432
Limit number of records users can add ... 433
Select multiple values from checkboxes or a list field and have them appear as individual database entries ... 434
Show list of customer orders ... 435
Store the date and time w hen a record is modified ... 435
Update multiple records on the List page ... 436
Update multiple tables ... 436
Search Master and Details tables together ... 437
Email ... 438
Email selected records ... 438
Send an email to selected users ... 439
Send an email w ith updated fields only ... 444
Send mass email to all users ... 444
Send an email w ith attachment from the database ... 445
Upload ... 446
Rename uploaded files ... 446
Misc ... 447
Check if start date is earlier than end date ... 447
Redirect to user info edit page ... 447
Restrict access to PHPrunner application by IP address ... 447
Save user data in session variables ... 448
Speed up data entry using events ... 448
Global events ... 449
Login page ... 449
Before process ... 449
Before login ... 449
After successful login ... 450
After unsuccessful login ... 451
After Logout ... 452
Before display ... 452
JavaScript OnLoad ... 453
Menu page ... 454
Before process ... 454
Before display ... 455
JavaScript OnLoad ... 456

Register page ... 457
Before process ... 457
Before registration ... 458
After successful registration ... 459
After unsuccessful registration ... 460
Before display ... 460
JavaScript OnLoad ... 461
Change passw ord page ... 463
Before process ... 463
Before change passw ord ... 463
After passw ord changed ... 464
Before display ... 465
JavaScript OnLoad ... 466
Remind passw ord page ... 467
Before process ... 467
Before passw ord reminder sent ... 468
After passw ord reminder sent ... 468
Before display ... 469
JavaScript OnLoad ... 470
After application initialized ... 471
Menu item: Modify ... 473
Before audit log ... 475
Table events ... 476
Add page ... 476
Before process ... 476
Copy page: OnLoad ... 477
Before record added ... 477
Custom add ... 479
After record added ... 480
Process record values ... 481
Before display ... 482
JavaScript OnLoad ... 483
Edit page ... 485
Before process ... 485
Before record updated ... 485
Custom record update ... 487
Process record values ... 488
After record updated ... 489
Before display ... 491
JavaScript OnLoad ... 491
List page ... 493
Before process ... 493
Before record deleted ... 493
After record deleted ... 494
After group of records deleted ... 495
Before record processed ... 496
After record processed ... 496
Before SQL query ... 497
Before display ... 498
JavaScript OnLoad ... 499
Get Row Count ... 500
Custom Query ... 501
Custom record fetch ... 502
Report page ... 503

Before process 503
Before display 503
Before SQL query 504
JavaScript OnLoad 505
Chart page 506
Before process 506
Before display 507
Before SQL query 508
ChartModify 508
JavaScript OnLoad 516
Printer-friendly page 517
Before process 517
Before record processed 518
After record processed 519
Before display 520
Before SQL query 521
JavaScript OnLoad 521
View page 523
Before process 523
Process record values 523
Before display 524
JavaScript OnLoad 525
Search page 526
Before process 526
Before display 527
JavaScript OnLoad 528
Import page 529
Before import started 529
Before record inserted 530
After import finished 531
Export page 531
Before process 531
JavaScript OnLoad 532
Before record exported 533
After table initialized 534
Get Table Permissions 535
Is Record Editable 536
Page life cycle overview 537
Common event parameters 541
Field events 542
2 Programming topics 546
Buttons 546
Button object 546
Button object 546
Methods 547
getCurrentRecord 547
getNextSelectedRecord 547
row Data object 548
row Data object 548
Data Access Layer (DAL) 549
About Data Access Layer 549
Using DAL functions in projects w ith multiple database connections 551
Methods 553
Add 553

CustomQuery 554
Delete 555
DBLookup 556
FetchByID 556
Query 557
QueryAll 558
TableName 559
Update 560
UsersTableName 560
w hereAdd 561
Database API 561
About Database API 561
Methods 562
SetConnection 562
Exec 563
Query 563
LastId 564
LastError 564
Insert 564
Delete 565
Select 566
PrepareSQL 567
QueryResult object 568
fetchAssoc 568
fetchNumeric 568
value 569
SQL variables in SQL and Lookup w izards 569
Grid Row Javascript API 570
About Grid Row Javascript API 570
Methods 571
fieldCell 571
getFieldValue 571
getFieldText 572
setFieldText 572
setFieldValue 572
record 573
setMessage 574
getMessage 574
Javascript API 575
About Javascript API 575
Control object 576
About Control object 576
Methods 579
addClass 579
addStyle 579
addValidation 580
clear 581
clearEvent 582
getDispElem 582
getValue 583
hide 584
invalid 585
isReadonly 585
makeReadonly 586

makeReadWrite ... 586
on ... 587
removeClass ... 588
removeValidation ... 588
reset ... 590
setDisabled ... 590
setEnabled ... 591
setFocus ... 591
setValue ... 592
show ... 593
validate ... 593
validateAs ... 594
InlineRow object ... 596
About InlineRow object ... 596
RunnerPage object ... 596
About RunnerPage object ... 596
Methods ... 597
getSearchController ... 597
hideField ... 597
show Field ... 598
getTabs ... 599
SearchController object ... 600
About SearchController object ... 600
Methods ... 601
addField ... 601
clear ... 602
deleteField ... 603
display ... 603
getSearchFields ... 604
toggleCriteria ... 604
toggleOptions ... 605
SearchField object ... 606
About SearchField object ... 606
Methods ... 607
addOption ... 607
getControl ... 608
getName ... 609
getOption ... 609
getOptions ... 610
getSecondControl ... 610
remove ... 611
removeOption ... 612
setOption ... 612
Examples ... 613
How to ask for confirmation before saving record ... 613
How to calculate values on the fly ... 613
How to change font size in text box ... 614
How to change font in "edit" controls ... 614
How to change w idth of edit box w ith AJAX popup ... 616
How to change w idth of text field on Quick Search panel ... 616
How to control Inline Add/Edit functionality from script ... 617
How to convert input into upper case ... 620
How to display all Options on Search panel ... 620
How to display any page in Bootstrap popup w indow ... 620

How to enable/disable a button ... 624
How to hide 'Edit selected'/'Delete selected' buttons ... 626
How to pass values from PHP code to Javascript ... 626
How to refresh List page after Edit in popup ... 627
How to show dropdown list of US states ... 628
How to work with tabs ... 628
How to control multi-step pages ... 630
How to return other field control in the inline mode ... 630
Tabs/Sections Javascript API ... 631
About Tabs/Sections API ... 631
Tabs methods ... 632
count ... 632
activate ... 632
activeIdx ... 633
hide ... 634
show ... 634
disable ... 635
enable ... 635
headerElement ... 636
bodyElement ... 637
addTab ... 637
moveTo ... 638
Sections methods ... 639
getSectionCount ... 639
getSection ... 639
headerElement ... 640
bodyElement ... 640
expand ... 641
collapse ... 642
Labels/Titles API ... 642
About Labels/Titles API ... 642
Methods ... 643
getFieldLabel ... 643
setFieldLabel ... 644
getTableCaption ... 644
setTableCaption ... 645
getProjectLogo ... 646
setProjectLogo ... 646
getFieldTooltip ... 647
setFieldTooltip ... 647
getPlaceholder ... 648
setPlaceholder ... 649
getPageTitleTempl ... 650
setPageTitleTempl ... 651
getBreadcrumbsLabelTempl ... 652
setBreadcrumbsLabelTempl ... 653
Page class ... 655
About RunnerPage class ... 655
Methods ... 656
getCurrentRecord ... 656
getMasterRecord ... 656
hideField ... 657
setProxyValue ... 658
showField ... 659

addTab 659
deleteTab 660
setTabTitle 661
setTabWhere 661
Search API 662
About Search API 662
Methods 663
getSearchObject 663
getFieldValue 664
setFieldValue 664
getSecondFieldValue 665
setSecondFieldValue 665
getSearchOption 666
setSearchOption 667
setSearchSQL 667
getAllFieldsSearchValue 669
setAllFieldsSearchValue 669
Security API 670
About Security API 670
Methods 671
getUserGroup 671
getUserGroups 671
getUserName 672
getDisplayName 672
setDisplayName 672
isGuest 673
isAdmin 673
isLoggedIn 673
loginAs 674
logout 674
getOw nerId 674
setOw nerId 675
checkUsernamePassw ord 675
getUserData 676
currentUserData 676
getPermissions 677
setPermissions 677
Passw ord hashing 678
SQLQuery class 679
About SQLQuery class 679
Methods 679
addField 679
addWhere 680
deleteField 681
replaceField 682
replaceWhere 683
Troubleshooting tips 683
Troubleshooting charts 683
Troubleshooting custom buttons 686
Troubleshooting Javascript errors 690
Troubleshooting tips 694
Data formatting 696
Email templates 696
How to access fields in the field events 698

How to create your own Edit control plugin 699
How to display message or tooltip 705
How to mark field invalid 705
How to display data returned by stored procedure 706
How to execute SQL Server/MySQL/ORACLE stored procedures 707
Master-details relationships 707
PHPRunner session variables 708
PHPRunner templates 709
runner_mail function 711
runner_sms function 714
Template files processing rules (Files.txt) 715
Template language 716
Useful links 722
Using JOIN SQL queries 723
3 Publishing PHP application to the remote Web server 724
Using FTP client to publish PHP pages to the remote Web server 724
Using FrontPage to publish PHP pages to the remote Web server 725
4 Demo Account 726
What is the Demo Account? 726
Terms and Conditions 727
5 Web reports 728
Online report/chart builder 728
Creating web report 733
Creating web chart 746
Custom SQL 758
6 Domain host instructions 762
Yahoo! 762
Connecting to MySQL 762
Configuring FTP 767
Publishing project via FTP 769
1&1 771
Connecting to MySQL 771
Configuring FTP 776
Publishing project via FTP 777
GoDaddy.com 779
Connecting to MySQL 779
Configuring FTP 782
Publishing project via FTP 784
WebHost4Life.com 786
Connecting to MySQL 786
Configuring FTP 789
Publishing project via FTP 790
MyHosting.com 792
Connecting to MySQL 792
Configuring FTP 796
Publishing project via FTP 798
InspiRunner.com 800
Connecting to MySQL 800
Configuring FTP 802
Publishing project via FTP 803
7 How to install local web server (XAMPP) 807
8 How to add external css/php/js files 811

9 Working with MS Access databases 813
10 Working with Oracle databases 813
11 Connect to remote MySQL database via PHP 814
12 AJAX-based Functionality 817
13 Localizing PHPRunner applications 823
14 Rich Text Editor plugins 829
15 PDF view settings 832
16 Web interface guide 833
17 Mobile Template 835
18 Testing web apps in mobile mode 837
19 Error reporting 842
Part IV Order PHPRunner online 845

1 Introduction

1.1 Welcome

Welcome and thank you for choosing PHPRunner!

PHPRunner creates a set of PHP pages to access and modify any MySQL, Oracle, MS SQL Server, PostgreSQL or MS Access database. Using generated PHP pages users can search, edit, delete, and add data into database. PHPRunner is extremely easy to learn, you can get started in just 15 minutes!

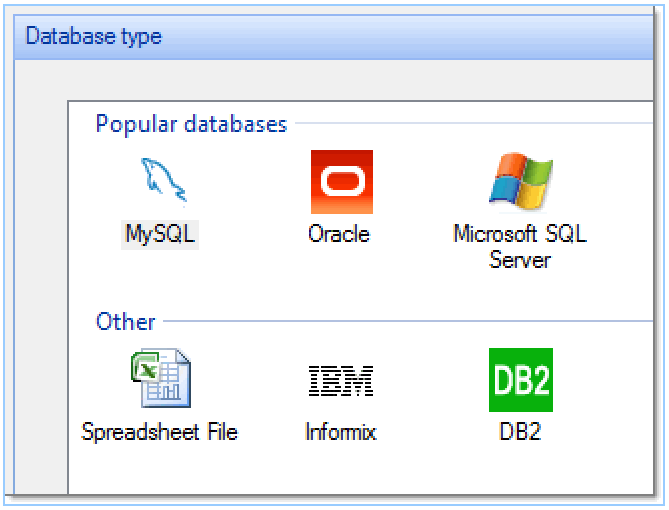
Templates

PHPRunner offers a large number of application templates - ready-made themed websites with complete graphical interface and database structure. All of the templates are easy to work with and completely customizable. The template could be used as a stand alone website, or could be integrated with other PHPRunner web applications. Some of the templates available with PHPRunner are: cars, classified ads, knowledge base, real estate, job listings, and news.

More info

Wide range of database support

PHPRunner supports four database types including MySQL, SQL Server, MS Access, and PostgreSQL. Even if you don't have a database, the software will help you create one.

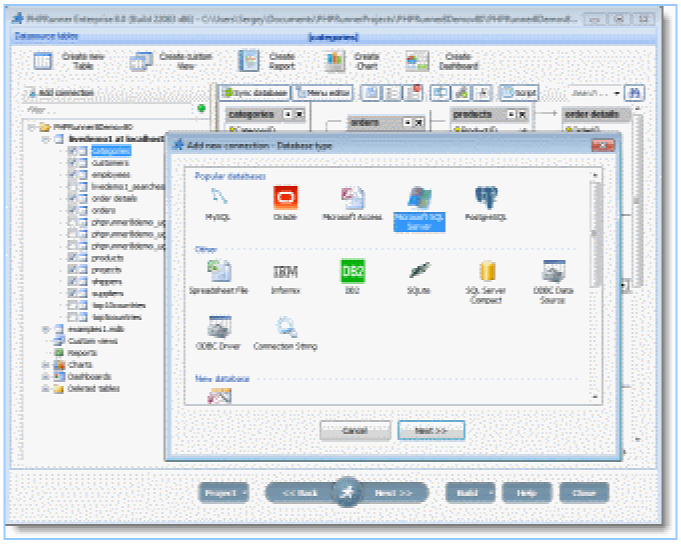


PHPRunner lets you connect to your local database or a database located on a remote server. If you have a remote MySQL database, which does not allow for a direct connection, you will be able to connect to it using "PHP proxy" method.

More info

Multiple database connections

PHPRunner lets you add multiple data sources and mix several database types like MS Access, SQL Server and MySQL in a single PHPRunner project. You can have master table in MySQL and details table in MS Access. The same applies to lookup tables.

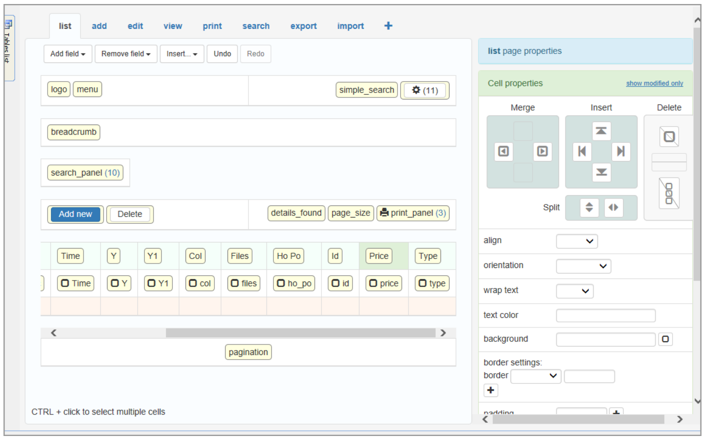


Multiple database connections feature is a part of the Enterprise Edition of PHPRunner.

More info

Page Designer

WYSIWYG Page Designer PHPRunner allows you to customize the look and feel of your application with ease of drag-n-drop. PHPRunner will present you with the proposed layout for each page. However, all of the elements on the pages can easily be modified. The Page Designer allows you to drag and drop, copy and paste the objects around the page. For all of the fields and labels on the page you can change the font, size, color, style, indentation, and alignment. The Page Designer allows you to jump into generated HTML code and make your modifications there.

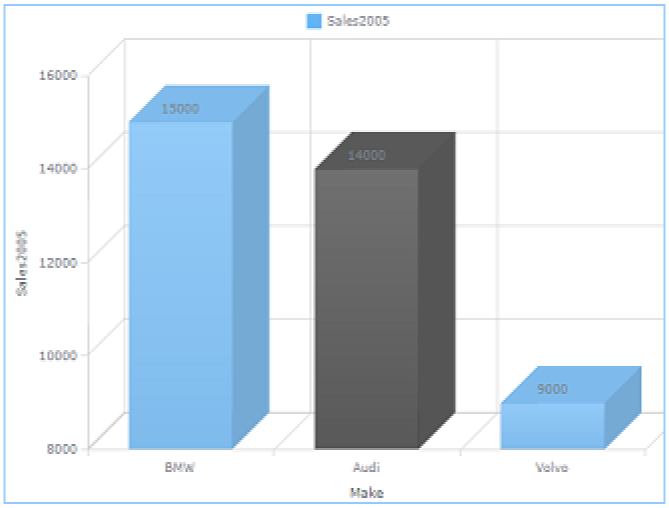


[More info](#)

Reports and Charts

PHPRunner lets you build sophisticated color-rich, highly customizable interactive charts and reports to complement your website. You will be able to choose from multiple chart and report types.

You can also get a Web Charts and Reports Builder as part of the [Enterprise Edition](#) of PHPRunner, which will let you build the charts and reports online. Just like in the software, Web Charts and Reports Builder offers a large selection of charts and reports that take just minutes to build. You can reuse the same security settings you have established in the program to decide which data sources you would like to expose to the users, and what permissions those users will have.

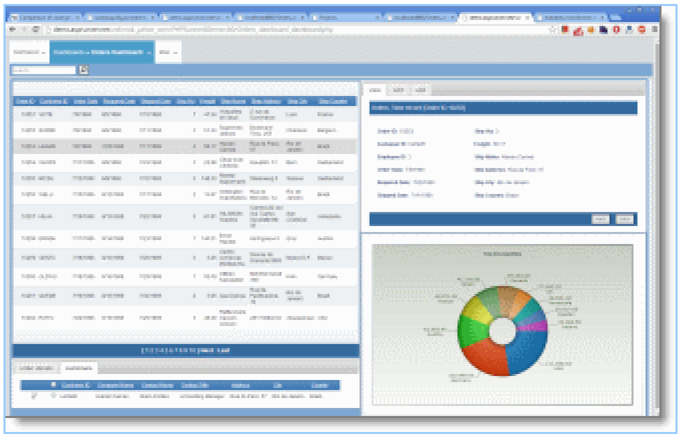


[More info](#)

Dashboards

In PHPRunner you can create various types of easy to read and powerful dashboards. They allow you to display multiple related or unrelated objects on the same page like grids, single record views, master and details together, charts, reports, search pages, etc. You can change the dashboard layout and customize its appearance.

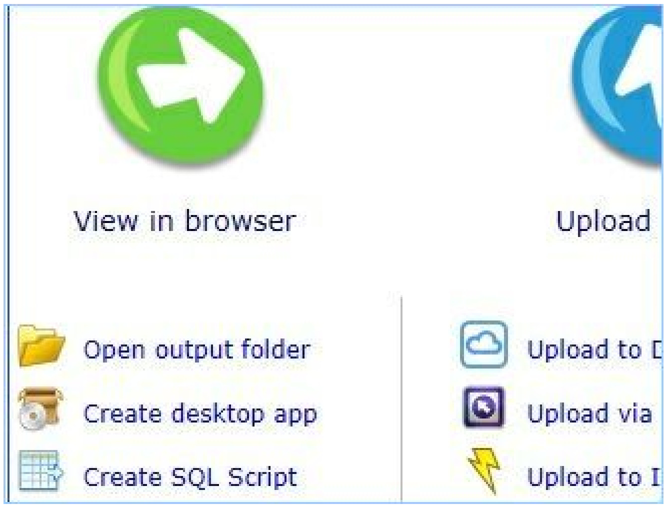
More info



Application Preview

After you have built your web application, PHPRunner provides you with multiple options how to preview it. You can view your application in browser locally. You can upload files to the remote Web server using built-in FTP client. You can publish your application to our Demo Server or simply sign up for our hosting specifically dedicated to PHPRunner customers.

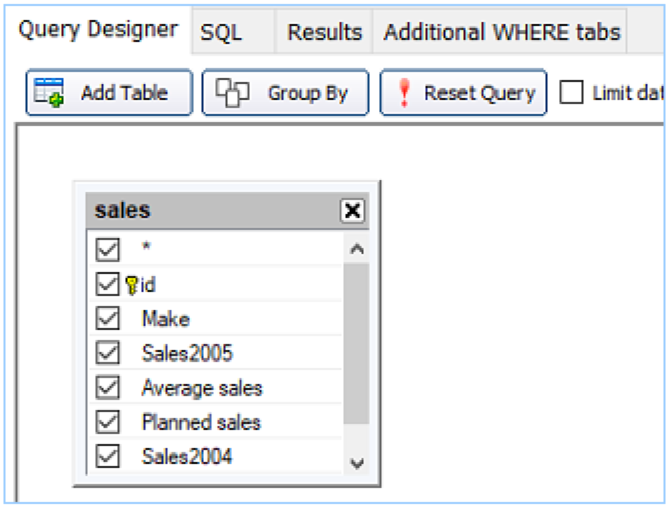
More info



SQL Editor

PHPRunner automatically creates a SQL query that can be easily modified in the graphical pane or as text. The SQL Editor also allows you to preview the results of your SQL statement, create joins with drag and drop and specify the criteria (where, order by, group by etc).

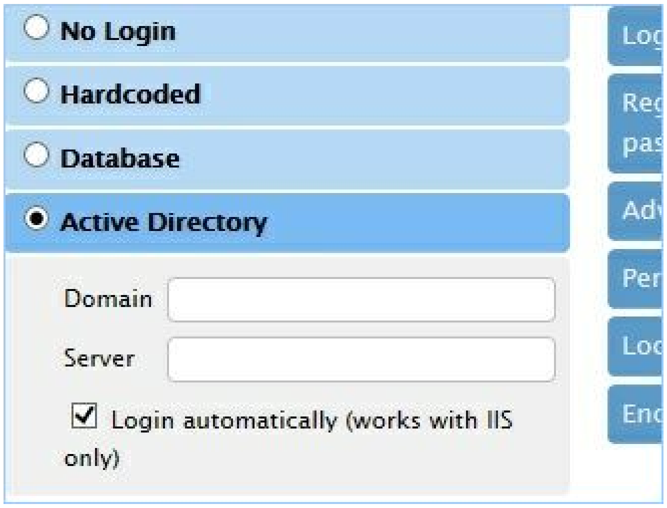
More info



Security

PHPRunner allows you to password-protect the access to your web application. You can either hardcode the username and password, store the login combinations in the database or use Active Directory authentication. You can add Login with Facebook option to your site. PHPRunner also allows you to set the user groups permission where you can restrict the tables and views, pages, and site functionality either right in the software or online.

Enterprise Edition of PHPRunner adds Active Directory support.



○ No Login
○ Hardcoded
○ Database
● Active Directory

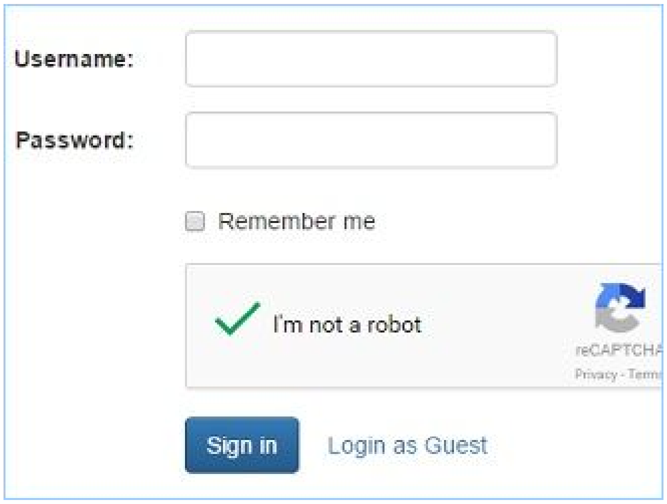
Domain
Server

☑ Login automatically (works with IIS only)

More info

User Login Settings

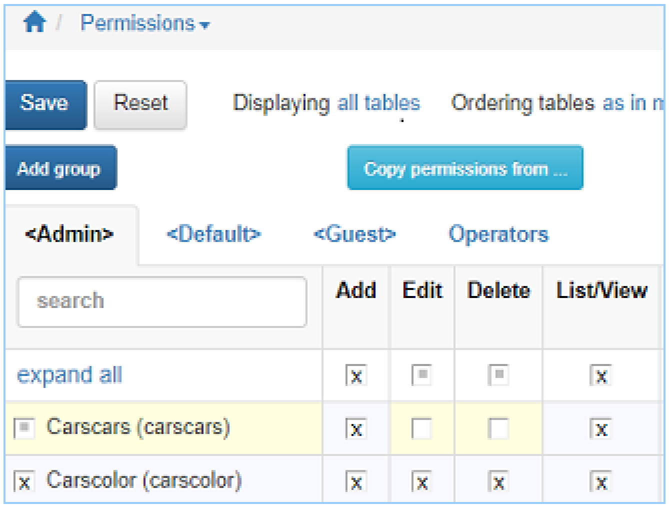
PHPRunner lets you create a New User Registration page as well as secure your pages from SPAM abusers using CAPTCHA protection, which easily determines whether the user is a computer or a human. You can also restrict the users from entering weak passwords and allow them to request password reminders.



More info

Dynamic Permissions

With dynamic permissions PHPRunner will let you create and modify the permissions and assign users to certain groups right from the generated web application. Every time you will need to modify the permissions, create a new group or assign users to groups you will not have to rebuild your project. Dynamic Permissions are especially helpful in larger corporations where the application security administrators are not the actual users of PHPRunner software.



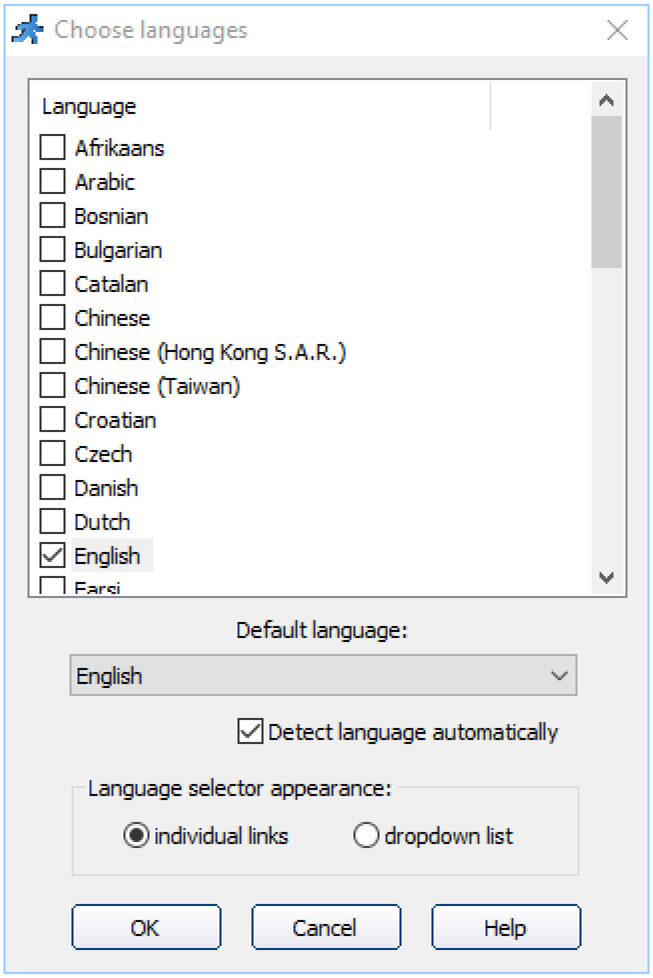
search	Add	Edit	Delete	List/View
expand all	☒	▣	▣	☒
▣ Carscars (carscars)	☒	□	□	☒
☒ Carscolor (carscolor)	☒	☒	☒	☒

More info

Multilanguage support

PHPRunner supports more than 30 languages giving your users an ability to choose the language while logging in. You can also add the translations for table names and fields. The list of supported languages includes: Afrikaans, Arabic, Bosnian, Catalan, Chinese, Chinese (Hong Kong S.A.R.), Chinese (Taiwan), Croatian, Danish, Dutch (Belgian), Dutch (Standard), English, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Malaysian, Norwegian (Bokmal), Polish, Portuguese (Brazil), Portuguese (Standard), Romanian, Slovak, Spanish, Swedish, Thai, Turkish, Urdu.

More info



Choose languages

Language

□ Afrikaans
□ Arabic
□ Bosnian
□ Bulgarian
□ Catalan
□ Chinese
□ Chinese (Hong Kong S.A.R.)
□ Chinese (Taiwan)
□ Croatian
□ Czech
□ Danish
□ Dutch
☑ English
□ Farsi

Default language: English

☑ Detect language automatically

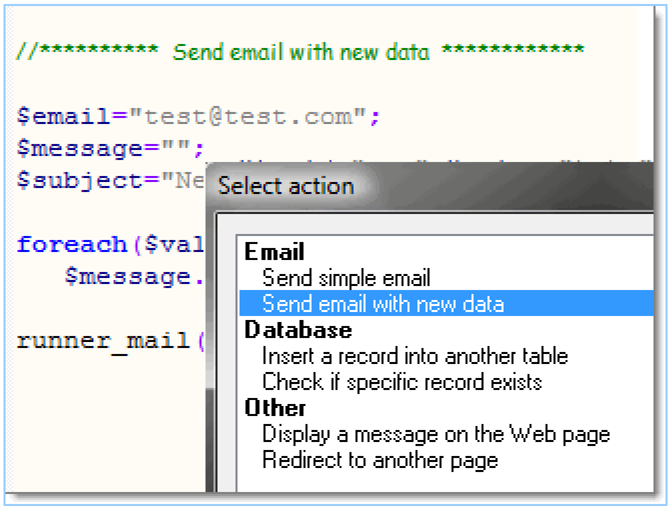
Language selector appearance:

● individual links ○ dropdown list

Events

With PHPRunner you can expand the functionality of your application by inserting events - fragments of PHP code. You will be able to define what an event should do and when it should be executed. A typical event would be to send an email with a new data, save data in another table, check record uniqueness, show related to current record info etc. You can either select one of predefined events or write your own from scratch.

More info



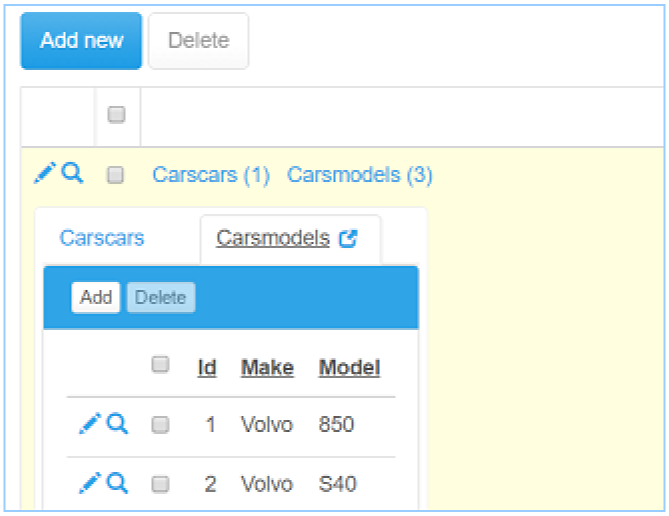
```
//********** Send email with new data ************

$email="test@test.com";
$message="";
$subject="Ne

foreach($val
    $message.

runner_mail(
```

Select action
Email
Send simple email
Send email with new data
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page

Master-Details Relationships

In PHPRunner all of your data sources and the relationships between them, including master-details, are visually displayed making them very intuitive to understand and manage. You will be able to link two or more data sets with easy of drag-n-drop. Once you have the relationships established, in the application you can navigate through master records and quickly jump over to the details of those records. Some basic scenarios of master-details relationships would be customer and orders data, patient and medical records or student-and courses information.

More info

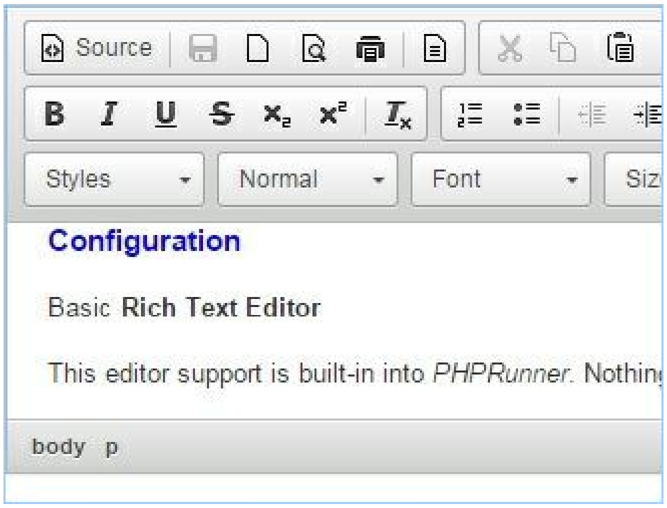


Id	Make	Model
1	Volvo	850
2	Volvo	S40

Rich Text Editor Controls

PHPRunner supports three third party rich text editors to allow for a great control over content formatting including common structural treatments like lists; formatting treatments like bold and italic text, and drag-and-drop inclusion and sizing of images. The three options include the Basic Rich Text Editor, CKEditor, and InnovaStudio Editor that vary in features, versatility, and footprint.

More info



Configuration

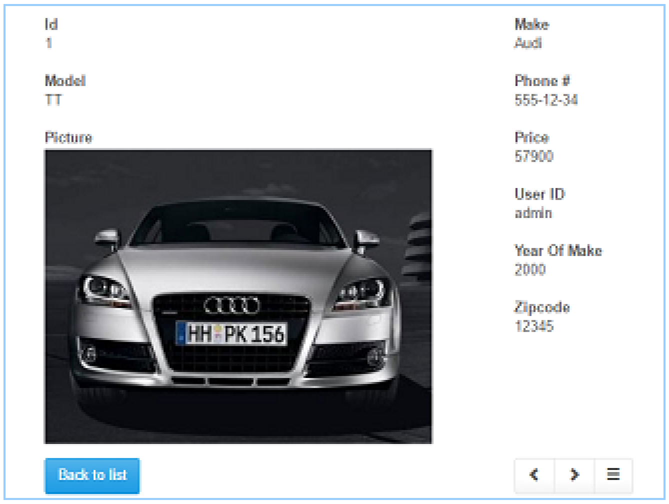
Basic Rich Text Editor

This editor support is built-in into *PHPRunner*. Nothin

Images and Documents

PHPRunner lets you upload the documents and images of any type to the database or to a directory on the web server. You can also create the image thumbnails on the fly, resize them on upload, and display them using iBox.

More info


Id
1

Model
TT

Picture

Make
Audi

Phone #
555-12-34

Price
57900

User ID
admin

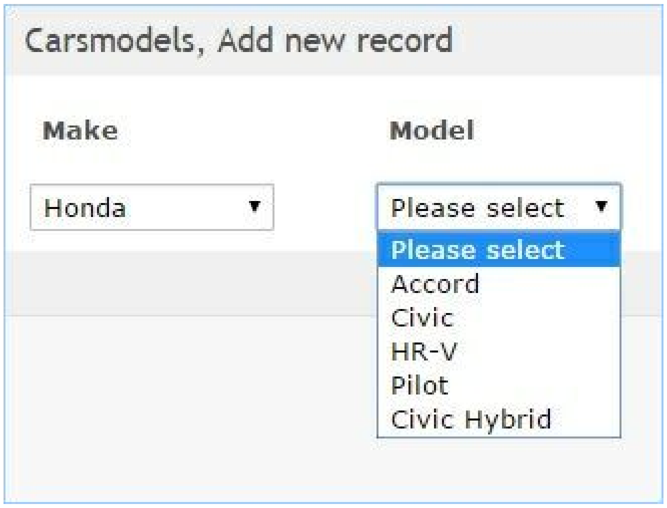
Year Of Make
2000

Zipcode
12345

Dependent Drop-down Boxes

With PHPRunner, you can use linked drop-down boxes, where values shown in the second drop-down box depend on the value you've chosen in the first one. You can link together as many drop-down boxes as you need in a linear chain or have multiple drop-down boxes link to the same master drop-down control.

More info



Carsmodels, Add new record

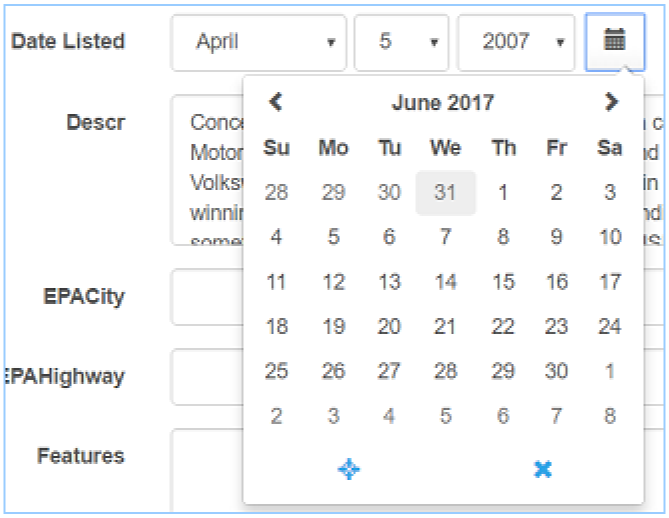
Make: Honda

Model: Please select

Edit Controls

PHPRunner offers a large variety of edit controls, which allow you to customize the appearance of the fields on Edit and Add pages. The field formats that you can choose from include text field, date, time, check box, radio button, file/image, lookup wizard and others.

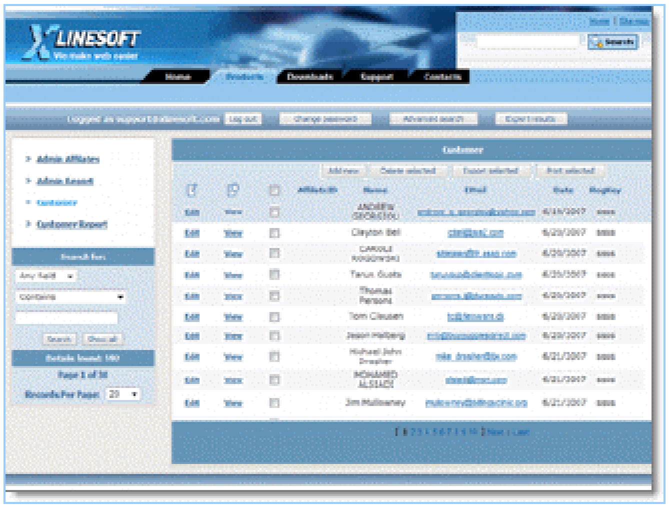
More info



Date Listed: April 5 2007

Integration With Existing Website

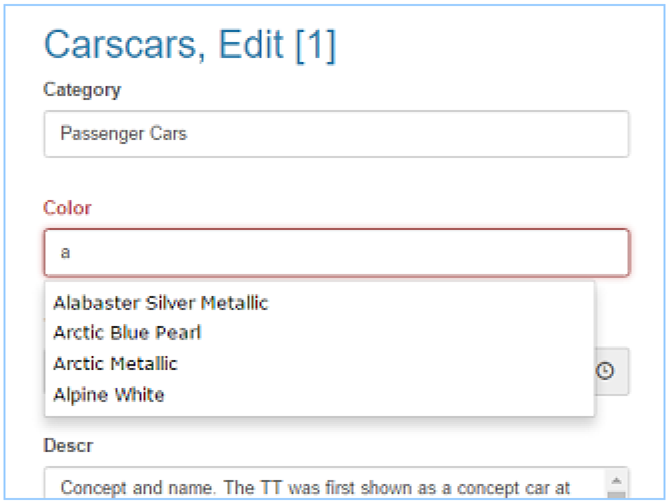
PHPRunner lets you seamlessly integrate the web applications you build into your existing website. You will be able to closely match the look and feel of all of your pages.



Ajax-based functionality

PHPRunner comes with built-in AJAX-based functionality making your websites much more user-friendly. You can search information more easily than ever with google-like auto-suggest feature. The AJAX driven dependent dropdown boxes also load much faster. With Ajax you will be able to preview the content by simply hovering over the links.

More info



Carscars, Edit [1]

Category

Passenger Cars

Color

a

Alabaster Silver Metallic
Arctic Blue Pearl
Arctic Metallic
Alpine White

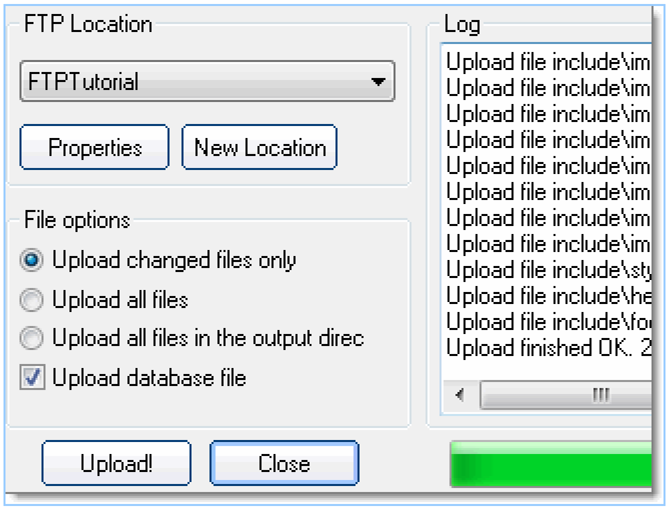
Descr

Concept and name. The TT was first shown as a concept car at

FTP Upload

PHPRunner will let you upload the entire set of pages to your web server via FTP. You will be able to set the upload properties based on your needs.

More info



FTP Location

FTPTutorial

File options

● Upload changed files only

○ Upload all files

○ Upload all files in the output direc

☑ Upload database file

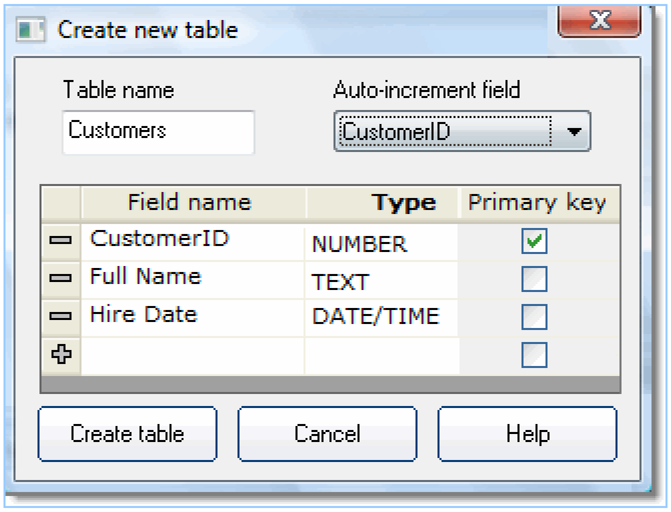
Log

```
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\im
Upload file include\sty
Upload file include\he
Upload file include\fo
Upload finished OK. 2
```

Create/Modify Database Tables

PHPRunner will let you create and modify tables right in the software. You will have access to table properties where you can set the field names, types, sizes, as well as set the primary key field. If you don't have a database at all, PHPRunner will let you create one too.

More info



Create new table

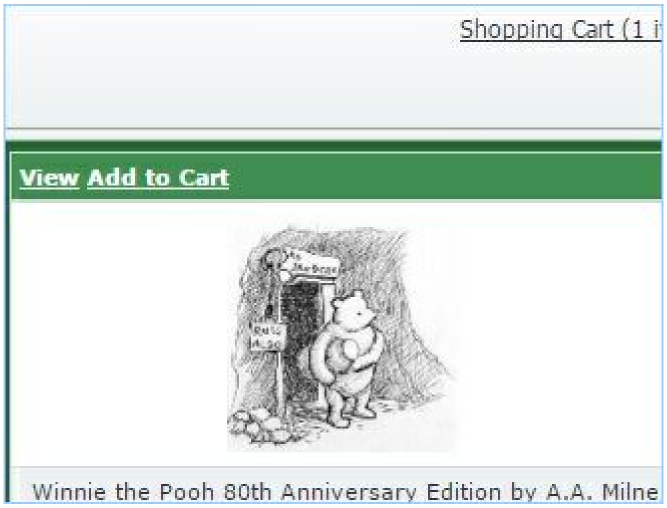
Table name: Customers

Auto-increment field: CustomerID

Field name	Type	Primary key
CustomerID	NUMBER	☑
Full Name	TEXT	□
Hire Date	DATE/TIME	□
		□

Additional Templates

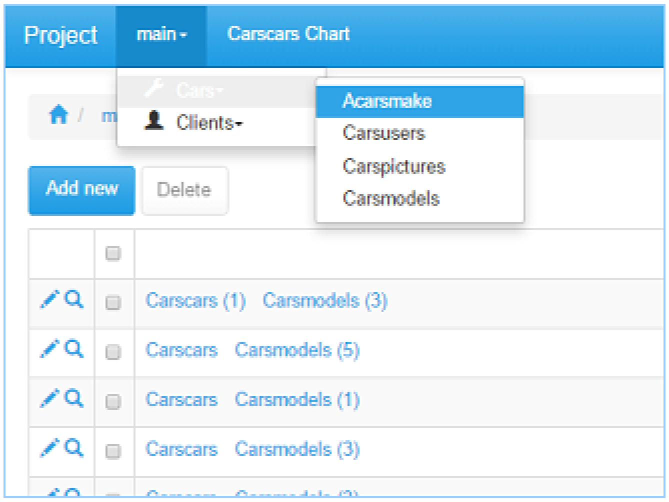
To complement the list of our built-in application templates we offer three more templates that make up the Templates Pack that could significantly enhance your web application. The Templates Pack includes the Shopping Cart template, Calendar template and Members template. The Templates Pack is available for purchase to all existing PHPRunner users and is 100% guaranteed to integrate with your other PHPRunner projects.



Cascade Menu Builder

Cascade Menu Builder lets you organize your tables into a multi-level menu for quicker navigation. This feature is particularly useful to users with large number of tables. You can have as many groups and subgroups as you need. Depending on your page layout the cascade menu will appear as a horizontal menu, vertical menu, or a tree-like vertical menu.

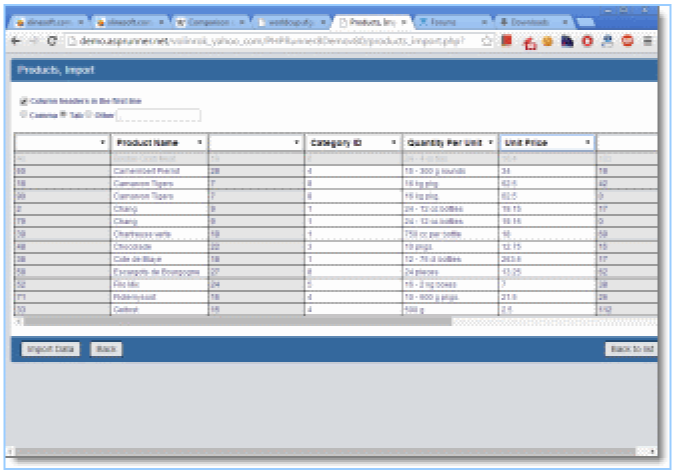
More info



Import

PHPRunner allows you to import CSV files, Excel 2007 (.xlsx) and Excel 97-2003 (.xls) files. You can copy and paste import data instead of uploading the file that can be extremely on mobile devices. Column mapping is supported.

More info



	Product Name		Category ID	Quantity Per Unit	Unit Price	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
60	Camembert Pierrot	28	4	15 - 300 g rounds	34	19
18	Carnarvon Tigers	7	8	16 kg pkg.	62.5	42
99	Carnarvon Tigers	7	8	16 kg pkg.	62.5	0
2	Chang	1	1	24 - 12 oz bottles	19.15	17
79	Chang	1	1	24 - 12 oz bottles	19.15	0
39	Chartreuse verte	18	1	750 cc per bottle	18	69
48	Chocolade	22	3	10 pkgs.	12.75	15
38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17
58	Escargots de Bourgogne	27	8	24 pieces	13.25	62
52	Filo Mix	24	5	16 - 2 kg boxes	7	38
71	Fløtemysost	15	4	10 - 500 g pkgs.	21.5	26
33	Geitost	15	4	500 g	2.5	112

1.2 System requirements

To run PHPRunner, your system should be equipped with the following:

- Windows 10, Windows 8, Windows 7, Windows 2008, Windows Vista.
- Internet Explorer 9 or better.

Web server requirements

- Internet Information Server (5.0 or later) or Apache. See How to install local server.
- PHP 5.x, PHP 7
- PHPRunner comes with built-in web server as well which is perfect for local testing.

Supported databases

- MySQL
- Maria DB
- Microsoft SQL Server
- PostgreSQL
- Oracle
- Microsoft Access

- DB2
- Informix
- SQLite
- Any ODBC-enabled database

1.3 Editions comparison

Table shows the difference between PHPRunner Trial, Standard and Enterprise Editions.

Features	Trial	Standard	Enterprise
Number of database objects per project *(objects are tables, charts, reports, custom views)*	20	Unlimited	Unlimited
Number of builds per project	200	Unlimited	Unlimited
"Unregistered" banner at the top of all pages	✓		
"Evaluation version" message on charts	✓		
Web report/chart builder			✓
Active Directory support			✓
Data encryption			✓
Multiple database connections			✓

1.4 Licensing details

License

By receiving and/or using PHPRunner, you accept the following Evaluation and Registered User Agreement. This agreement is a binding legal agreement between XLineSoft and the purchasers, users or evaluators of XlineSoft's software and products. If you do not intend to honor this agreement, remove all installed XLineSoft products from your computer now.

Evaluation (Unregistered) and Registered User Agreement

You may evaluate this program for maximum of twenty one calendar days, after which you must register the program with XLineSoft or remove the software from your computer.

You may allow other users to evaluate copies of the unregistered software. All evaluation users are subject to the terms of this agreement.

The evaluator/user/buyer/owner is not allowed to attempt to reverse engineer, disassemble or decompile any XLineSoft products.

XLineSoft name and any logo or graphics file that represent our software may not be used in any way to promote products developed with our software. All parts of XLineSoft products are copyright protected. No program, code, part, image, video clip, audio sample, text or computer generated sequence of images may be copied or used in any way by the user except as intended within the bounds of the single user program.

The evaluator/user/buyer/owner of XLineSoft will indemnify, hold harmless, and defend XLineSoft against lawsuits, claims, costs associated with defense or accusations that result from the use of XLineSoft products.

XLineSoft is not responsible for any damages whatsoever, including loss of information, interruption of business, personal injury and/or any damage or consequential damage without limitation, incurred before, during or after the use of our products. Our entire liability, without exception, is limited to the customers' reimbursement of the purchase price of the software (maximum being the suggested retail price as listed by XLineSoft) in exchange for the return of the product, all copies, registration papers and manuals, and all materials that constitute a transfer of ownership from the customer back to XLineSoft.

Each registered copy of the PHPRunner may be used in only one single location by one user. Use of the software means that you have loaded the program and run it or have installed the program onto a computer. If you install the software onto a multi-user platform or network, each and every individual user of the software must be registered separately.

You may make one copy of the registered software for backup purposes, providing you only have one copy installed on one computer being used by one person. If any person other than yourself uses XLineSoft software registered in your name, regardless of whether it is at the same time or different times, then this agreement is being violated!

The sale of and/or distribution of registered copies of this software is strictly forbidden. It is a violation of this agreement to loan, rent, lease, borrow, or transfer the use of registered copies of XLineSoft products.

1.5 What is the registration

PHPRunner is "Try before you buy" software. This means that we have made the software available to you for free evaluation. You are entitled to evaluate the software for up to 21 days without obligation to pay. After 21 days, if you decide to keep the software, you must register your copy with us.

Demo version (non-registered) of PHPRunner is a "full-featured" release. This means that the same capabilities available in the registered software are present in the non-registered software. This allows you to try out all the features in PHPRunner to confirm that they work to your satisfaction.

Registration entitles you free technical support for 90 days and one year of free upgrades. Registration may also entitle you to discounts on new software releases from XLineSoft. We will also send you information bulletins by email to let you know about what’s happening with other XLineSoft products.

Finally, by registering the software, you provide us with the resources and incentive to support the software with updates and to develop additional quality software products in the future.

How can I register?

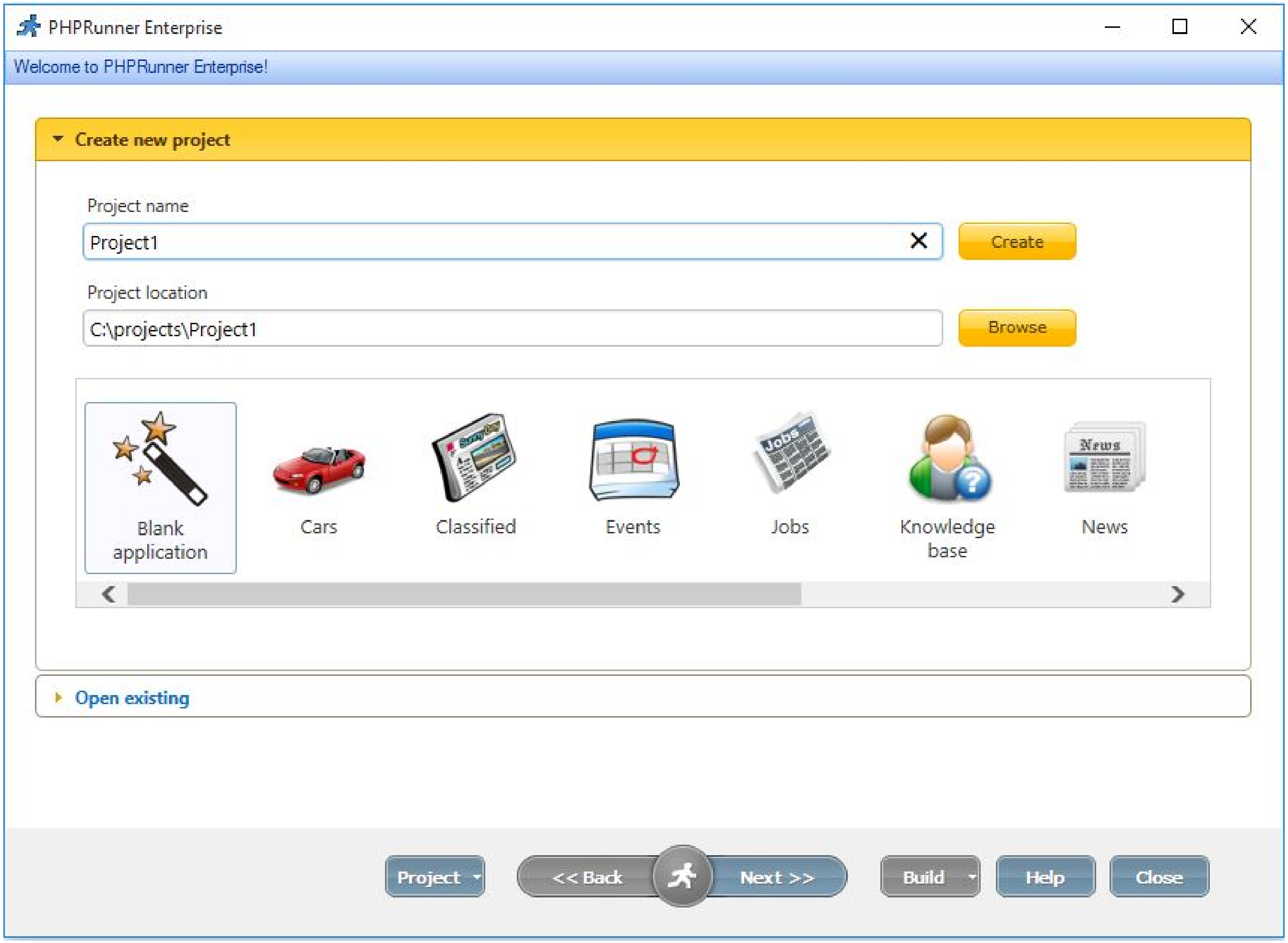
1.6 Quick start guide

This is step-by-step tutorial that will help you build your first PHP pages quickly. Hit **Help** button if you need more information during the process.

Note: To get more familiar with PHPRunner, you can also watch flash tutorials that are available at http://www.xlinesoft.com/phprunner/php-database.htm.

Run PHPRunner application after you installed it on your computer. It can be found at **Start -> Programs -> PHPRunner -> PHPRunner**.

On the first screen enter a project name and a project location. If you already have a database, select **Blank application** to build the project from scratch. Otherwise, you can select one of the predefined templates and get a themed web site and database created for you. Click **Next**>>.



PHPRunner Enterprise

Welcome to PHPRunner Enterprise!

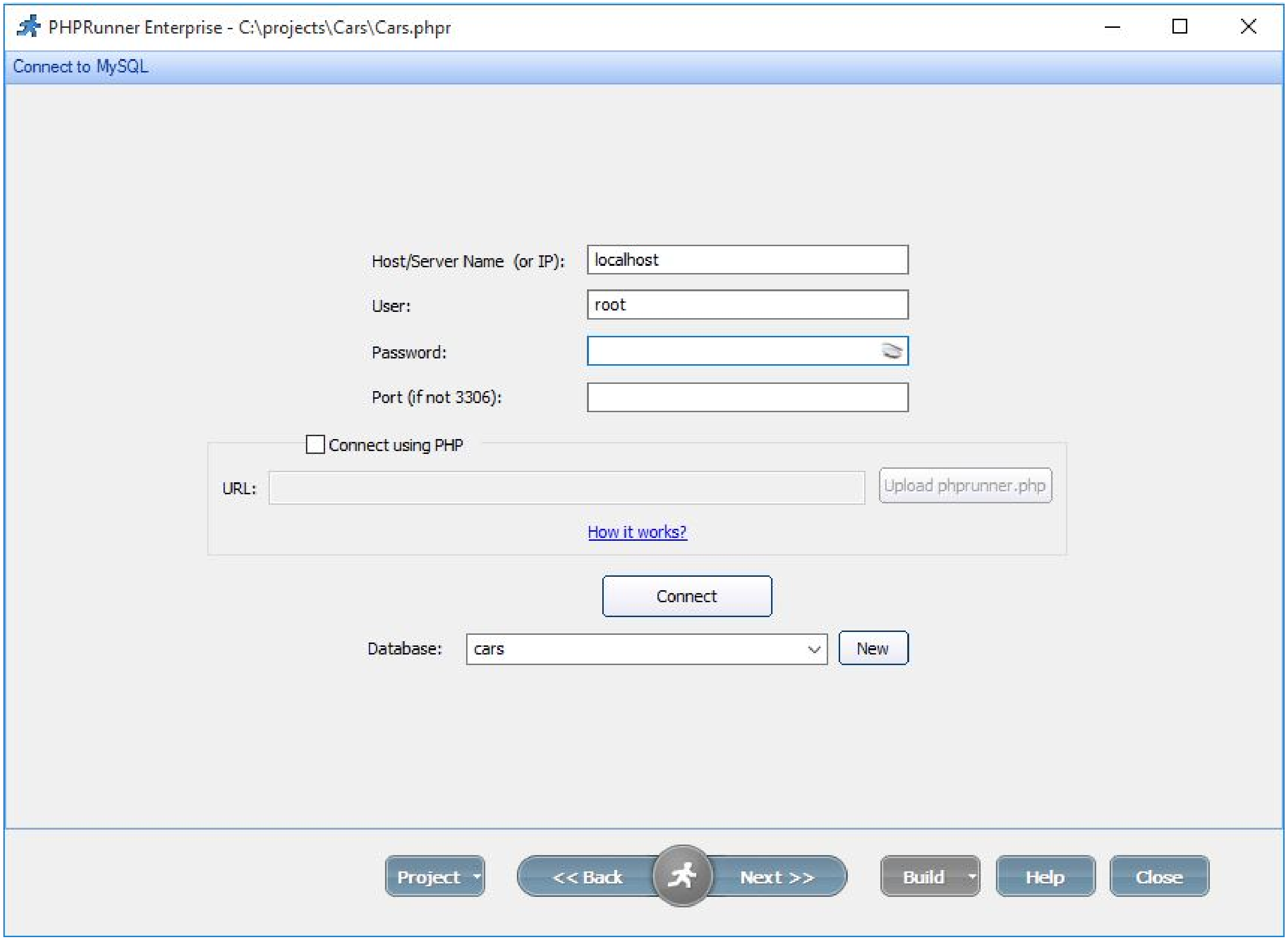
Create new project

Project name: Project1

Project location: C:\projects\Project1

On the next screen choose database type and click **Next**>>. In our example, we will connect to MySQL. If you do not have MySQL server, see How to install local server.

Then select a database or create new one. When finished click **Next**>>.



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Connect to MySQL

Host/Server Name (or IP): localhost

User: root

Password:

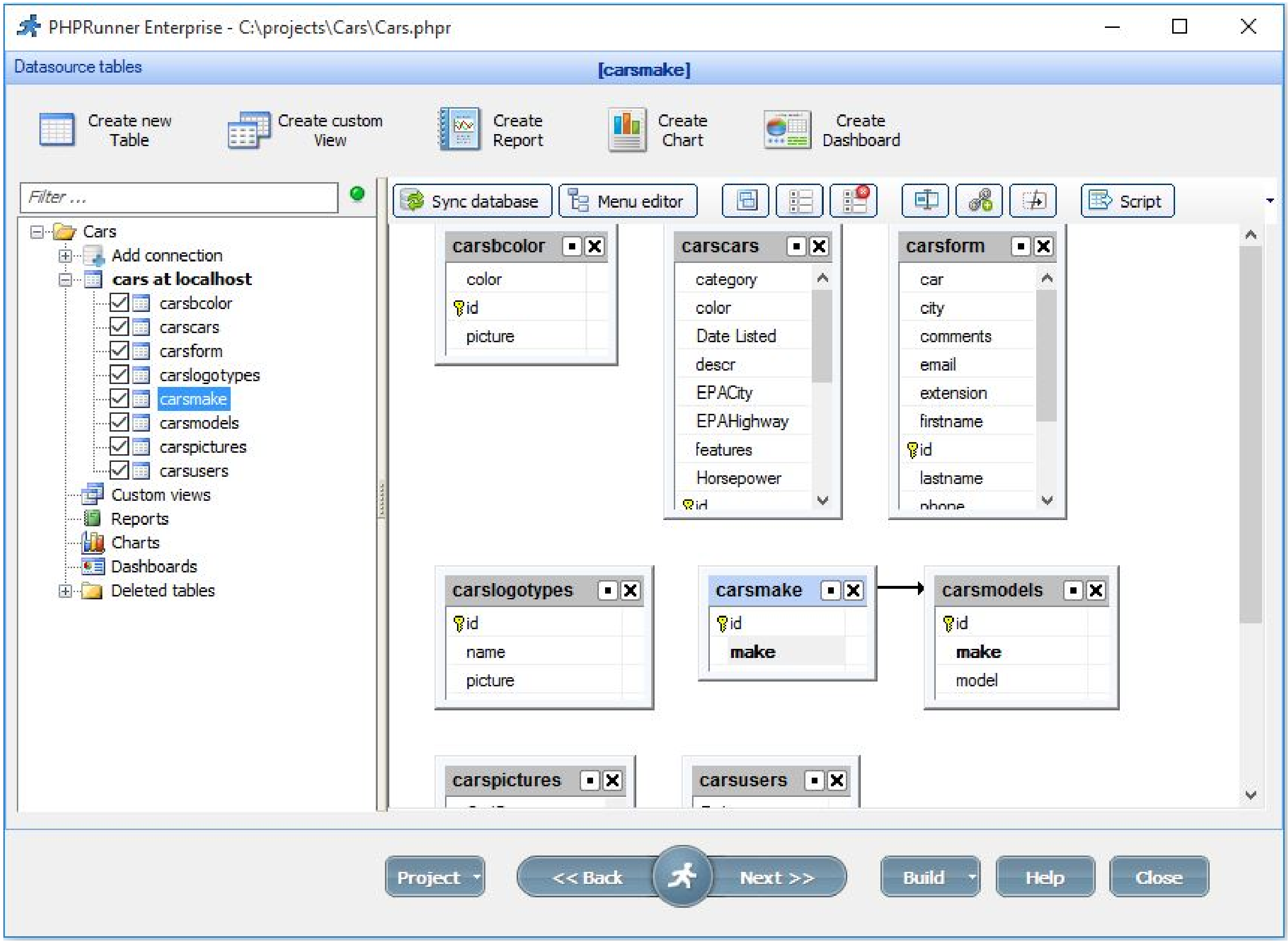
Port (if not 3306):

□ Connect using PHP

URL:

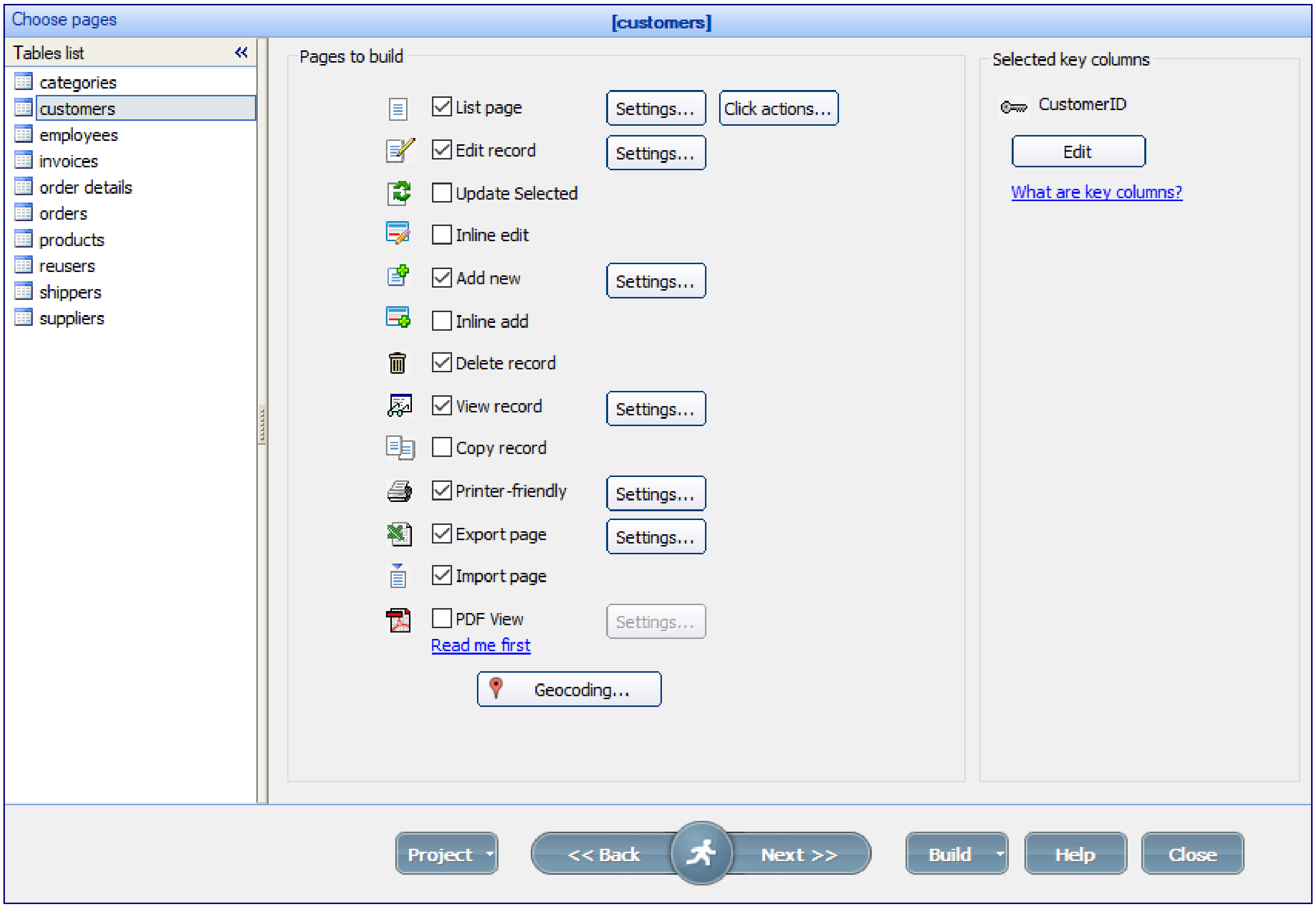
Database: cars

Select datasource tables from the list of available tables. Click **Next**>>. On the next screen you can modify SQL query.



On the next screen choose pages you would like to create and select key fields.

Note: Key column is the datasource column which lets you query each table row individually and modify each row without altering other rows in the same table. The values that compose a key column are always unique. When finished click **Next**>>.



Choose pages [customers]

Tables list

- categories
- customers
- employees
- invoices
- order details
- orders
- products
- reusers
- shippers
- suppliers

Pages to build

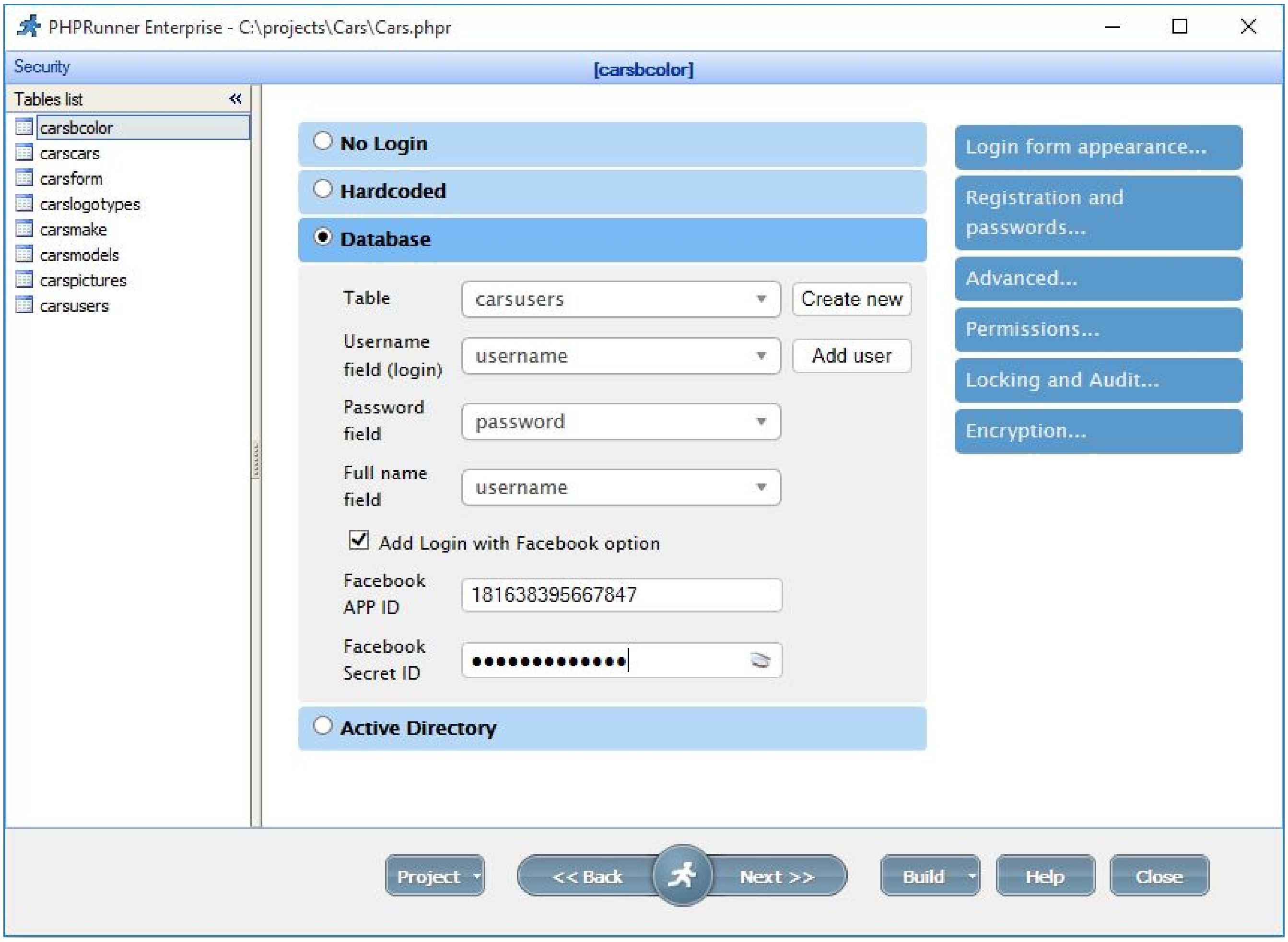
- ☑ List page
- ☑ Edit record
- □ Update Selected
- □ Inline edit
- ☑ Add new
- □ Inline add
- ☑ Delete record
- ☑ View record
- □ Copy record
- ☑ Printer-friendly
- ☑ Export page
- ☑ Import page
- □ PDF View

Selected key columns

CustomerID

On the next several screens you can choose the fields to appear on each page, customize your data appearance and set the fields order on each page. You can also modify the miscellaneous settings.

Security screen allows you to protect an access to your database. Refer to security settings for more information. Click **Next**>>.



○ No Login

○ Hardcoded

● Database

Table: carsusers

Username field (login): username

Password field: password

Full name field: username

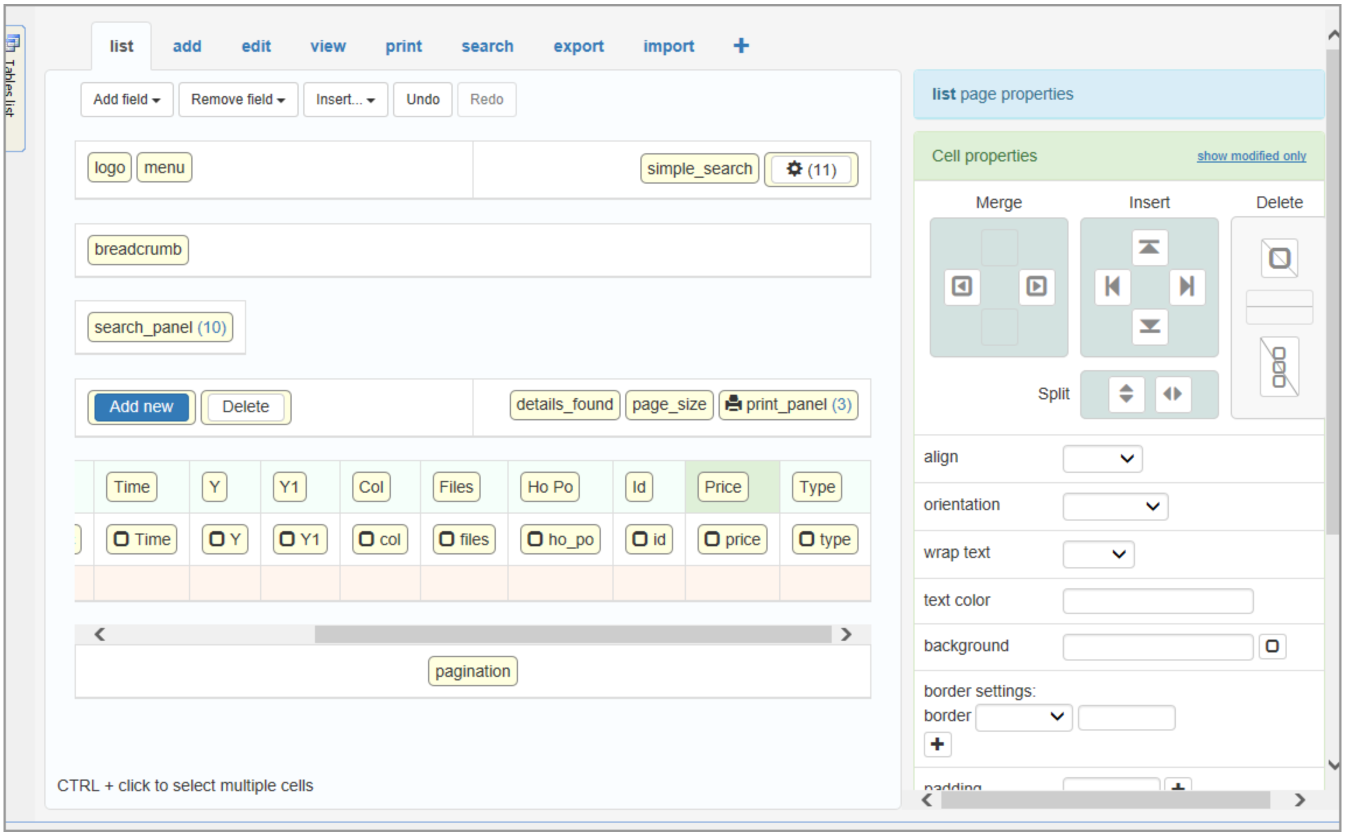
☑ Add Login with Facebook option

Facebook APP ID: 181638395667847

Facebook Secret ID: •••••••••••••

○ Active Directory

Page Designer allows you to customize the look and feel of your web pages in a more user-friendly way.



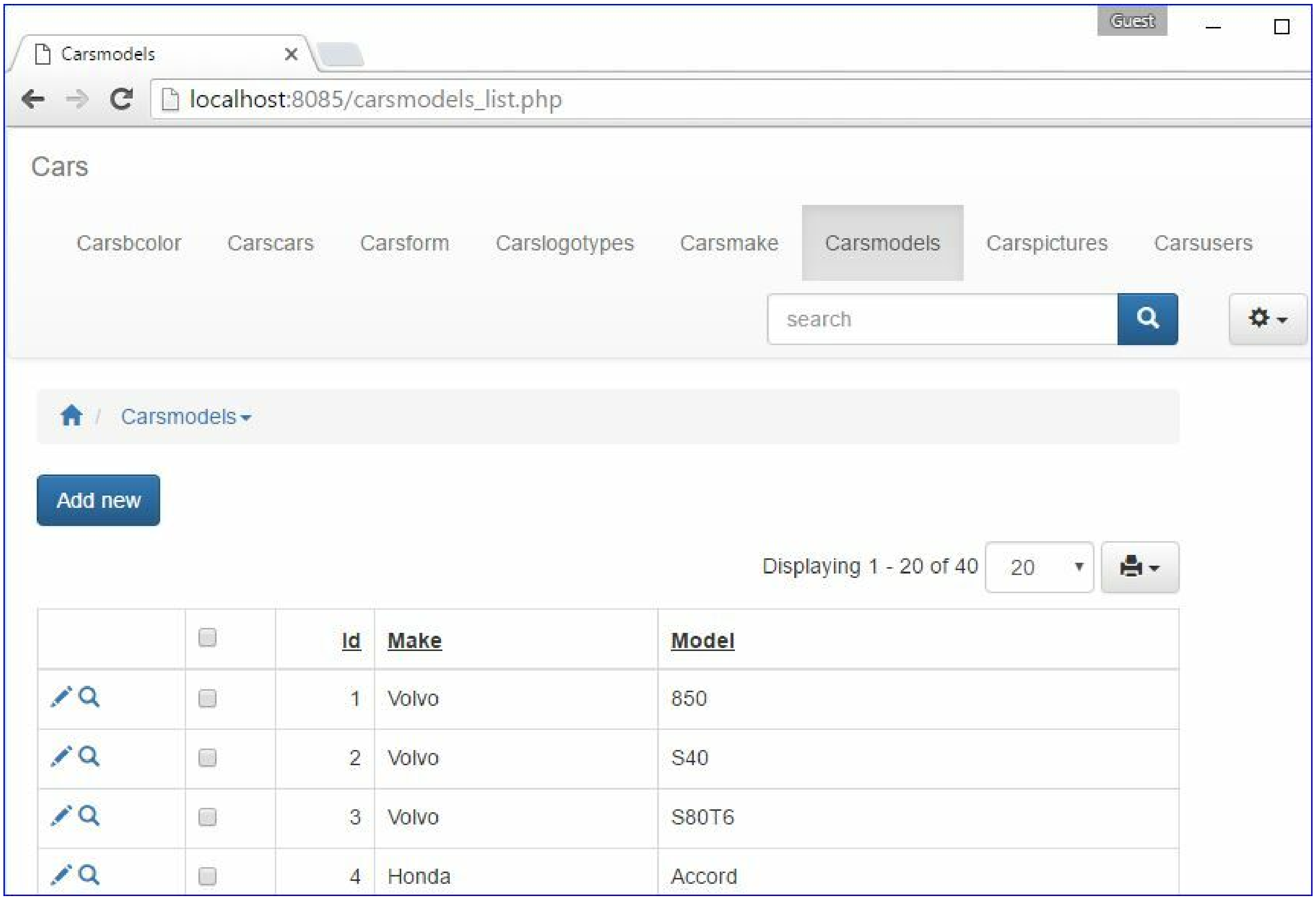
On the next screen you can [choose and modify style and color scheme](#) for displaying the pages.

On the next screen you can enhance your web site functionality by adding [events](#).

You are almost finished! On the next screen select [output directory](#) where you like to put generated PHP pages and image files. Press **Build** to generate files. You are done! Now you are ready to [test the generated application](#).

To preview your application locally, you need the Internet Information Server or Apache up and running on your computer. If you do not have a web server, see [How to install a local Web server](#). You can also upload the files to the remote Web server using built-in [FTP client](#). You can publish your application to our [Demo Server](#) or simply sign up for our hosting specifically dedicated to PHPRunner customers.

Here is the sample generated page:



Displaying 1 - 20 of 40

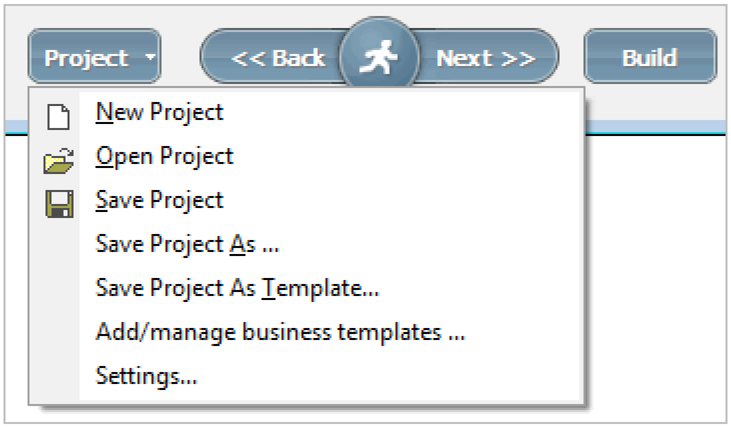
		Id	Make	Model
	□	1	Volvo	850
	□	2	Volvo	S40
	□	3	Volvo	S80T6
	□	4	Honda	Accord

2 Using PHPRunner

2.1 Working with projects

Quick jump	
Creating new project	Saving your project as a Template
Project structure	Add template to project
Opening an existing project	Project settings
Saving a project	

Click **Project** button and select one of the option from the drop down list to create new project, open or save an existing one, define project settings.



Creating new project

To create new project, click **Project** button and select **New Project**. When you create a new project you have two options:

- Create a new project from scratch by selecting **Blank application**.
- Create a project from the template.

For more information about templates that are shipped with PHPRunner, see What are templates.

Project structure

With PHPRunner, you can save all your settings in a single project file and do not have to go through individual files if you simply need to change a single label or field format.

Each project in PHPRunner saves to its own directory, which contains the following subdirectories:

1. *visual* - Contains modified visual templates.
2. *tmp* - Temporary storage of visual templates and other files. *tmp\backup* directory stores backup copies of your project. Project backup name (i.e. Project4.2011-11-09 10_08_00.w.zip) contains date and time of when backup was created.
3. *output* - Directory with output files. You can point output directory to another folder on Output directory screen.
4. *source* - contains additional files to be included in the build process.
5. *styles* - contains project styles and color schemes

The Default directory for a new project is *C:\Users\<username>\Documents\PHPRunnerProjects\project_name*. The Project file will be saved in the project directory as *project_name.phpr*.

When you open a project created with PHPRunner 4.x or older, you are prompted to choose a project directory. After you select a directory, your project file is copied to it. Then the next time you open your project it will be from the folder you selected, not from the original location.

If you upload files to the Web server using third party FTP client software, you must upload the entire contents of the output directory.

When you make a backup of your project, you should include all files in the project directory together with all subdirectories. At a minimum, you should backup the project file itself along with all files in the visual directory.

Opening an existing project

To open existing project, select **Open project**.

Saving a project

If you want to save the current project under a different name - for example, development purposes or to create a backup - select **Save Project As** option. A new project directory will be created and all necessary project files will be copied to it.

Note: PHPRunner creates a new project automatically upon startup.

Saving your project as a Template

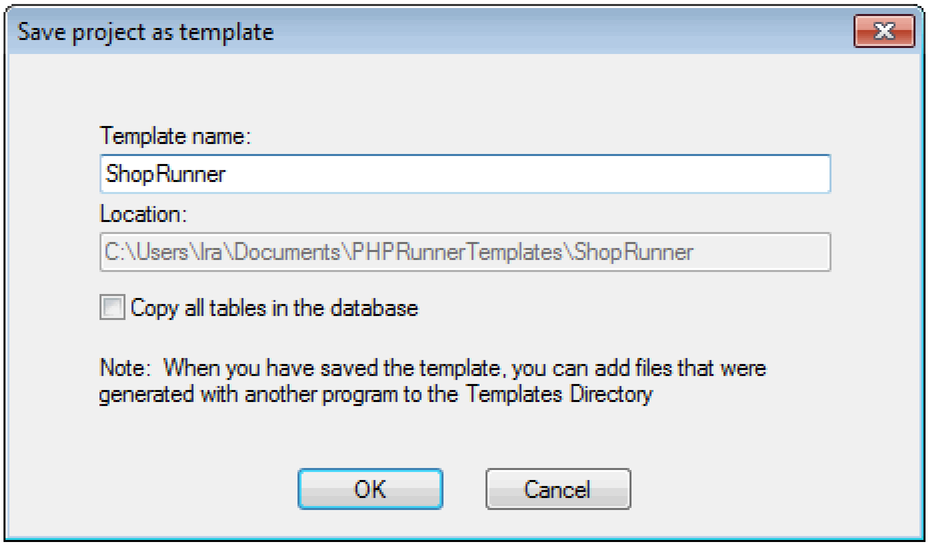
You can save your project as a template.

In this case, your database along with your project file and all files you have edited with the visual editor will be saved.

Note: Save Project As Template option is available only in MySql projects.

While saving Project As Template you need to type in the template name you wish your project to be saved under. The template will be saved in the Business Templates Directory (*by default in C:\Users\<username>\Documents\PHPRunnerTemplats\project_name*).

Note: After your template is saved, you can add files that are not generated by PHPRunner to the template directory.



Save project as template

Template name: ShopRunner

Location: C:\Users\Ira\Documents\PHPRunnerTemplates\ShopRunner

□ Copy all tables in the database

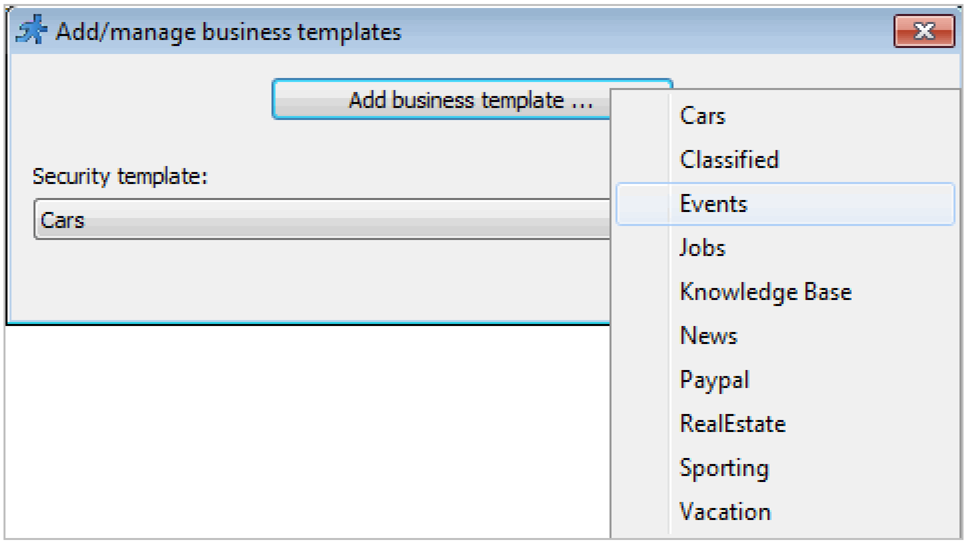
Note: When you have saved the template, you can add files that were generated with another program to the Templates Directory

After that saved Template will be available on the list of templates when you create a new project.

You can also add a thumbnail image to the template that will be displayed on templates list. Place an image named *preview.gif* (JPG and PNG formats are supported also) to template folder. Image size should be 130x97.

Add template to project

You can create new project using two templates or add a template to existing project. To avoid replacing template tables with existing all PHPRunner business templates (tables and files) have prefix.



Add/manage business templates

Security template: Cars

If you have added several templates to the project you can choose one to inherit security settings from in **Security template** dropdown box.

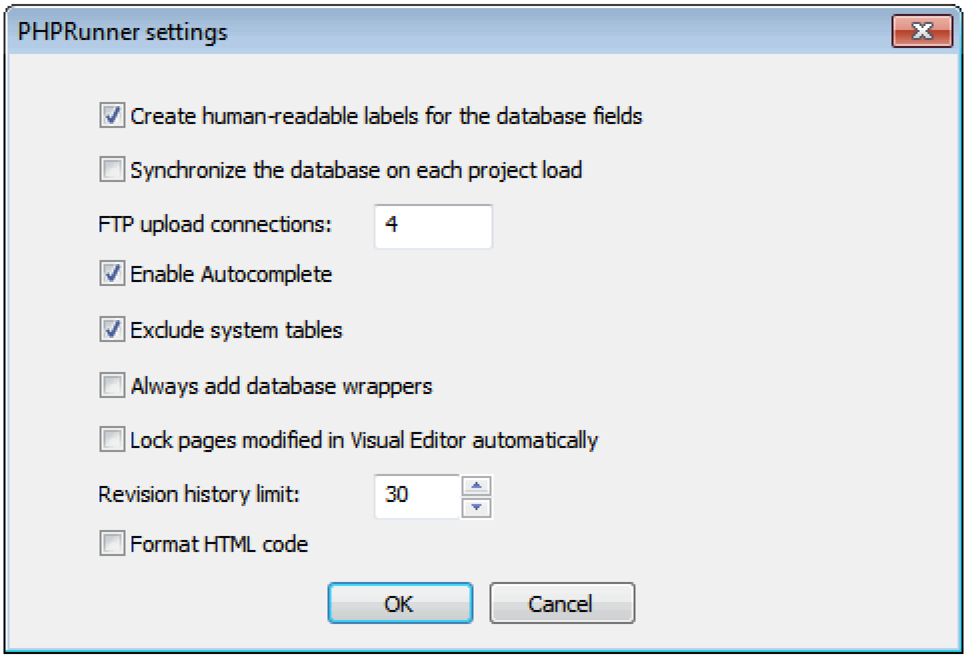
If you add more than one template to the project you can get some errors trying to run your project right after that. This what you need to do in order to add two or more templates to the same project.

1. Create a temporary test project using second template

2. Proceed to AfterAppInit event and copy all code there. Close temporary project without saving.

3. Go to your original project and paste this code to the end of AfterAppInit event.

Project settings



PHPRunner settings

☑ Create human-readable labels for the database fields

☐ Synchronize the database on each project load

FTP upload connections: 4

☑ Enable Autocomplete

☑ Exclude system tables

☐ Always add database wrappers

☐ Lock pages modified in Visual Editor automatically

Revision history limit: 30

☐ Format HTML code

Use **Create human-readable labels for the database fields** option to convert field names into the more human readable format. For example, if this option is enabled, the field name *id* will be displayed as *Id*, *last_name* as *Last Name, FirstName* as *First Name* etc.

Use **Synchronize the database on each project load** option to enable/disable automatic database synchronization. We recommend to use this option for small or local databases. For more information about database synchronization, see Datasource tables: Synchronize database.

To increase or decrease the files upload speed, change the number of **FTP upload connections**.

Enable Autocomplete option enables Intellisense that provides autocomplete popups and function calltips in Event Editor.

Exclude system tables option excludes system tables from the list of tables for Users table, Lookup tables etc.

Use **Always add database wrappers** option to add wrappers to all names of fields and tables. When this option is disabled, database wrappers will be added only to the field names containing spaces and service field names.

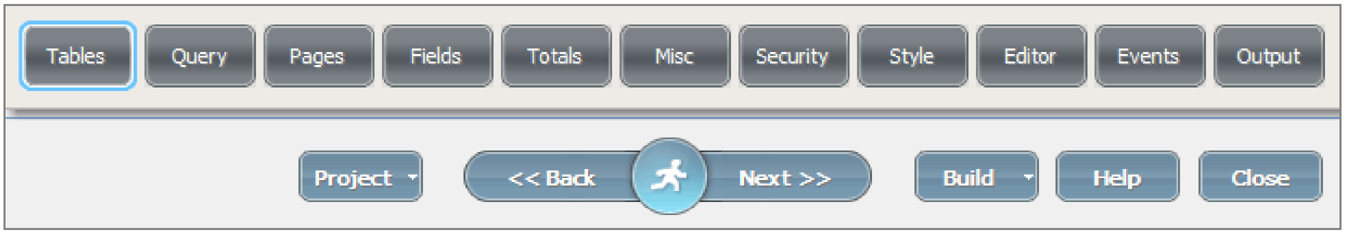
After you enabled **Lock pages modified in Visual Editor automatically** option, all pages that you manually modify will be locked from further automatic modifications. You can still modify locked pages manually.

Revision history limit defines the number of page revisions to be saved in Visual Editor.

Use **Format HTML code** option to auto-format HTML code of pages in Visual Editor.

2.2 Navigation bar

Navigation bar provides an easy way to navigate between PHPRunner pages and manage your projects. Navigation bar is located at the bottom of PHPRunner main window and available all the time you work with the project.



Button	Description
Project	Open/Save projects. Click the arrow to view all options. More info about working with projects.
<< Back	Jump to the previous page.
🏃	Quick jump to another page. Select a page from popup navigation bar.
Next >>	Jump to the next page.
Build	Build the project. You can select between two options: Build and proceed to 'Finished' screen and Build and stay on the same page. Click the arrow to change the selected option.
Help	Open PHPRunner manual.

Close:	Exit PHPRunner.

2.3 Templates

2.3.1 What are templates

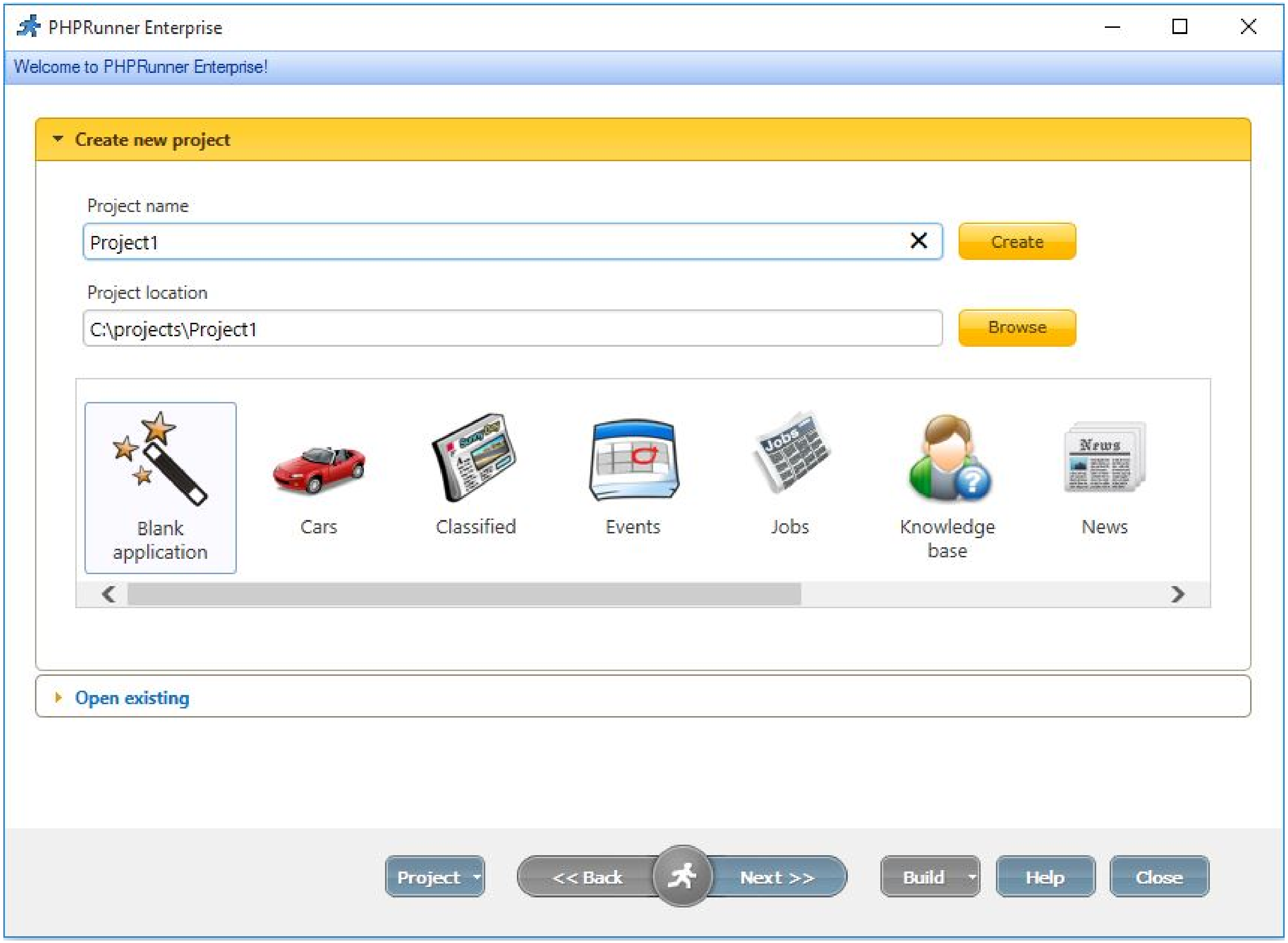
Upgrading templates.

Template helps you to create a themed web site easily and quickly.

It consists of PHPRunner project file, pre-built pages, and a script for creating tables in MySQL database.

Template is comfortable to work with, since a user doesn't need to take care about tables and database structure or about placing content on the web site pages. All you have to do is to choose template you need and generate the project.

To create a project from the template, on the first welcome page in PHPRunner select the template you want to use and click **Next**. On database connection point PHPRunner to the database where template tables to be created. PHPRunner can create a new database named after template name or use any existing database to create database tables.



PHPRunner Enterprise

Welcome to PHPRunner Enterprise!

Create new project

Project name: Project1

Project location: C:\projects\Project1

Currently, the following templates are available for free:

- Cars
- Classified
- Events
- Jobs
- Knowledge Base
- News
- Paypal
- Real Estate
- Sporting
- Vacation

Also, the paid templates are available:

MassMailer

DocManager

EmailReader

Survey

Shopping Cart

Calendar

Members

Invoice

Quiz Forum

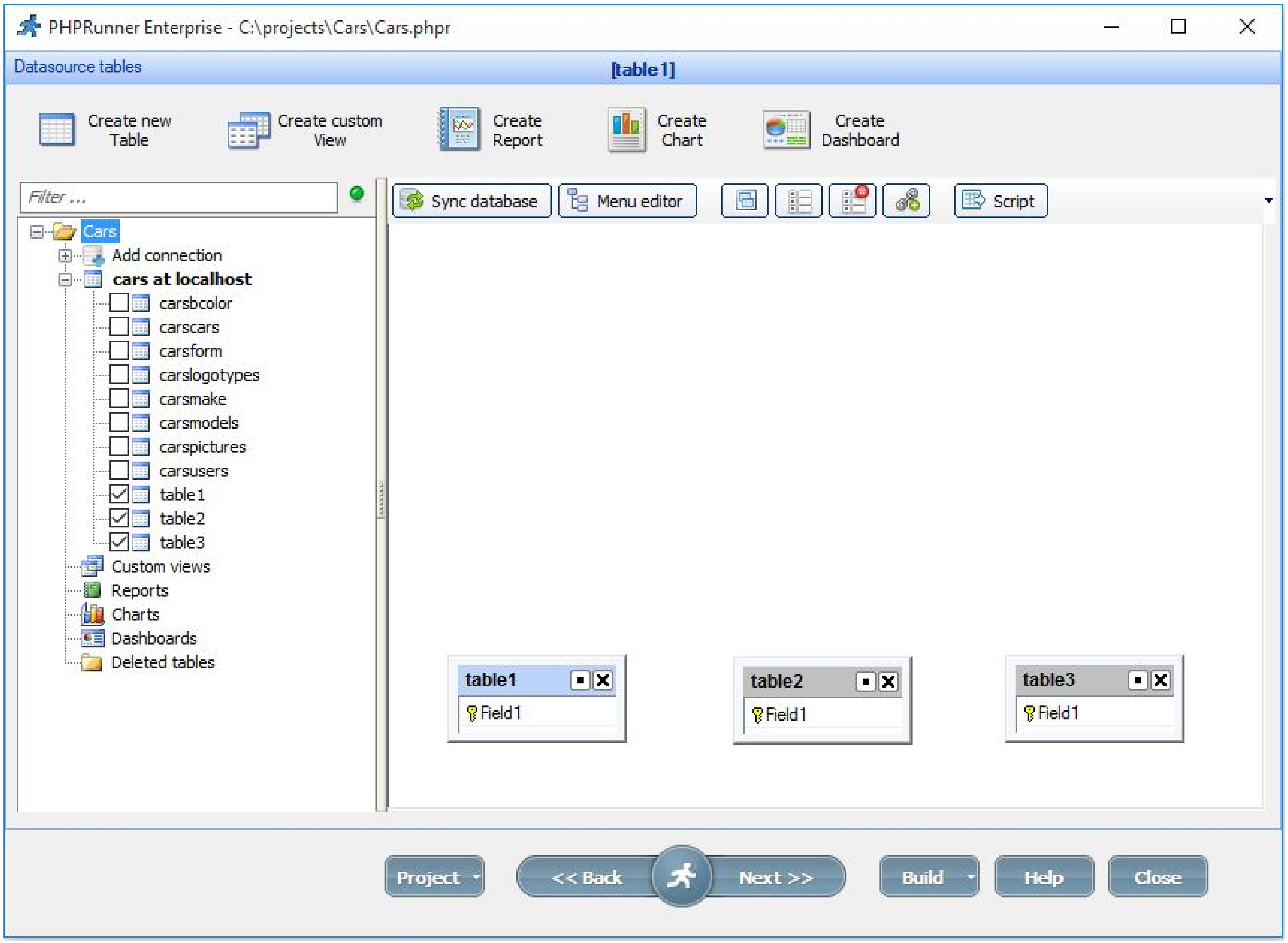
You can create new project using two templates or add a template to existing project. For more information, see Working with projects.

Upgrading templates

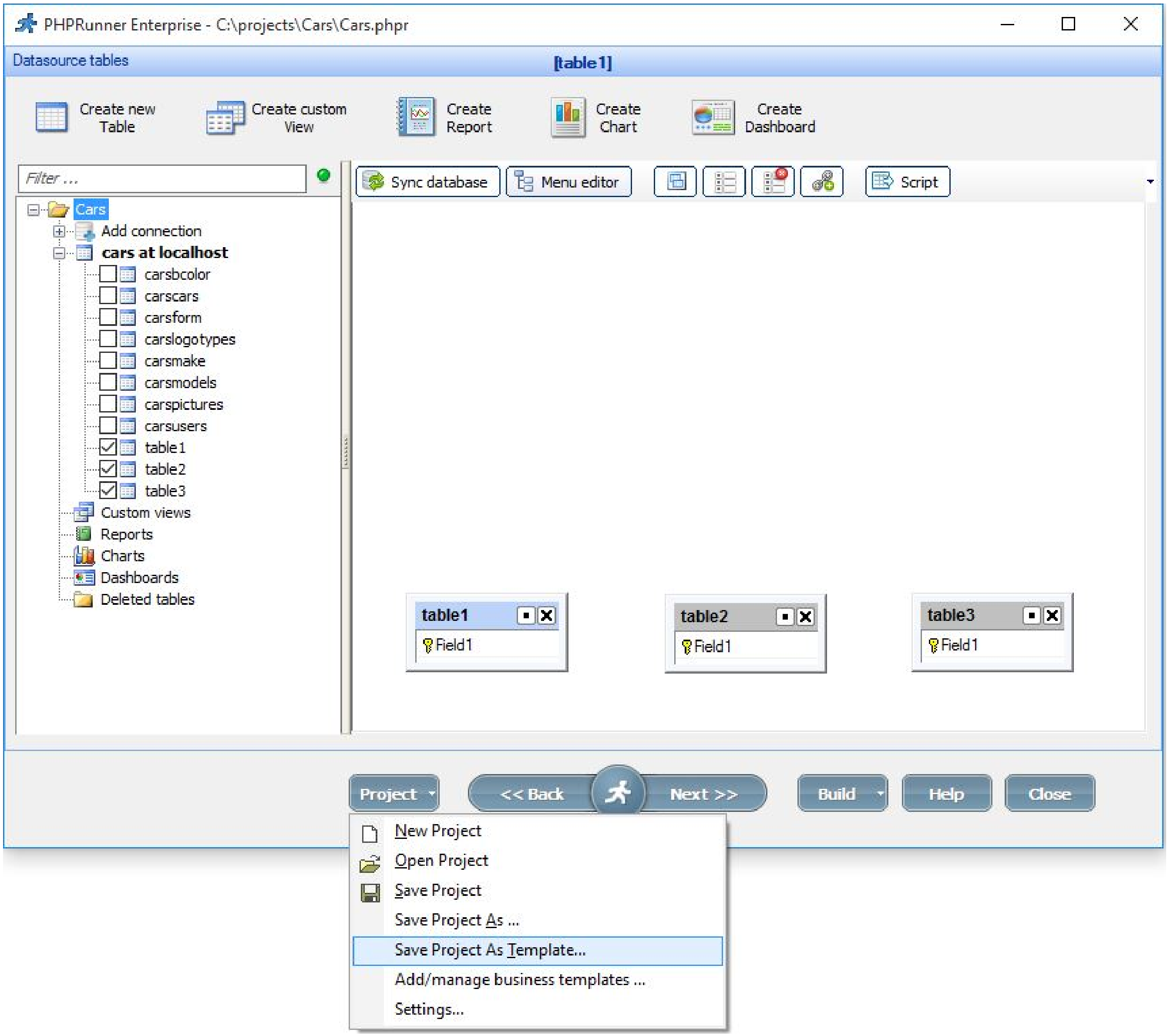
To upgrade template, remove template tables from the project ('Datasource tables' screen) and then add template to the project again.

To avoid replacing template tables with existing all PHPRunner business templates (tables and files) have the unique prefix. For example, all tables of the project 'Cars' have prefix 'cars'.

1. Remove template tables from the project



2. Add business template back to the project



2.3.2 Cars

This template is designed to build New/Used cars listings.

Administrator (**admin/admin**) has full access to all tables.

Guest users can search/view cars listings and send a quote request to a dealer/car owner.

[View live demo]

This template uses the following tables:

- carscars - main table that holds all cars listings
- carsmake - lookup table with car makes
- carsmodels - lookup table with car models
- carsusers - login table
- carsform - table to store quote requests
- carspictures - cars images
- carsbcolors - lookup table with cars body colors

2.3.3 Classified ads

This template is designed to build a classified ads website.

Administrator (**admin/admin**) has full access to all tables.

Registered users can add and edit their own ads.

Guest users can search/view ads, contact ad author, and tell a friend about this ad.

[View live demo]

This template uses the following tables:

- clmain - main table that holds all cars listings
- clcategory - lookup table with list of categories
- clsubcategory - lookup table with list of subcategories
- clusers - login table
- clreply - table to store "contact the author" requests
- clreplyfriend - table to store "tell a friend" emails

2.3.4 Events

Events

Subject :	Mike's 27th B
Date :	2/25/2008
Time :	4:00 PM - 10:
Location :	Mike's House 1111 Garlanc Plymouth, MN

This template is designed to build events listings.

Administrator (**admin/admin**) has full access to all tables.

Guest users can search/view events and tell friends about this event.

View live demo

This template uses the following tables:

- evevents - main table that holds events
- evcategories - lookup table with list of categories
- evusers - login table
- evtellafriend - table to store "tell a friend" emails

2.3.5 Jobs

Jobs

Job	Location
Financial Analyst	Minneapolis, MN 55
Project Manager	Bloomington, MN 55
Programmer	Washington, DC 98
Engineering	Washington, DC 65

This template is designed to build "company jobs available" website.

Administrator (**admin/admin**) has full access to all tables.

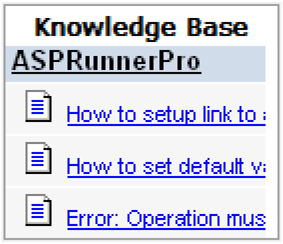
Guest users can search/view jobs and contact the company in regards to a specific open position.

View live demo

This template uses the following tables:

- jobsjobs - main table that holds all jobs listings
- jobsjobtype - lookup table with list of job types (categories)
- jobsusers - login table
- jobsstate - lookup table with the list of US states

2.3.6 Knowledge base



This template is designed to build a knowledge base website.

Administrator (**admin/admin**) has full access to all tables.

Registered user can add comments to knowledge base articles and edit his own comments.

Guest users can search/view knowledge base articles.

View live demo

This template uses the following tables:

- kbarticles - main table that holds knowledge base articles
- kbcategories - knowledge base categories
- kbusers - login table
- kbcoments - article comments

2.3.7 News

This template is designed to build a news website.

Administrator (**admin/admin**) has full access to all tables.

Guest users can search/view news.

View live demo

This template uses the following tables:

- newsmain - main table that holds news articles
- newscategory - news categories
- newssubcategory - news subcategories
- newsusers - login table

2.3.8 Paypal

This template is designed to build a simple e-commerce website integrated with Paypal shopping cart.

Administrator (**admin/admin**) has full access to all tables.

Guest users can search/view items, add items to shopping cart, and pay for items using Paypal.

View live demo

This template uses the following tables:

- ppmain - main table that holds items description, price, images etc
- ppcategory - item categories
- ppsubcategory - item subcategories
- ppusers - login table
- pppaypal_info - paypal account info

2.3.9 Real estate

This template is designed to build a real estate listings website.

Administrator (**admin/admin**) has full access to all tables.

Guest users can search/view property listings.

View live demo

This template uses the following tables:

- retblresults - main table that holds property description, price, images etc
- reusers - login table
- retblbathrooms - lookup table (number of bathrooms)
- retblbedrooms - lookup table (number of bedrooms)
- retblcooling - lookup table (AC type)
- retblgarage - lookup table (garage type)
- retblheating - lookup table (heating type)
- retblstyle - lookup table (property style)
- retbltype - lookup table (property type)

2.3.10 Sporting

Sporting

Date	Game
04.01.2007	Minnesota Wi
10.02.2007	Buffalo Sabre
01.03.2007	Minnesota Wi
10.04.2007	Minnesota Wi

This template is designed to build and maintain sport events listings.

Administrator (**admin/admin**) has full access to all tables. Admin can enter schedule, standings and game results.

Guest users can search/view team standings and schedule.

View live demo

This template uses the following tables:

- spschedule - games schedule
- spstandings - teams standings
- spusers - login table

2.3.11 Vacation houses

This template is designed to build vacation house listings.

Administrator (**admin/admin**) has full access to all tables. Admin can add/edit properties, make/cancel reservations etc.

Guest users can search/view property listings and availability.

View live demo

This template uses the following tables:

- vacproperties - main table that holds vacation house listings
- vacreservations - house reservations
- vacusers - login table
- vacbedrooms - lookup table (number of bedrooms)
- vacbathrooms - lookup table (number of bathrooms)
- vacareas - lookup table (area type)
- vaclocations - lookup table (location type)
- vacpropertytype - lookup table (property type)
- vacstate - lookup table (states)

- vacstatus - lookup table (reservation status)

2.4 Connecting to the database

Quick jump

Connecting to MySQL

Connecting to PostgreSQL

Connecting to Oracle, Microsoft SQL Server, Informix, DB2

Connecting to MS Access, spreadsheet file, SQLite

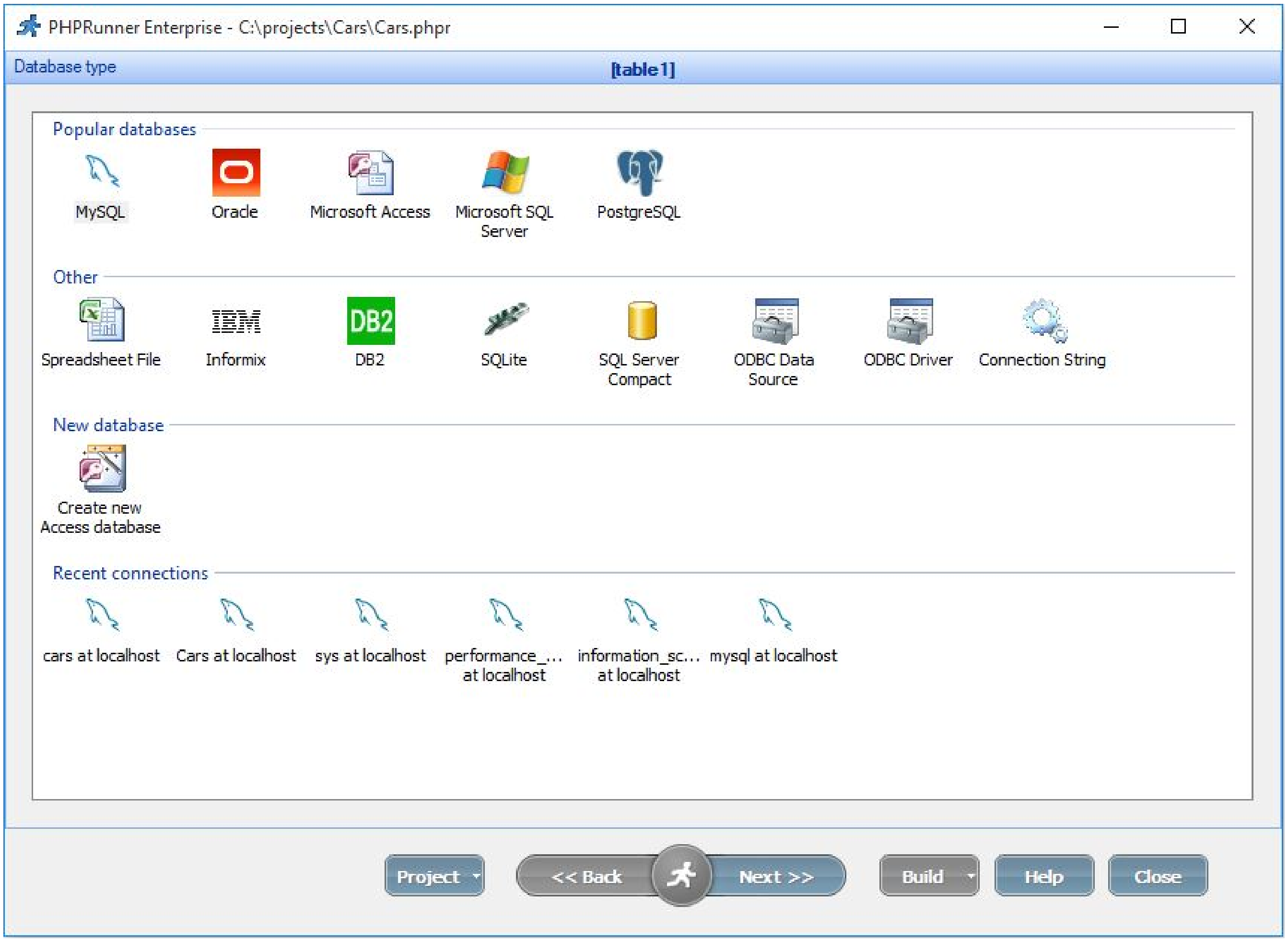
Connecting to ODBC Data Source

Connecting through ODBC driver dialog

Create a new MS Access database

Downloading drivers

PHPRunner supports the following databases: MySQL, Oracle, Microsoft SQL Server, MS Access, PostgreSQL, spreadsheet file, Informix, DB2, SQLite, any other ODBC-enabled database.

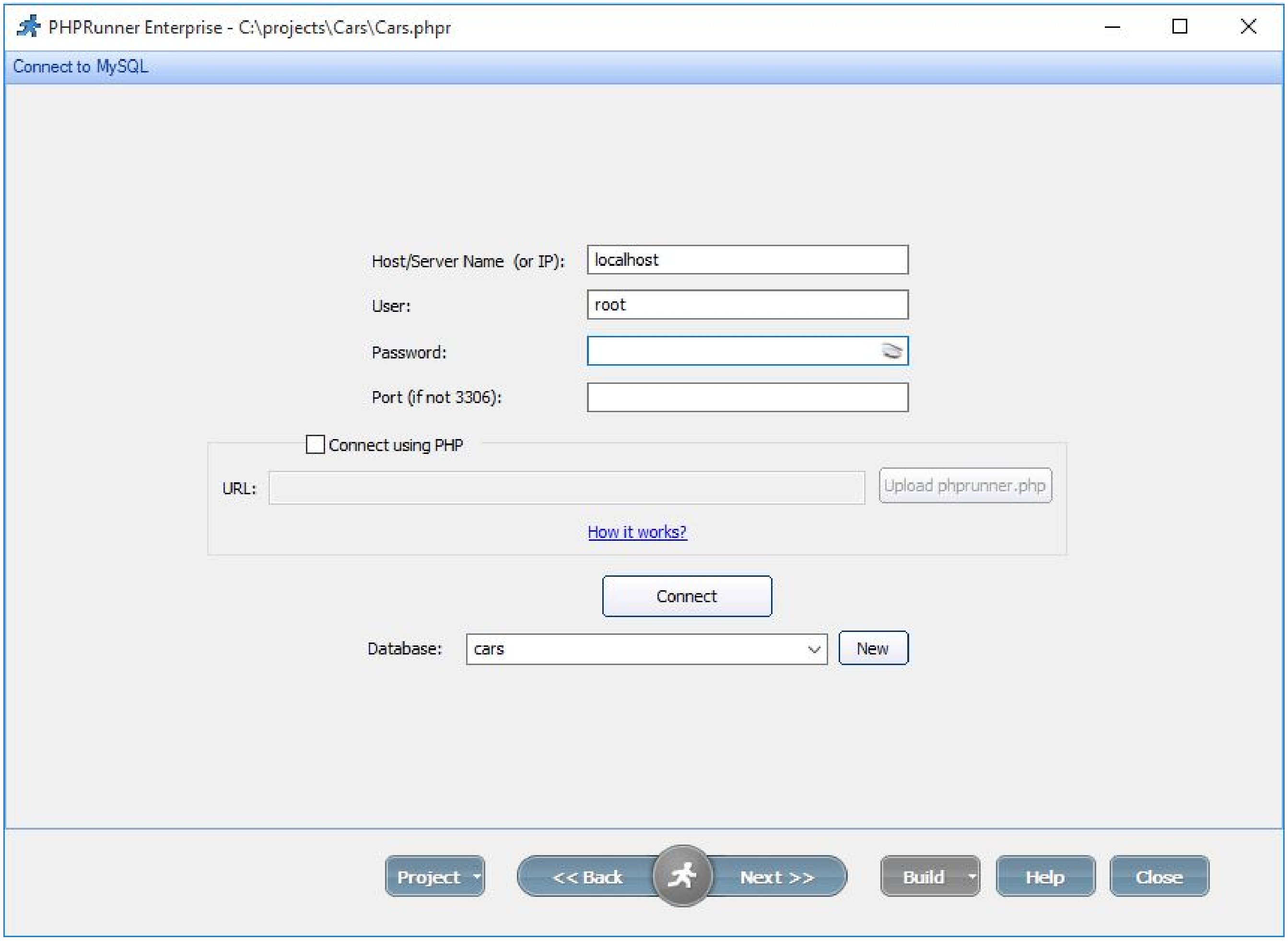


Select your database type and press **Next>>**. Depending on the selected database type you will see one of the database-specific dialog boxes shown below.

Use **Recent connections** to quickly connect to previously opened databases.

Connecting to MySQL

Type the Host/Server Name (usually **localhost**), user name, password and click **Connect**. More info about how to install local web server and MySQL server.



Host/Server Name (or IP): localhost

User: root

Password:

Port (if not 3306):

□ Connect using PHP

URL:

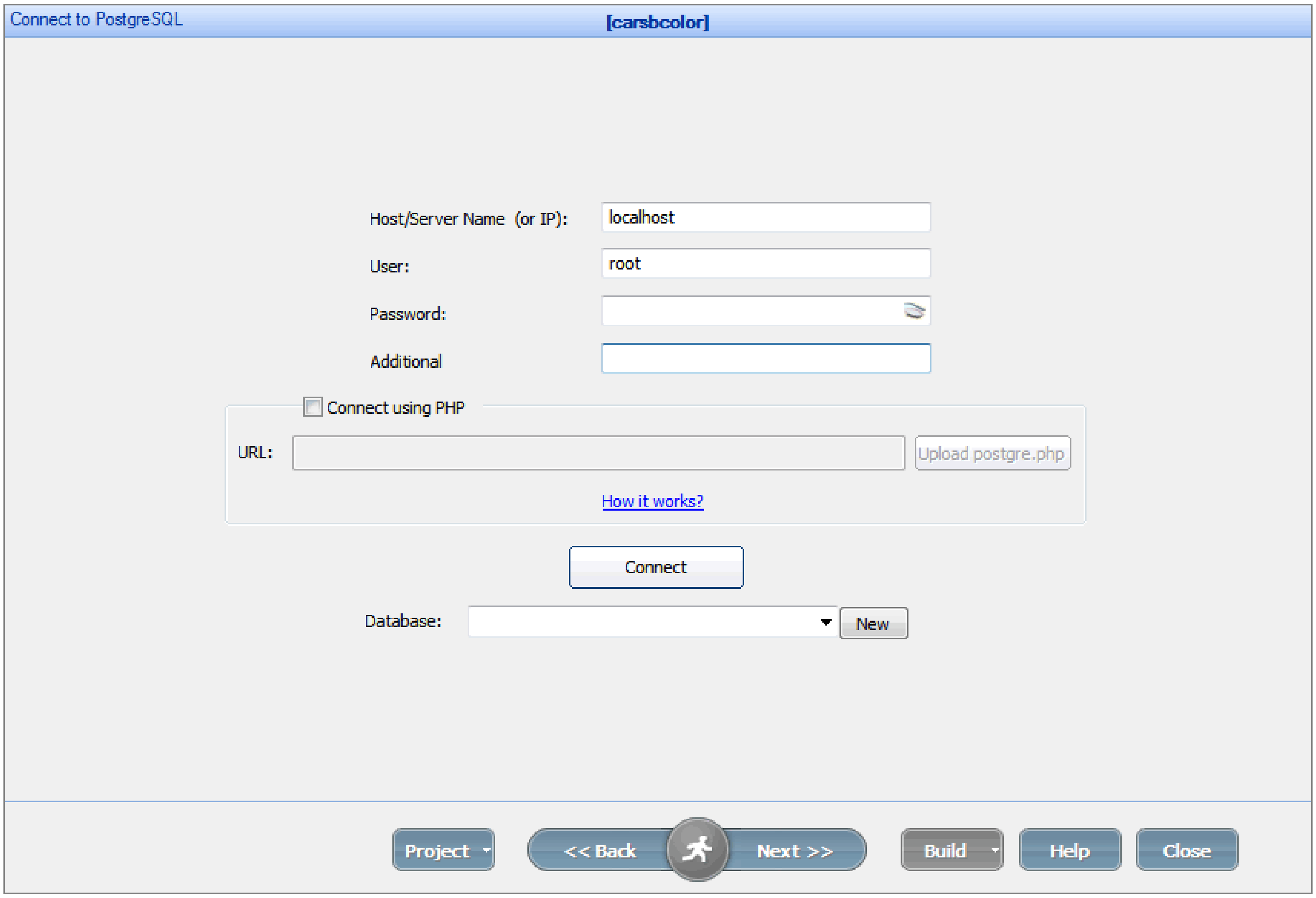
Database: cars

Select **Database** and click **Next >>**.

If your MySQL server doesn't allow remote connection, you can connect via PHP.

Connecting to PostgreSQL

Type the Host/Server Name (usually **localhost**), user name, password, set additional parameters if required (e.g. *port=5439*) and click **Connect**.



Host/Server Name (or IP): localhost

User: root

Select **Database** and click **Next >>**.

If your PostgreSQL server doesn't allow remote connection, you can connect via PHP.

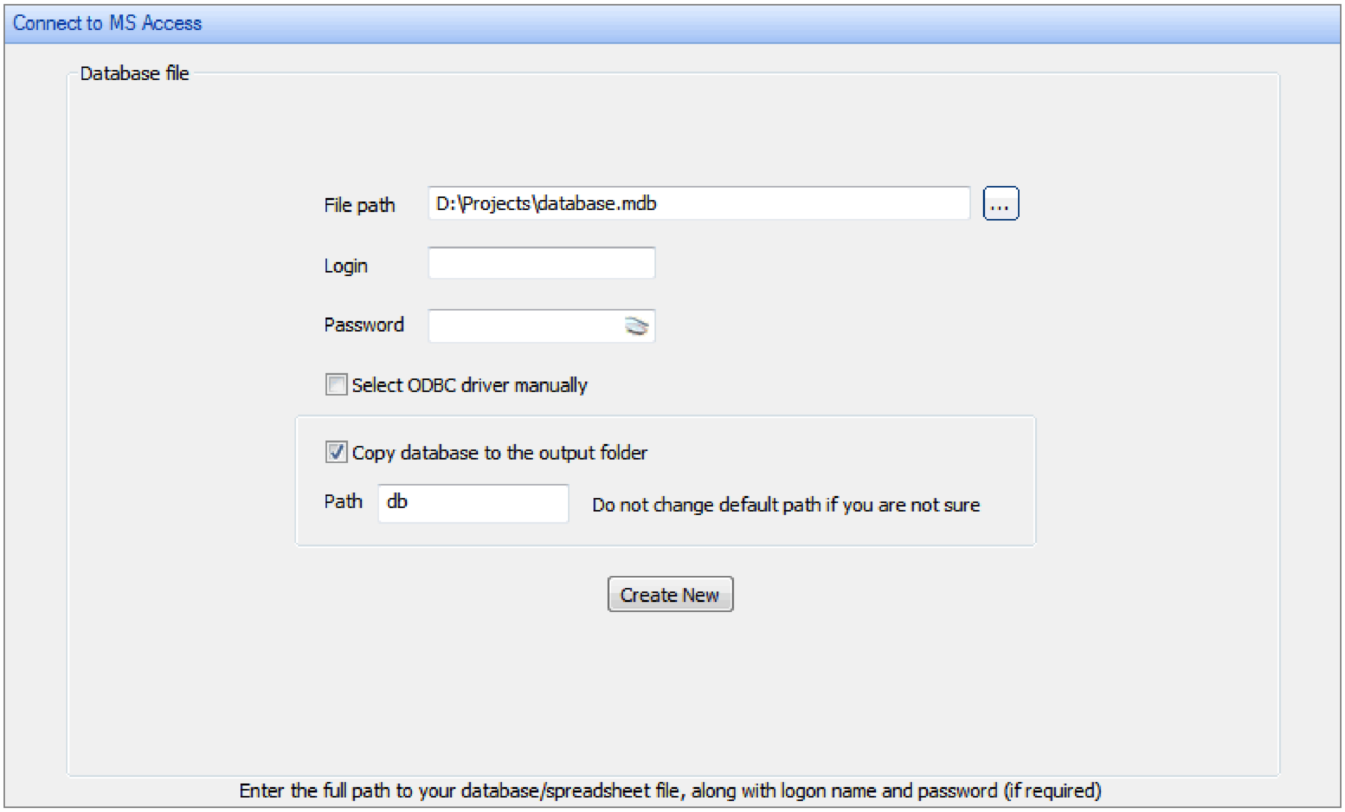
Connecting to Oracle, Microsoft SQL Server, Informix, DB2

Type the Host/Server name, Database Name, username, password and click **Connect**.

Connecting to MS Access, spreadsheet file, SQLite

Choose **Spreadsheet File** option to select Excel, FoxPro, DBase, Paradox or text database file.

Define the file path to your database/spreadsheet file, enter login name and password if required. Click **Next >>**.



File path: D:\Projects\database.mdb

Login:

Password:

☐ Select ODBC driver manually

☑ Copy database to the output folder

Path: db

Do not change default path if you are not sure

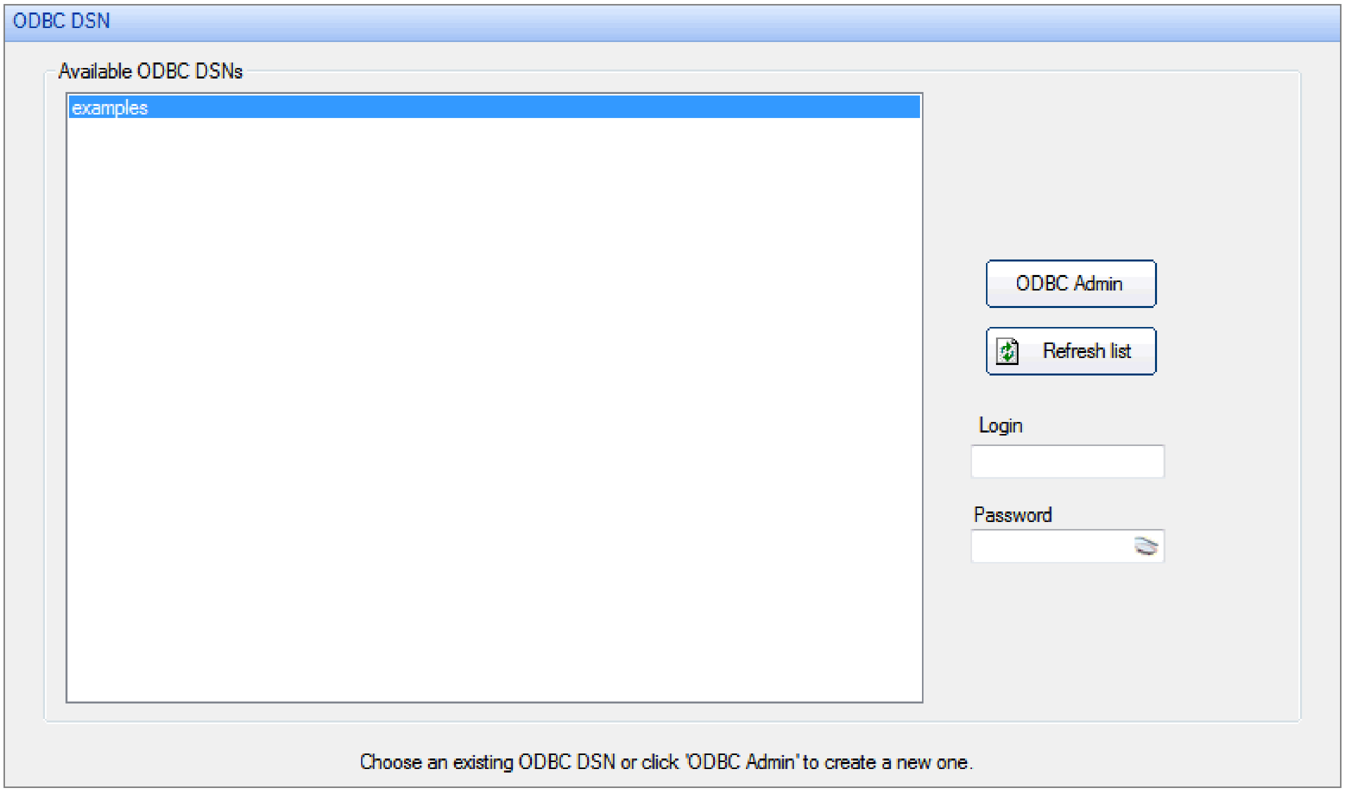
Enter the full path to your database/spreadsheet file, along with logon name and password (if required)

PHPRunner will try to find what ODBC driver to use to connect to selected database file. Select the **Select ODBC driver manually** check box if you'd like to select ODBC driver manually.
You can read more about working with MS Access connection in this topic: Connecting to MS Access database

Connecting to ODBC Data Source

If you already have an ODBC Data Source Name (DSN) associated with your database, select ODBC DSN from the list box. Enter your Login and Password if required. Press **Next>>** to continue.

To create new DSN, press the **ODBC Admin** button. In the window opened add new DSN. Then click **Refresh list**.

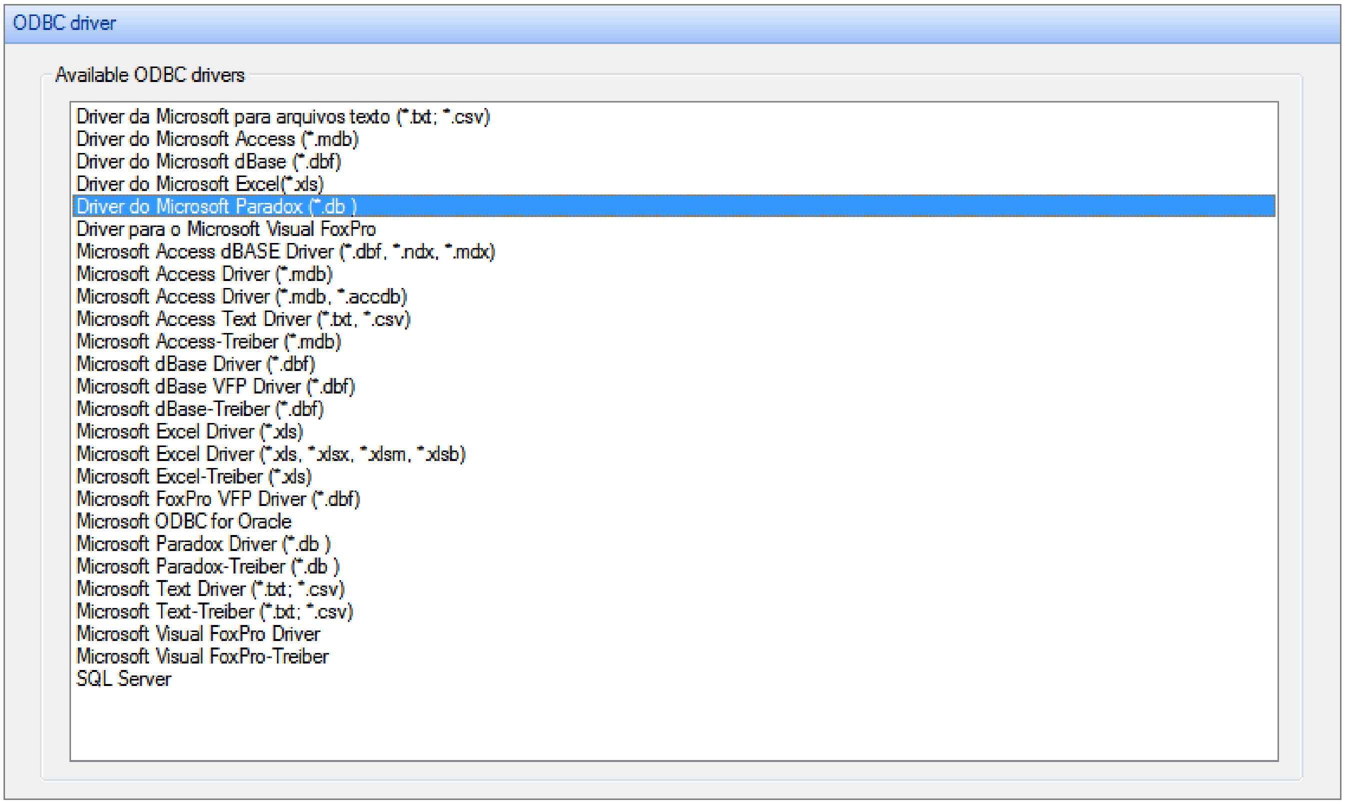


Choose an existing ODBC DSN or click 'ODBC Admin' to create a new one.

Connecting through ODBC driver dialog

Choose this option to connect directly through ODBC driver specific dialog.

Select ODBC driver from the list of available ODBC drivers and click **Next**>>.



Driver da Microsoft para arquivos texto (*.txt; *.csv)
Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)
Driver do Microsoft Paradox (*.db)
Driver para o Microsoft Visual FoxPro
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)
Microsoft Access Driver (*.mdb)
Microsoft Access Driver (*.mdb, *.accdb)
Microsoft Access Text Driver (*.txt, *.csv)
Microsoft Access-Treiber (*.mdb)
Microsoft dBase Driver (*.dbf)
Microsoft dBase VFP Driver (*.dbf)
Microsoft dBase-Treiber (*.dbf)
Microsoft Excel Driver (*.xls)
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
Microsoft Excel-Treiber (*.xls)
Microsoft FoxPro VFP Driver (*.dbf)
Microsoft ODBC for Oracle
Microsoft Paradox Driver (*.db)
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
SQL Server

Create a new MS Access database

If you do not have a database yet choose this option and click **Next**>>. PHPRunner will help you to create a new empty MS Access database.

Using **Create new table** window you can create new tables in your database.

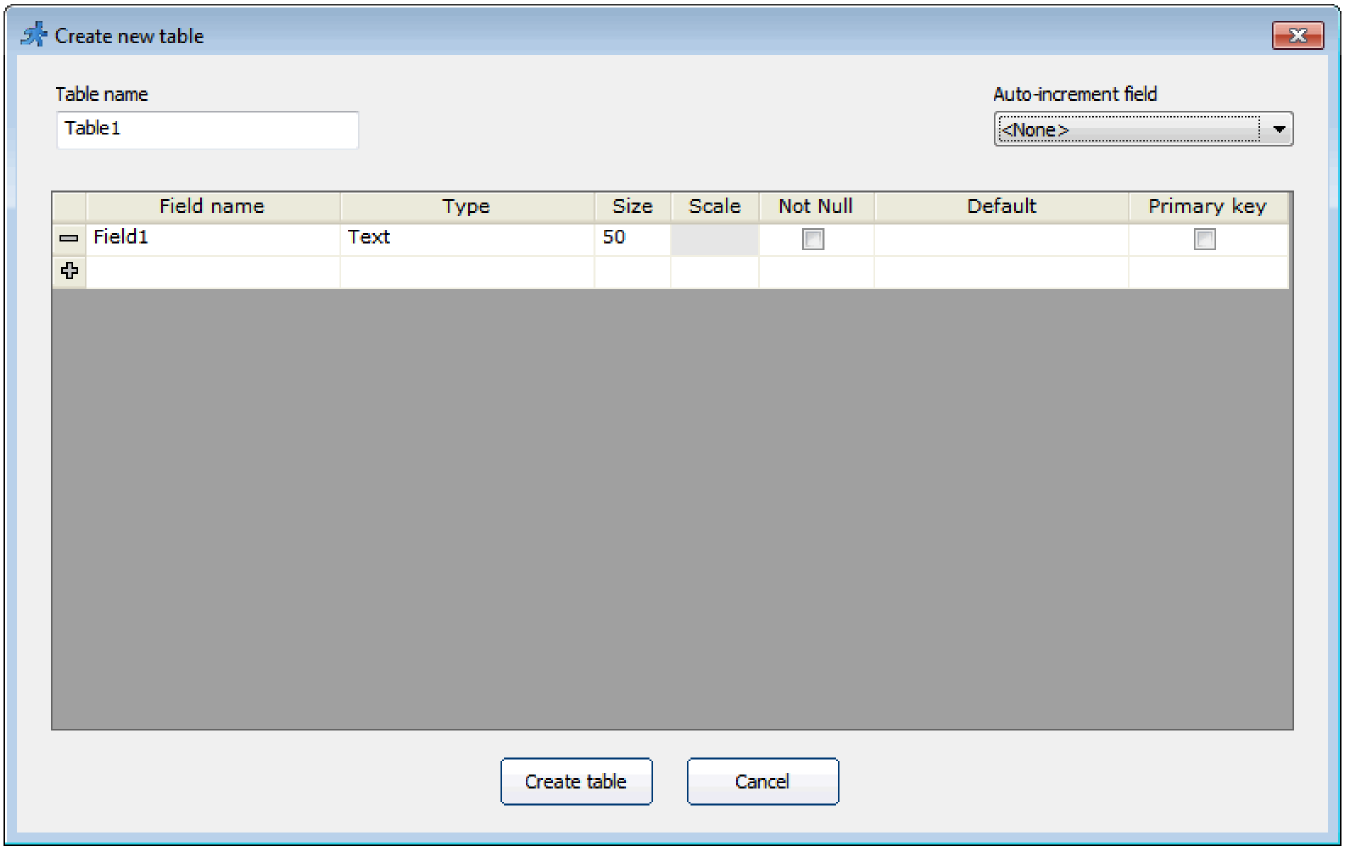


Table name: Table1

Auto-increment field: <None>

	Field name	Type	Size	Scale	Not Null	Default	Primary key
▭	Field1	Text	50		□		□
✚							

For each field type name, type, size, scale (applies to DECIMAL datatype in SQL Server, Oracle and MySQL only). Set the Primary key field. Databases created with PHPRunner will be saved to the project directory (database.mdb, database1.mdb etc.).

Note: Don't change table settings after you've built your project and added data records. All data will be lost after table modification.

After you successfully connected to database, it is time to select datasource tables.

Downloading drivers

If you use 32-bit version of PHPRunner, you need to download 32-bit drivers. If you use 64-bit version of PHPRunner, download 64-bit drivers.

32-bit drivers:

- **Oracle** - Download link (requires Oracle account).
- **Informix** - Download link (requires IBM account).
- **DB2** - Download link.

64-bit drivers:

- **Oracle** - Download link (requires Oracle account).
- **Informix** - Download link (requires IBM account).
- **DB2**:
  - version 9.7 - Download link;
  - version 10.1 - Download link;
  - version 10.5 - Download link.

2.5 Datasource tables

Quick jump

Create custom view

Create report

Create chart

Create dashboard

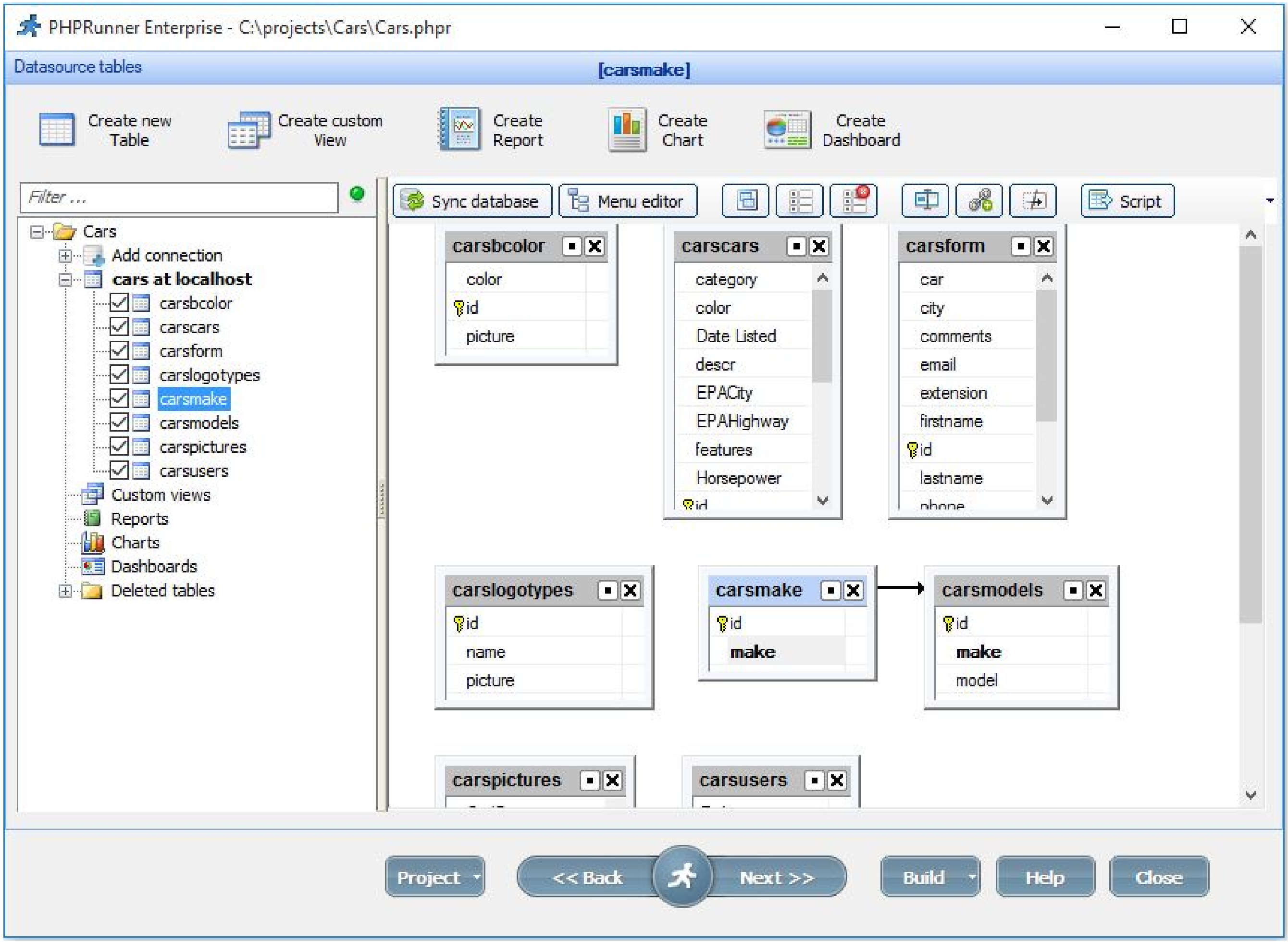
Synchronize database

Renamed/deleted tables

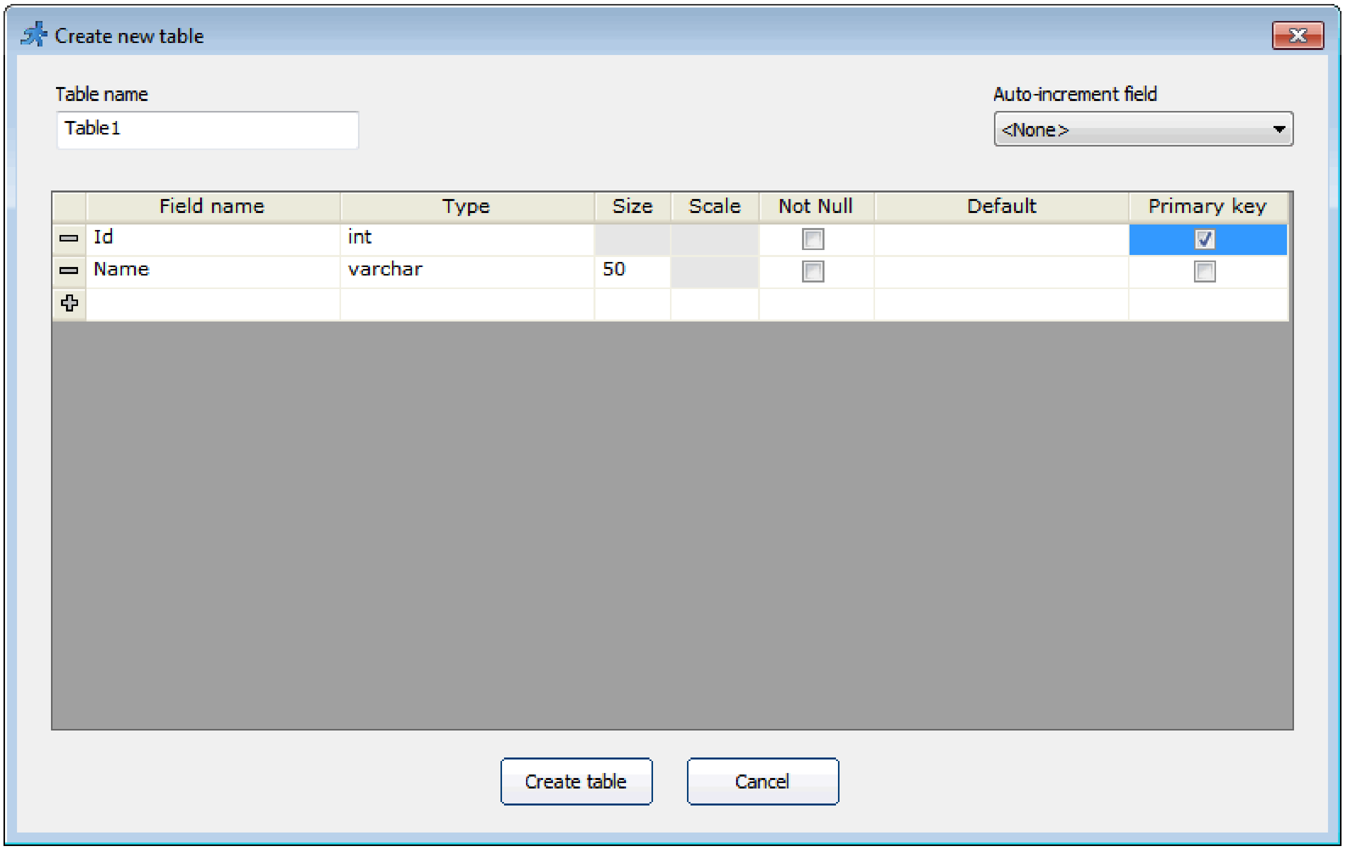
Multiple database connections

Setting master-details relationship between tables

After you successfully connected to database, select all datasource tables you'd like to build PHP code for. After that highlight one of selected tables and step through other screens in PHPRunner adjusting settings for selected table. You can always see name of currently selected table in the middle of blue info pane. To switch between selected tables use **Table list** pane on the left.



Click **Create new Table** to add new table to the existing database. Then for each field type name, type, size, scale (applies to DECIMAL datatype in SQL Server, Oracle and MySQL only). Set the Primary key field. Click **Create table**. Also you can add new custom view, report, chart or dashboard to your project.



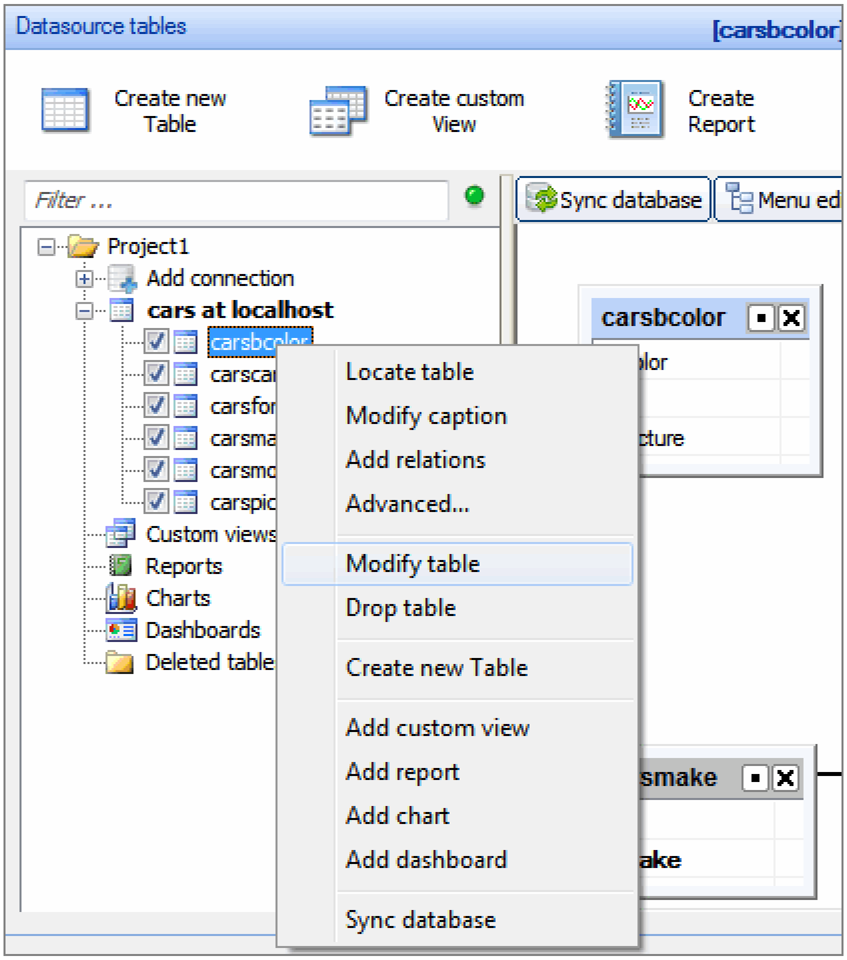
Create new table

Table name: Table1

Auto-increment field: <None>

	Field name	Type	Size	Scale	Not Null	Default	Primary key
−	Id	int			□		☑
−	Name	varchar	50		□		□
+							

To edit or delete tables right-click the table name in the list.

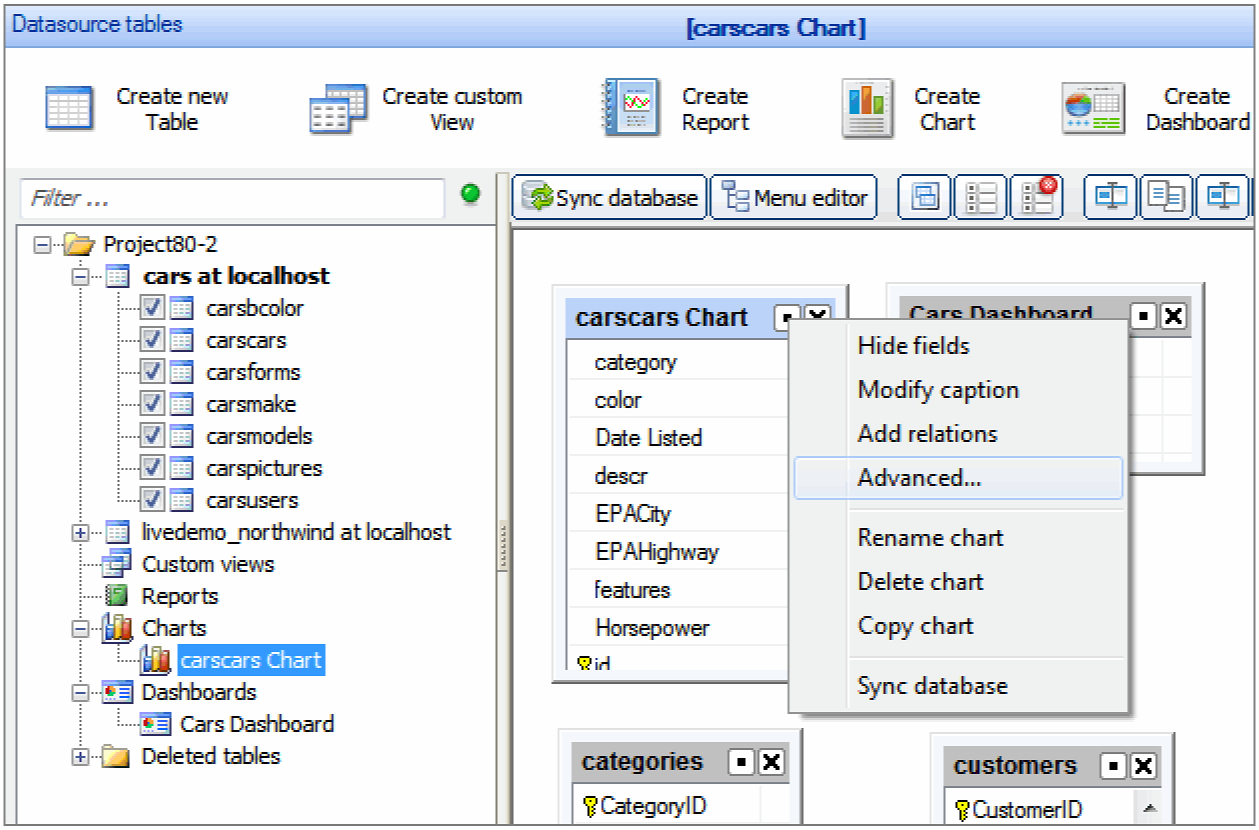


Toolbar description:

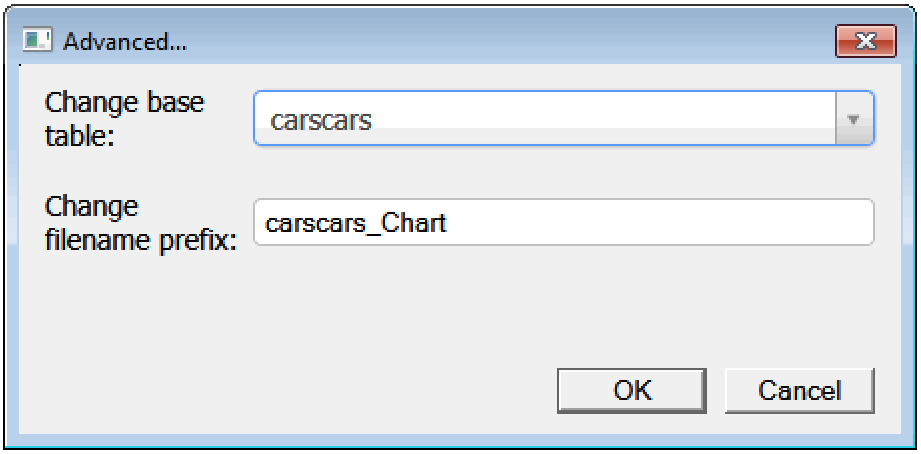
Button	Description
Sync database	Synchronizes database.
Menu editor	Opens Menu Builder.
	Arranges tables alphabetically.
	Shows all fields in all tables.
	Hides all fields in all tables.
	Opens Label Editor where you can change table captions.

🔗	Opens Table link properties window where you can add new table relations.
⊞	Shows/Hides fields for currently selected table.
Script	Opens Create SQL script window where you can create SQL script for tables/data transfer to another server.
🔍	Searches within table names.

To open context menu of a table/view/chart/report/dashboard, click ⊡ near its name or right-click it.



Change caption option allows to edit caption and field labels. Click **Advanced** option to change a base table (is available only for view/chart/report) or filename prefix.



Change base table: carscars

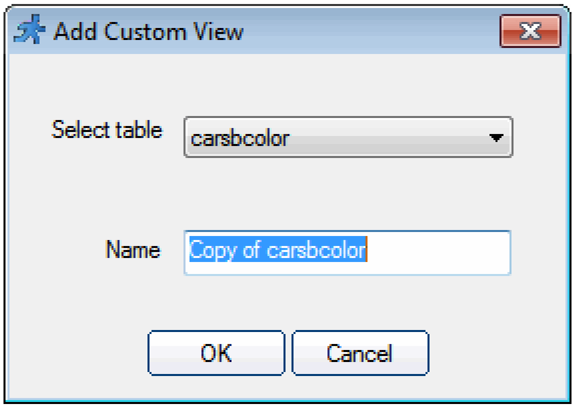
Change filename prefix: carscars_Chart

Since view, chart and report may include fields from several tables, a base table for them is the table where data is added, edited and deleted. Key columns are also selected from the base table on **Choose pages** screen.

By default, prefix for each generated file name is the table name. E.g. if the table name is *OrderDetails*, the file name of List page will be *OrderDetails_list.php*. If you change the prefix to *details*, the file name will be *details_list.php.*

Create custom View

Click **Create custom View** to create an additional view of the same table. This feature is useful when you like to present several views of the same data.



Select table: carsbcolor

Name: Copy of carsbcolor

When you create new custom view, all current settings of the table, on the basis of which view is created, are copied into it except for visual templates and events. You can create a copy of existing custom view (right-click the custom view and select **Copy**). Note that custom views are not created in the database and only exist in the project.

Example:

Table Cars shows all cars in the database. SQL query: Select * from Cars.

View **Active listings** displays active listings only. **SQL Query:** *Select * from Cars where status='active'*

View **Closed listings** displays closed listings only. **SQL Query:** *Select * from Cars where status='closed'*

SQL query can be modified later on SQL query tab in PHPRunner.

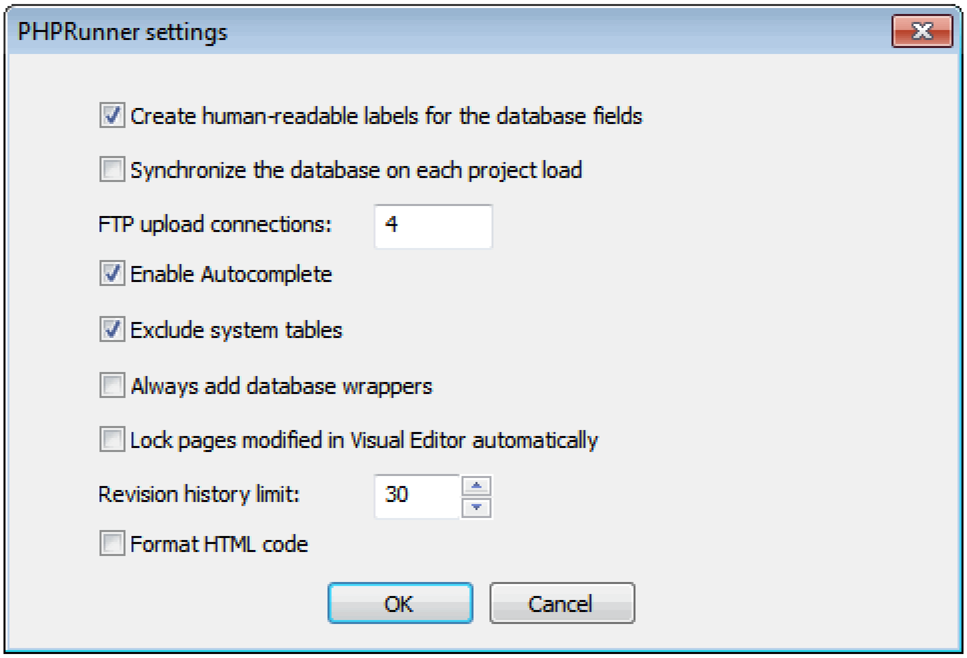
Note: when you create a custom view all table settings are copied to Custom view settings.

Synchronize database

Every time you make changes to the database in PHPRunner (e.g. you create new table or view, you add or edit table fields etc.), these changes should be implemented in the database. In other words, the structures of the database and PHPRunner project should be synchronized.

To synchronize database manually, use **Sync database** button or right-click tables tree or blank area near tables tree and select **Sync database**.

Also you can use the option to synchronize the database automatically on each project load (select **Project** -> **Settings)**. Use this option for small or local databases. In the case of remote or large databases the automatic synchronization will take some time when opening the project.



PHPRunner settings

☑ Create human-readable labels for the database fields

□ Synchronize the database on each project load

FTP upload connections: 4

☑ Enable Autocomplete

☑ Exclude system tables

□ Always add database wrappers

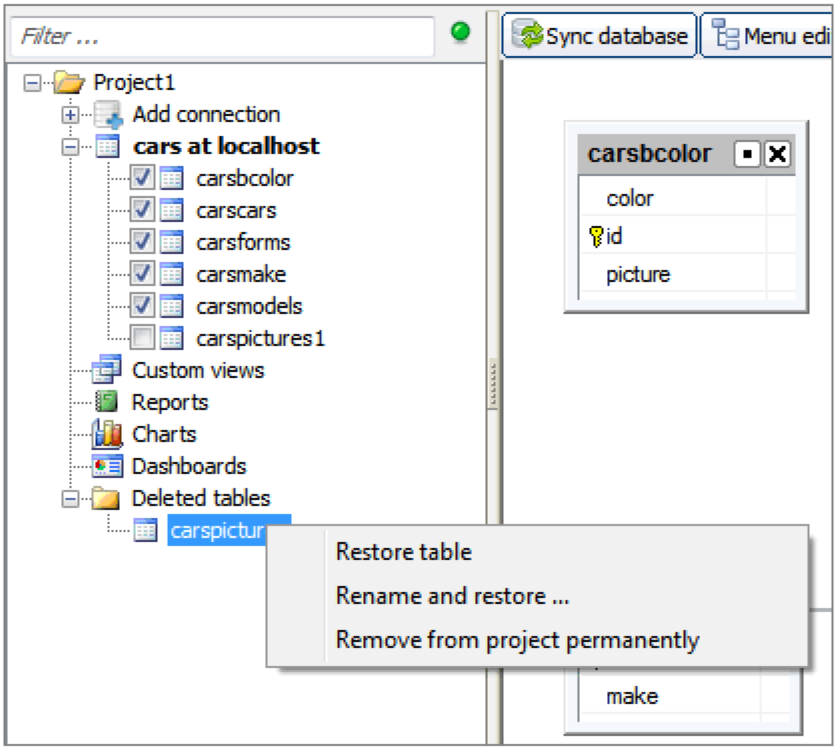
□ Lock pages modified in Visual Editor automatically

Revision history limit: 30

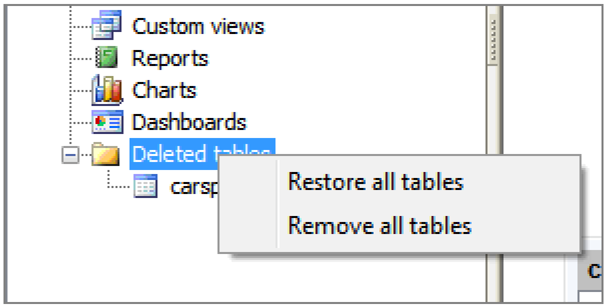
□ Format HTML code

Renamed/deleted tables

Tables that were renamed in the database (outside PHPRunner) or deleted in PHPRunner are moved to the **Deleted tables** folder. You can restore table, rename and then restore table, remove table from the project permanently.



Right-click the **Deleted tables** to get two more options: Remove all tables and Restore all tables.



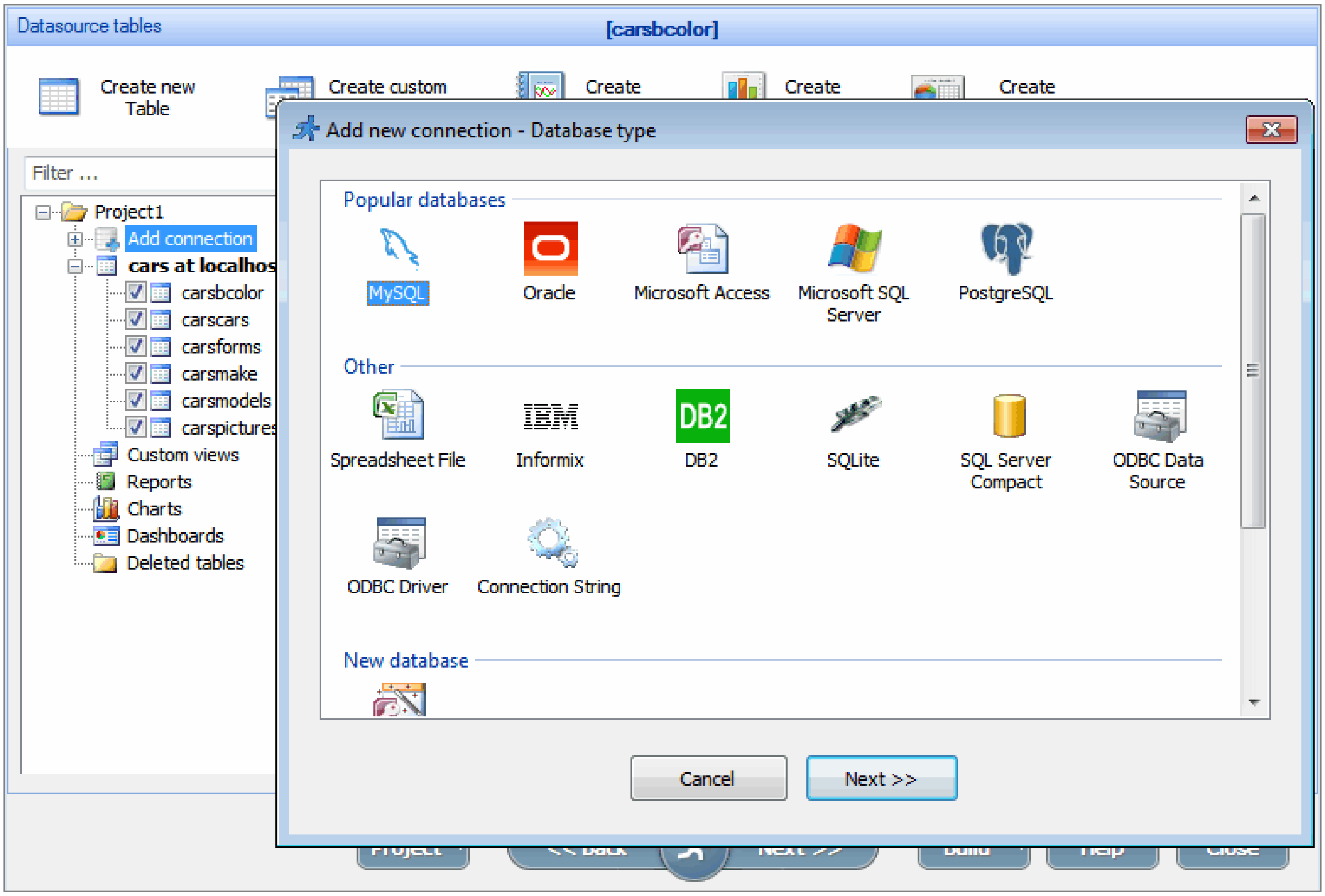
Multiple database connections

You can add multiple data sources and mix several database types like MS Access, SQL Server and MySQL in a single PHPRunner project. You can have master table in MySQL and details table in MS Access. The same applies to lookup tables.

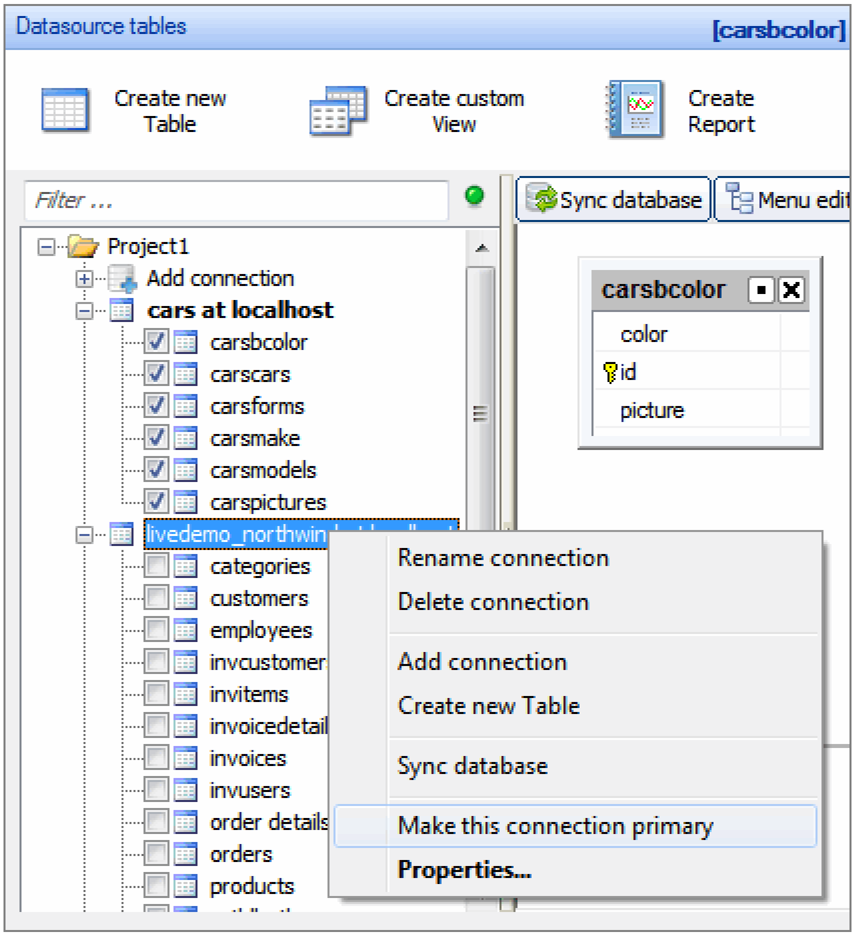
Note: Multiple database connections feature is available only in the Enterprise Edition of PHPRunner. See Editions Comparison.

To add a new database connection:

- click **Add connection**;
- select database type and connect to the database.



The first database connection is considered as "primary" one. But you can make any other connection primary by right-clicking the database name and selecting the corresponding option.



The following features work only for the primary database connection:

- Data Access Layer;
- Add template to project - template will be added only to the primary database;
- Upload to demo account - tables from the primary database will be downloaded to the demo account.

All other features work for all database connections, including **Save project as template** option.

We do not advise to add tables with the same names that belong to different databases to avoid conflicts.

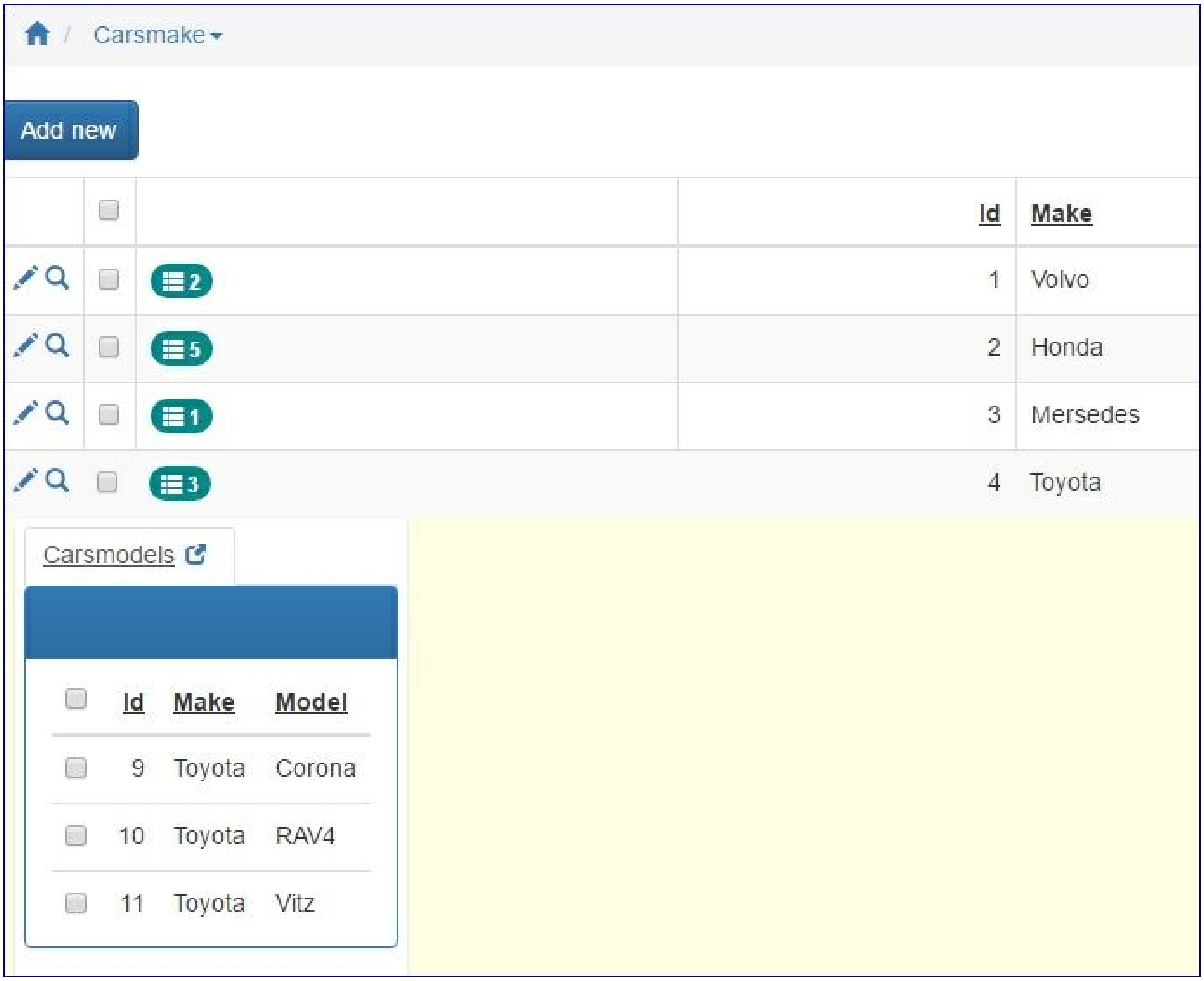
2.6 Master-details relationship between tables

Quick jump
Changing order of child tables
Print master table data with details

Charts and reports as master and details tables

Master-details relationships are commonly used in applications. An example of this type of relationship is an order with a header and line items. In PHPRunner you can join, or link, two or more tables that have at least one common field. You can Add/Edit records in the master-details tables on the same page.

Note that one master table can have multiple details tables. And you can display data from multiple details tables (nested or one master - many details) on the same page. I.e. you can display customers, orders by customer, order details per order and edit all three tables at the same time without leaving the page.



	□		Id	Make
	□	2	1	Volvo
	□	5	2	Honda
	□	1	3	Mersedes
	□	3	4	Toyota

□	Id	Make	Model
□	9	Toyota	Corona
□	10	Toyota	RAV4
□	11	Toyota	Vitz

To create master-details relationship in PHPRunner:

1. Go to the **Datasource tables** page and select both master and details tables in the list of datasource tables..

2. Drag field from one table to another. **Table link properties** dialog opens.

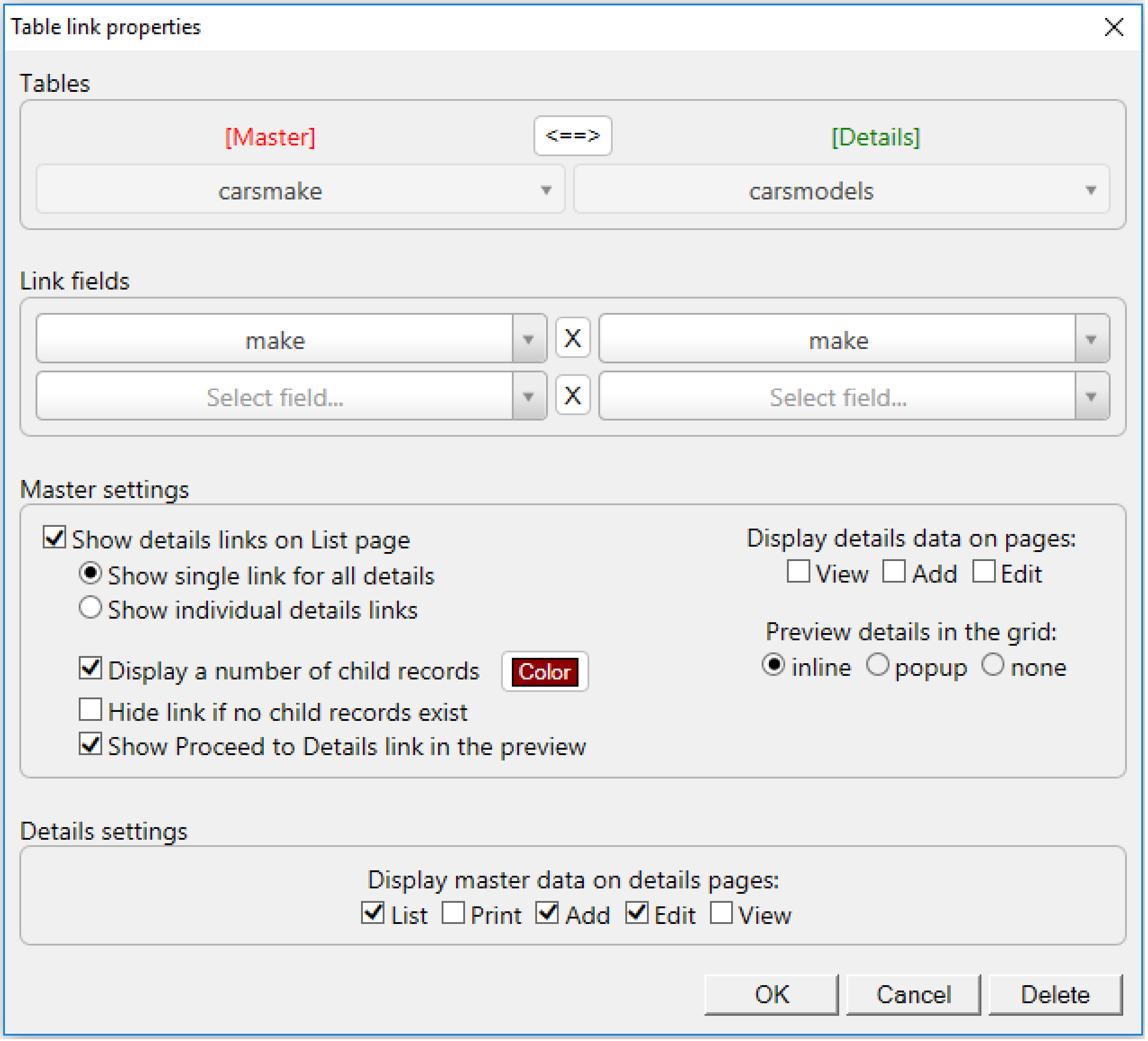


Table link properties

Tables

[Master] <==> [Details]

carsmake ▾ | carsmodels ▾

Link fields

make ▾ X make ▾

Select field... ▾ X Select field... ▾

Master settings

☑ Show details links on List page

● Show single link for all details

○ Show individual details links

☑ Display a number of child records Color

□ Hide link if no child records exist

☑ Show Proceed to Details link in the preview

Display details data on pages:

□ View □ Add □ Edit

Preview details in the grid:

● inline ○ popup ○ none

Details settings

Display master data on details pages:

☑ List □ Print ☑ Add ☑ Edit □ View

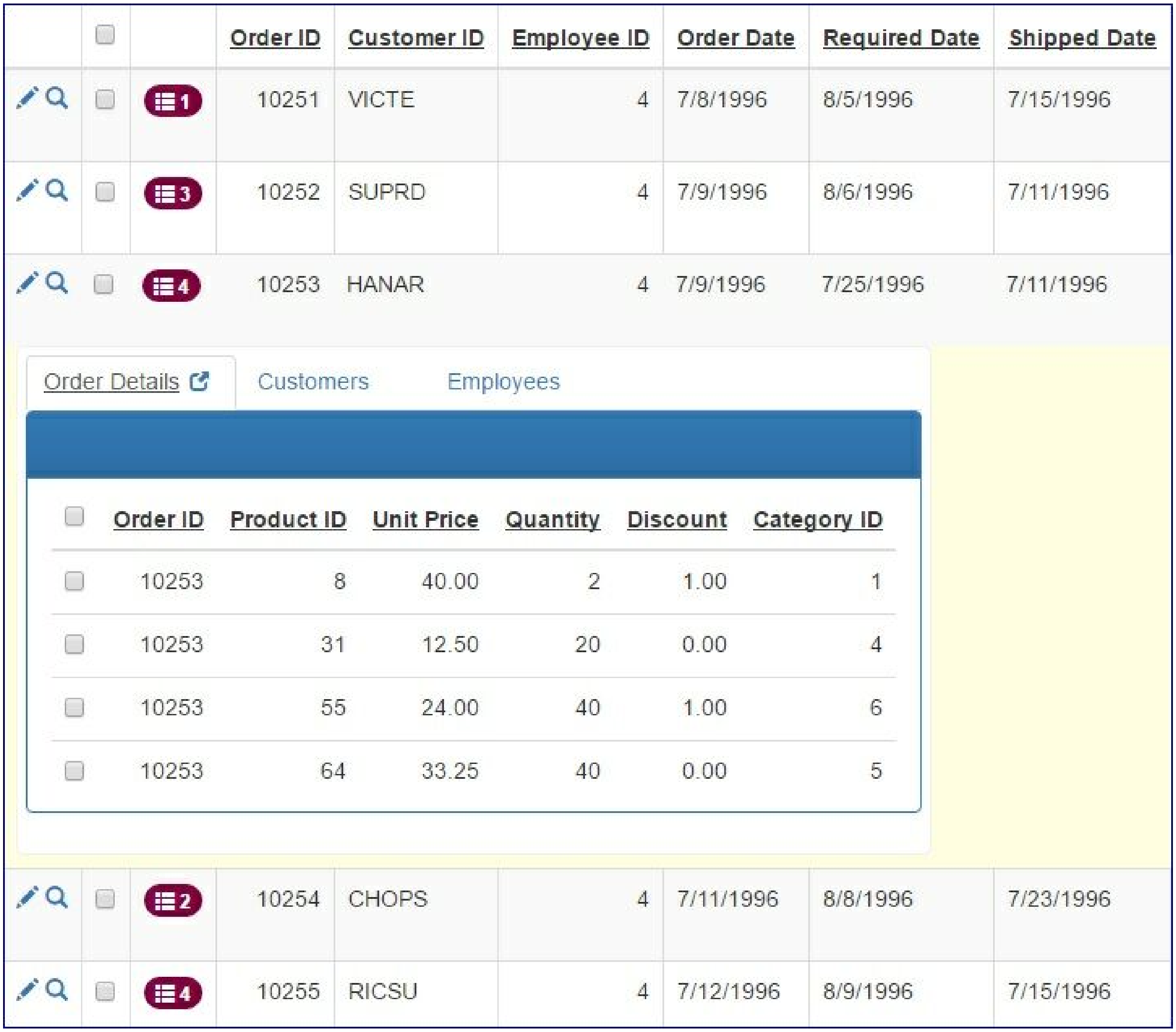
OK Cancel Delete

3. Choose what table is Master and select link fields in both tables. If you'd like to display master table data on the details page select the corresponding check box. Click **OK**.

See examples of Master and Details settings below.

Show single link for all details:

Link for details is displayed as ☰3 icon. Click it to view details tables. Use this option if master table has multiple details tables.

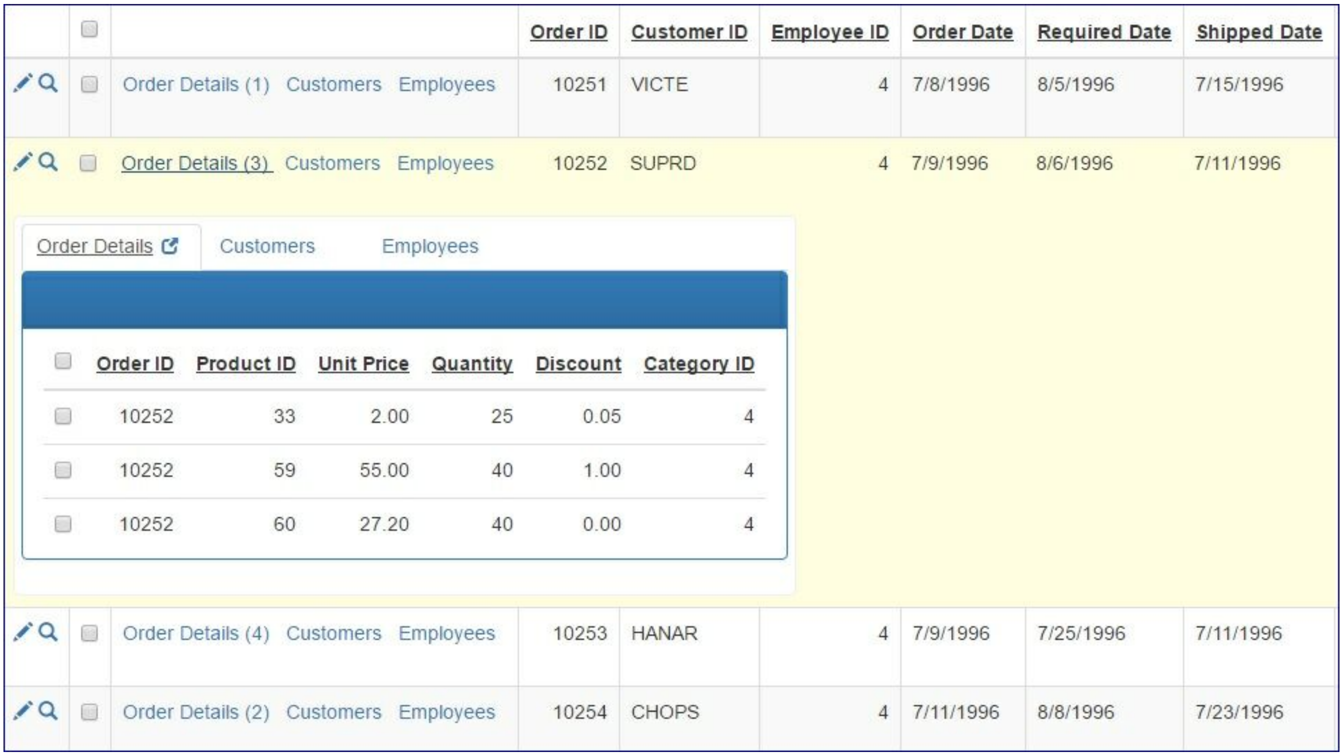


	□		Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date
	□	1	10251	VICTE	4	7/8/1996	8/5/1996	7/15/1996
	□	3	10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996
	□	4	10253	HANAR	4	7/9/1996	7/25/1996	7/11/1996

□	Order ID	Product ID	Unit Price	Quantity	Discount	Category ID
□	10253	8	40.00	2	1.00	1
□	10253	31	12.50	20	0.00	4
□	10253	55	24.00	40	1.00	6
□	10253	64	33.25	40	0.00	5

	□		Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date
	□	2	10254	CHOPS	4	7/11/1996	8/8/1996	7/23/1996
	□	4	10255	RICSU	4	7/12/1996	8/9/1996	7/15/1996

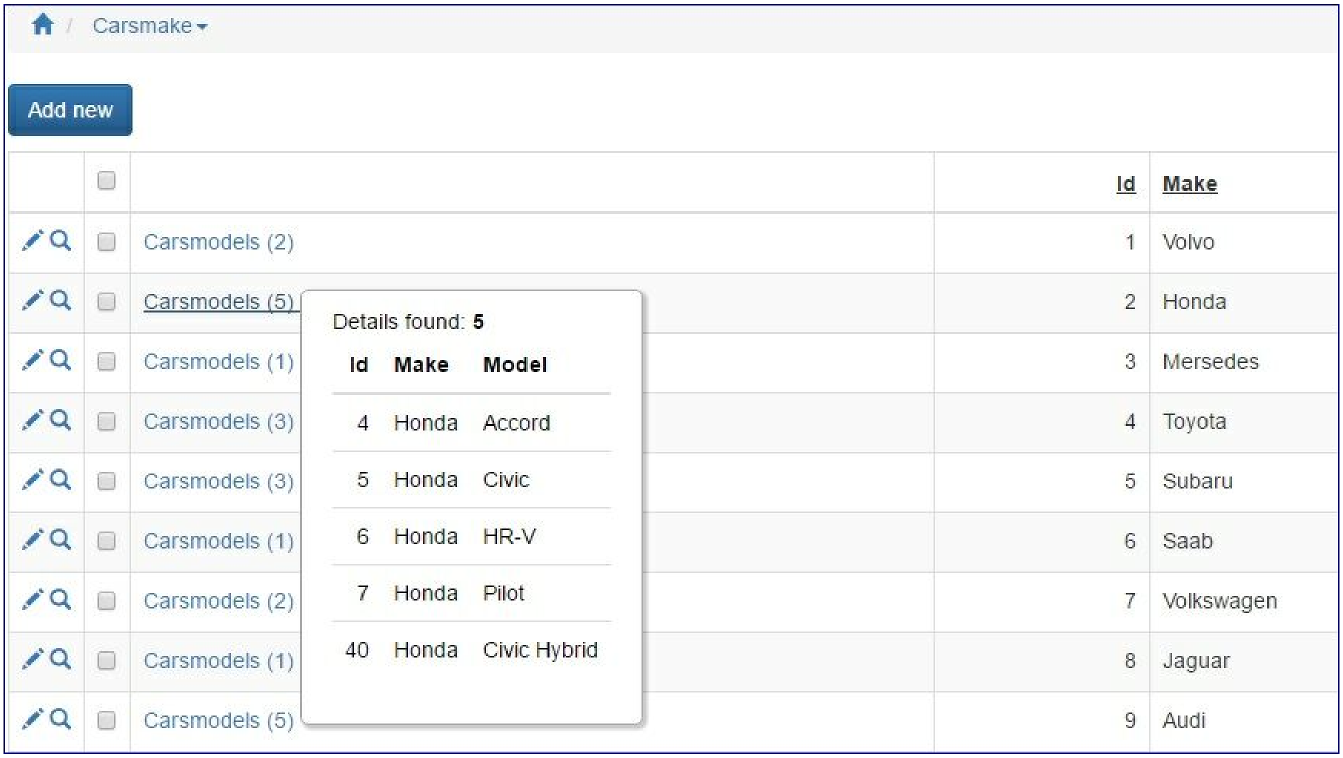
Show individual details links:



	□		Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date
✎ 🔍	□	Order Details (1) Customers Employees	10251	VICTE	4	7/8/1996	8/5/1996	7/15/1996
✎ 🔍	□	Order Details (3) Customers Employees	10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996
✎ 🔍	□	Order Details (4) Customers Employees	10253	HANAR	4	7/9/1996	7/25/1996	7/11/1996
✎ 🔍	□	Order Details (2) Customers Employees	10254	CHOPS	4	7/11/1996	8/8/1996	7/23/1996

□	Order ID	Product ID	Unit Price	Quantity	Discount	Category ID
□	10252	33	2.00	25	0.05	4
□	10252	59	55.00	40	1.00	4
□	10252	60	27.20	40	0.00	4

Preview details records in popup:



	□		Id	Make
✎ 🔍	□	Carsmodels (2)	1	Volvo
✎ 🔍	□	Carsmodels (5)	2	Honda
✎ 🔍	□	Carsmodels (1)	3	Mersedes
✎ 🔍	□	Carsmodels (3)	4	Toyota
✎ 🔍	□	Carsmodels (3)	5	Subaru
✎ 🔍	□	Carsmodels (1)	6	Saab
✎ 🔍	□	Carsmodels (2)	7	Volkswagen
✎ 🔍	□	Carsmodels (1)	8	Jaguar
✎ 🔍	□	Carsmodels (5)	9	Audi

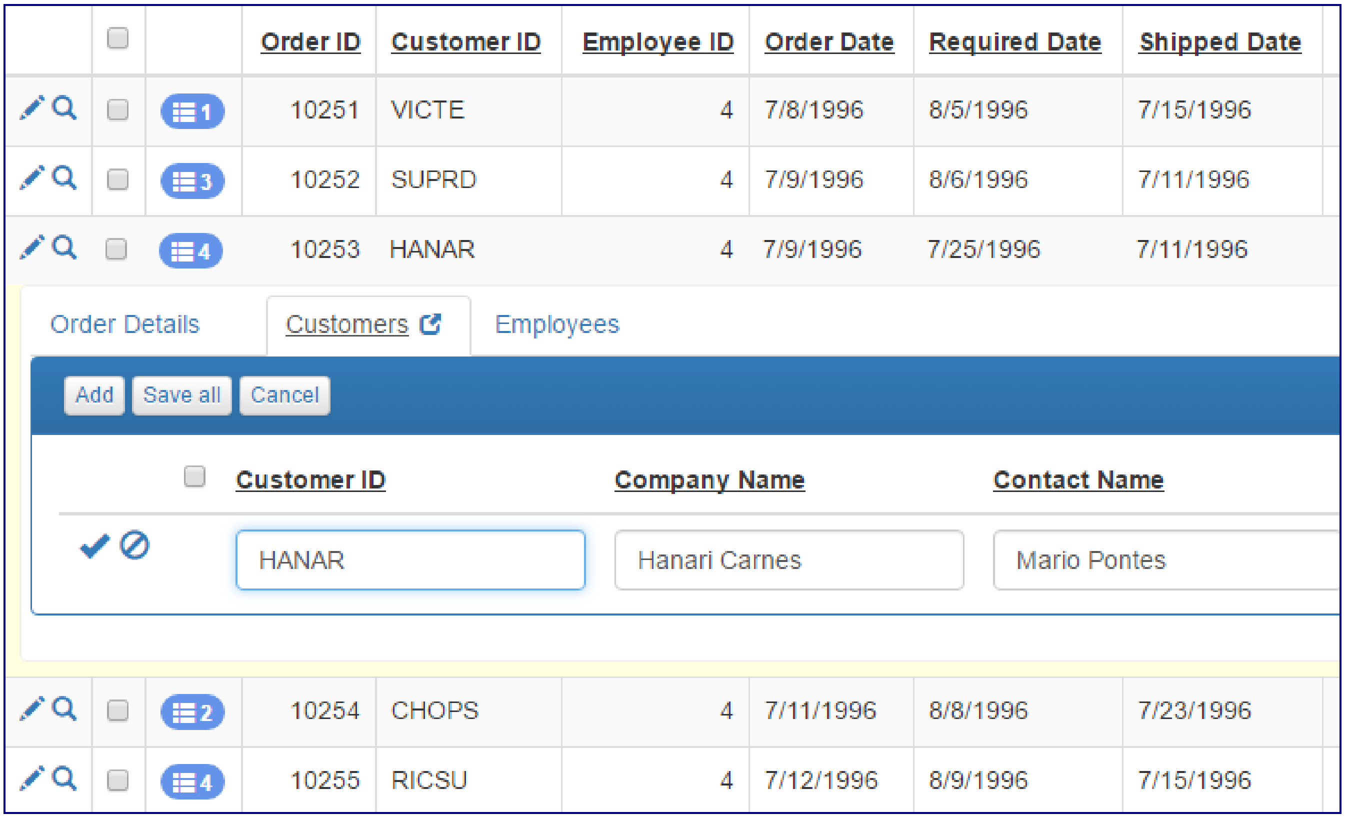
Details found: **5**

Id	Make	Model
4	Honda	Accord
5	Honda	Civic
6	Honda	HR-V
7	Honda	Pilot
40	Honda	Civic Hybrid

Note: If you select to display details records in popup, you can define the popup page appearance in **Visual Editor** -> **Details preview** page for details table.

Preview details records inline:

If you select to display details records inline, you will be able to add/edit/delete details in inline mode on master table.

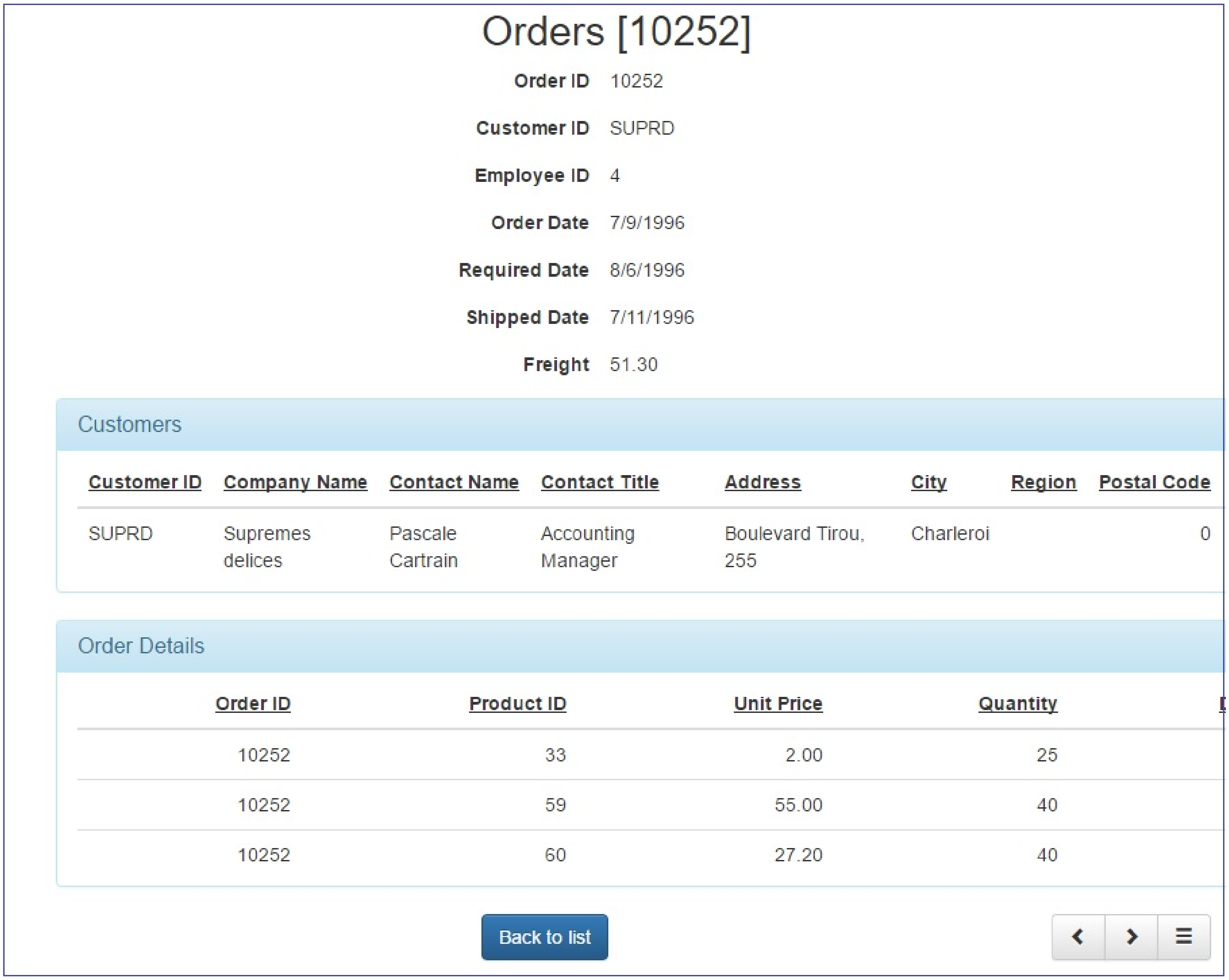


	□		Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date
	□	1	10251	VICTE	4	7/8/1996	8/5/1996	7/15/1996
	□	3	10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996
	□	4	10253	HANAR	4	7/9/1996	7/25/1996	7/11/1996

□	Customer ID	Company Name	Contact Name
✓ ⊘	HANAR	Hanari Carnes	Mario Pontes

	□		Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date
	□	2	10254	CHOPS	4	7/11/1996	8/8/1996	7/23/1996
	□	4	10255	RICSU	4	7/12/1996	8/9/1996	7/15/1996

Display details records on the View page:



Orders [10252]

Order ID	10252
Customer ID	SUPRD
Employee ID	4
Order Date	7/9/1996
Required Date	8/6/1996
Shipped Date	7/11/1996
Freight	51.30

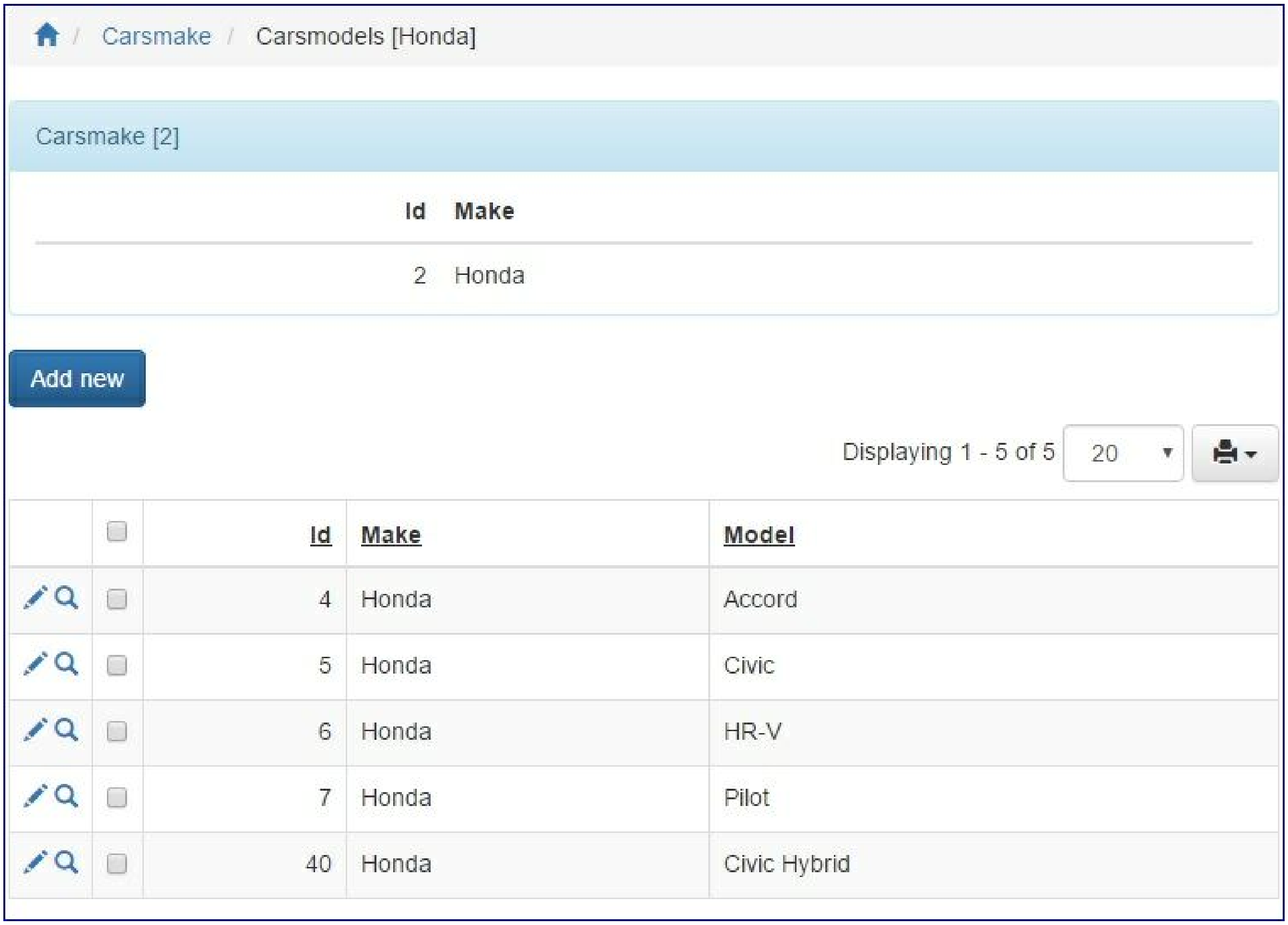
Customers

Customer ID	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code
SUPRD	Supremes delices	Pascale Cartrain	Accounting Manager	Boulevard Tirou, 255	Charleroi		0

Order Details

Order ID	Product ID	Unit Price	Quantity
10252	33	2.00	25
10252	59	55.00	40
10252	60	27.20	40

Display master table data on the details page:



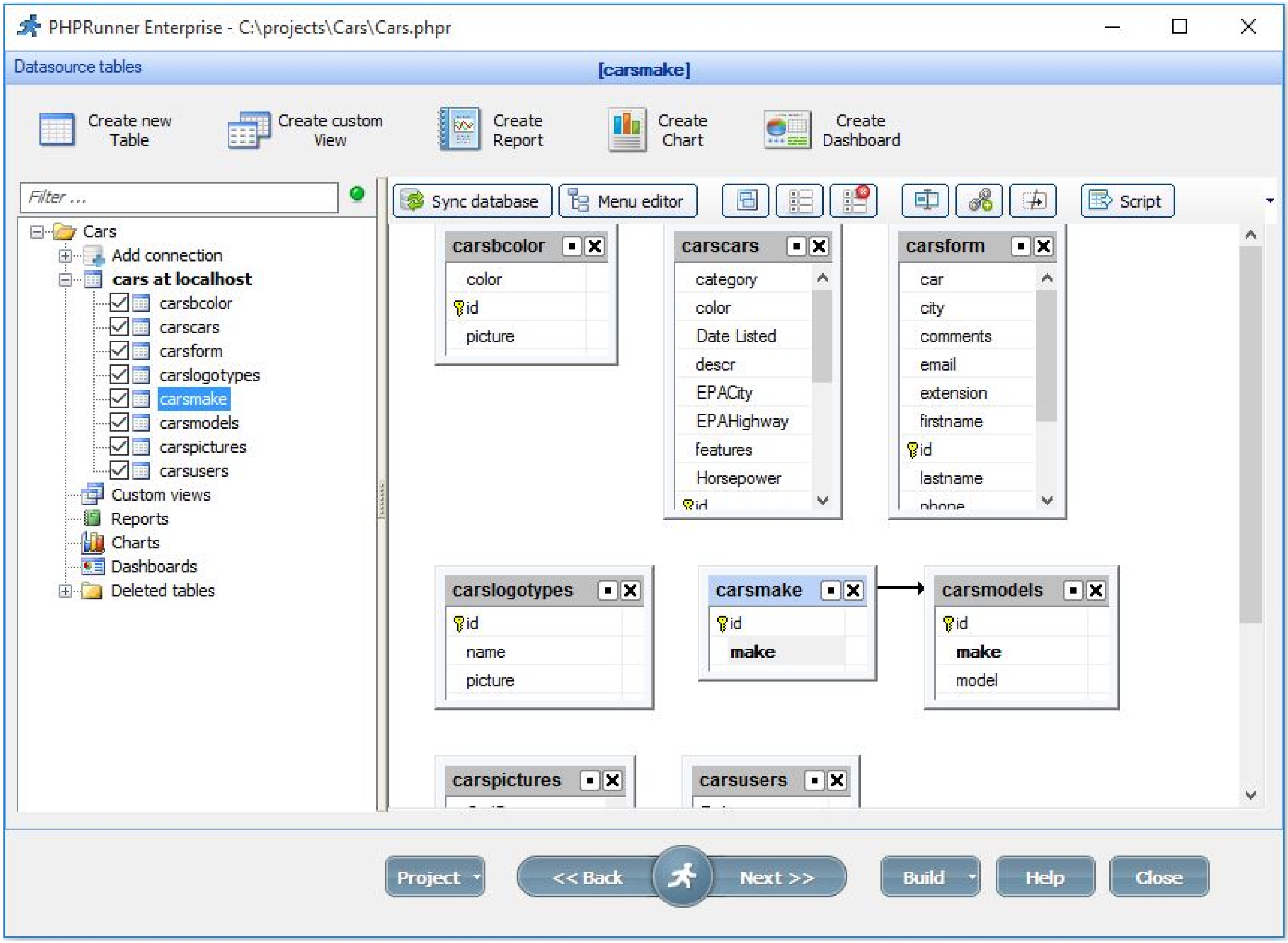
Carsmake [2]

Id	Make
2	Honda

Displaying 1 - 5 of 5

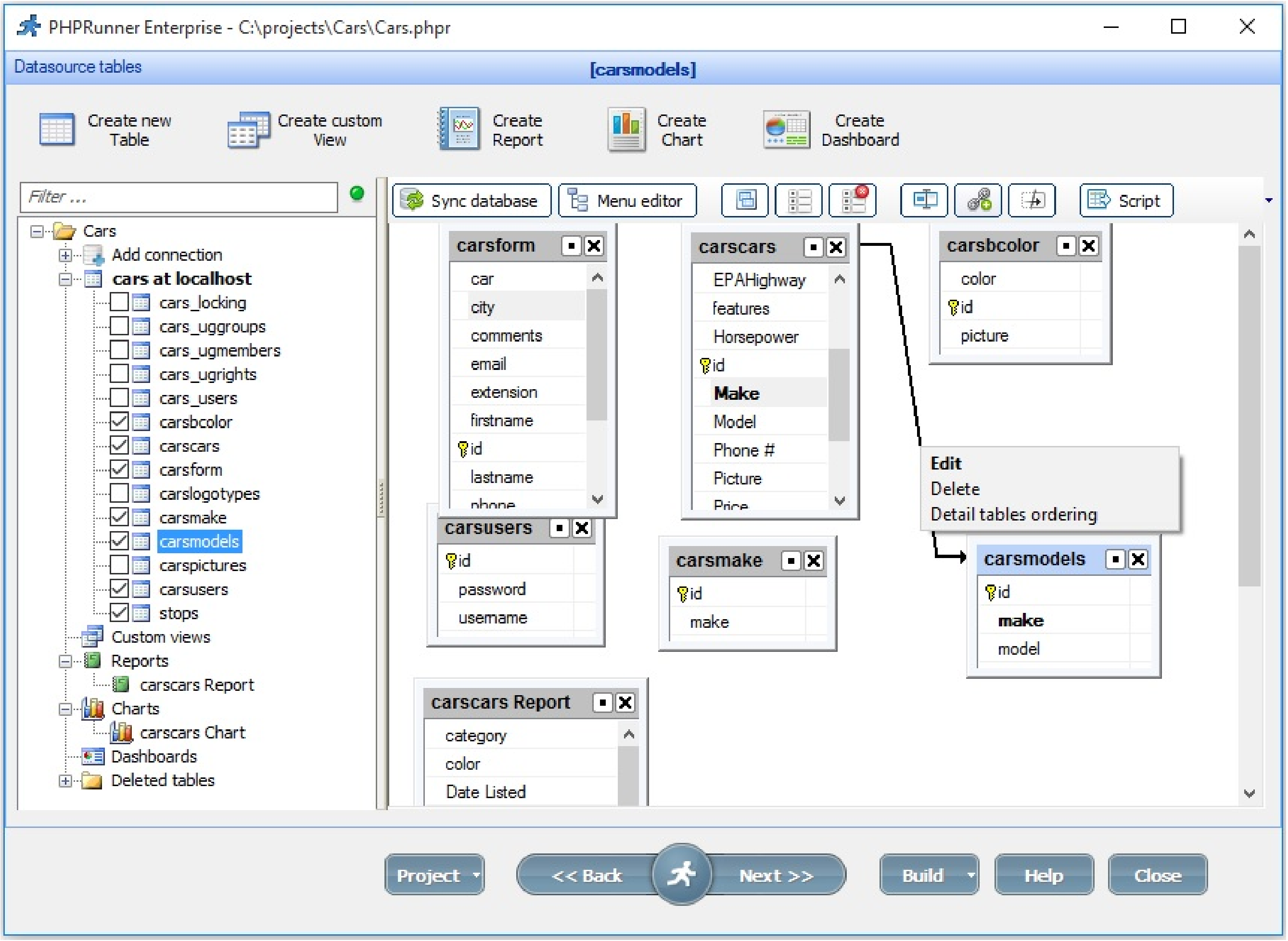
	□	Id	Make	Model
	□	4	Honda	Accord
	□	5	Honda	Civic
	□	6	Honda	HR-V
	□	7	Honda	Pilot
	□	40	Honda	Civic Hybrid

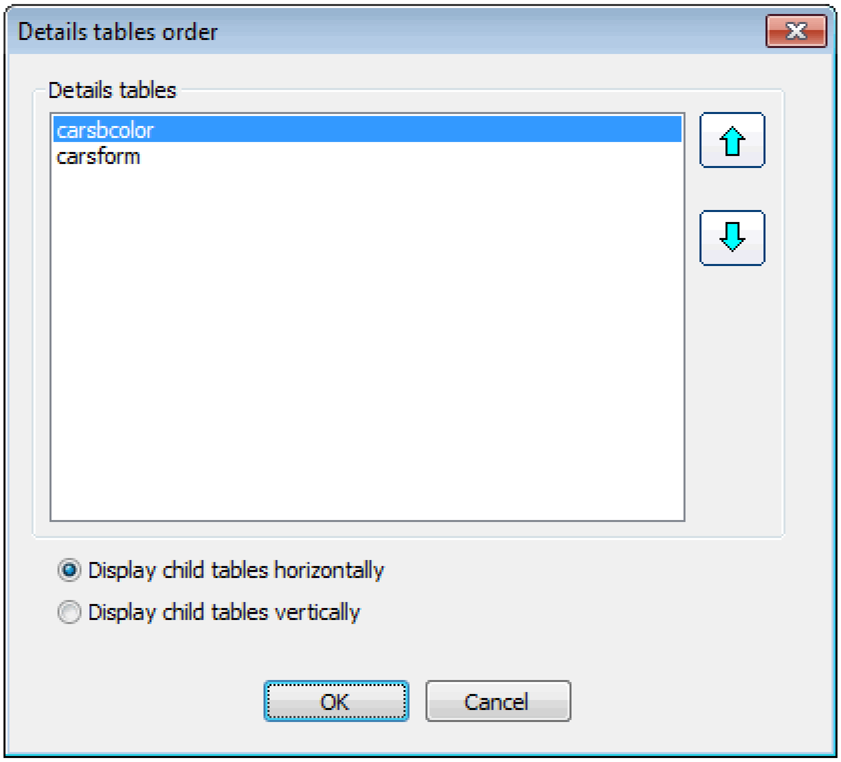
4. Now master-details relationship is created and displayed as arrow between tables.



Changing order of child tables

If master table has two or more details tables, you can re-order details tables as well as choose orientation (vertical or horizontal). To do so right click a link between tables and choose "Detail tables ordering".


PHPRunner Enterprise - C:\projects\Cars\Cars.phpr
Datasource tables
[carsmodels]
Create new Table
Create custom View
Create Report
Create Chart
Create Dashboard
Filter ...
Cars
Add connection
cars at localhost
cars_locking
cars_uggroups
cars_ugmembers
cars_ugrights
cars_users
carsbcolor
carscars
carsform
carslogotypes
carsmake
carsmodels
carspictures
carsusers
stops
Custom views
Reports
carscars Report
Charts
carscars Chart
Dashboards
Deleted tables
Sync database
Menu editor
Script
carsform
car
city
comments
email
extension
firstname
id
lastname
phone
carscars
EPAHighway
features
Horsepower
id
Make
Model
Phone #
Picture
Price
carsbcolor
color
id
picture
Edit
Delete
Detail tables ordering
carsusers
id
password
username
carsmake
id
make
carsmodels
id
make
model
carscars Report
category
color
Date Listed
Project
<< Back
Next >>
Build
Help
Close



Details tables order

Details tables

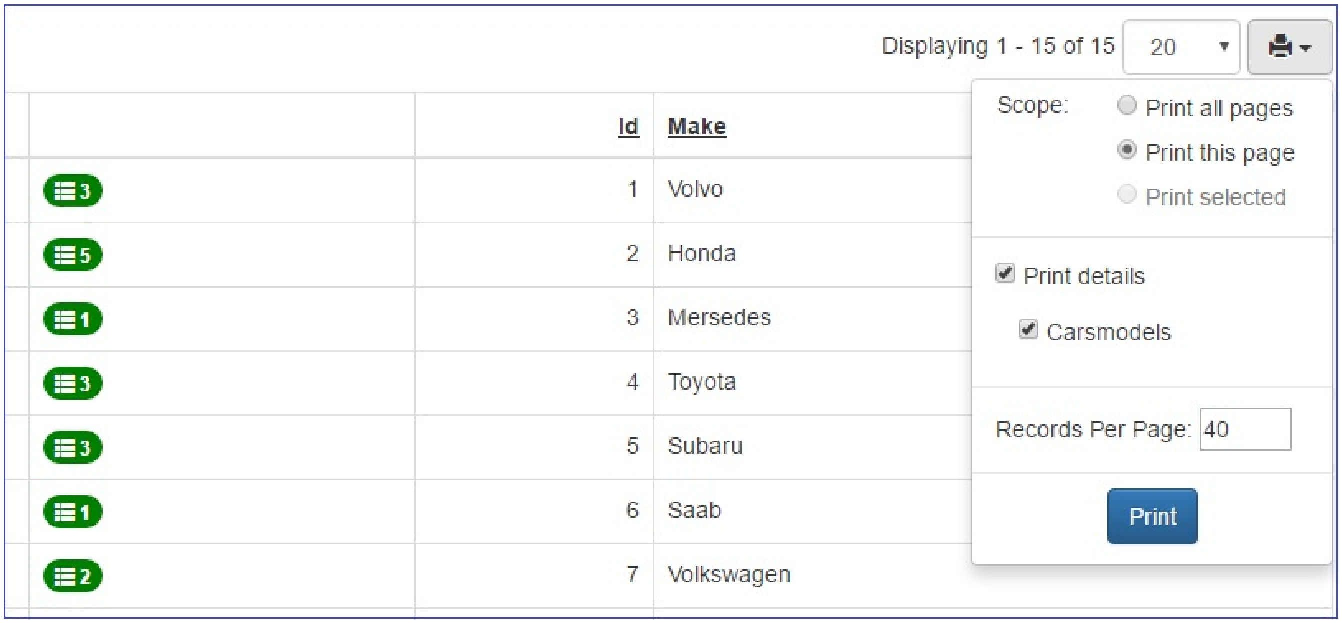
carsbcolor
carsform

● Display child tables horizontally
○ Display child tables vertically

For more information, see Master-details relationship.

Print master table data with details

You can print the master table data on the List page with data from details tables.



Displaying 1 - 15 of 15 20

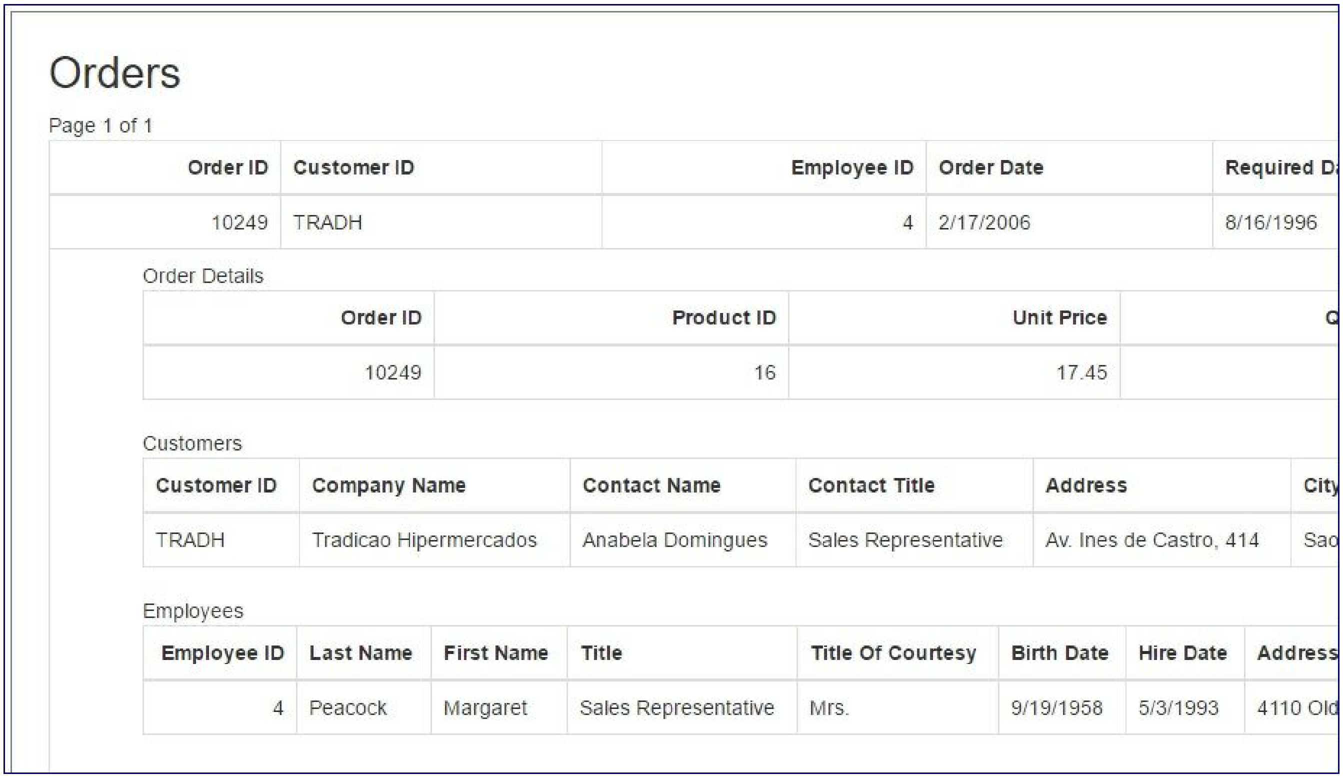
	Id	Make
3	1	Volvo
5	2	Honda
1	3	Mersedes
3	4	Toyota
3	5	Subaru
1	6	Saab
2	7	Volkswagen

Scope: ○ Print all pages
● Print this page
○ Print selected

☑ Print details
☑ Carsmodels

Records Per Page: 40

Example of page to be printed:



Orders

Page 1 of 1

Order ID	Customer ID	Employee ID	Order Date	Required Da
10249	TRADH	4	2/17/2006	8/16/1996

Order Details

Order ID	Product ID	Unit Price	Q
10249	16	17.45	

Customers

Customer ID	Company Name	Contact Name	Contact Title	Address	City
TRADH	Tradicao Hipermercados	Anabela Domingues	Sales Representative	Av. Ines de Castro, 414	Sao

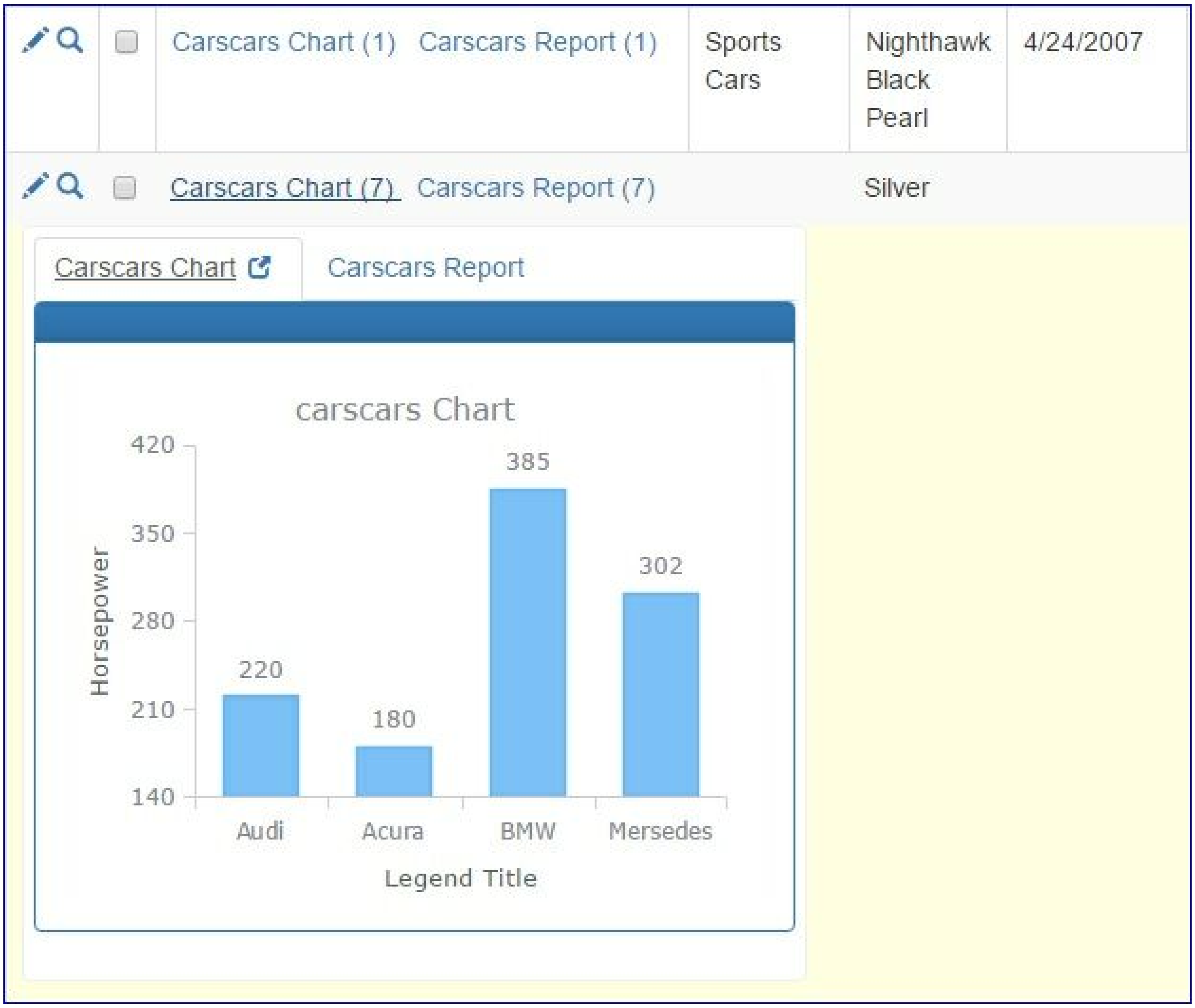
Employees

Employee ID	Last Name	First Name	Title	Title Of Courtesy	Birth Date	Hire Date	Address
4	Peacock	Margaret	Sales Representative	Mrs.	9/19/1958	5/3/1993	4110 Old

Charts and reports as master and details tables

Now you can use charts and reports as both master and details tables.

Chart as details table:

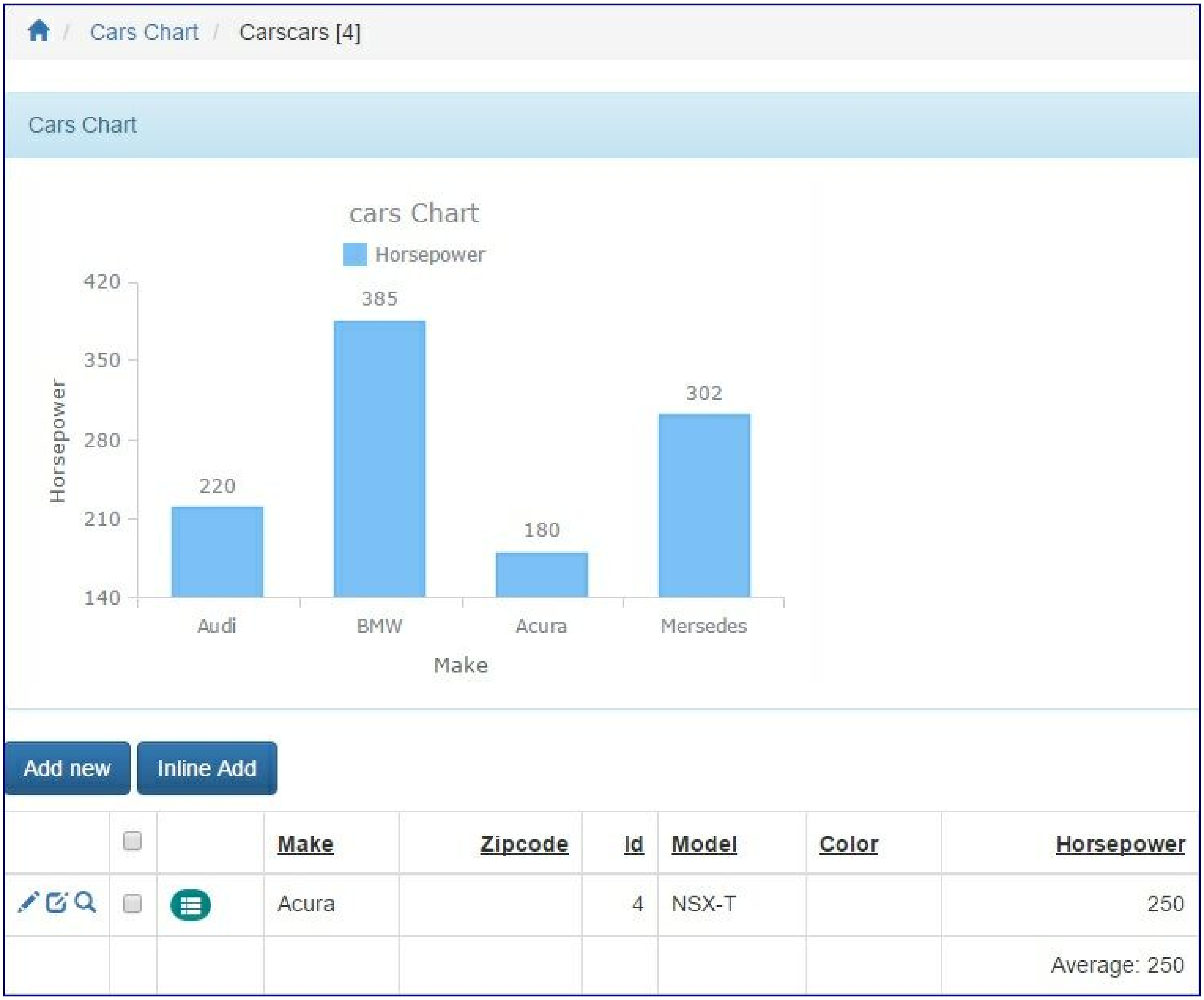


	□	Carscars Chart (1)	Carscars Report (1)	Sports Cars	Nighthawk Black Pearl	4/24/2007
	□	Carscars Chart (7)	Carscars Report (7)		Silver	

Carscars Chart

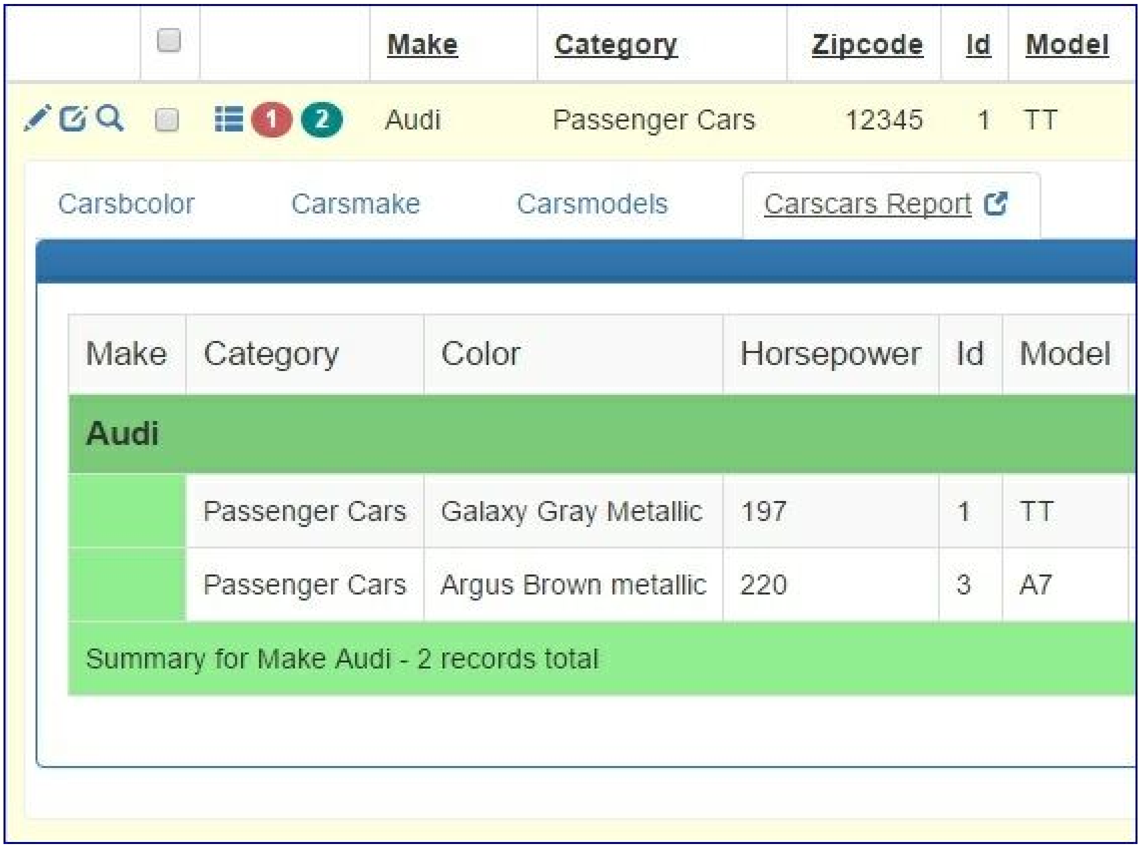
Carscars Report

Chart as master table:



	□		Make	Zipcode	Id	Model	Color	Horsepower
✎ ☑ 🔍	□	☰	Acura		4	NSX-T		250
								Average: 250

Report as details table:



	□		Make	Category	Zipcode	Id	Model
	□	1 2	Audi	Passenger Cars	12345	1	TT

Make	Category	Color	Horsepower	Id	Model
Audi					
	Passenger Cars	Galaxy Gray Metallic	197	1	TT
	Passenger Cars	Argus Brown metallic	220	3	A7
Summary for Make Audi - 2 records total					

2.7 SQL query page

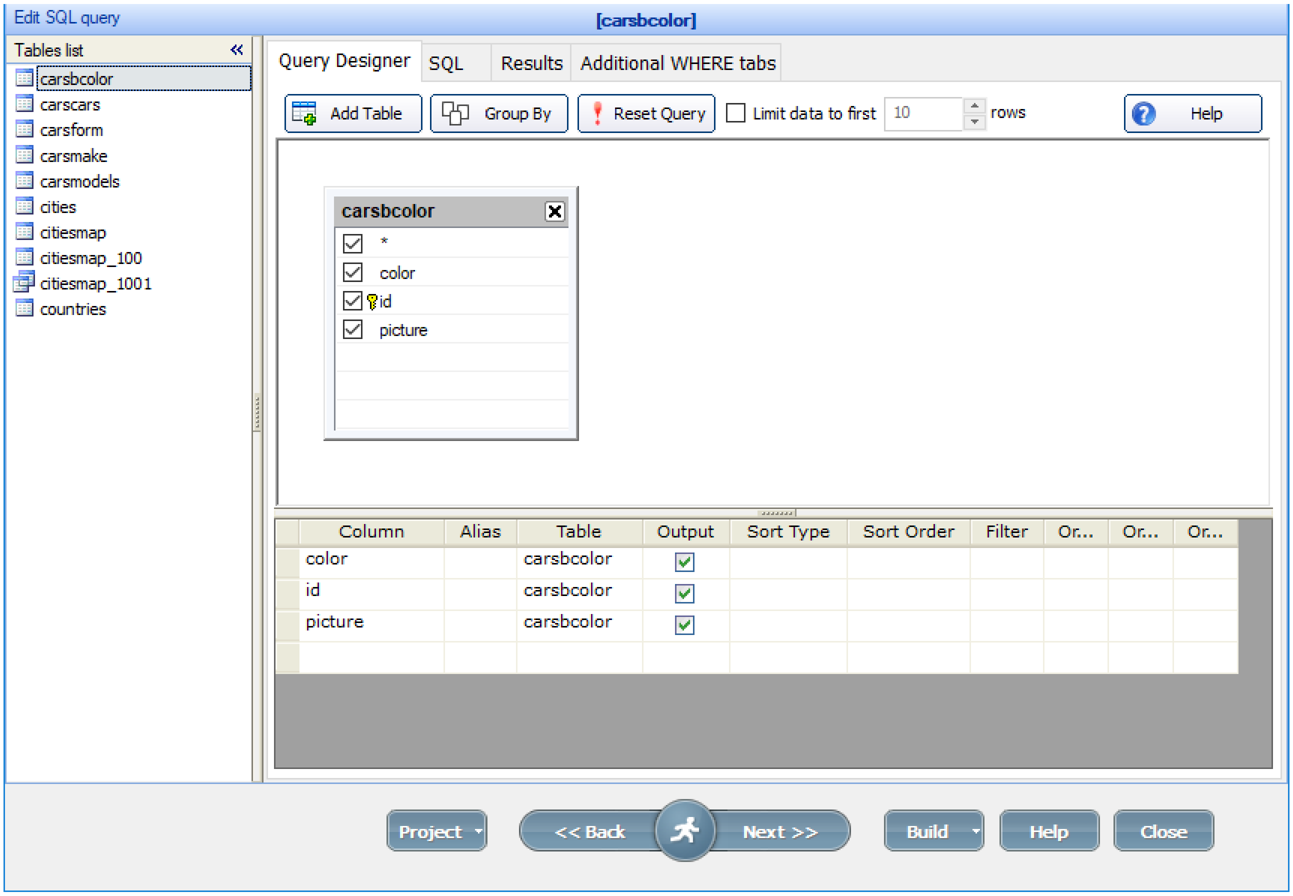
2.7.1 About SQL query designer

On the **Edit SQL query** page you can modify SQL query that PHPRunner has built for you automatically. In most cases you can proceed with default SQL query.

This page includes graphical panes that display your SQL statement (Query Designer tab), a text pane that displays the text of your SQL statement (SQL tab) and result of edited query as table with values (Results tab). You can work in either the graphical or text panes and then check result of executed query. Query Designer synchronizes the views so they are always current.

Features:

1. a visual interface to design queries
2. automatic SQL Statement generation
3. create joins with drag and drop
4. the grid pane to specify criteria (Order By, Group By, Where etc.)
5. SQL parsing - enter the SQL statement, and the diagram and grid will be populated



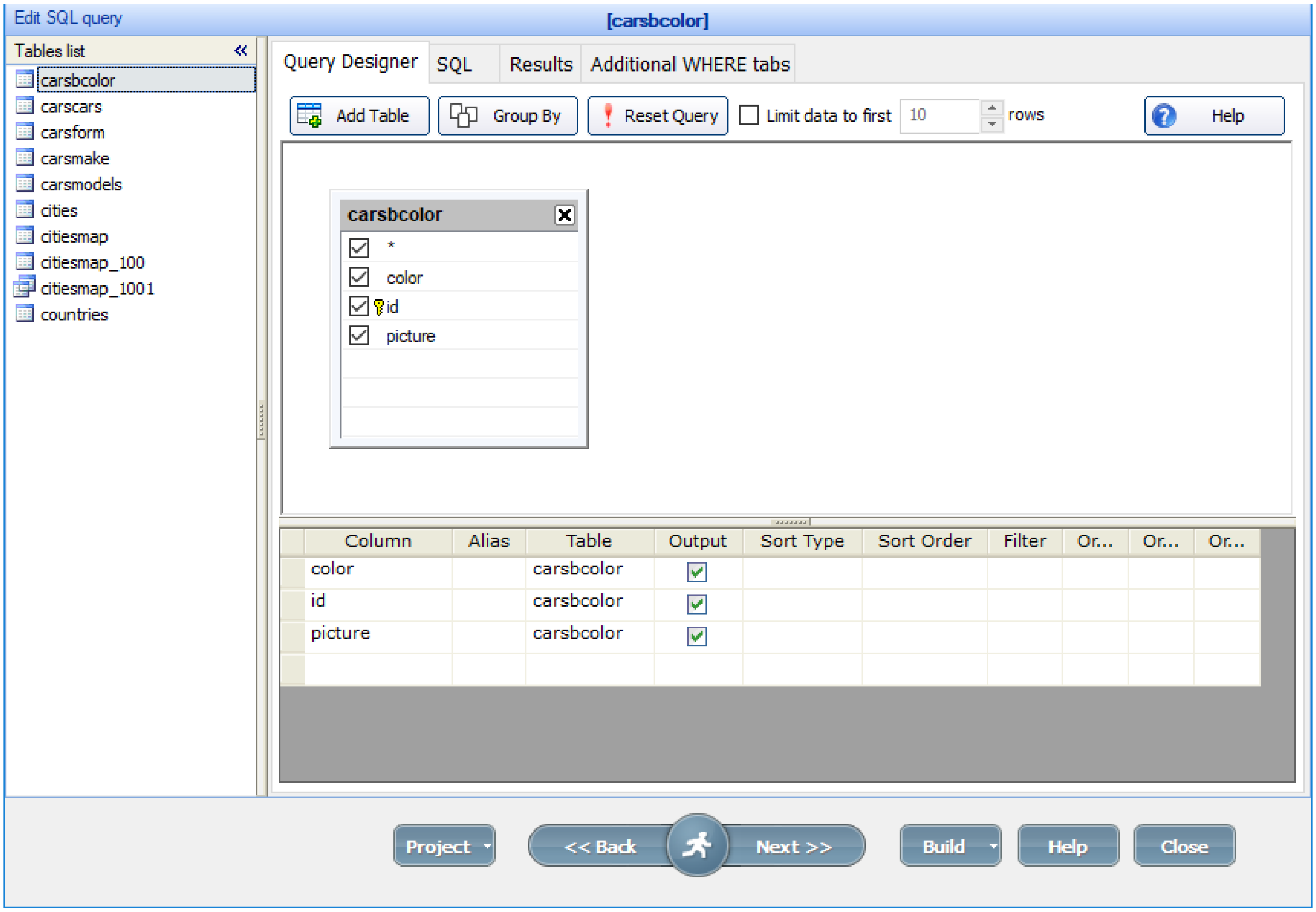
☐ Limit data to first 10 rows

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
color		carsbcolor	☑						
id		carsbcolor	☑						
picture		carsbcolor	☑						

2.7.2 Query Designer

Query Designer gives you the ability to use the simple graphical interface to construct SQL statements.

Note: if you modify default SQL query, make sure that key column(s) are included into fields list. This is required to provide edit/delete functionality.



Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
color		carsbcolor	☑						
id		carsbcolor	☑						
picture		carsbcolor	☑						

To switch between tables use **Tables list** panel on the left. Note that all fields marked with a tick in the *Output* column are added to the SELECT clause.

Limit data to first "N" rows.
This new cool feature allows you to limit the number of records to be displayed on List, Print, Export, Details pages.

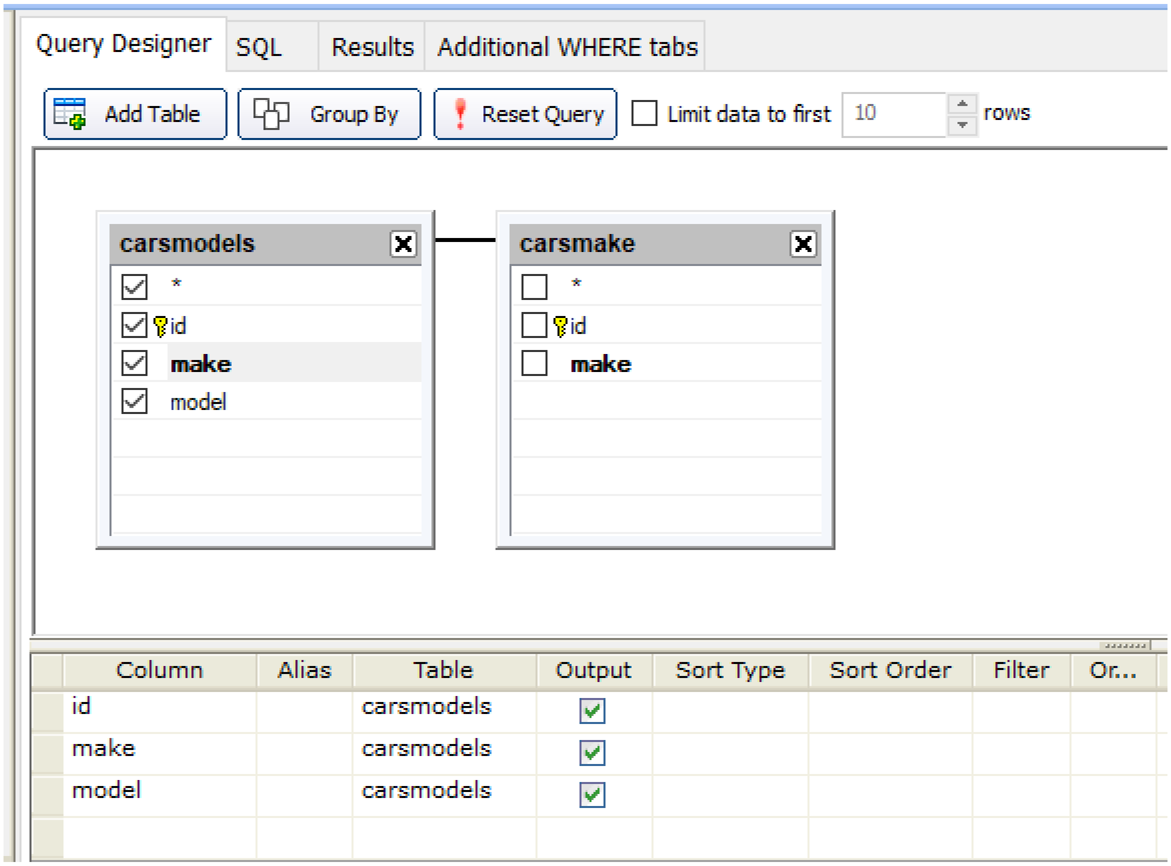
It can be useful when you need to speed up the load of the webpage or your chart has too many items on it and you only need to show the most significant ones. It also works with list pages added to dashboard.

Search and filters will be applied first and then results will be limited to first "N" rows.
This option will work with all PHPRunner project items except reports with that have group fields selected.

What is supported

Inner joins, outer joins

To add join click the **Add Table** button, select table and then drag and drop any field from first table to the joined table. To setup join type click the line between tables, select foreign keys on the **Table link properties** dialog in both tables and choose join type.



□ Limit data to first 10 rows

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...
id		carsmodels	☑				
make		carsmodels	☑				
model		carsmodels	☑				

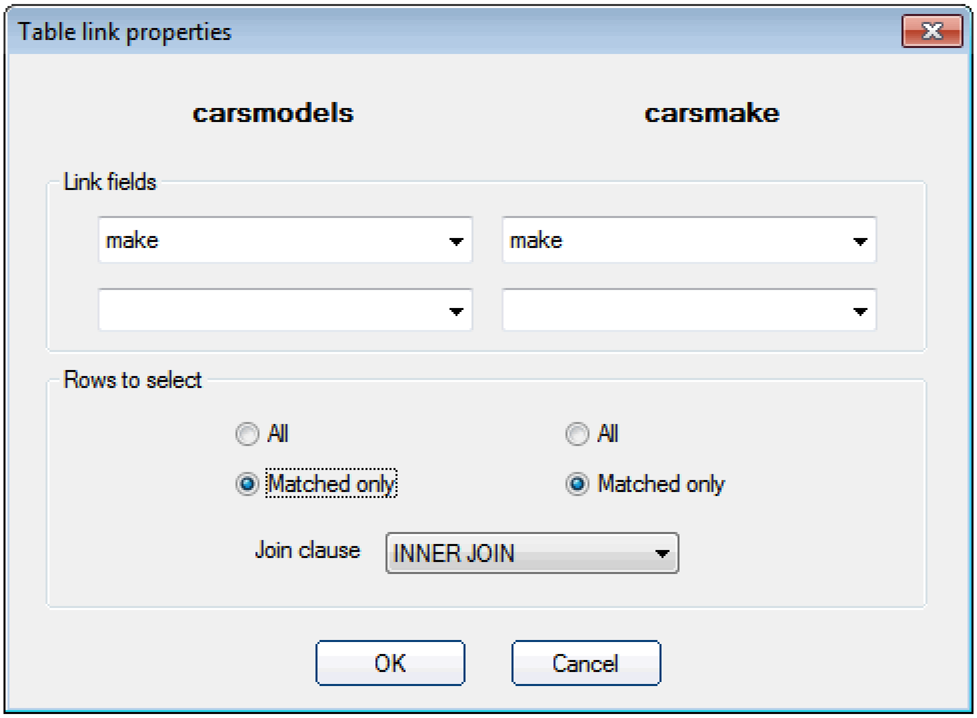


Table link properties

carsmodels	carsmake
Link fields: make	make
○ All ● Matched only	○ All ● Matched only

Join clause: INNER JOIN

More info about join types.

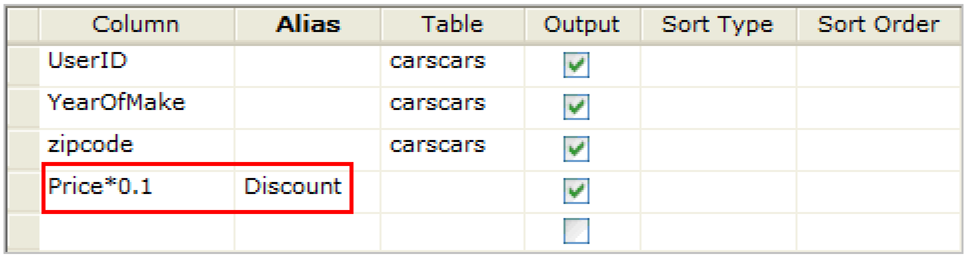
SQL query:

```
SELECT
carsmodels.id,
carsmodels.model,
carsmodels.make
FROM carsmodels
INNER JOIN carsmake ON carsmodels.make = carsmake.make
```

Note: it's recommended to use aliases for fields from joined tables to avoid confusion when two fields from different tables have the same name.

Calculated fields

To add calculated fields use empty grid below all field names:



Column	Alias	Table	Output	Sort Type	Sort Order
UserID		carscars	☑		
YearOfMake		carscars	☑		
zipcode		carscars	☑		
Price*0.1	Discount		☑		
			□		

SQL query:

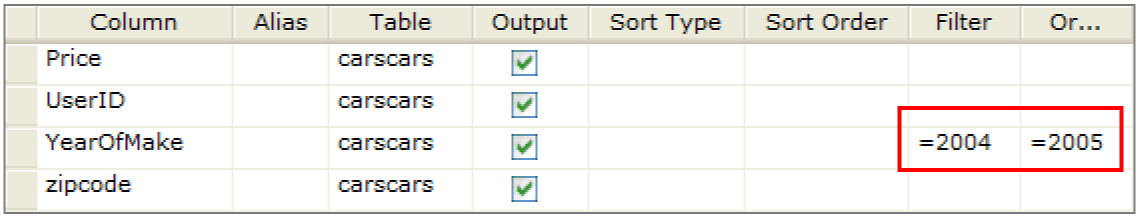
```
SELECT
category,
color,
Date Listed,
descr,
EPACity,
EPAHighway,
features,
UserID,
YearOfMake,
zipcode,
Price*0.1 AS Discount
FROM carscars
```

In the example above the alias *Discount* is assigned to the calculated field *Price*0.1*. Note that if the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. So you should use *$values["Discount"]* instead of *$values["Price*0.1"]* in your events. For more information about events, see Events.

We do not recommend using aliases to give a field another name. If you have very long or complex field names, you can assign a label to the field on Choose fields page or in the Label editor instead of using aliases.

WHERE clause

You can add where clause in the **Filter** column. If you need to add two or more conditions use **Or** columns.



Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...
Price		carscars	☑				
UserID		carscars	☑				
YearOfMake		carscars	☑			=2004	=2005
zipcode		carscars	☑				

SQL query:

```
SELECT *
FROM carscars
WHERE YearOfMake =2004
```

For more complicated queries wrap the condition by parentheses:

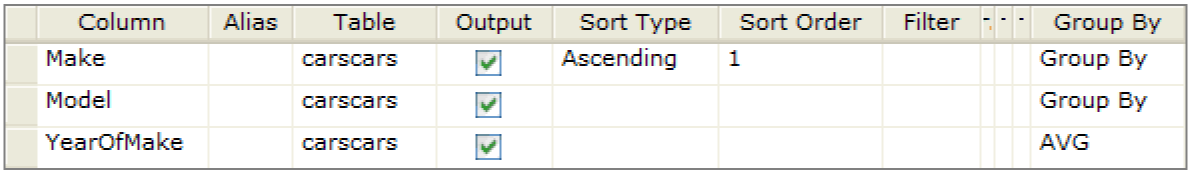
```
SELECT *
FROM carscars
WHERE ( YearOfMake =2004 OR YearOfMake =2005 )
```

ORDER BY clause

If you'd like to specify default sorting order on the list page (ascending or descending) select **Sort Type** in the corresponding column for necessary fields.

GROUP BY clause

To add GROUP BY clause click the **Group By** button and select one of grouping function in **Group By** column.



Column	Alias	Table	Output	Sort Type	Sort Order	Filter	-	-	-	Group By
Make		carscars	☑	Ascending	1					Group By
Model		carscars	☑							Group By
YearOfMake		carscars	☑							AVG

SQL query:

```
SELECT
Make,
Model,
AVG (YearOfMake)
FROM carscars
GROUP BY Make, Model
ORDER BY Make
```

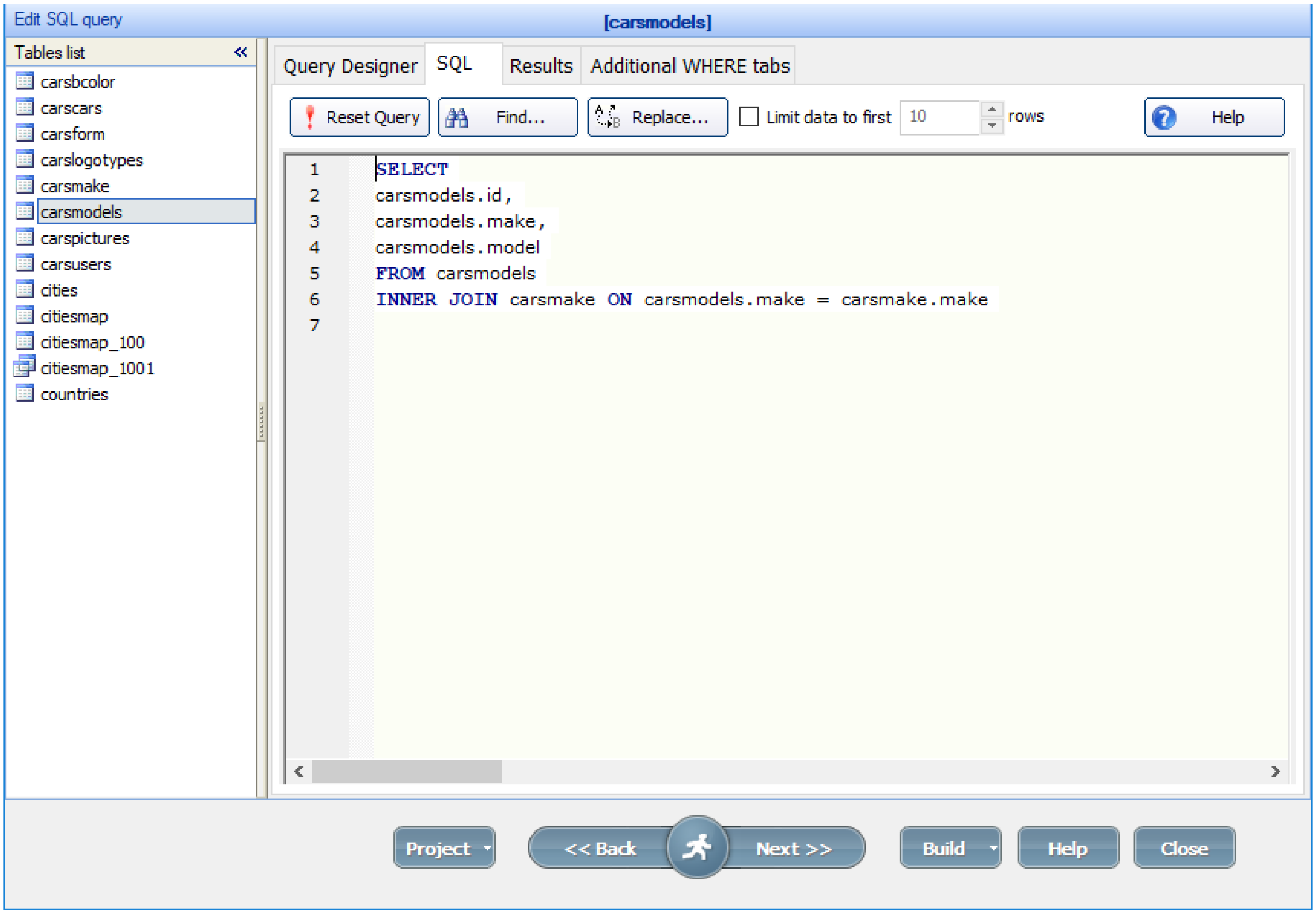
What is not supported

- Stored procedure calls
- Update/Delete/Insert/Create queries
- Unions
- DISTINCT keyword

2.7.3 SQL

On the **SQL** tab you can modify SQL query manually. All changes on this tab automatically transfer to the Query Designer.

Note: if you modify default SQL query, make sure that key column(s) are included. This is required to provide edit/delete functionality. If table participates in Master-Details relationship make sure link fields (primary/foreign key) stay on the list of fields.



```
SELECT
carsmodels.id,
carsmodels.make,
carsmodels.model
FROM carsmodels
INNER JOIN carsmake ON carsmodels.make = carsmake.make
```

To switch between tables use **Tables list** panel on the left. **Find** and **Replace** buttons let you quickly search within your SQL code and modify it.

What is supported

Inner joins, outer joins

SQL query:

```
SELECT
carsmodels.id,
carsmodels.model,
carsmodels.make
FROM carsmodels
INNER JOIN carsmake ON carsmodels.make = carsmake.make
```

Note: it's recommended to use aliases for fields from joined tables to avoid confusion when two fields from different tables have the same name.

Calculated fields

SQL query:

```
SELECT
category,
color,
Date Listed,
descr,
EPACity,
EPAHighway,
features,
UserID,
YearOfMake,
zipcode,
Price*0.1 AS Discount
FROM carscars
```

In the example above the alias *Discount* is assigned to the calculated field *Price*0.1*. Note that if the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. So you should use *$values["Discount"]* instead of *$values["Price*0.1"]* in your events. For more information about events, see Events.

Do not use aliases as field labels in your app. If you have very long or complex field names, you can assign a label to the field on Choose fields page or in the Label editor instead of using aliases.

WHERE clause

SQL query:

```
SELECT *
FROM carscars
WHERE YearOfMake =2004
```

For more complicated queries wrap the condition by parentheses:

```
SELECT *
FROM carscars
WHERE ( YearOfMake =2004 OR YearOfMake =2005 )
```

ORDER BY and GROUP BY clauses

SQL query:

```
SELECT
Make,
Model,
AVG (YearOfMake)
FROM carscars
GROUP BY Make, Model
```

```
ORDER BY Make
```

Aliases

When you connect to databases like DB2, Oracle or Postgre and your SQL query contains aliases, we recommend to enclose them in double quotes. Here is an example:

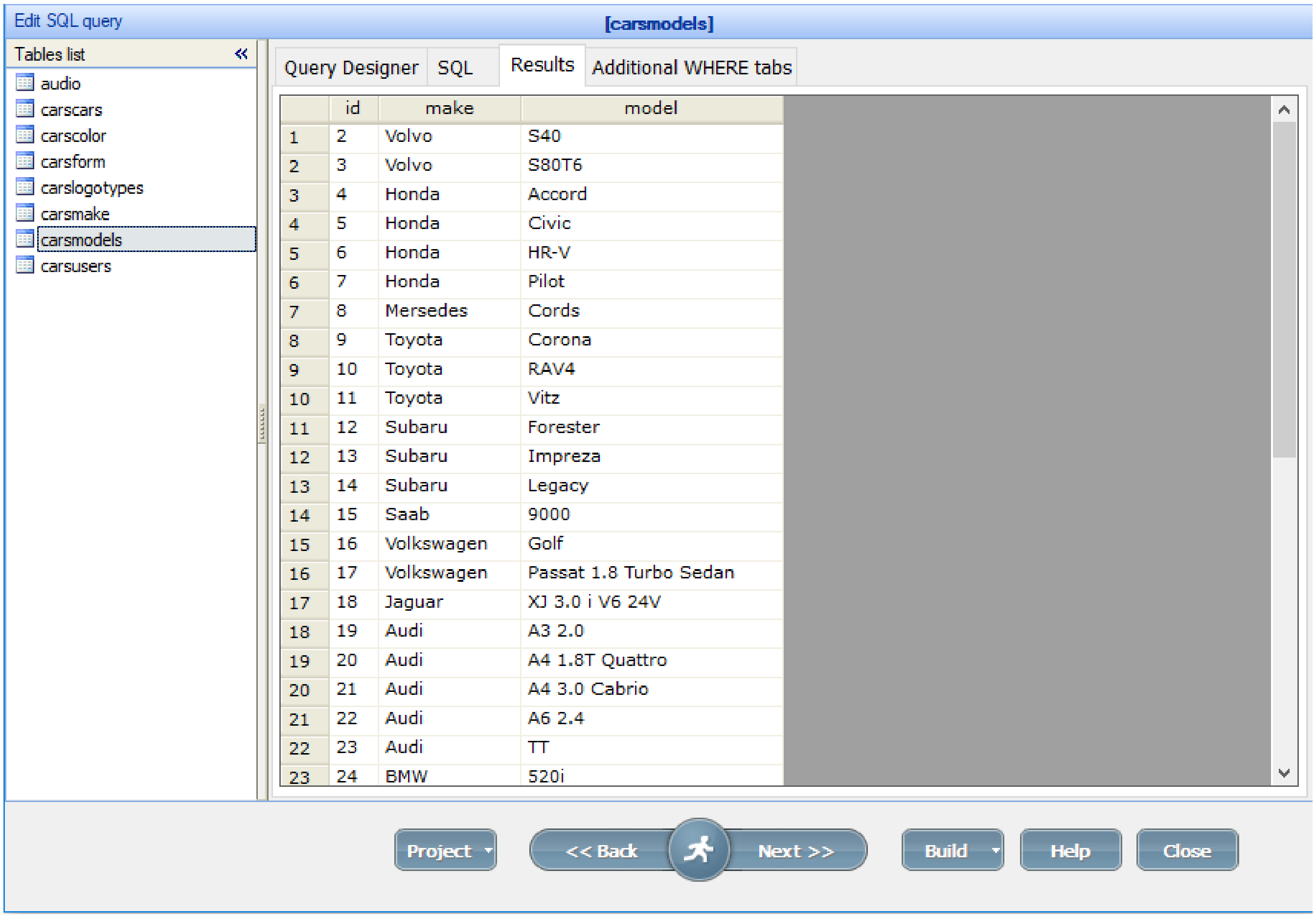
```
select FieldName as "FieldAlias"
from TableName
```

What is not supported

- Stored procedure calls
- Update/Delete/Insert/Create queries
- Unions
- DISTINCT keyword

2.7.4 Results

The **Results tab** lets you see the results of an SQL query. Up to 200 records will be displayed.



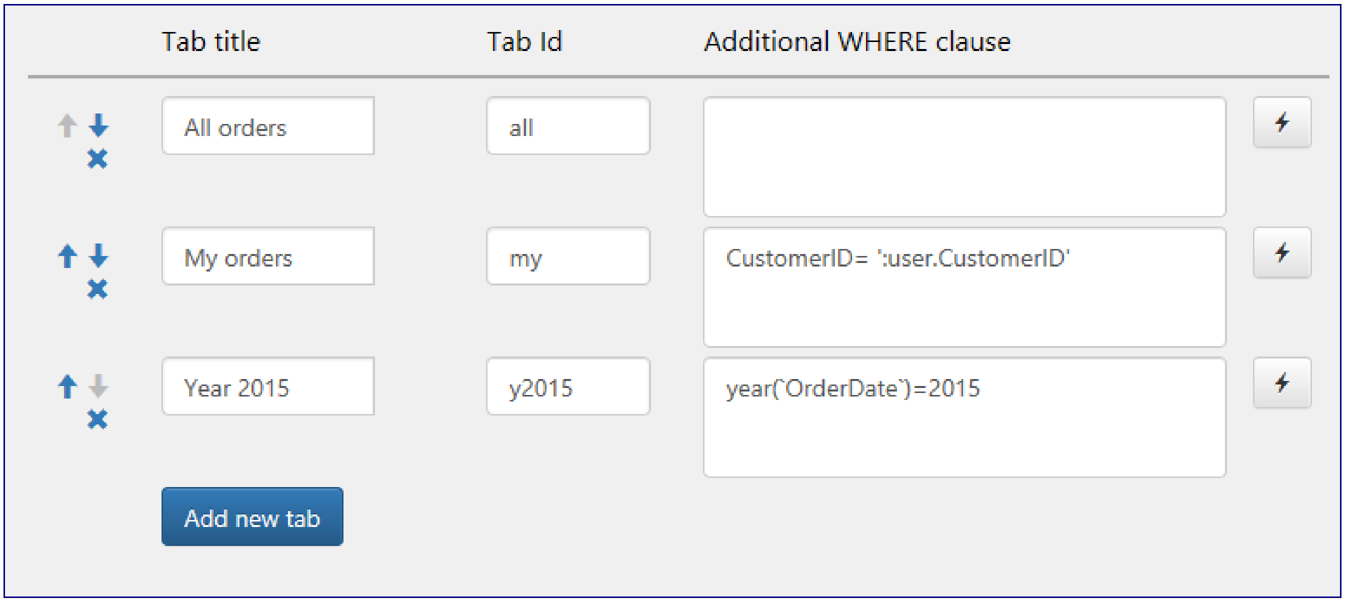
	id	make	model
1	2	Volvo	S40
2	3	Volvo	S80T6
3	4	Honda	Accord
4	5	Honda	Civic
5	6	Honda	HR-V
6	7	Honda	Pilot
7	8	Mersedes	Cords
8	9	Toyota	Corona
9	10	Toyota	RAV4
10	11	Toyota	Vitz
11	12	Subaru	Forester
12	13	Subaru	Impreza
13	14	Subaru	Legacy
14	15	Saab	9000
15	16	Volkswagen	Golf
16	17	Volkswagen	Passat 1.8 Turbo Sedan
17	18	Jaguar	XJ 3.0 i V6 24V
18	19	Audi	A3 2.0
19	20	Audi	A4 1.8T Quattro
20	21	Audi	A4 3.0 Cabrio
21	22	Audi	A6 2.4
22	23	Audi	TT
23	24	BMW	520i

You may use the **Tables list** panel on the left to switch between tables.

2.7.5 Tabs and additional WHERE clause

Tabs and additional WHERE on List page

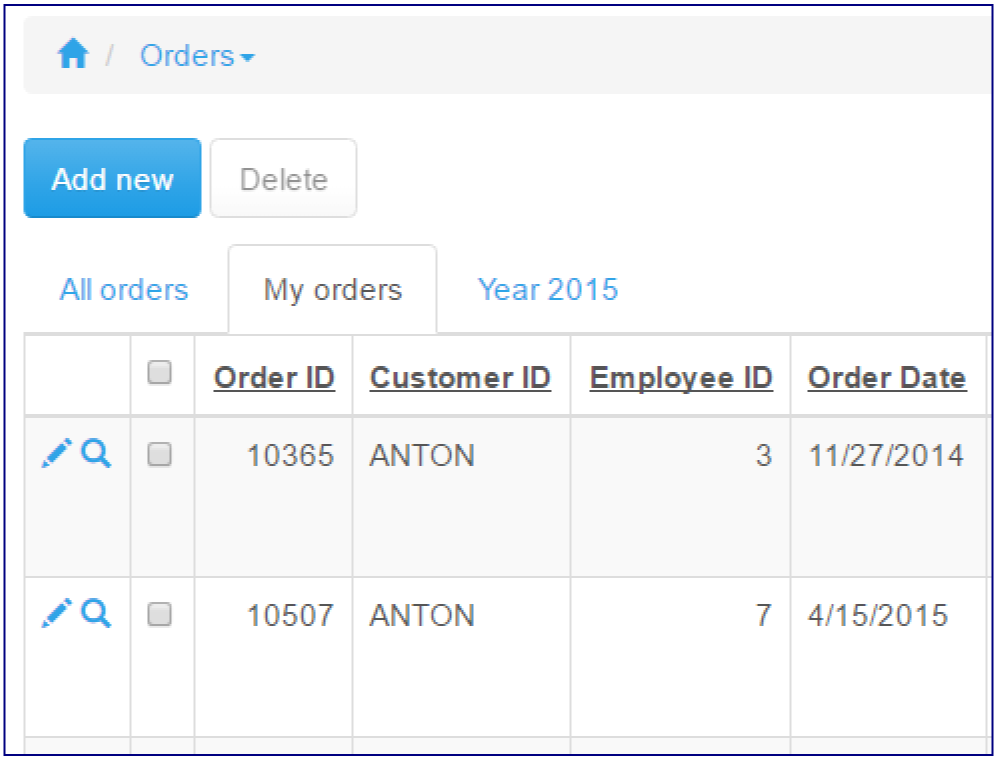
This feature will only work with Bootstrap layout.



Tab title	Tab Id	Additional WHERE clause
All orders	all	
My orders	my	CustomerID= ':user.CustomerID'
Year 2015	y2015	year(`OrderDate`)=2015

You can also use SQL variables in the WHERE tabs.

View in generated application



Order ID	Customer ID	Employee ID	Order Date
10365	ANTON	3	11/27/2014
10507	ANTON	7	4/15/2015

Sample WHERE expressions

"phone like '%555%'"

"city like 'M%'"

"salary>50000"

You can find some other examples of using SQL here.

You can also create and manage WHERE tabs from your code to:

- create a tab dynamically use addTab() function.
- delete tab title use deleteTab() function.
- change tab title use setTabTitle() function.
- change tab where clause use setTabWhere() function.

See also SQL variables in the WHERE tabs.

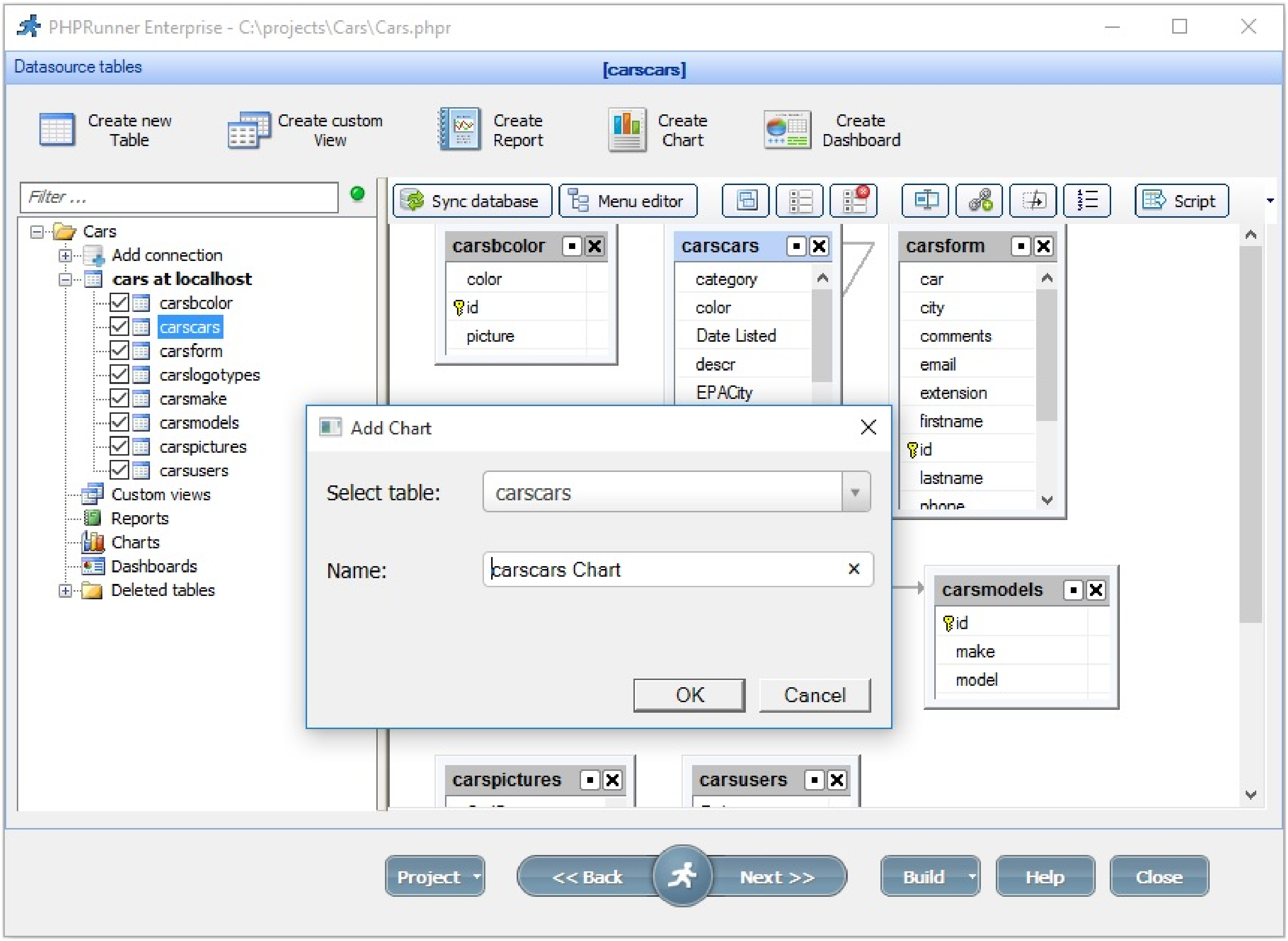
2.8 Charts

2.8.1 Creating chart

To create a chart:

1. Proceed to the **Datasource tables** page and click **Create Chart**.

2. Select datasource table and chart name. Click **OK**.



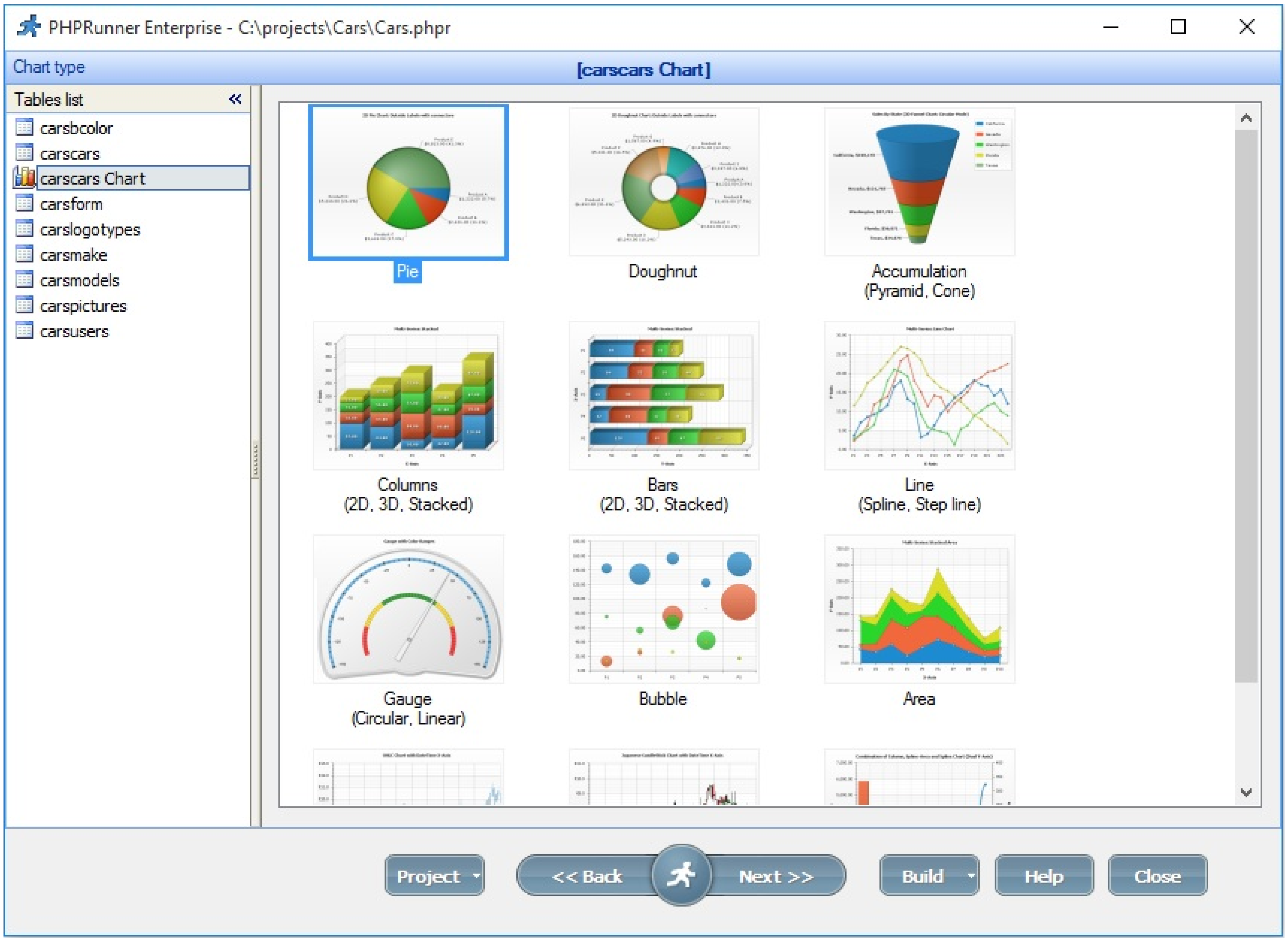
Select table: carscars

Name: carscars Chart

Note that you can create a copy of existing chart (right-click the chart and select **Copy**).

On the next several pages (use **Next** button to navigate) you can:

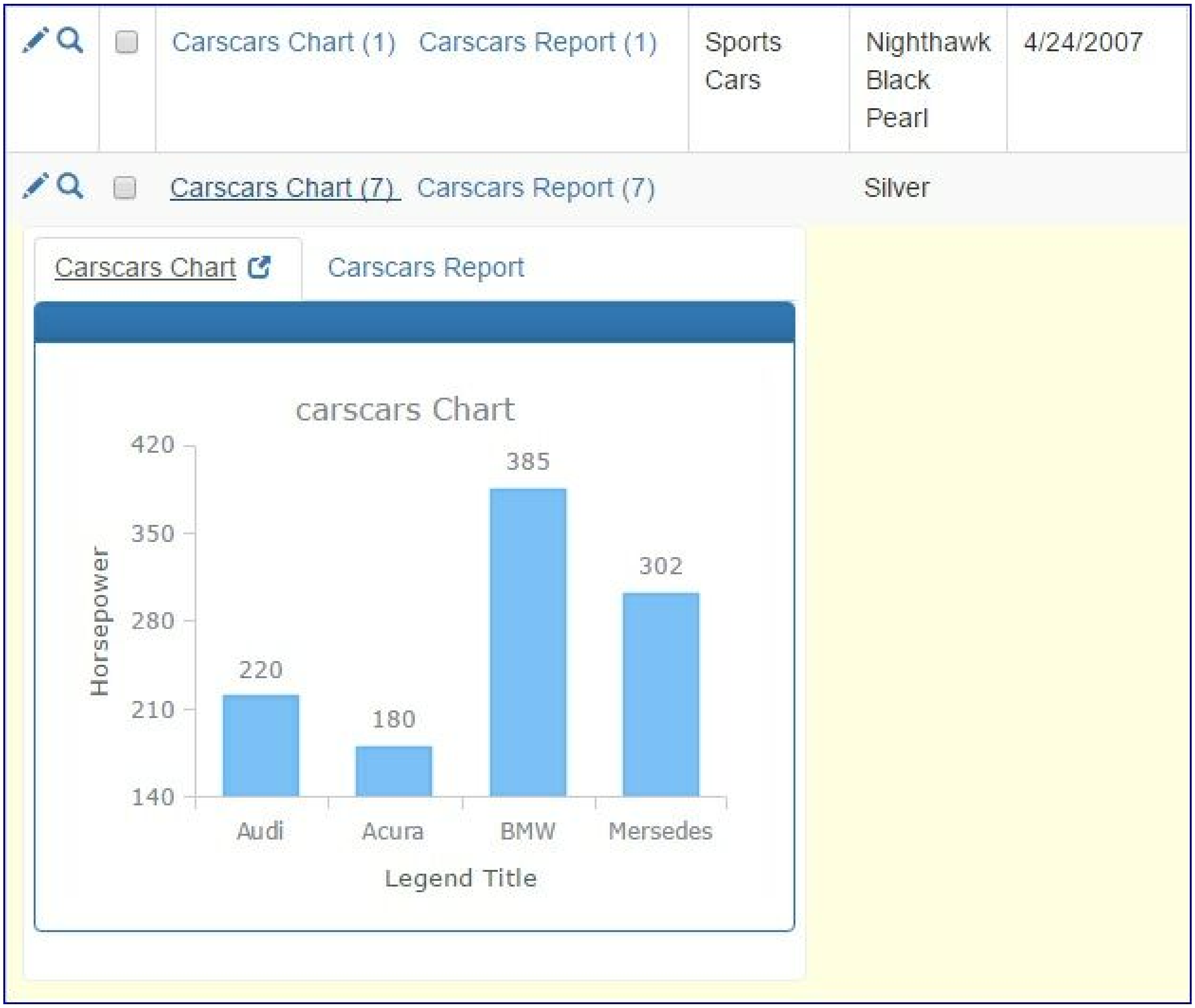
- make necessary changes to SQL query. More info about editing SQL queries;
- select the type of chart to build. More info about chart types;



- choose Data Series fields (more info about setting chart parameters);
- modify chart appearance options.

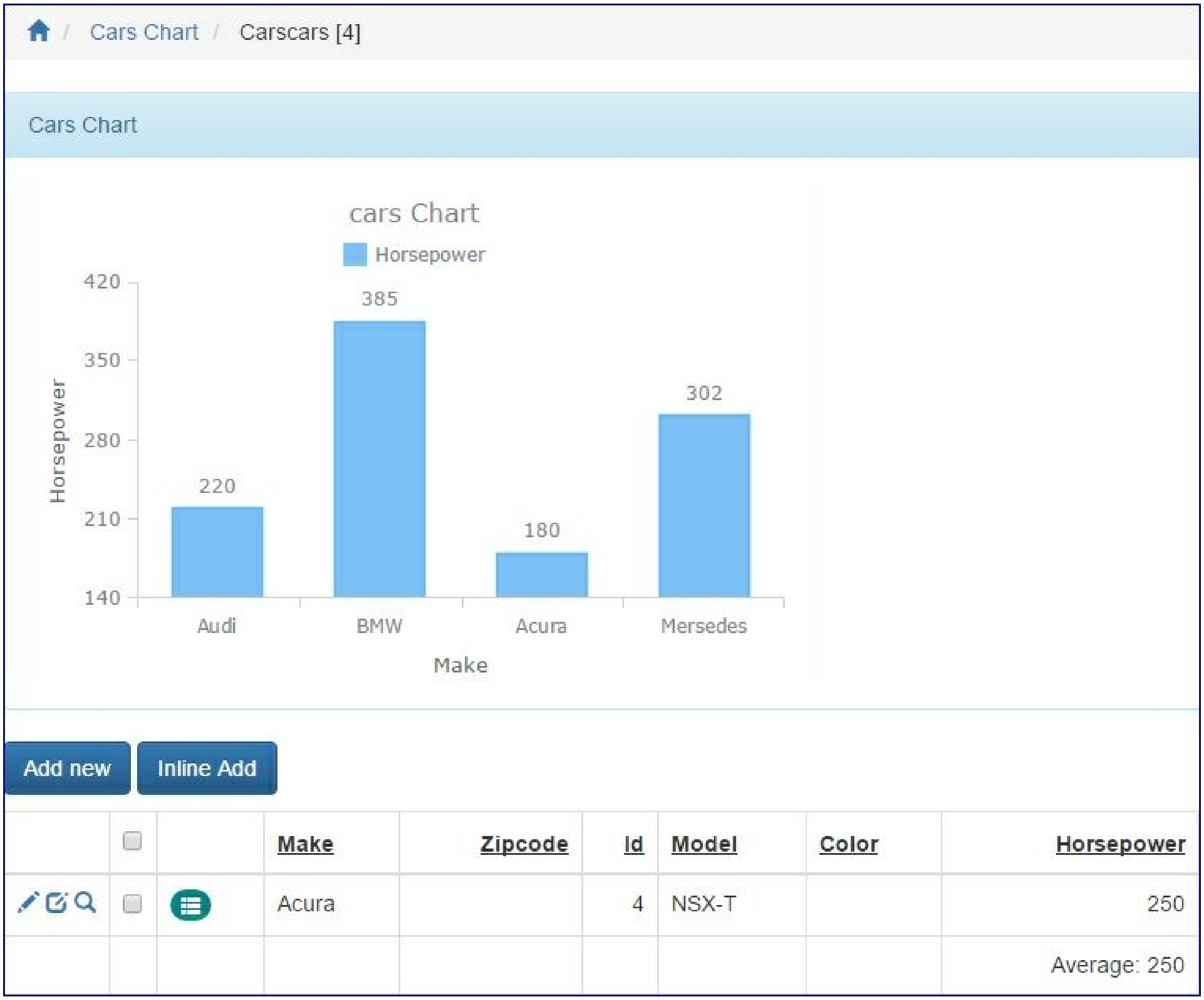
Note that you can use charts as both master and details tables. For more information, see Master-details relationship between tables.

Chart as details table:



□	Carscars Chart (1) Carscars Report (1)	Sports Cars	Nighthawk Black Pearl	4/24/2007
□	Carscars Chart (7) Carscars Report (7)		Silver	

Chart as master table:



			Make	Zipcode	Id	Model	Color	Horsepower
	□		Acura		4	NSX-T		250
								Average: 250

To further customize chart appearance like setting colors, fonts or changing chart title use ChartModify event.

2.8.2 Chart types

2.8.2.1 List of chart types

List of chart types:

Chart type	Single Series	Multi Series	Horizontal	Vertical	3D
Accumulation	Yes	No	Yes	No	No
Area	Yes	Yes	Yes	No	No

Bubble	Yes	Yes	Yes	No	Yes
Column/Bar	Yes	Yes	Column	Bar	Yes
Combined	Yes	Yes	Yes	No	No
Financial OHLC/Candlestick	Yes	Yes	-	-	No
Gauge	Yes	No	-	-	No
Line	Yes	Yes	Yes	No	No
Pie/Doughnut	Yes	Yes	-	-	No

2.8.2.2 Accumulation chart

Accumulation charts are typically single series charts representing the data in percentages and do not utilize axes. The height of a chart segment is proportional to the y-coordinate value of the corresponding point.

Chart settings:

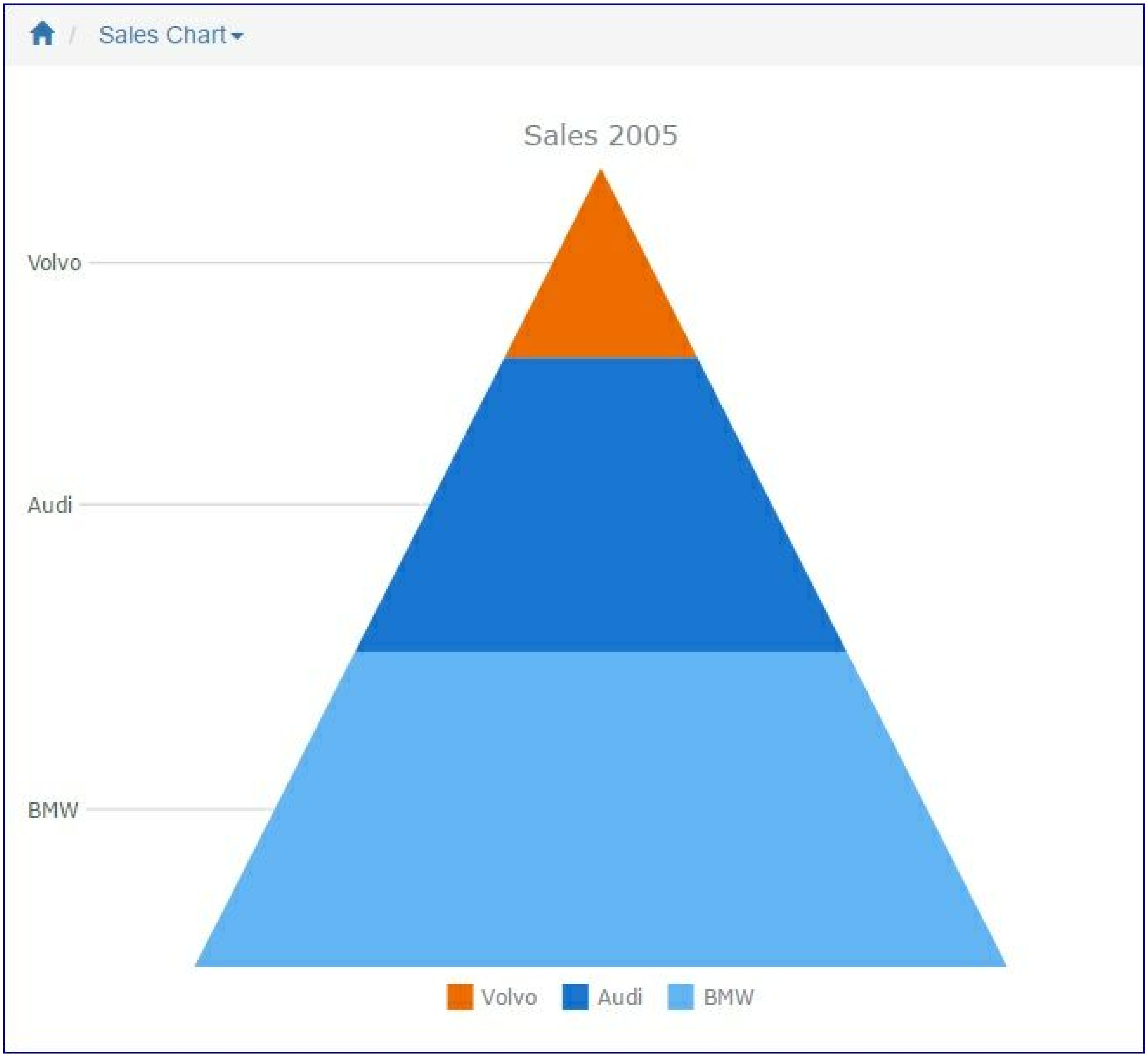
- **Accumulation appearance** - this option defines the chart form (cone, flat, pyramid).
- **Accumulation inverted** - if this option is enabled, chart image is shown inverted.

Example of accumulation chart

Sample data table:

Make	Sales2005
BMW	15000
Audi	14000
Volvo	9000

In this example we choose *Sales2005* as **Data Series** field, *Make* as **Label** field.



2.8.2.3 Area chart

An area chart is based on the line chart. The area between axis and line are emphasized with colors. Area charts are used to represent cumulated totals using numbers or percentages (stacked area charts in this case) over time.

Area chart may contain single series or multi series.

Chart settings:

- **Chart Scrolling** - this option allows you display a scrollable chart. Don't forget to define the number of points to show on the chart screen.

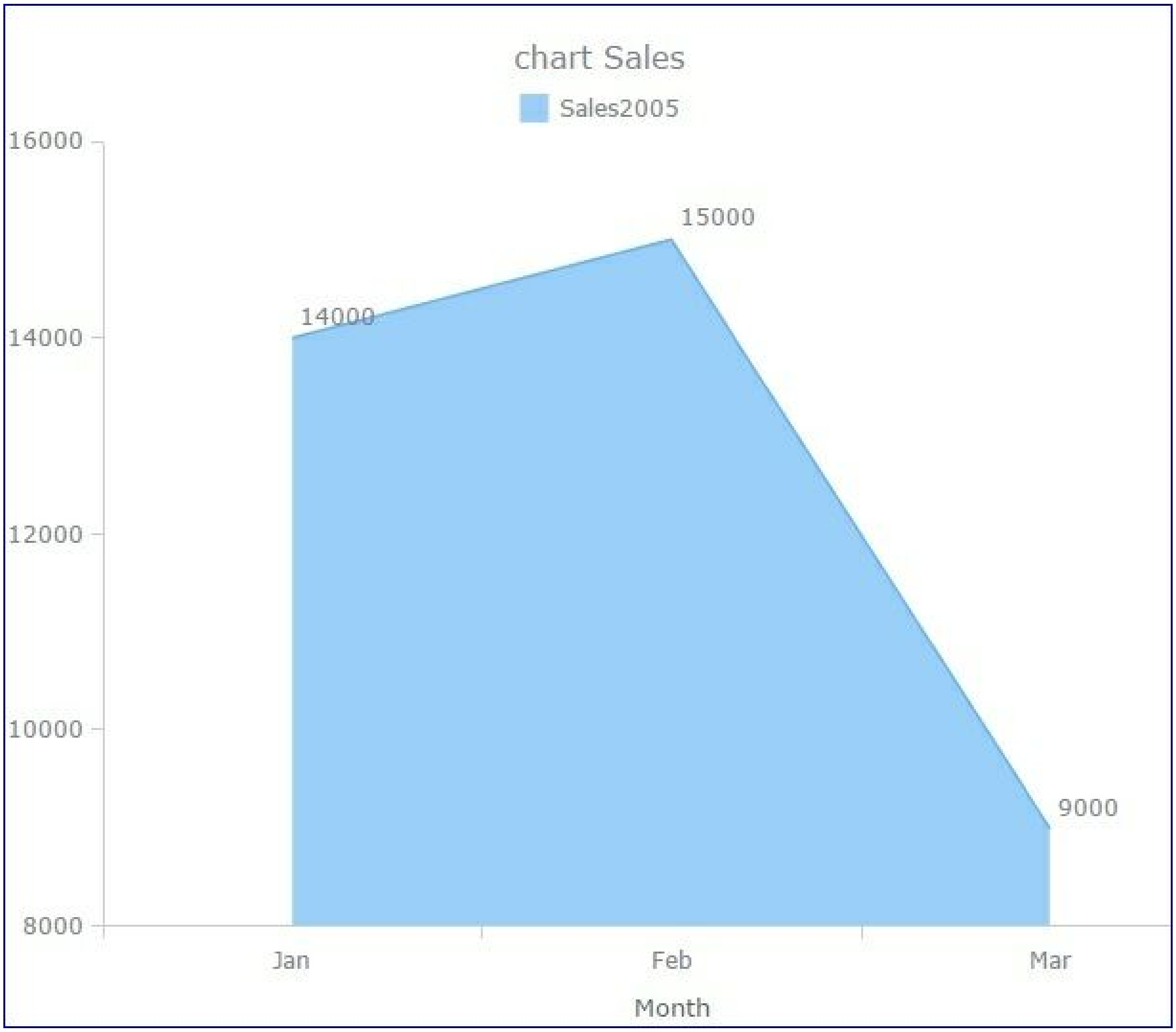
- **Chart stacked** - this option allows building stacked chart where a single bar on the chart shows data for more than one category of data. Stacked chart requires two or more Data series.

Example of single-series area chart

Sample data table:

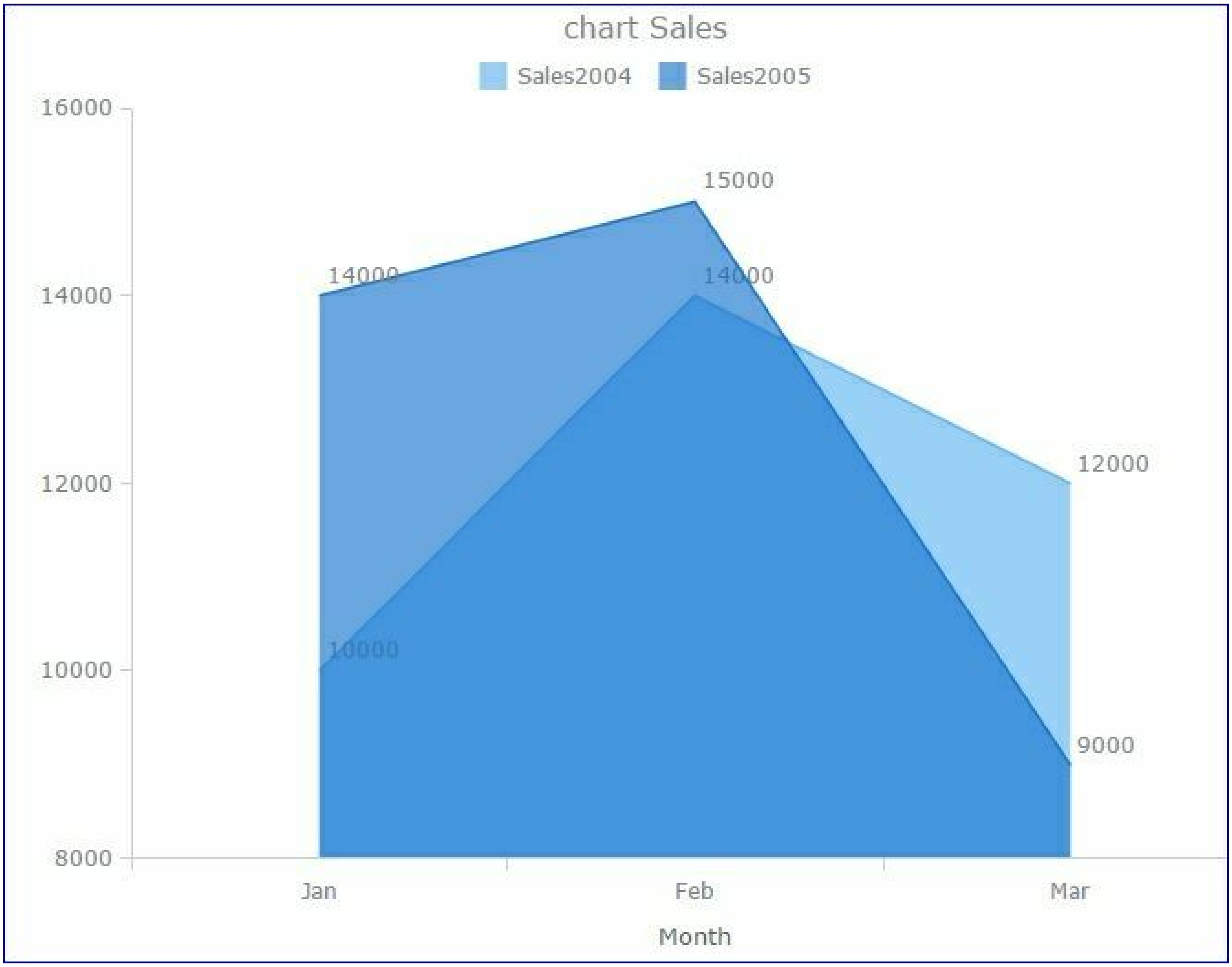
Month	Sales2004	Sales2005
Jan	10000	14000
Feb	14000	15000
Mar	12000	9000

In this example we choose *Sales2005* as **Data Series** field, *Month* as **Label** field.



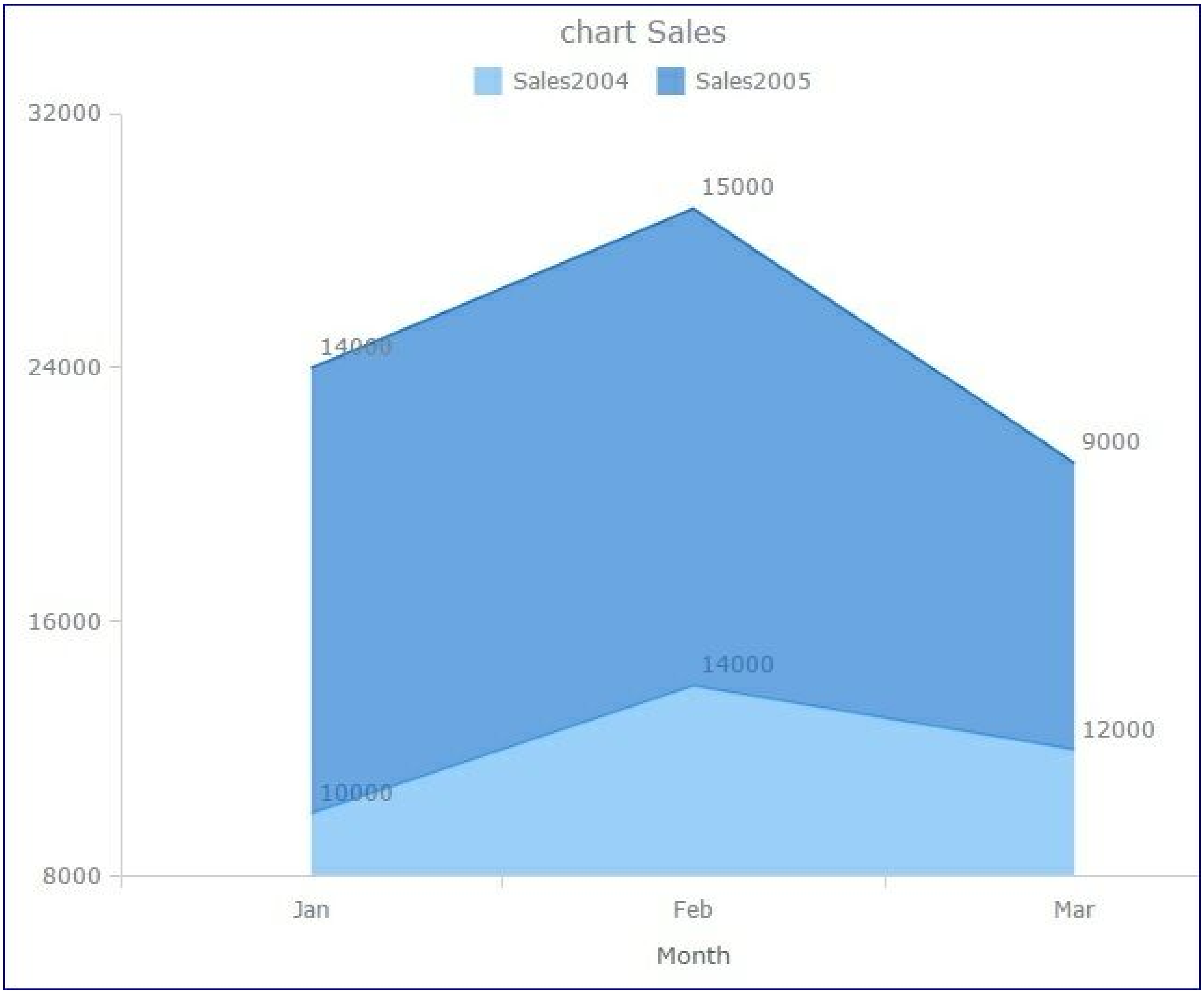
Example of multi-series area chart

In this example we choose *Sales2004* and *Sales2005* as **Data Series** fields, *Month* as **Label** field.



Example of multi-series stacked area chart

In this example we choose *Sales2004* and *Sales2005* as **Data Series** fields, *Month* as **Label** field.



2.8.2.4 Bubble chart

A Bubble chart is a variation of a Scatter chart in which the data points are replaced with bubbles. Bubble charts are often used to present financial data. Use a Bubble chart when you want specific values to be more visually represented in your chart by different bubble sizes.

Bubble chart need 3 values (x, y and size) to show bubbles. Depending on data model and the visualization purpose the bubble chart may contain single series or multi series.

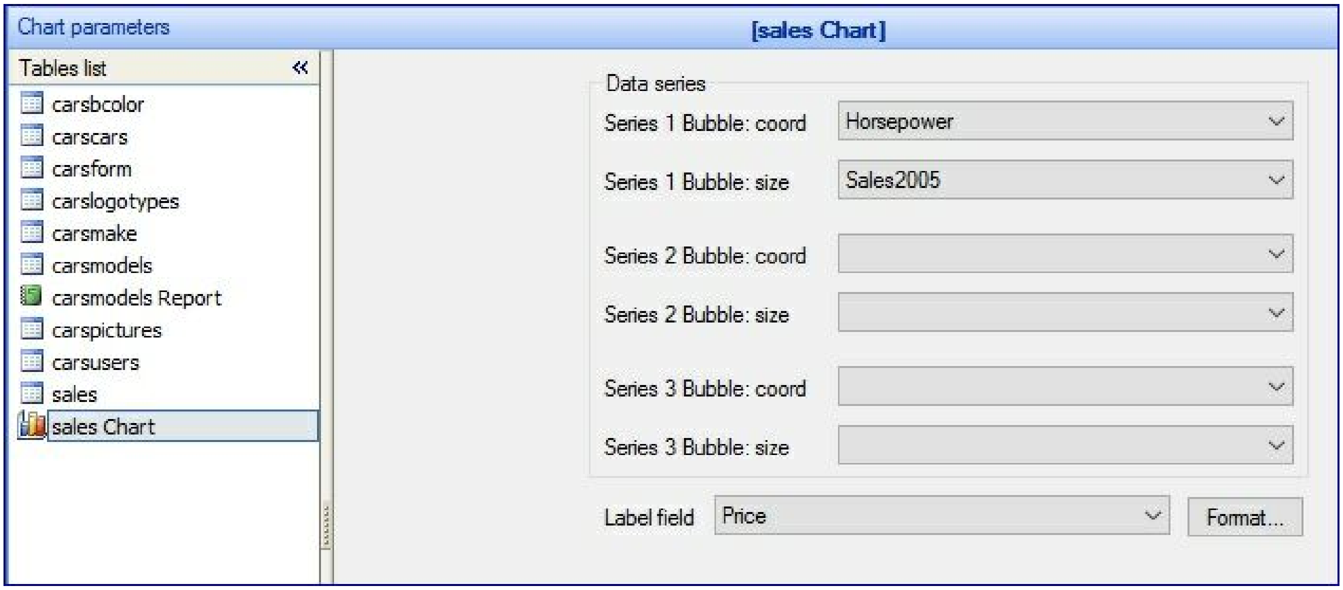


Chart parameters

[sales Chart]

Tables list

- carsbcolor
- carscars
- carsform
- carslogotypes
- carsmake
- carsmodels
- carsmodels Report
- carspictures
- carsusers
- sales
- sales Chart

Data series

Series 1 Bubble: coord	Horsepower
Series 1 Bubble: size	Sales2005
Series 2 Bubble: coord	
Series 2 Bubble: size	
Series 3 Bubble: coord	
Series 3 Bubble: size	
Label field	Price

Chart settings:

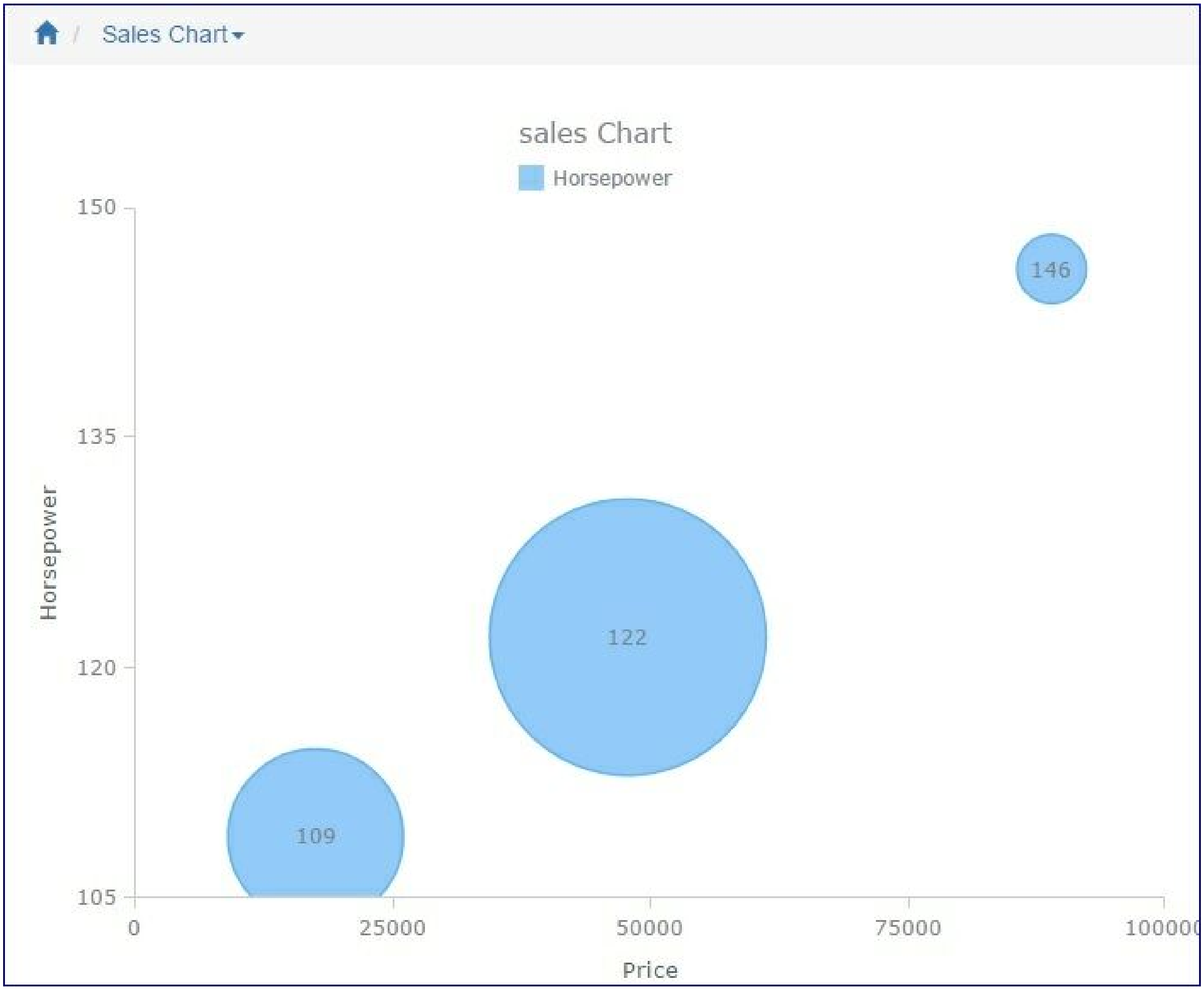
- **Chart 3D** - this option allows building 3D (three-dimensional) chart. If this option is disabled, 2D (two-dimensional) chart will be built.

Example of bubble chart

Sample data table:

Make	Horse power	Sales2005	Price
Audi	109	120000	17599
BMW	146	109500	88999
Volvo	122	130000	47899

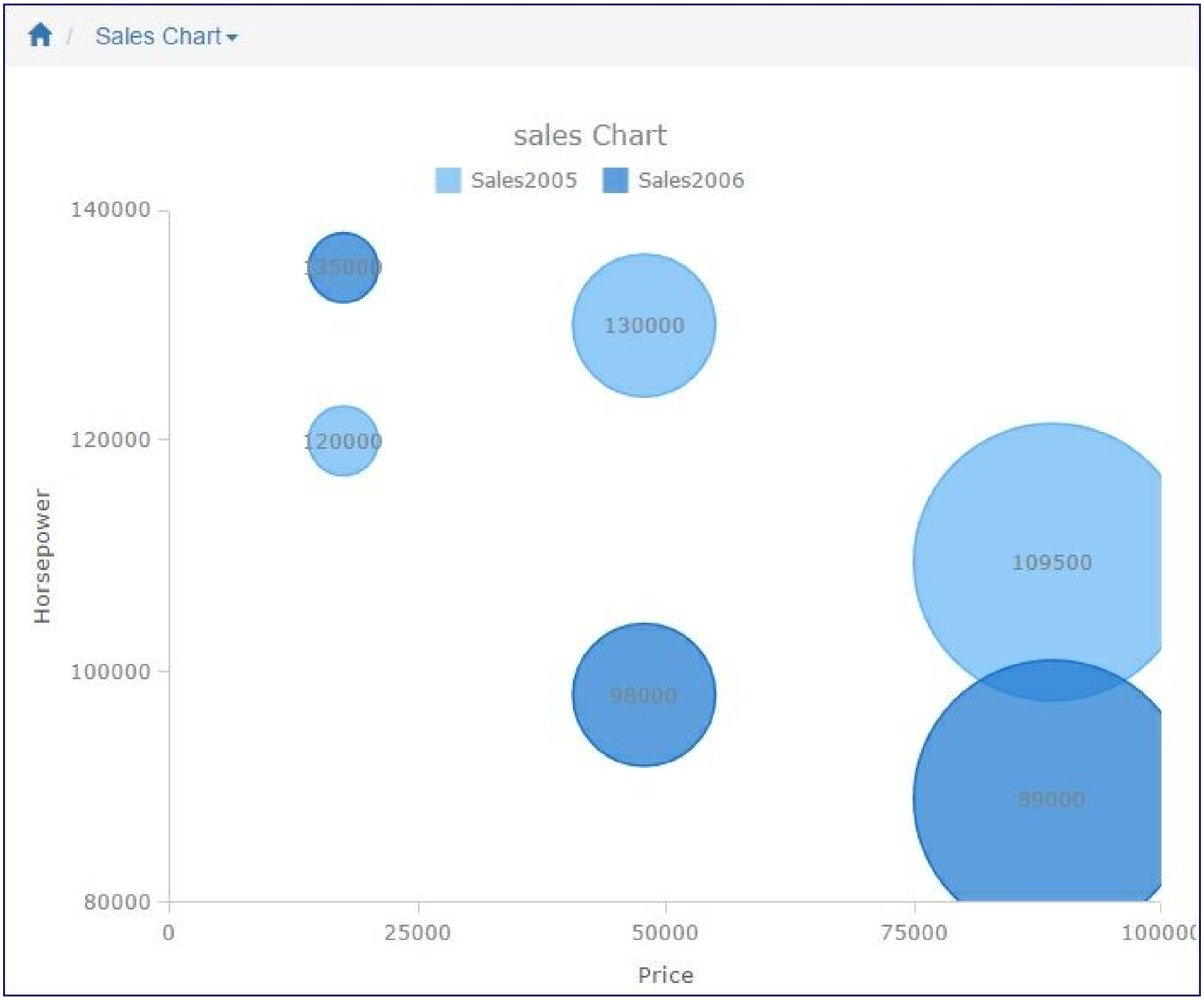
In this example we choose *Horsepower* as Y-coordination field, *Sales2005* as bubble size field, *Price* as **Label** field (X-coordination field).



Example of chart with two data series selected

Make	Horsepower	Sales 2005	Sales 2006	Price
Audi	109	120000	135000	17599
BMW	146	109500	89000	88999
Volvo	122	130000	98000	47899

In this example we choose *Sales2005* and *Sales2006* as Y-coordination fields, *Horsepower* as bubble size field, *Price* as **Label** field (X-coordination field).



2.8.2.5 Column/Bar charts

A column/bar chart is a chart with rectangular bars of lengths usually proportional to the magnitudes or frequencies of what they represent.

The column chart is vertically oriented bars. In column charts, categories are typically organized along the horizontal axis and values along the vertical axis.

The bar chart is horizontally oriented bars. In bar charts, categories are typically organized along the vertical axis and values along the horizontal axis. Consider using a bar chart when:

- The axis labels are long.
- The values that are shown are durations.

The column/bar chart may contain single series (one Data series field and label field selected - one for Y-Axis and another one for X-Axis) or multi series (two or more Data series fields selected).

Chart settings:

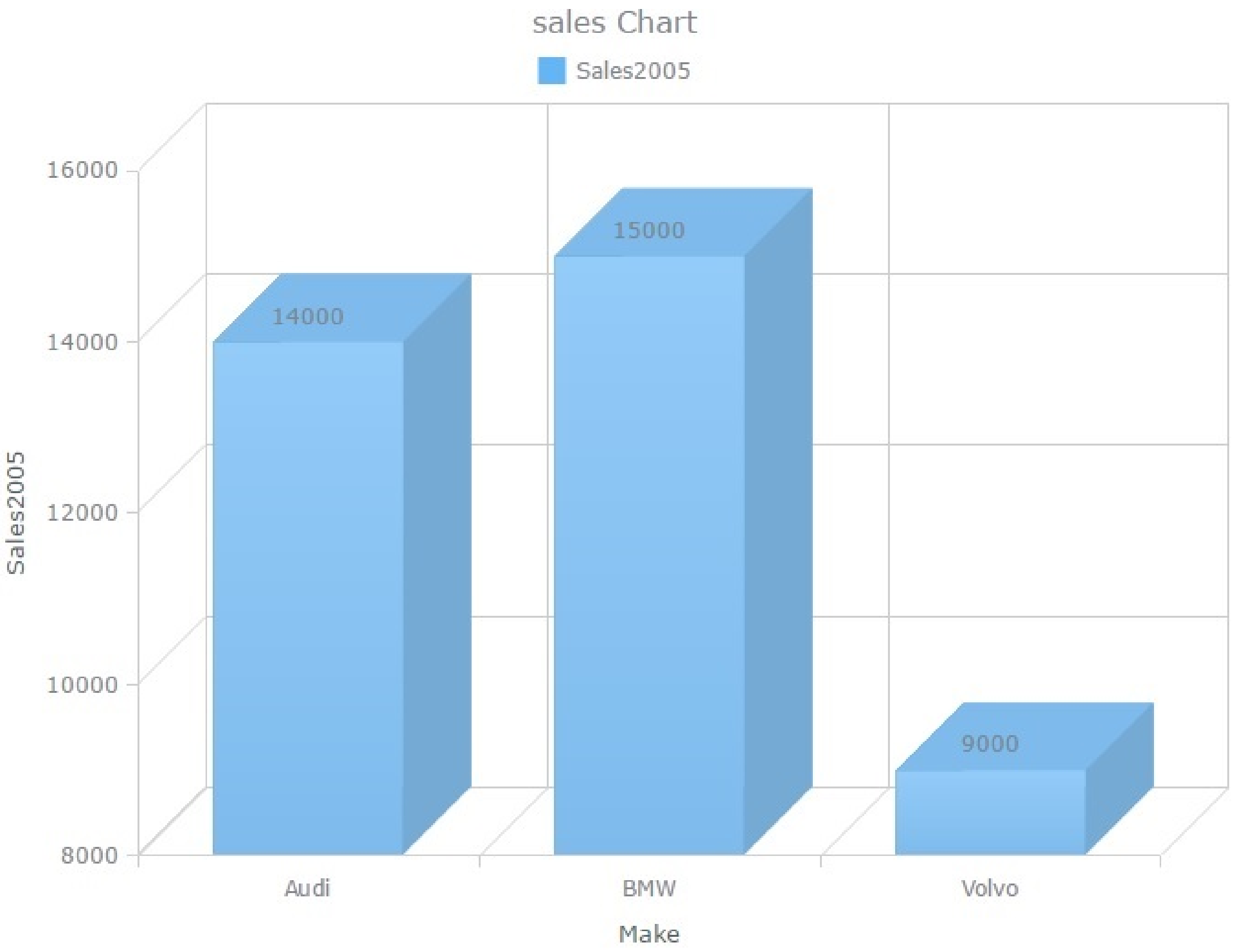
- **Chart 3D** - this option allows building 3D (three-dimensional) chart. If this option is disabled, 2D (two-dimensional) chart will be built.
- **Chart stacked** - this option allows building stacked chart where a single bar on the chart shows data for more than one category of data. Stacked chart requires two or more Data series.

Example of 3D bar chart

Sample data table:

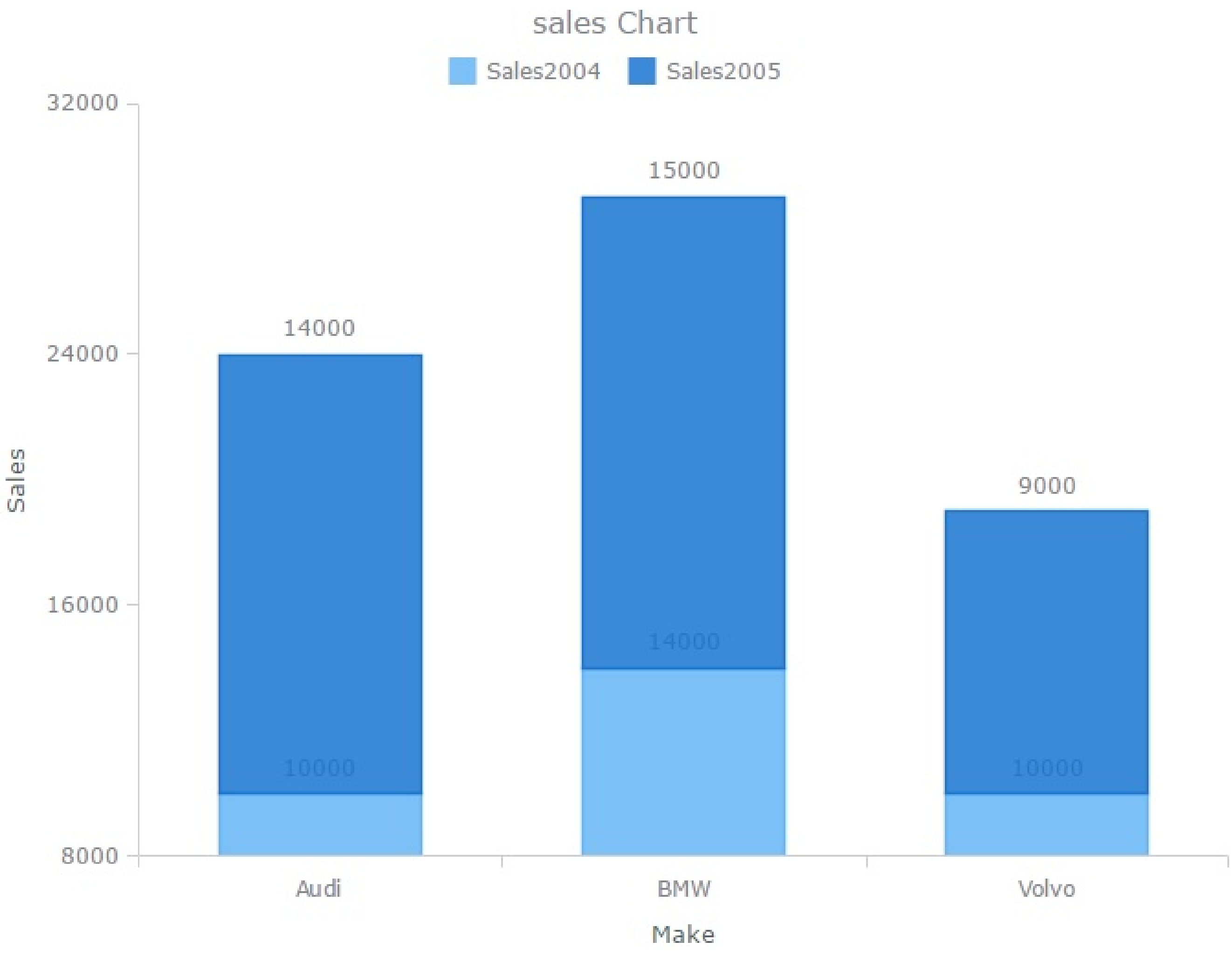
Make	**Sales2004**	**Sales2005**
Audi	10000	14000
BMW	14000	15000
Volvo	10000	9000

In this example we choose *Sales2005* as **Data Series** field, *Make* as **Label** field.



Example of stacked column chart

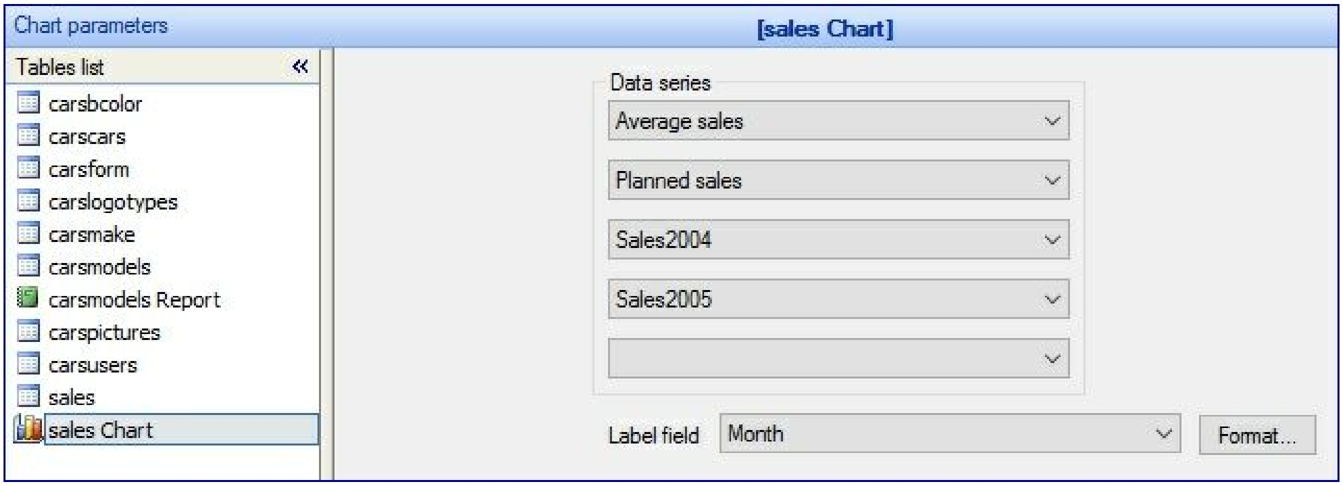
In this example we choose *Sales2004* and *Sales2005* as **Data Series** fields, *Make* as **Label** field.



2.8.2.6 Combined chart

A combined chart allows you to join several types of charts.

If you select one Data series field, you will get a Line chart. If you select two Data series fields, you will get Line-Area chart. If three or more Data series fields - Line-Area-Column chart.



Data series: Average sales, Planned sales, Sales2004, Sales2005

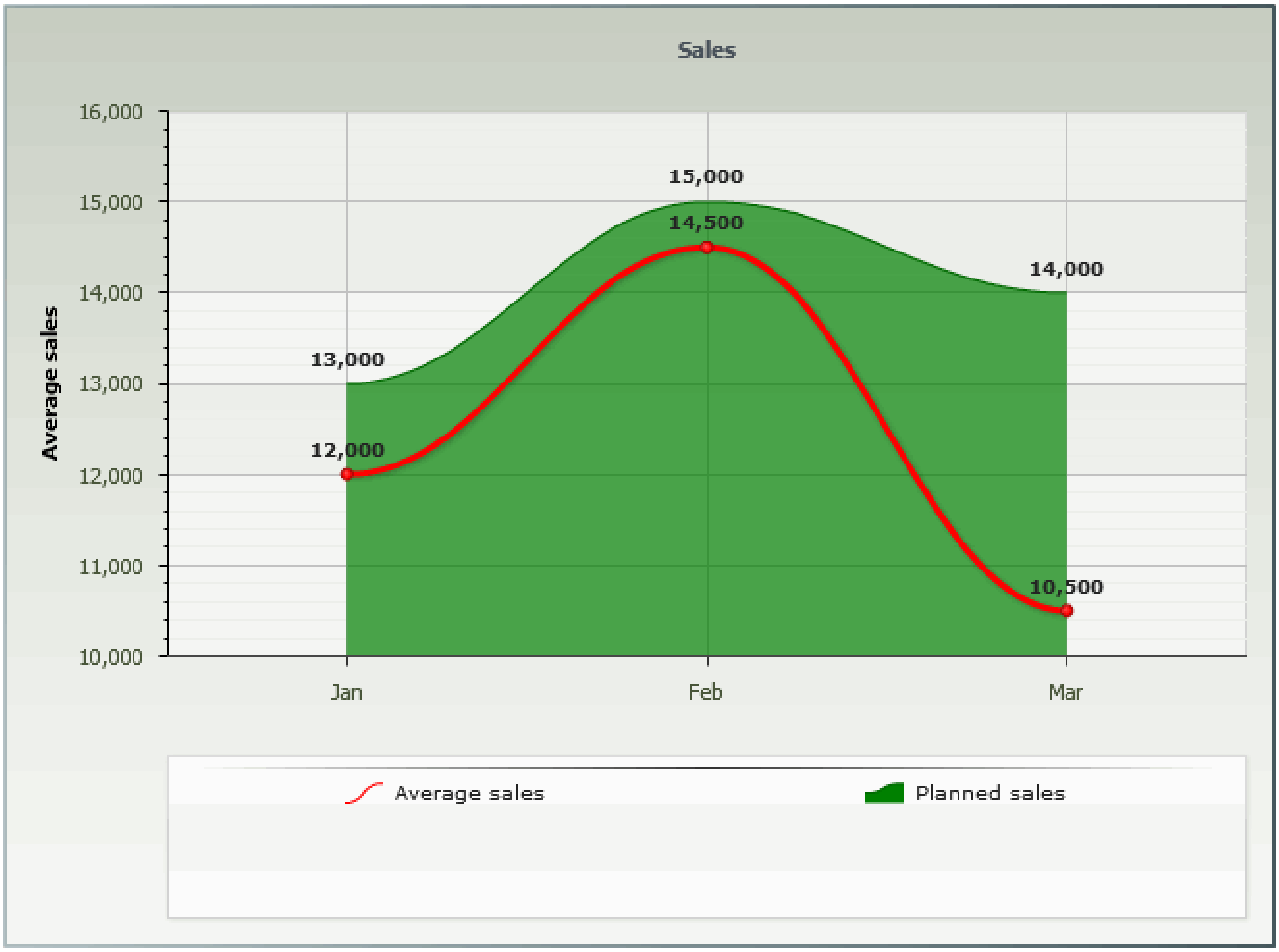
Label field: Month

Example of combined chart

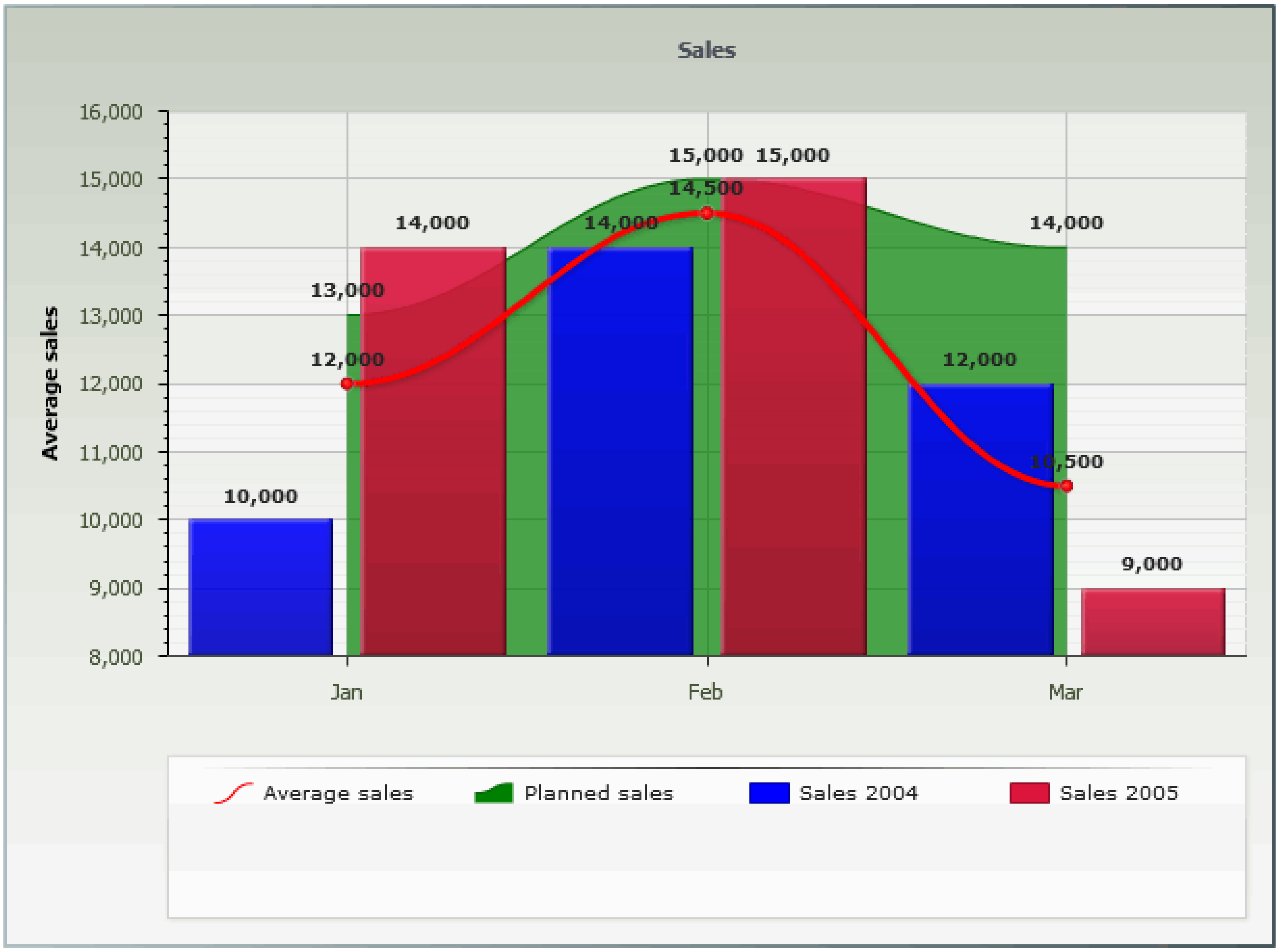
Sample data table:

Month	Average sales	Planned sales	Sales2004	Sales2005
Jan	12000	13000	10000	14000
Feb	14500	15000	14000	15000
Mar	10500	14000	12000	9000

In this example we choose *Average sales* and *Planned sales* as **Data Series** fields, *Month* as **Label** field.



In this example we choose *Average sales*, *Planned sales*, *Sales2004* and *Sales2005* as **Data Series** fields, *Month* as **Label** field.



2.8.2.7 Financial OHLC/Candlestick charts

An open-high-low-close chart (also known as OHLC chart) is a type of chart typically used to illustrate movements in the price of a financial instrument over time. Each vertical line on the chart shows the price range (the highest and lowest prices) over one unit of time, e.g. one day or one hour. Tick marks project from each side of the line indicating the opening price (e.g. for a daily OHLC chart this would be the starting price for that day) on the left, and the closing price for that time period on the right.

The Japanese Candlestick chart is another way of displaying market price data, with the opening and closing prices defining a rectangle within the range for each time unit. The rectangles have different colors depending on whether prices rose or fell in that period.

Both charts show the exact same data, i.e. the opening, high, low, and closing prices during a particular time frame. Some traders find the candlestick chart easier to read.

OHLC/Candlestick chart may contain single series or multi series. These charts use four values, so you need to pass opening, high, low and closing price values. Also you need to select *Label* field.

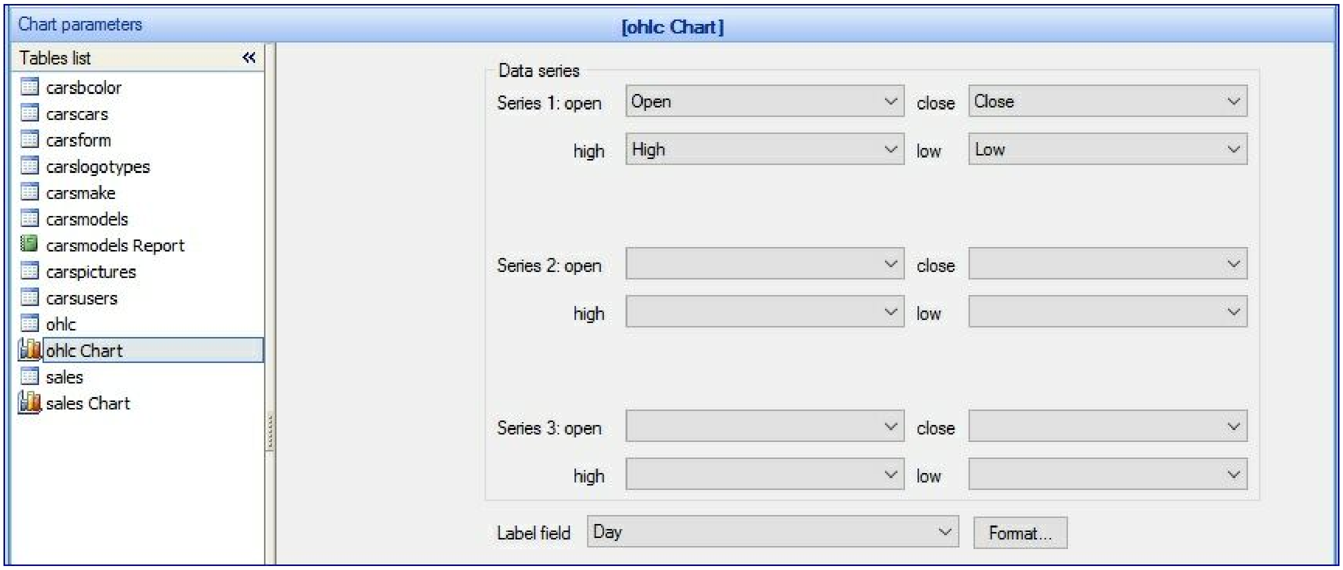


Chart parameters [ohlc Chart]

Tables list: carsbcolor, carscars, carsform, carslogotypes, carsmake, carsmodels, carsmodels Report, carspictures, carsusers, ohlc, ohlc Chart, sales, sales Chart

Data series

Series 1: open: Open; close: Close; high: High; low: Low

Series 2: open; close; high; low

Series 3: open; close; high; low

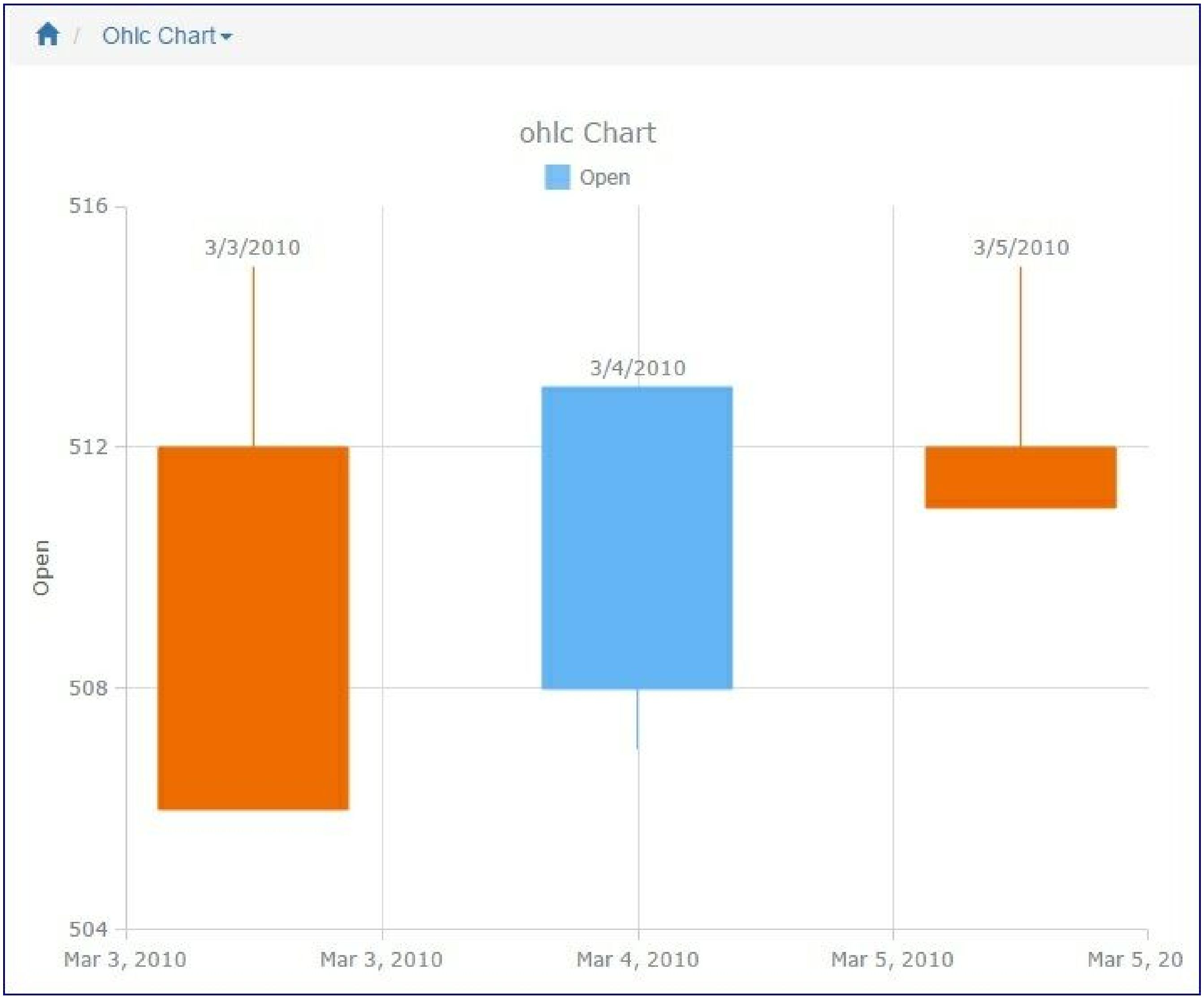
Label field: Day

Example of Candlestick chart

Sample data table:

Day	Open	High	Low	Close
03-Mar-10	512.00	515.00	506.10	506.00
04-Mar-10	508.00	513.00	507.00	513.00
05-Mar-10	512.00	515.00	511.00	511.00

In this example we choose *Open*, *High*, *Low*, *Close* as **Data Series 1** fields, *Day* as **Label** field.



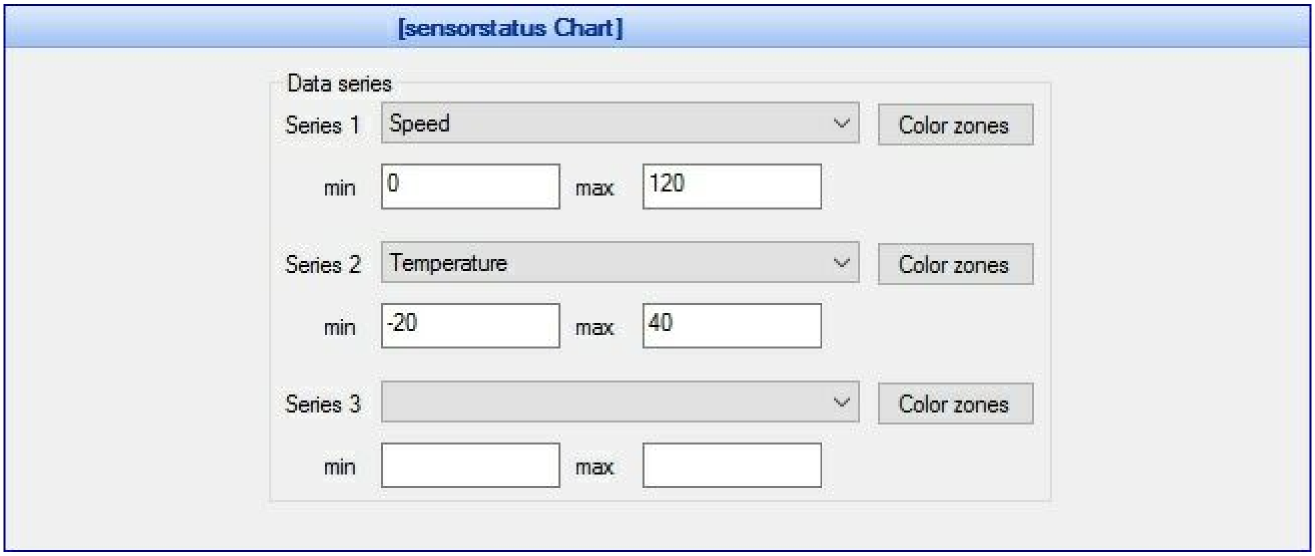
2.8.2.8 Gauge chart

A Gauge chart represents a value on a graduated scale or dial. The Gauge chart can be of two types: Circular Gauge and Linear Gauge.

The circular Gauge looks like gauges on a car dashboard. It consists of radial scale, holding your data range, pointer, and might support different color ranges for classifying your data.

The linear Gauge is a wide line, that can hold much information and possess a variety features. It can be either vertical or horizontal.

You can display one or several Gauges on a chart by choosing one or several Data series. Also you can define minimum and maximum values.



[sensorstatus Chart]

Data series

Series 1: Speed

min: 0 max: 120

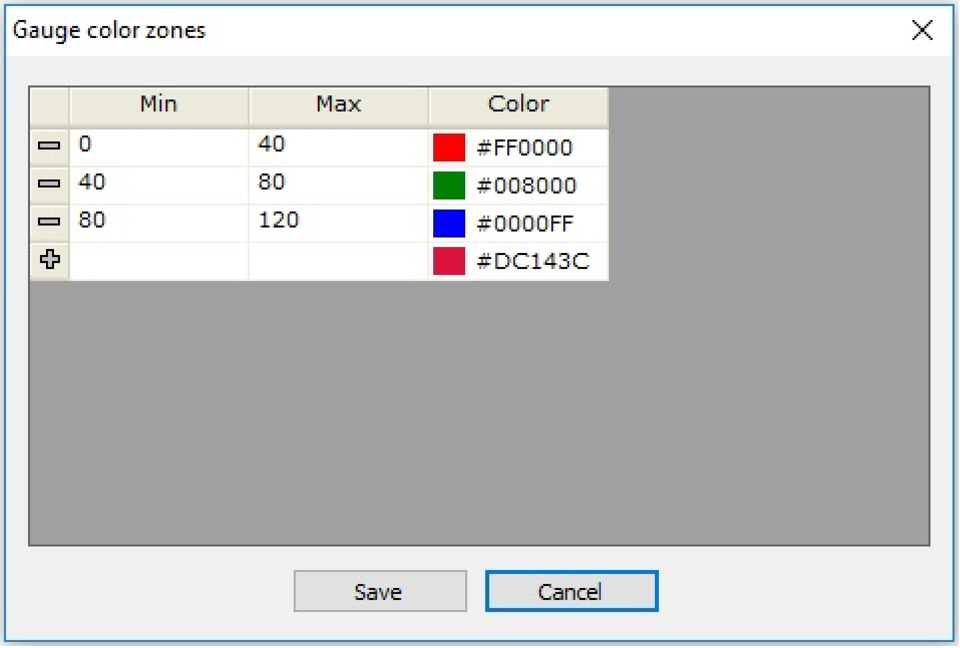
Series 2: Temperature

min: -20 max: 40

Series 3:

min: max:

Color zones are used to decide whether your data is in preset limits. You can set the desired number of different colors depending on the number of your conditions.



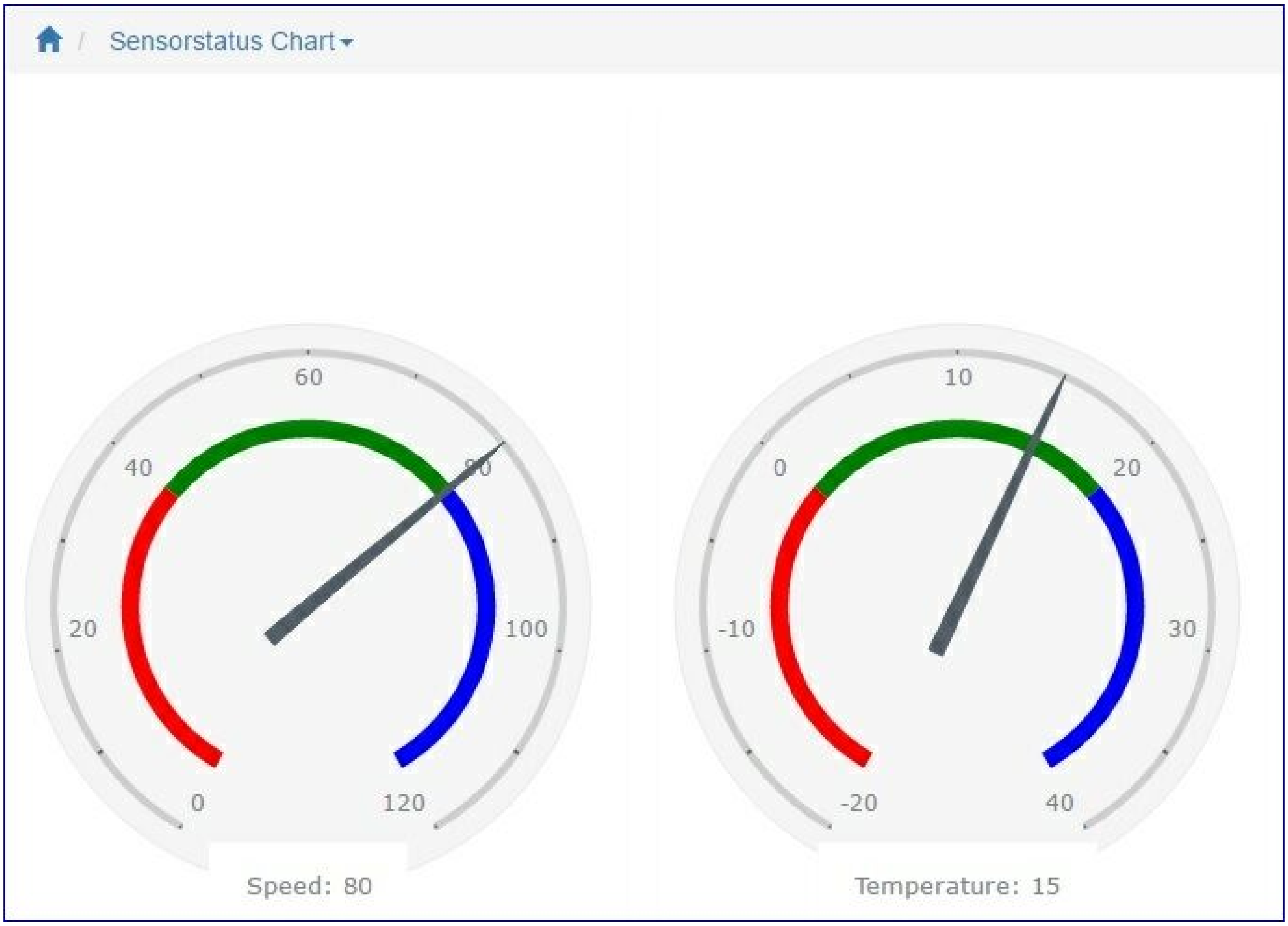
Gauge color zones

	Min	Max	Color
−	0	40	🟥 #FF0000
−	40	80	🟩 #008000
−	80	120	🟦 #0000FF
+			🟥 #DC143C

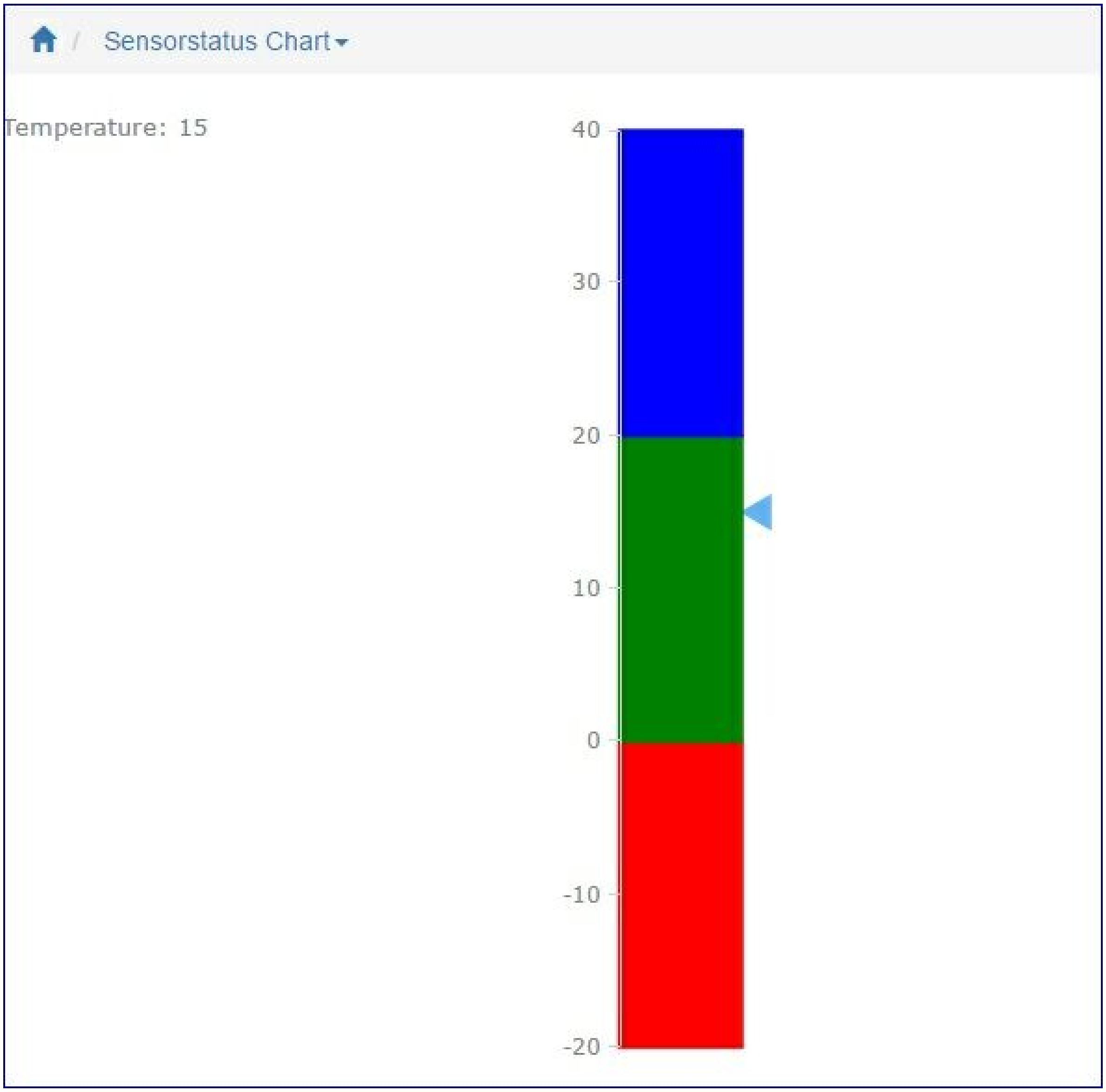
Chart settings:

- **Gauge style** - this option defines the gauge appearance (circle, horizontal linear, vertical linear).

Example of circular Gauge chart with two Data series selected



Example of vertical linear Gauge chart



2.8.2.9 Line chart

A line chart displays information as a series of data points connected by line segments. A line chart is often used to visualize a trend in data over intervals of time. In a line chart, category data is distributed evenly along the horizontal axis, and all value data is distributed evenly along the vertical axis.

Line chart may contain single series or multi series.

Chart settings:

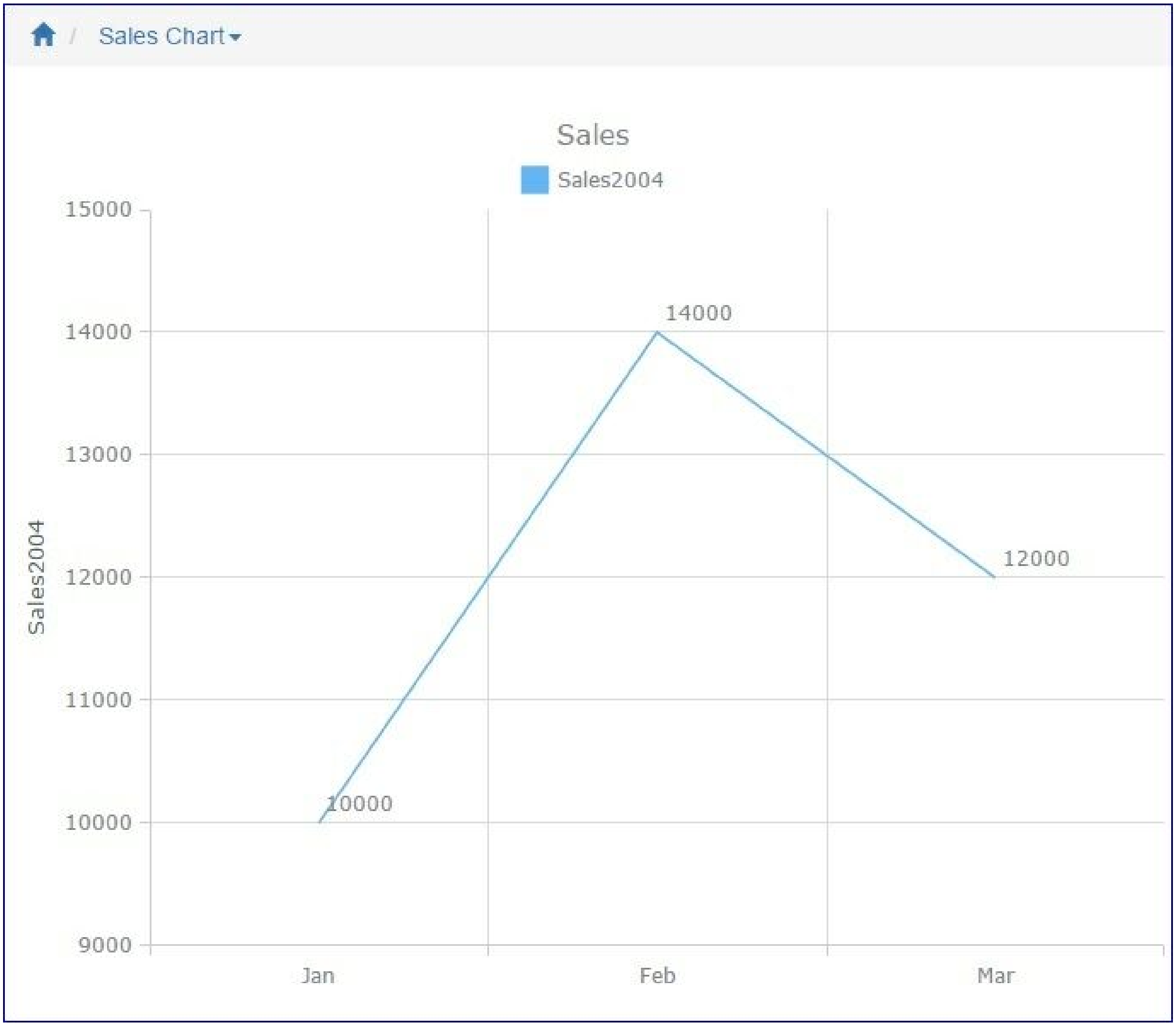
- **Line style** - this option defines the line segments style (normal, spline, step).

Example of single-series line chart

Sample data table:

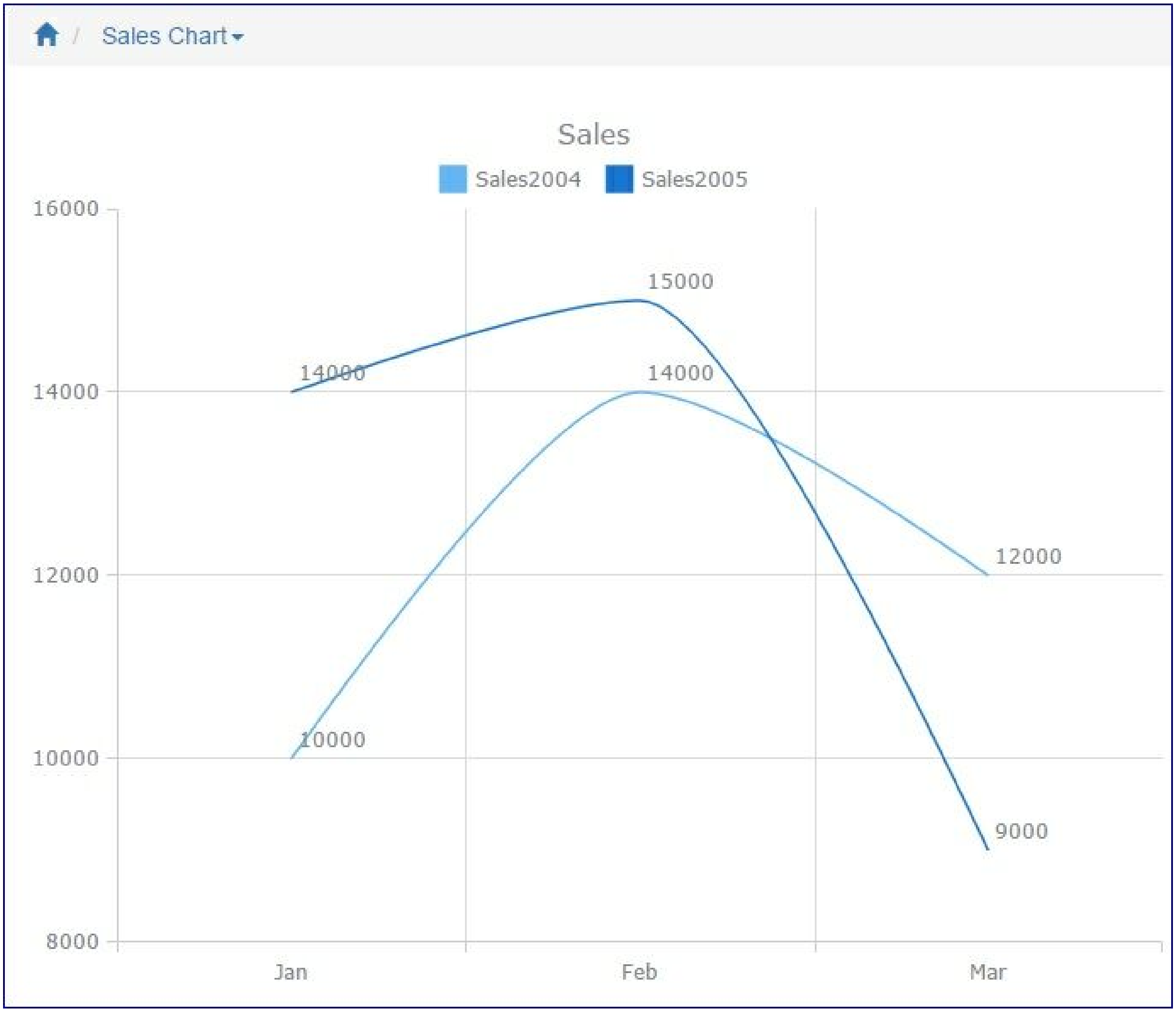
Month	Sales2004	Sales2005
Jan	10000	14000
Feb	14000	15000
Mar	12000	9000

In this example we choose *Sales2004* as **Data Series** field, *Month* as **Label** field.



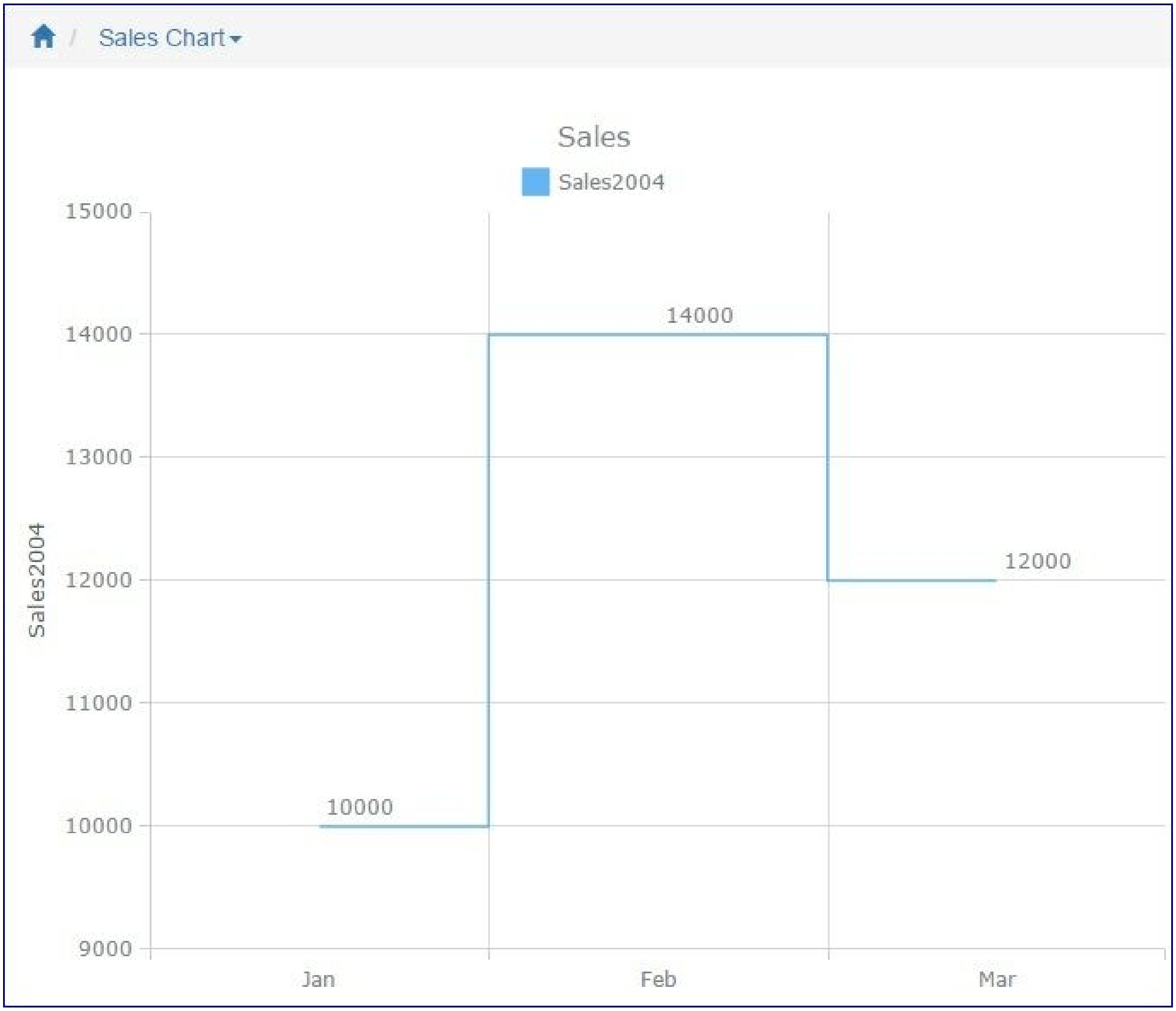
Example of multi-series spline chart

In this example we choose *Sales2004* and *Sales2005* as **Data Series** fields, *Month* as **Label** field.



Example of step line chart

In this example we choose *Sales2004* as **Data Series** field, *Month* as **Label** field.



2.8.2.10 Pie/Doughnut charts

A pie chart is a circular chart divided into sectors, illustrating percents. A doughnut chart is functionally identical to a pie chart, with the exception of a blank center.

You can create single-series and multi-series pie and doughnut charts by choosing one or several Data series fields.

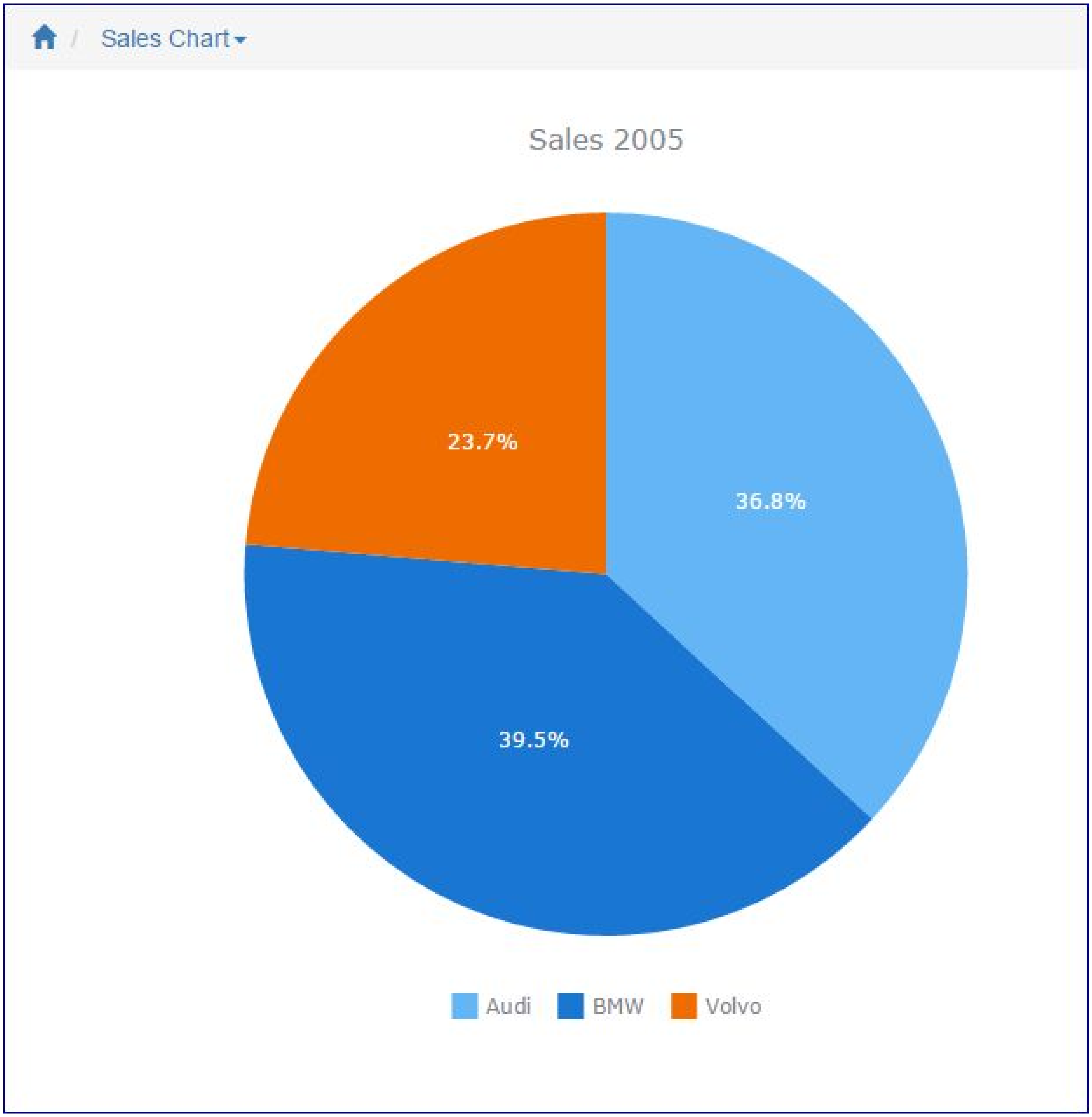
Example of pie chart

Sample data table:

Make	Sales2005

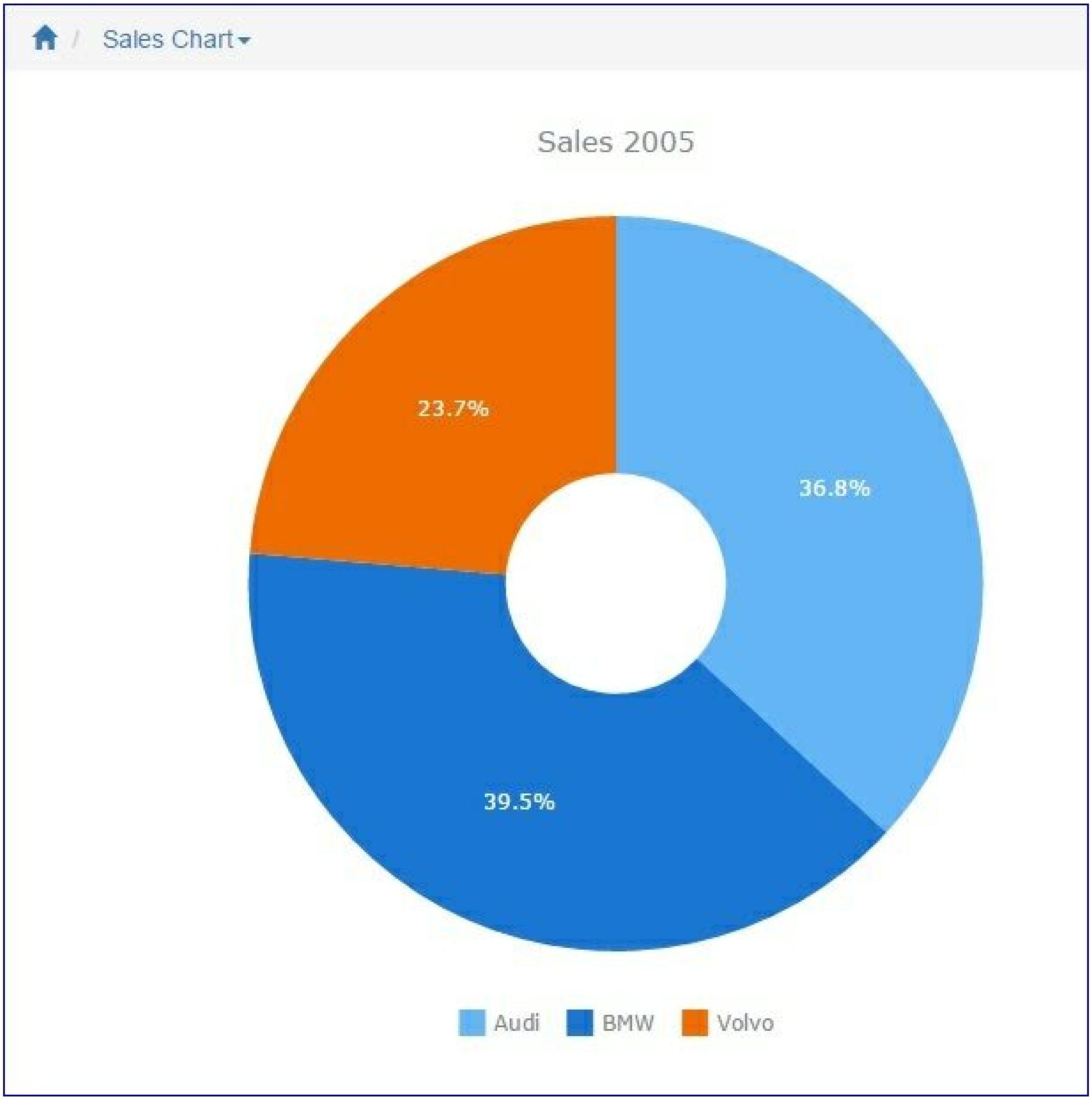
Audi	14000
BMW	15000
Volvo	9000

In this example we choose *Sales2005* as **Data Series** field, *Make* as **Label** field.



Example of doughnut chart

In this example we choose *Sales2005* as **Data Series** field, *Make* as **Label** field.

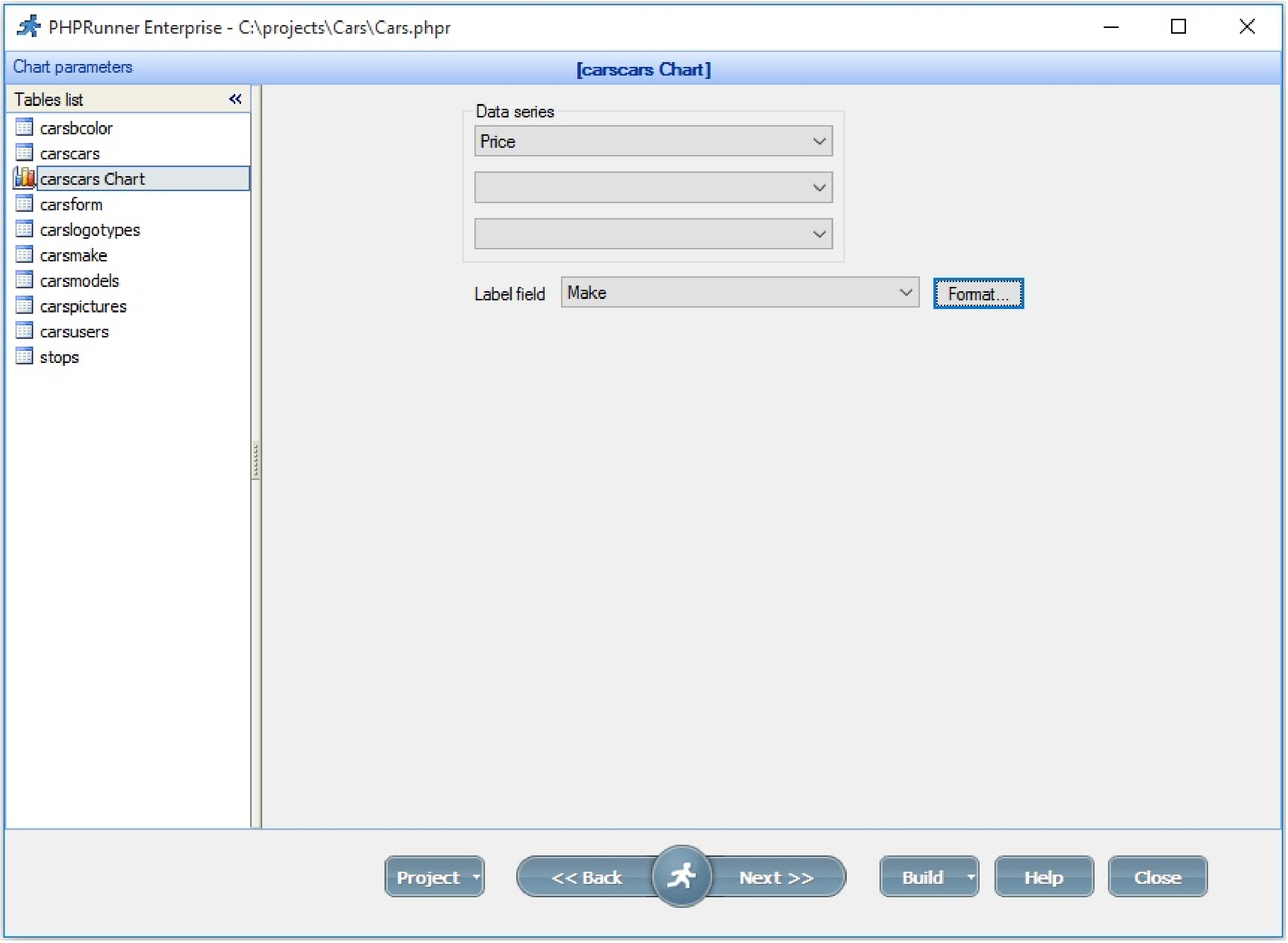


2.8.3 Chart parameters

On the **Chart parameters** page you can choose Data series fields (fields with data) and Label field (field with data labels).

You can add unlimited number of data series. Additional Data series dropdown list boxes are added automatically once you used available ones.

For more information about choosing data series for certain chart type, see Chart types.



Data series: Price

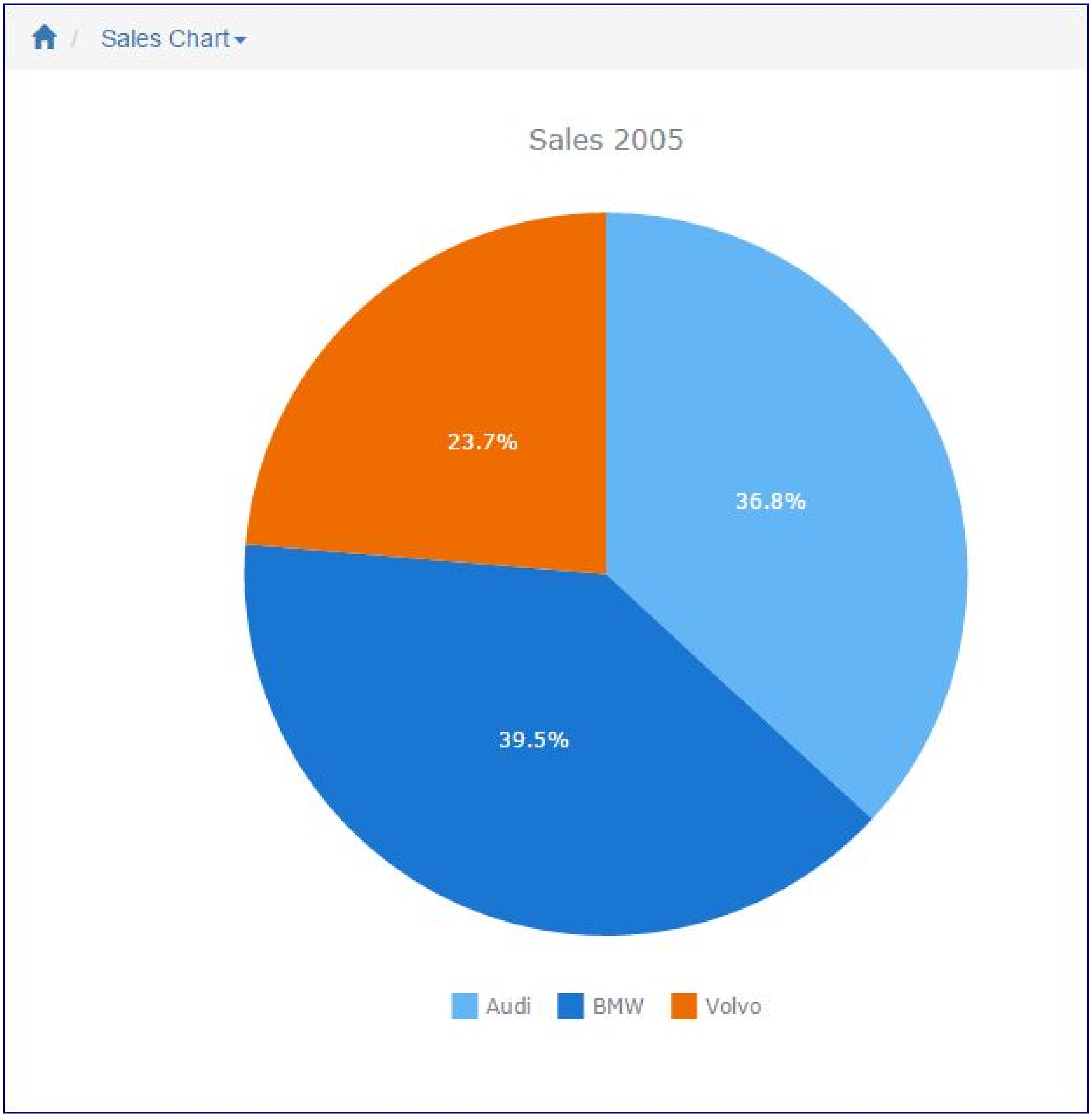
Label field: Make

Note: only numeric fields can be chosen as a Data series. Therefore only numeric fields are available for selection in Data series dropdown list box.

Here is a sample data table:

Make	Sales2005
Audi	14000
BMW	15000
Volvo	9000

In this example we choose *Sales2005* as **Data Series** field, *Make* as **Label** field.

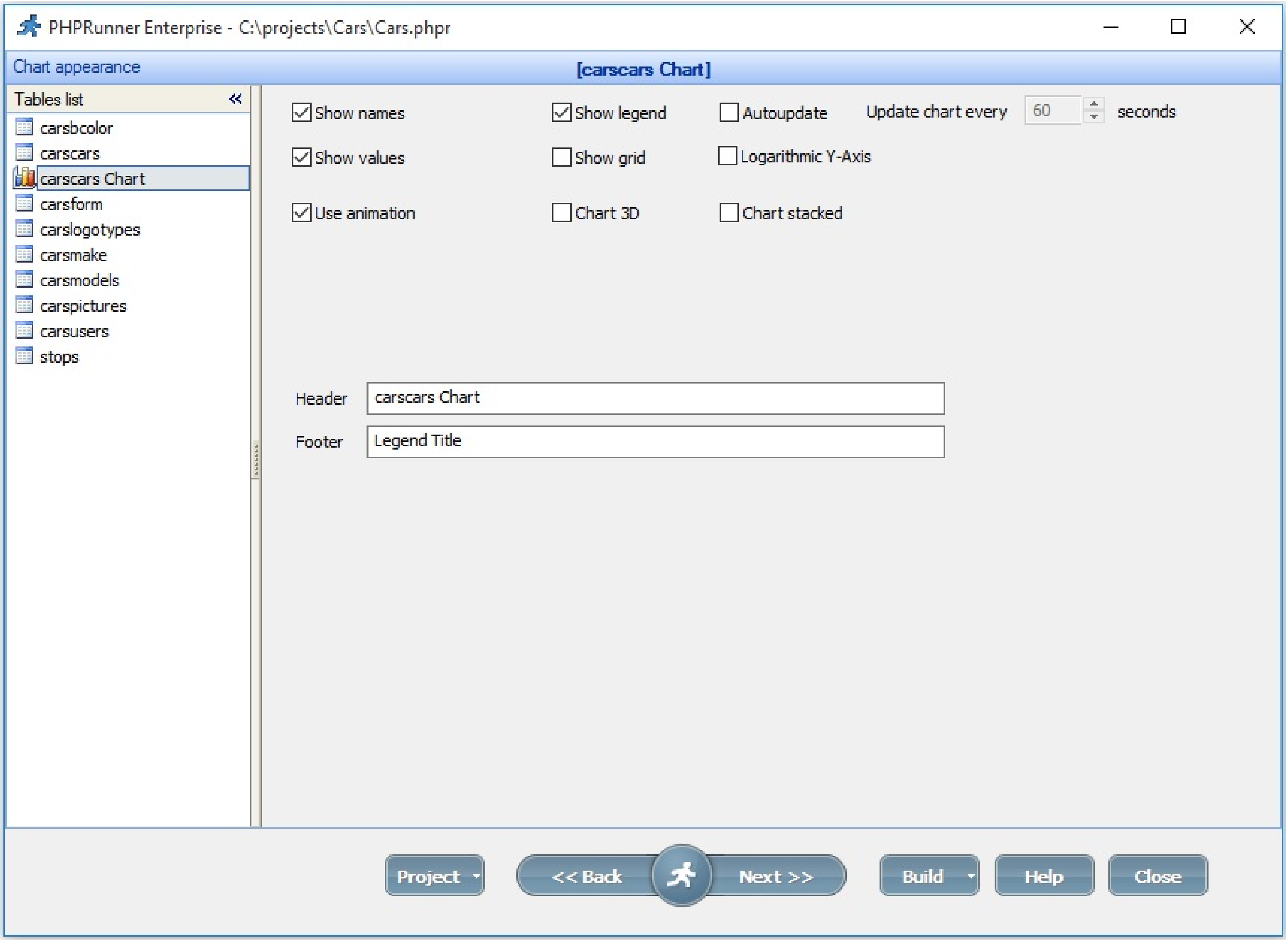


Sometimes the data in your database needs to be processed before it is used in the chart. For more information, see Using SQL to shape chart data.

To further customize chart appearance like setting colors, fonts or changing chart title use ChartModify event.

2.8.4 Chart appearance

On the **Chart Appearance** page you can control how your chart is displayed on the web page.



☑ Show names ☑ Show legend ☐ Autoupdate Update chart every 60 seconds

☑ Show values ☐ Show grid ☐ Logarithmic Y-Axis

☑ Use animation ☐ Chart 3D ☐ Chart stacked

Header carscars Chart

Footer Legend Title

Use the **Autoupdate** check box to enable chart auto-refresh by specified time interval.

The **Use animation** check box enables the chart animation while opening a chart.

If you select one of the 2D charts (e.g. 2D Column chart), additional options will be available. Use **Chart 3D** option to display data columns as the 3D images. Use **Chart stacked** option to display stacking chart where chart elements are stacked on top of each other. The **Chart scrolling** option allows you display a scrollable chart. Don't forget to define the number of bars to show on the chart screen.

Use the **Logarithmic Y-Axis** option to convert a linear value axis to a logarithmic value axis. If you have several data series on the chart, you can use the **Multiple Y-Axes** option to position each data series relative to its own Y axis.

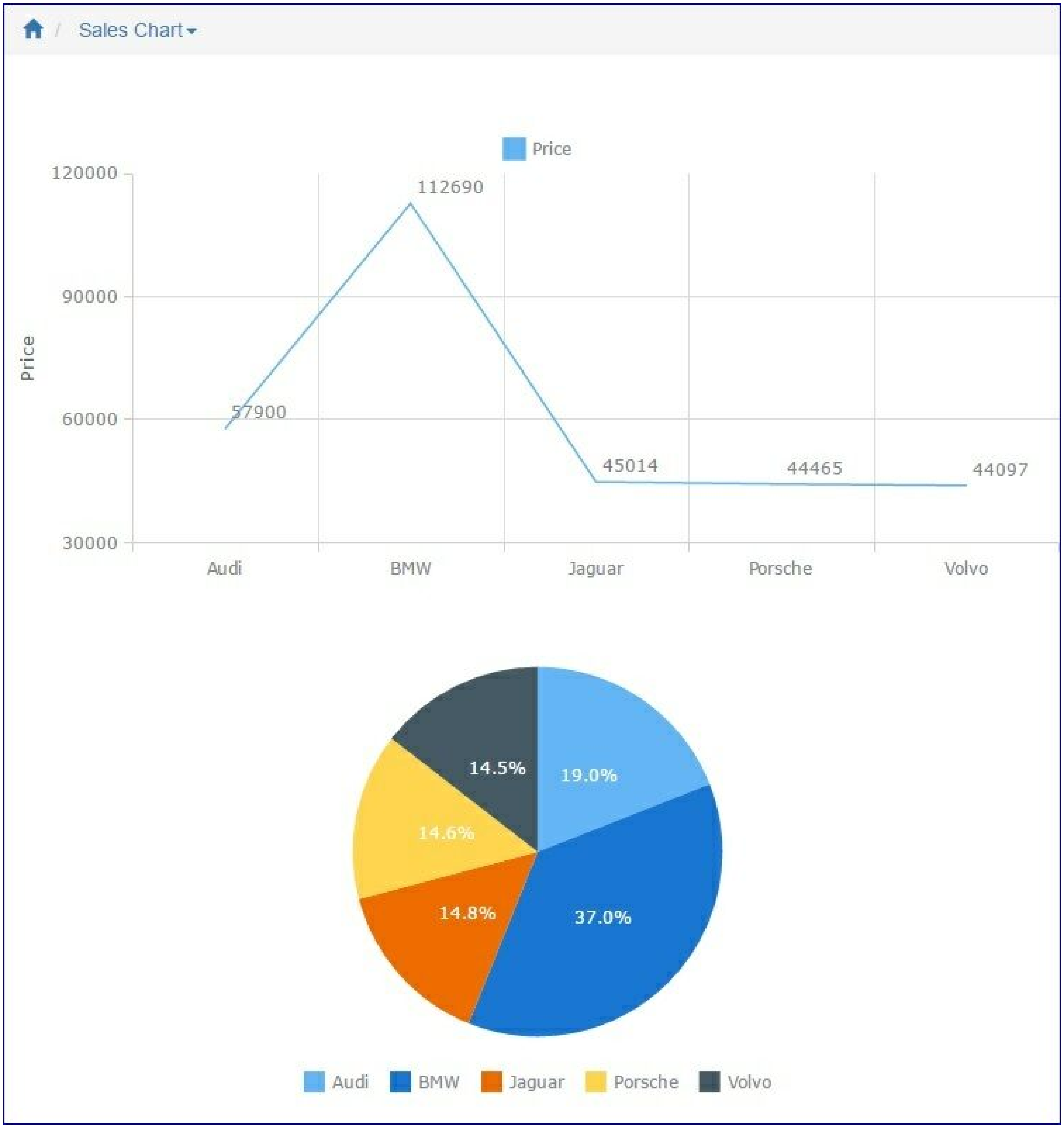
For more information about settings for certain chart type, see Chart types.

Y-axis label input box will only appear if you have multiple data series.

Notes:

1. Chart can be resized on the **Visual Editor** page.

2. In Visual Editor you can copy and paste chart to any other page (report/list/view/another chart). It can be useful to build a dashboard with several charts on it.



To further customize chart appearance like setting colors, fonts or changing chart title use ChartModify event.

2.8.5 Using SQL to shape chart data

Charts are all about visualizing your data. Charts need data fields that store numeric values however using aggregate functions you can build charts on any data.

Consider the following Orders table:

Customer	Country	Total
Andrew Peters	USA	$250
Katie Bradshow	Australia	$85
Jeff Simpson	USA	$150
Arnold Matteus	Germany	$120
Arnold Matteus	Germany	$160
Jeff Montgomery	GB	$150
Andrew Peters	USA	$65
Jeff Simpson	USA	$95
Luke Sohu	France	$40
Jeff Montgomery	GB	$120

Example 1: Total Sales per country

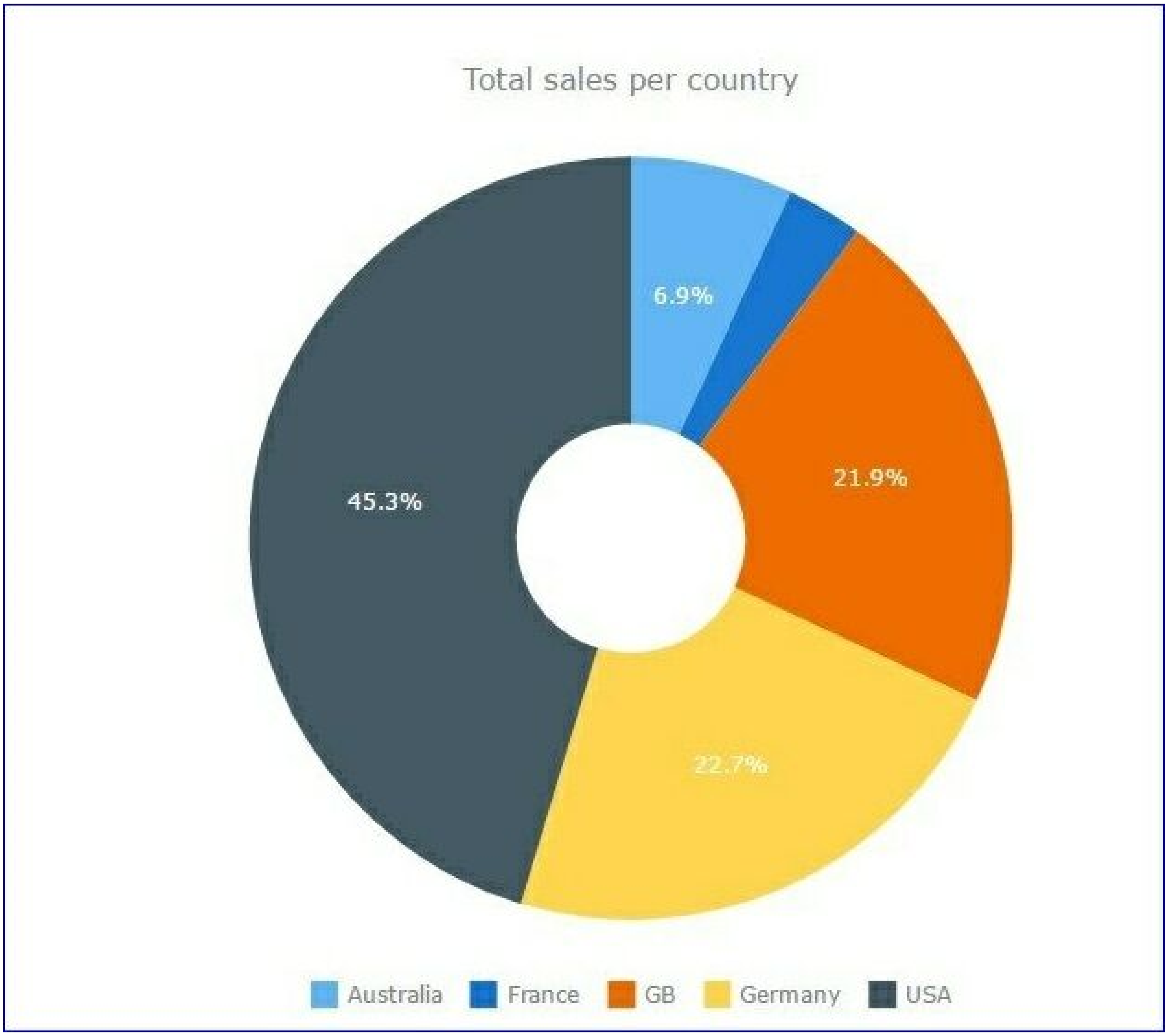
SQL query:

```
select Country, sum(total) as STotal
from Orders
group by country
order by 2 desc
```

Results:

Country	Stotal
USA	$560
Germany	$280
GB	$270
Australia	$85
France	$40

Chart:



Example 2: Number of orders per country

SQL query:

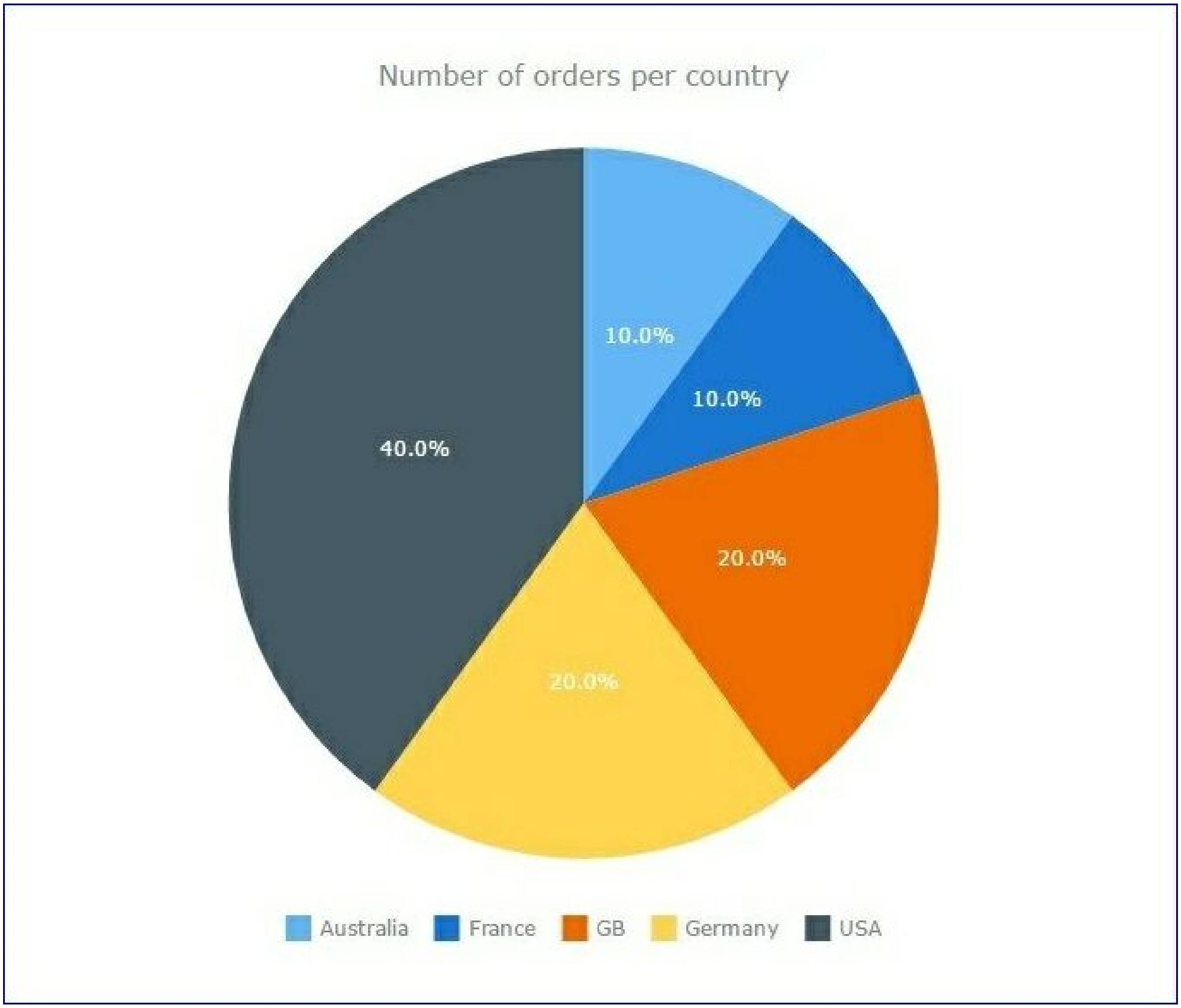
```
select Country, count(total) as CTotal
from Orders
group by country
order by 2 desc
```

Results:

Country	CTotal
USA	4

Germany	2
GB	2
Australia	1
France	1

Chart:



Example 3: Shaping your data more complex way

This example shows how to use GROUP BY in conjunction with INNER JOIN. For example we have the following data and like to display a diagram that illustrates how many flags each client has.

clientid	flag
1001	Green
1001	Green
1001	Green
1001	Green
1001	Amber
1001	Amber
1001	Red
1002	Green
1002	Amber
1002	Amber
1002	Amber
1002	Red
1003	Green
1003	Amber
1003	Red

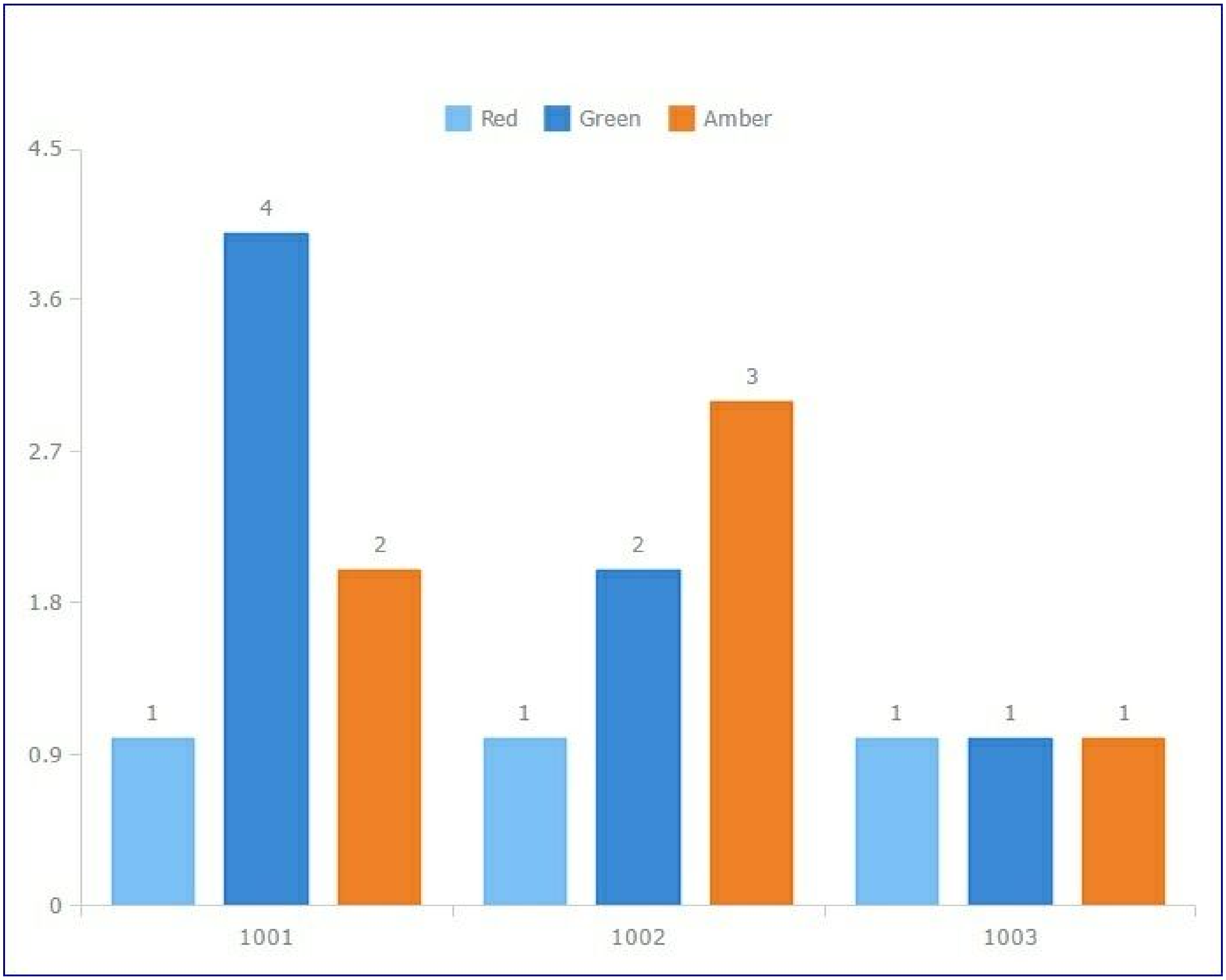
To shape our data we use the following SQL query:

```
select a.clientid, a.green,b.red, c.amber
from (select count(flag) as green, clientid from sensorstatus
where flag='Green' group by clientid) a
inner join (select count(flag) as red, clientid from sensorstatus
where flag='Red' group by clientid) b on a.clientid=b.clientid
inner join (select count(flag) as amber, clientid from sensorstatus
where flag='Amber' group by clientid) c on a.clientid=c.clientid
```

which gives us the following results:

clientid	green	red	amber
1001	4	1	2
1002	2	1	3

1003	1	1	1



Related info:

- Interactive SQL tutorial

2.9 Reports

2.9.1 Creating report and setting parameters

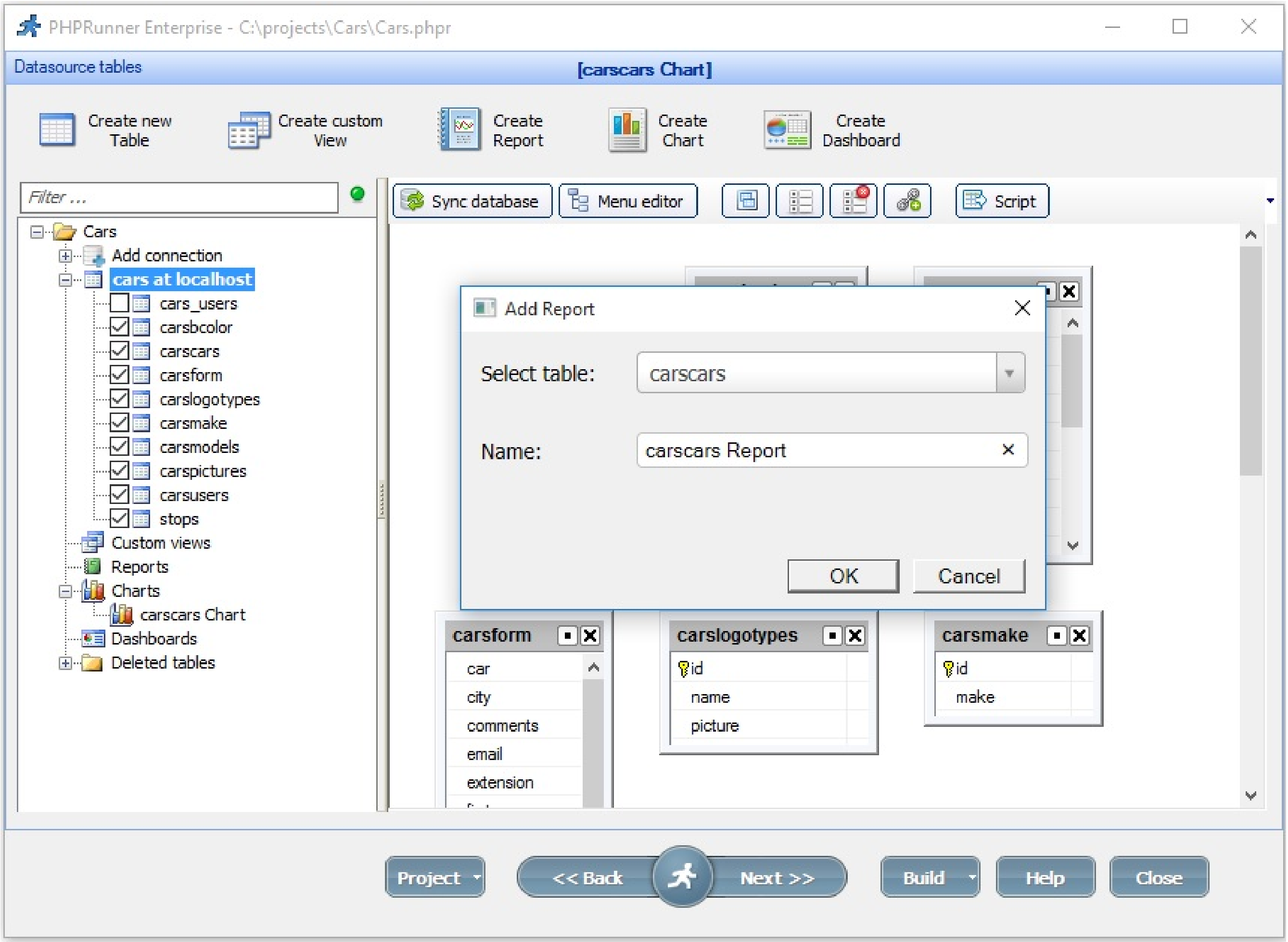
Quick jump

Grouping fields in report

To create a report:

1. Proceed to the **Datasource tables** page and click **Create Report**.

2. Select datasource table and report name. Click **OK.**



Select table: carscars

Name: carscars Report

Note that you can create a copy of existing report (right-click the report and select **Copy**).

On the next several pages (use **Next** button to navigate) you can:

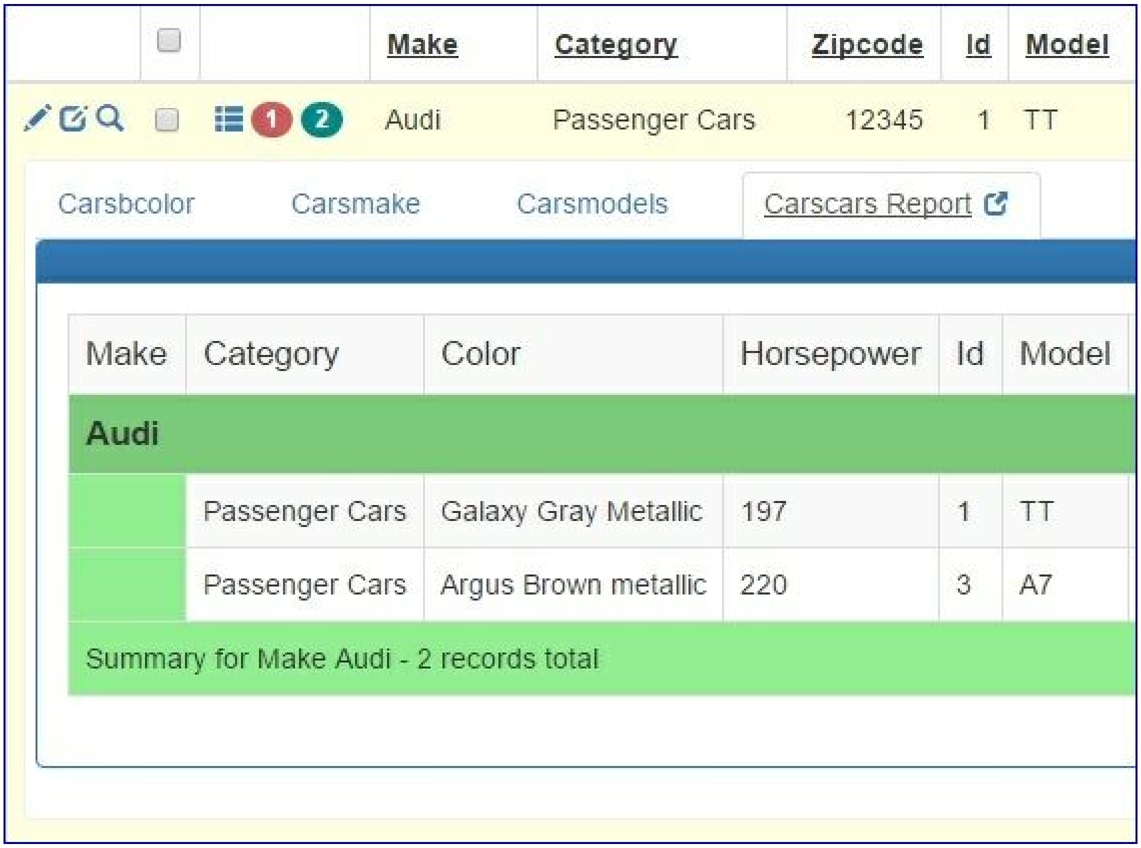
- make necessary changes to SQL query. More info about editing SQL queries;

- choose group fields and grouping types;

- set report totals and layout.

Note that you can use reports as both master and details tables. For more information, see Master-details relationship between tables.

Report as details table:



			Make	Category	Zipcode	Id	Model
	□	1 2	Audi	Passenger Cars	12345	1	TT

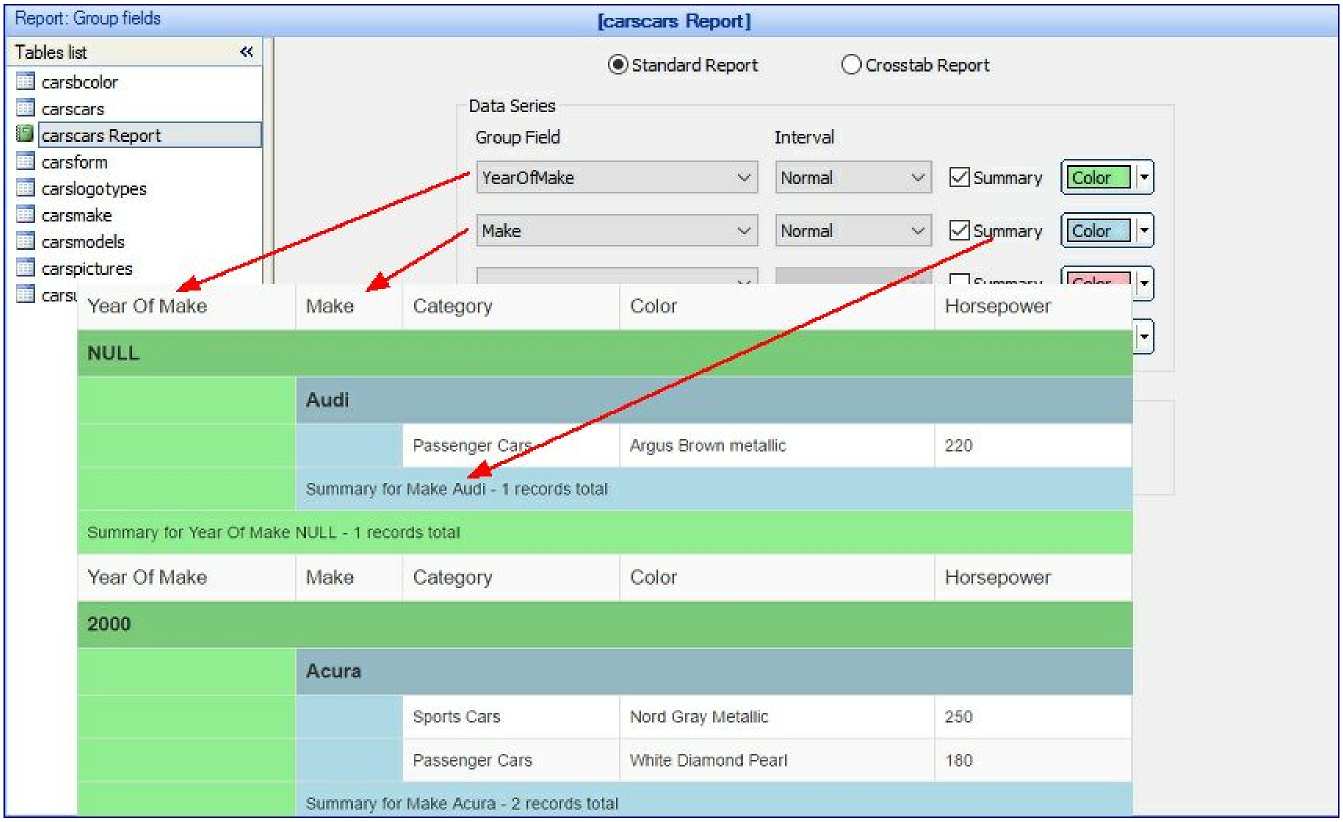
Make	Category	Color	Horsepower	Id	Model
Audi					
	Passenger Cars	Galaxy Gray Metallic	197	1	TT
	Passenger Cars	Argus Brown metallic	220	3	A7
Summary for Make Audi - 2 records total					

Grouping fields in report

On the **Report: Group fields** page you can choose group fields, grouping types, what summary types to show etc.

You can choose between standard and cross-tab reports.

Standard reports



● Standard Report ○ Crosstab Report

Data Series

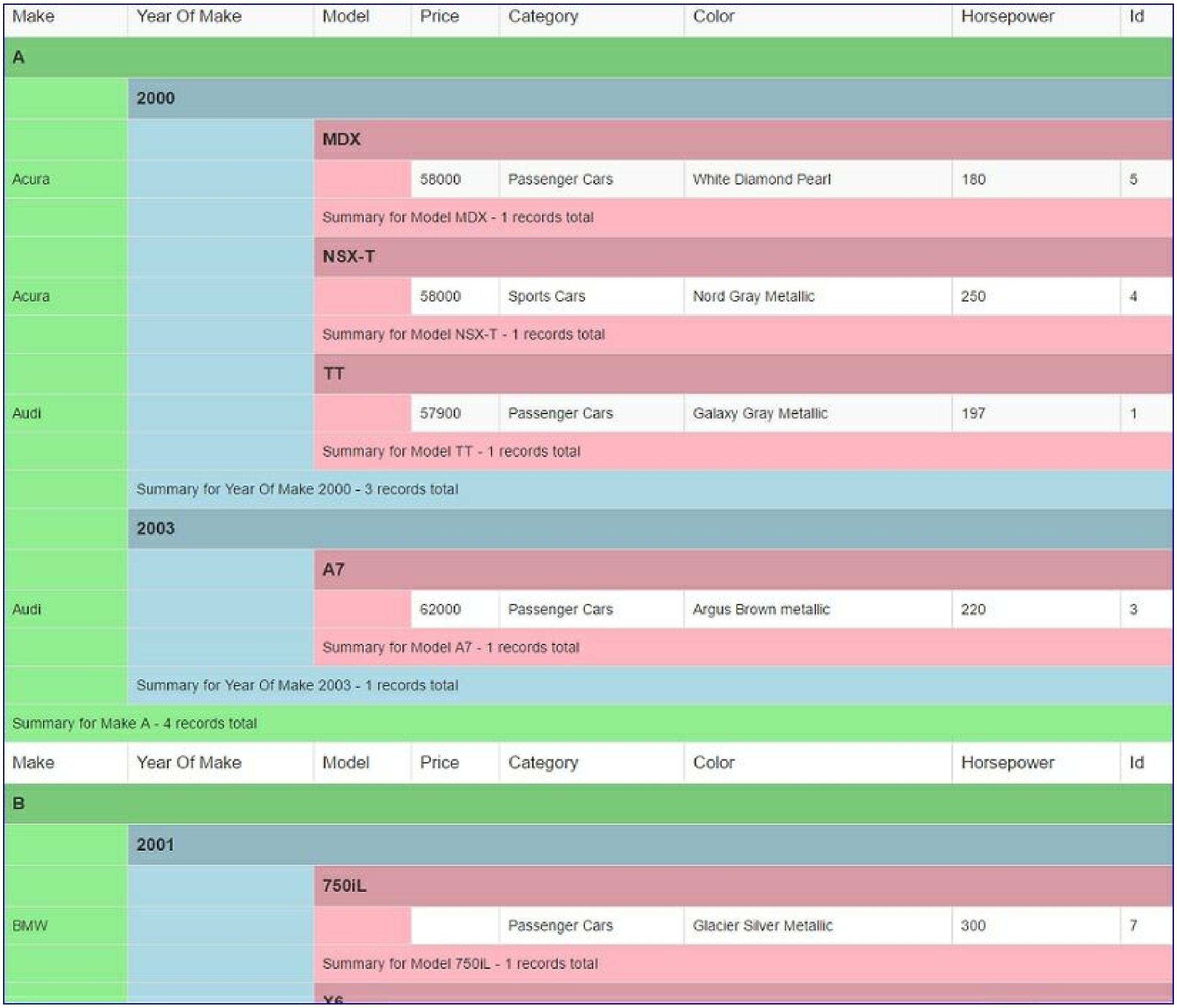
Group Field	Interval		
YearOfMake	Normal	☑ Summary	Color
Make	Normal	☑ Summary	Color
		□ Summary	Color

Year Of Make	Make	Category	Color	Horsepower
NULL				
	Audi			
		Passenger Cars	Argus Brown metallic	220
	Summary for Make Audi - 1 records total			
Summary for Year Of Make NULL - 1 records total				
Year Of Make	Make	Category	Color	Horsepower
2000				
	Acura			
		Sports Cars	Nord Gray Metallic	250
		Passenger Cars	White Diamond Pearl	180
	Summary for Make Acura - 2 records total			

If you clear the **Show details and summary** check box, only summary will be shown on report page.

Besides standard intervals (new group starts when group field value changes) you can use other interval types. Available interval types are different for each data type.

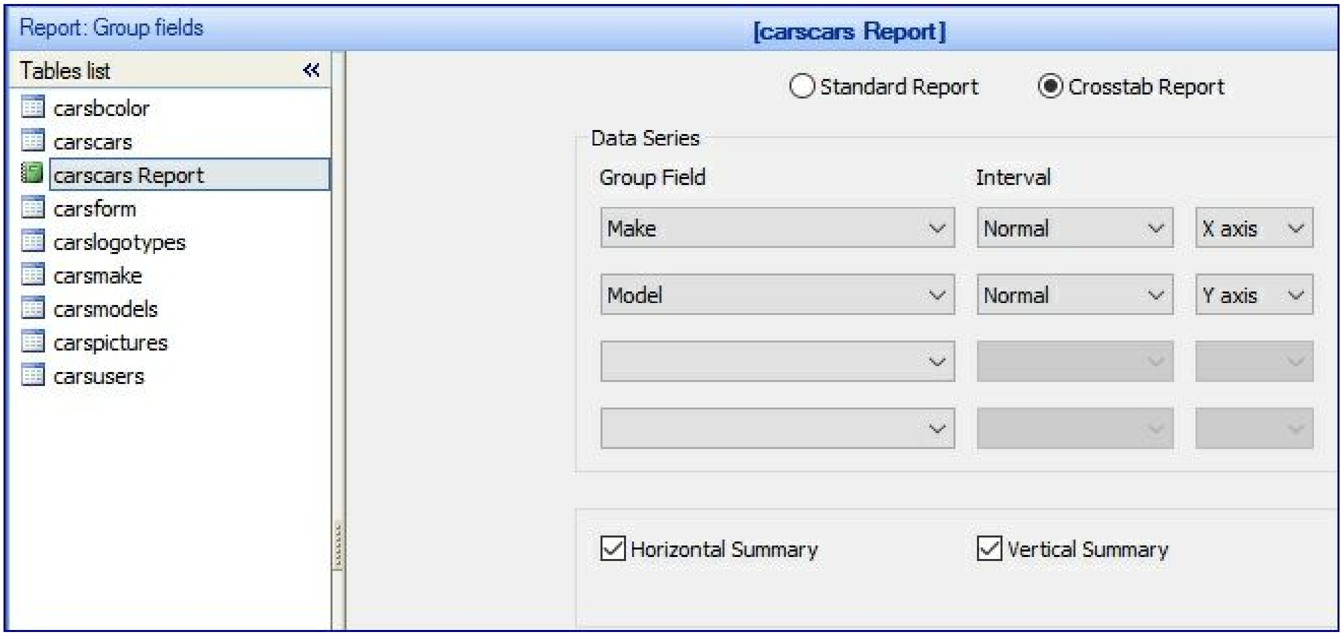
Here is the example of text group field using first letter as an interval.



Make	Year Of Make	Model	Price	Category	Color	Horsepower	Id
A							
	2000						
		MDX					
Acura			58000	Passenger Cars	White Diamond Pearl	180	5
		Summary for Model MDX - 1 records total					
		NSX-T					
Acura			58000	Sports Cars	Nord Gray Metallic	250	4
		Summary for Model NSX-T - 1 records total					
		TT					
Audi			57900	Passenger Cars	Galaxy Gray Metallic	197	1
		Summary for Model TT - 1 records total					
	Summary for Year Of Make 2000 - 3 records total						
	2003						
		A7					
Audi			62000	Passenger Cars	Argus Brown metallic	220	3
		Summary for Model A7 - 1 records total					
	Summary for Year Of Make 2003 - 1 records total						
Summary for Make A - 4 records total							
Make	Year Of Make	Model	Price	Category	Color	Horsepower	Id
B							
	2001						
		750iL					
BMW				Passenger Cars	Glacier Silver Metallic	300	7
		Summary for Model 750iL - 1 records total					
		X6					

Cross-tab reports

PHPRunner supports cross-tab reports (often called pivot tables). On the **Group fields** screen you can choose two or more variables (columns) and assigned them to either X axis or to Y axis. You can assign more than one variable to the axis and switch between them in runtime.



Report: Group fields

[carscars Report]

Tables list

- carsbcolor
- carscars
- carscars Report
- carsform
- carslogotypes
- carsmake
- carsmodels
- carspictures
- carsusers

○ Standard Report ● Crosstab Report

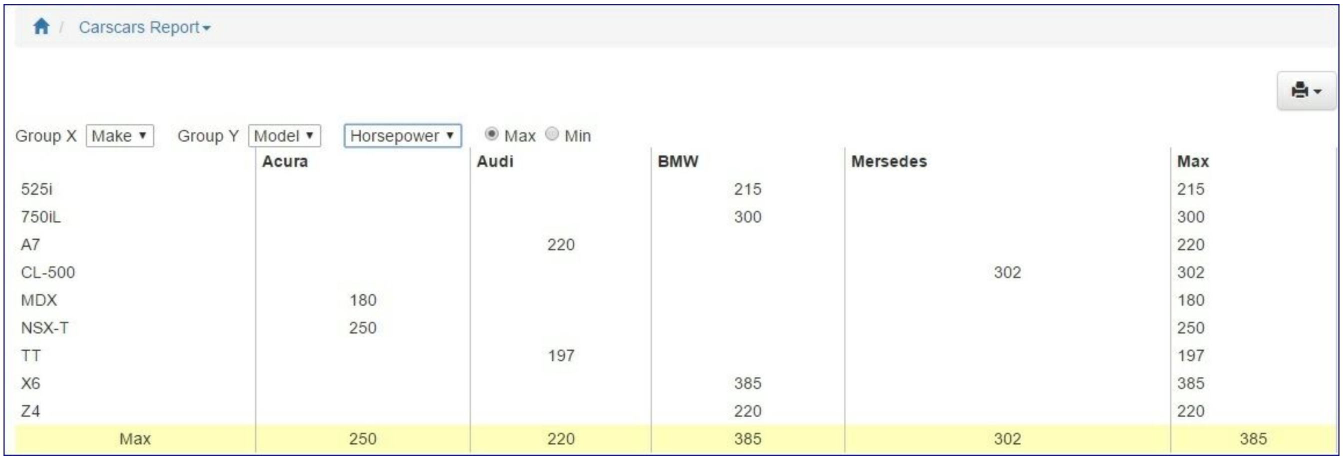
Data Series

Group Field	Interval	
Make	Normal	X axis
Model	Normal	Y axis

☑ Horizontal Summary ☑ Vertical Summary

On **Report: Totals** screen you can choose totals options like Min, Max, Sum and Average.

Here is how cross-tab reports looks in generated application. You can select variables, totals types and data fields to display.



Carscars Report

Group X Make Group Y Model Horsepower ● Max ○ Min

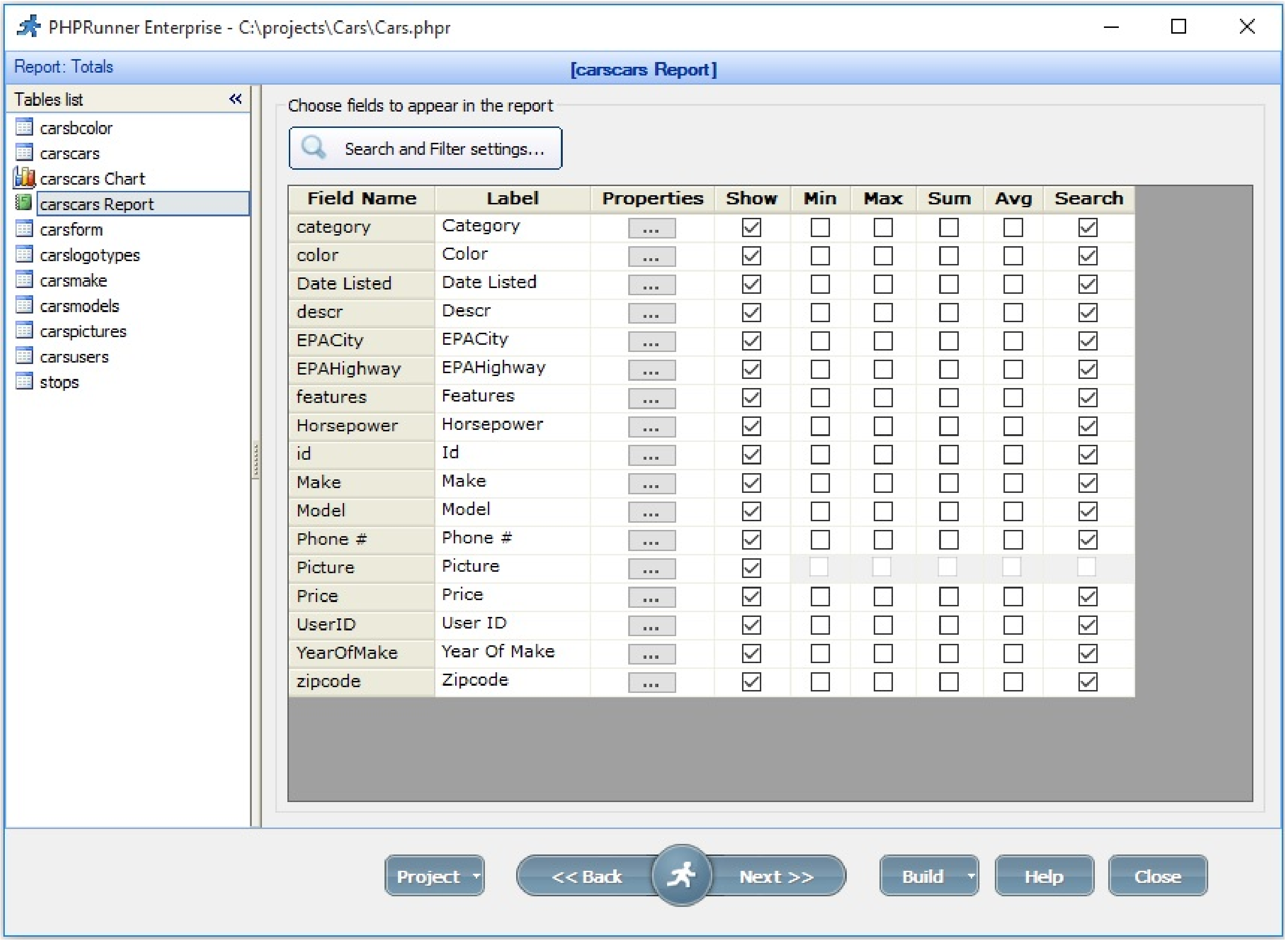
	Acura	Audi	BMW	Mersedes	Max
525i			215		215
750iL			300		300
A7		220			220
CL-500				302	302
MDX	180				180
NSX-T	250				250
TT		197			197
X6			385		385
Z4			220		220
Max	250	220	385	302	385

2.9.2 Report totals and layout

Quick jump

The Report: Miscellaneous page

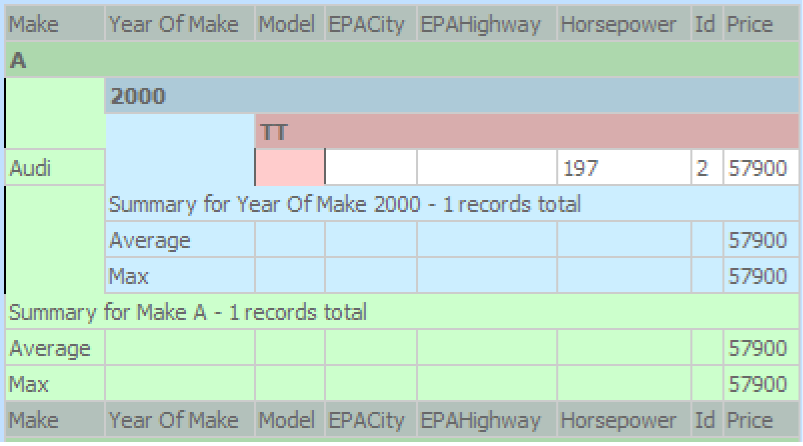
On the **Report Totals** page you can choose what fields to display on the report/search pages.



Choose fields to appear in the report

Field Name	Label	Properties	Show	Min	Max	Sum	Avg	Search
category	Category	...	☑	□	□	□	□	☑
color	Color	...	☑	□	□	□	□	☑
Date Listed	Date Listed	...	☑	□	□	□	□	☑
descr	Descr	...	☑	□	□	□	□	☑
EPACity	EPACity	...	☑	□	□	□	□	☑
EPAHighway	EPAHighway	...	☑	□	□	□	□	☑
features	Features	...	☑	□	□	□	□	☑
Horsepower	Horsepower	...	☑	□	□	□	□	☑
id	Id	...	☑	□	□	□	□	☑
Make	Make	...	☑	□	□	□	□	☑
Model	Model	...	☑	□	□	□	□	☑
Phone #	Phone #	...	☑	□	□	□	□	☑
Picture	Picture	...	☑	□	□	□	□	□
Price	Price	...	☑	□	□	□	□	☑
UserID	User ID	...	☑	□	□	□	□	☑
YearOfMake	Year Of Make	...	☑	□	□	□	□	☑
zipcode	Zipcode	...	☑	□	□	□	□	☑

Also you can apply aggregate functions like MIN, MAX, SUM and AVERAGE. The results of these calculations will be displayed after each group and at the end of page/report.



Make	Year Of Make	Model	EPACity	EPAHighway	Horsepower	Id	Price
A							
	2000						
		TT					
Audi					197	2	57900
	Summary for Year Of Make 2000 - 1 records total						
	Average						57900
	Max						57900
Summary for Make A - 1 records total							
Average							57900
Max							57900
Make	Year Of Make	Model	EPACity	EPAHighway	Horsepower	Id	Price

The **Report: Miscellaneous** page allows you to choose report layout. If you use grouping you can choose between Stepped, Block, Outline and Align layouts. If you don't use grouping you can only use Tabular layout which is very similar to the basic List page.

You can also change the way your data is sorted. Note that you can only control data sort order inside the group, as groups are sorted by Group fields.

Number of lines per page option determines where to insert the page break, when you print the whole report.

Additional info on reports setup

If you need to display a value from a lookup table instead of ID on the Report page, you have two options:

1. Modify SQL query to include fields from joined tables.

2. Enable report search page, proceed to **Visual Editor**, open **Search** page, double-click the field and make it **Lookup Wizard**.

Either option allows you to display a value from another table on the Report page (i.e. Customer Name instead of Customer ID).

2.10 Dashboards

2.10.1 Creating dashboards

Quick jump

Create new dashboard

Customize dashboard layout

Master-details dashboard

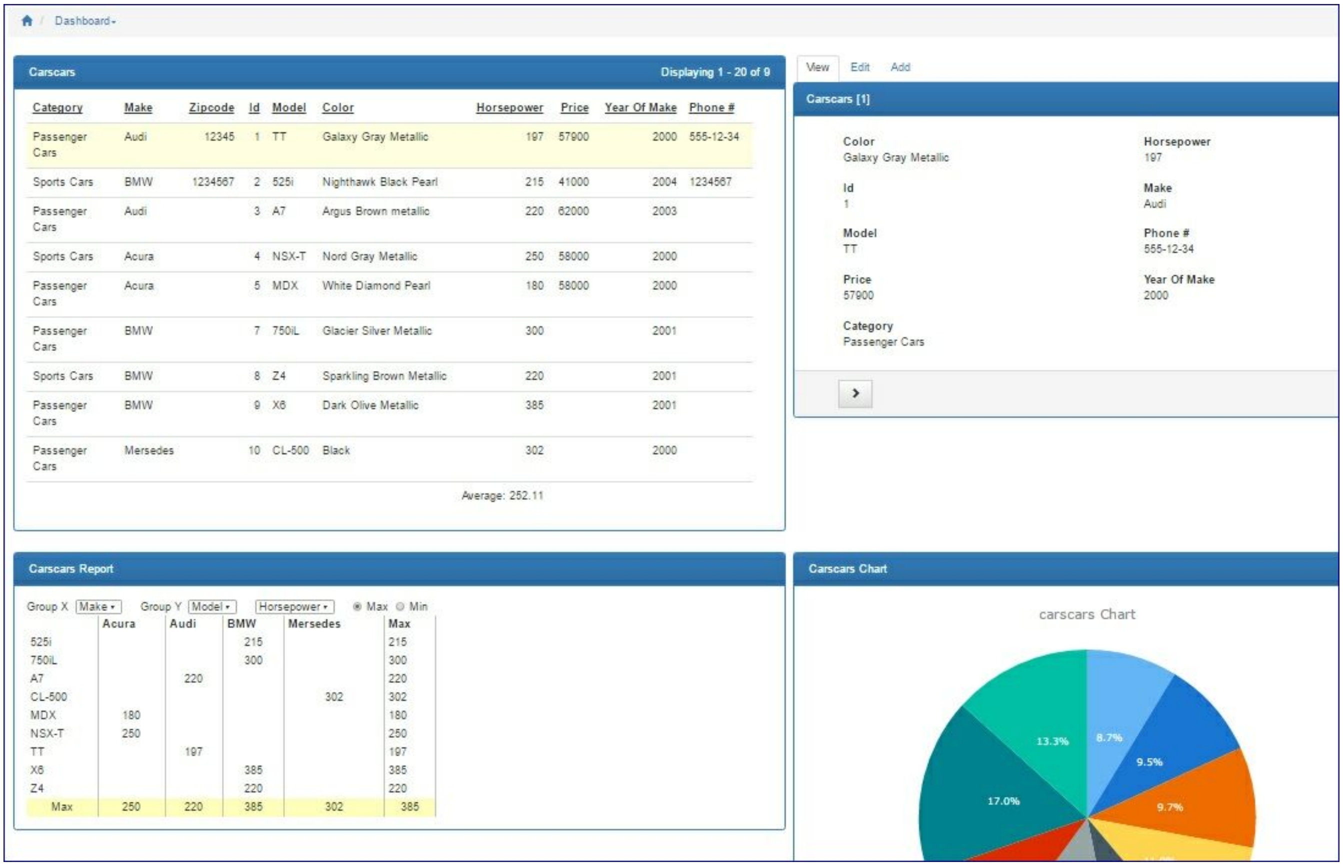
Dashboard with data grid and single record view

Dashboard with map and single record view

Dashboard with data grid and map

Dashboards allow you to display multiple related or unrelated objects on the same page such as grids, single record views, master and details together, charts, reports, search pages, maps, etc.

Here is an example of a dashboard with an Orders table, along with details (Order Details and Customers) and a single Orders table record view:



Dashboard

Carscars Displaying 1 - 20 of 9

Category	Make	Zipcode	Id	Model	Color	Horsepower	Price	Year Of Make	Phone #
Passenger Cars	Audi	12345	1	TT	Galaxy Gray Metallic	197	57900	2000	555-12-34
Sports Cars	BMW	1234567	2	525i	Nighthawk Black Pearl	215	41000	2004	1234567
Passenger Cars	Audi		3	A7	Argus Brown metallic	220	62000	2003	
Sports Cars	Acura		4	NSX-T	Nord Gray Metallic	250	58000	2000	
Passenger Cars	Acura		5	MDX	White Diamond Pearl	180	58000	2000	
Passenger Cars	BMW		7	750iL	Glacier Silver Metallic	300		2001	
Sports Cars	BMW		8	Z4	Sparkling Brown Metallic	220		2001	
Passenger Cars	BMW		9	X6	Dark Olive Metallic	385		2001	
Passenger Cars	Mersedes		10	CL-500	Black	302		2000	
						Average: 252.11			

View Edit Add

Carscars [1]

Color Galaxy Gray Metallic	Horsepower 197
Id 1	Make Audi
Model TT	Phone # 555-12-34
Price 57900	Year Of Make 2000
Category Passenger Cars	

Carscars Report

Group X Make ▾ Group Y Model ▾ Horsepower ▾ ● Max ○ Min

	Acura	Audi	BMW	Mersedes	Max
525i			215		215
750iL			300		300
A7		220			220
CL-500				302	302
MDX	180				180
NSX-T	250				250
TT		197			197
X6			385		385
Z4			220		220
Max	250	220	385	302	385

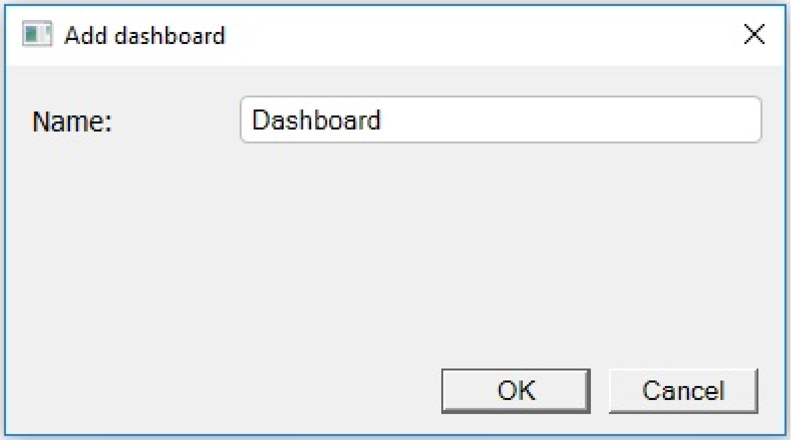
Carscars Chart

Creating a new dashboard

To create a dashboard:

1. Proceed to the **Datasource tables** page and click **Create Dashboard**.

2. Enter the dashboard name. Click **OK**.



Add dashboard

Name: Dashboard

Note: It is possible to create a copy of an existing dashboard (right-click the dashboard and select **Copy**).

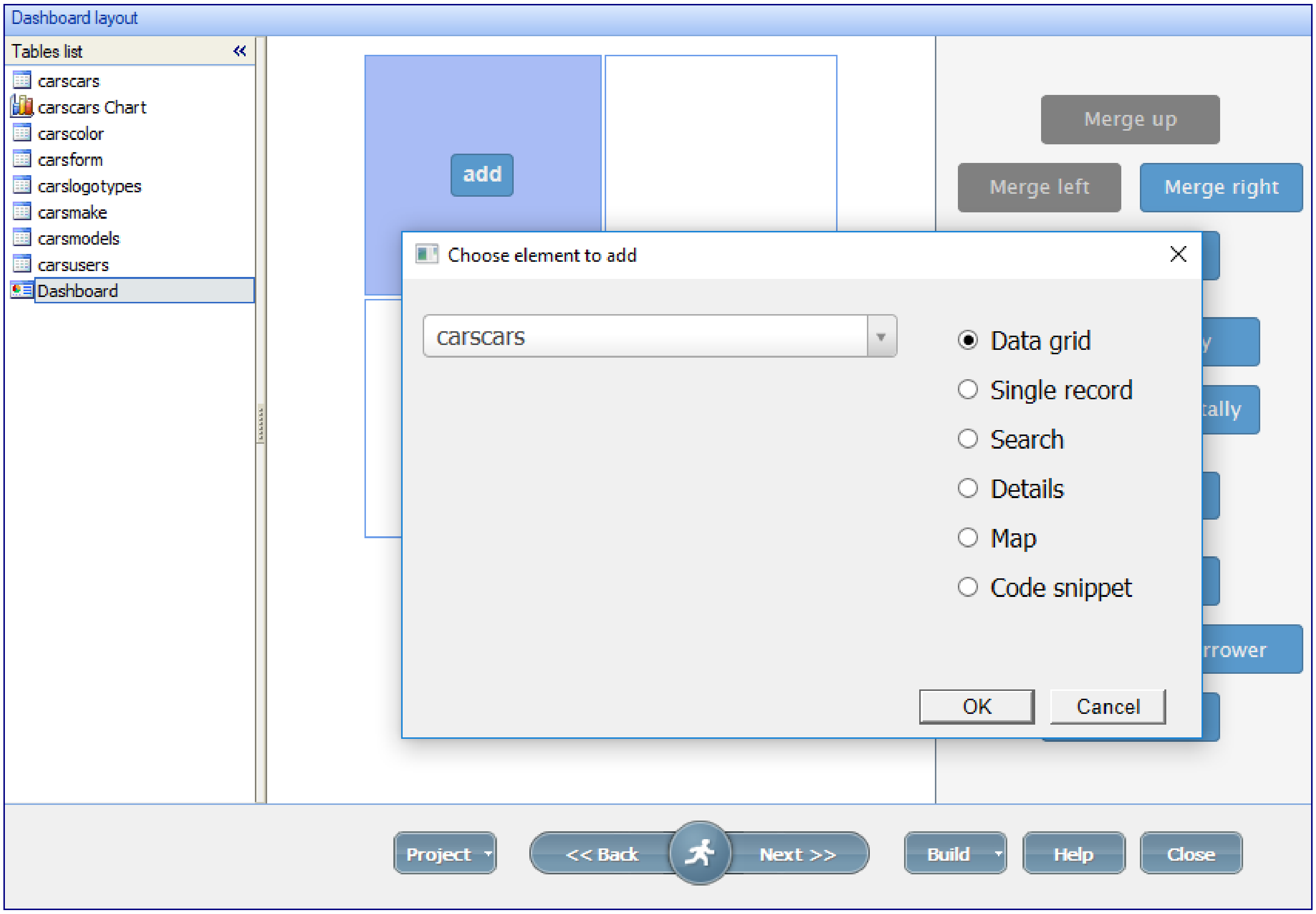
On the next several pages (use **Next** button to navigate) you can:

- set the dashboard layout;
- define your dashboard's search settings.

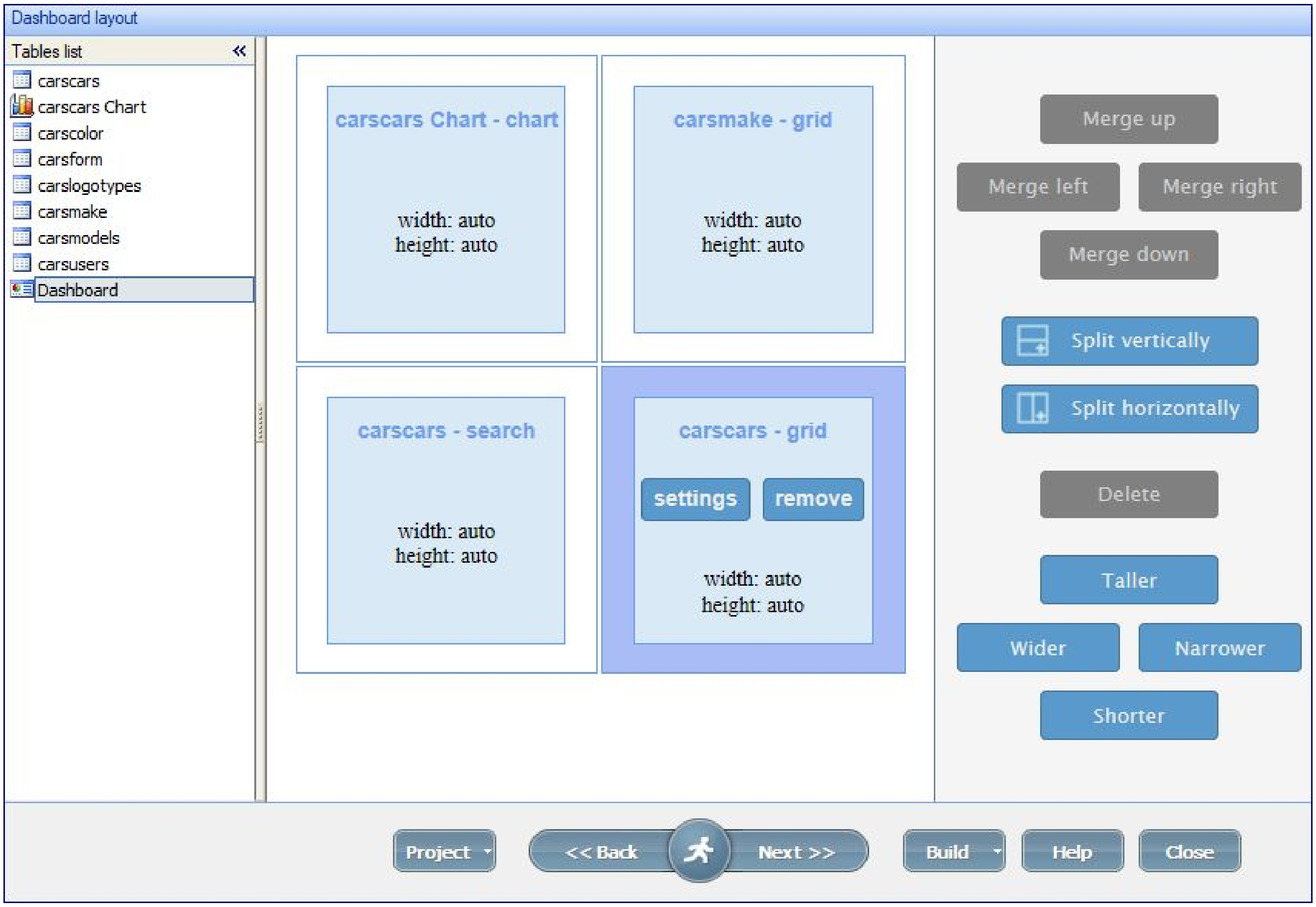
Customizing an existing dashboard's layout

On the **Dashboard layout** screen you may select dashboard elements and define your dashboard's layout. You can change the dashboard layout using merge and split buttons.

Clicking **add** allows you select a data grid, single record view, chart, report, search page or map to use as a dashboard element.



Click **settings** on the added dashboard element to customize its appearance. Click **remove** to delete it.

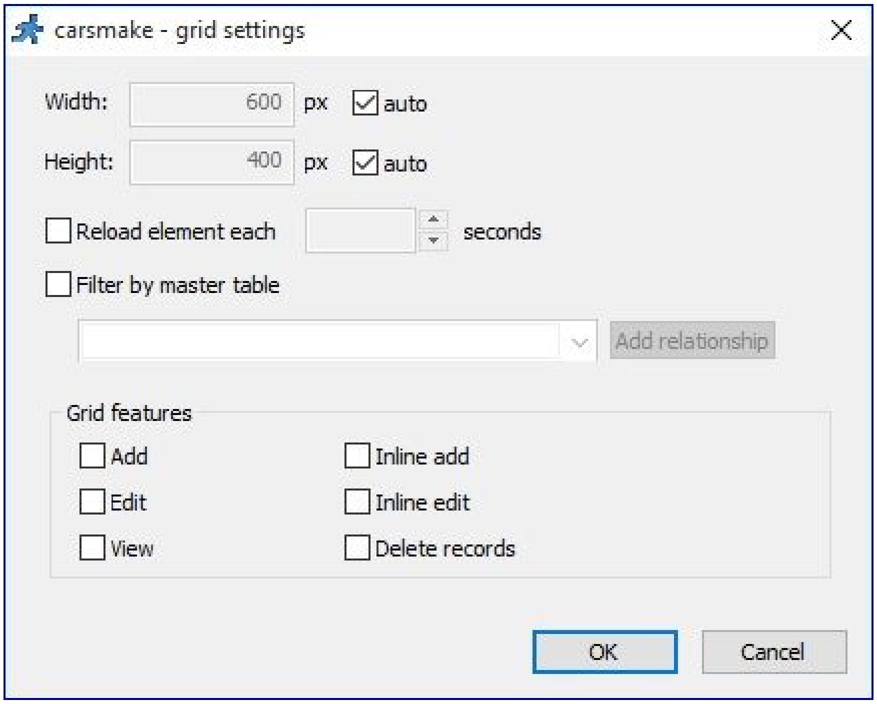


Code snippets in dashboards

You can insert any sort of code snippet as a dashboard element. This code could be used to display current weather, calculate order totals, show the number of active users, or even embed a Youtube video.
See this live demo for inspiration.

Master-details dashboard

You can select grid features to be displayed in a browser: Add/Edit/View/Delete record buttons may be used for a data grid, View/Add/Edit tabs for a single record. Use the **Filter by master table** option to make dashboard elements display data depending on other dashboard elements. You can use this option for cascading master-details (for example, customers-orders-order details) or when a map is used as details.



carsmake - grid settings

Width: 600 px ☑ auto

Height: 400 px ☑ auto

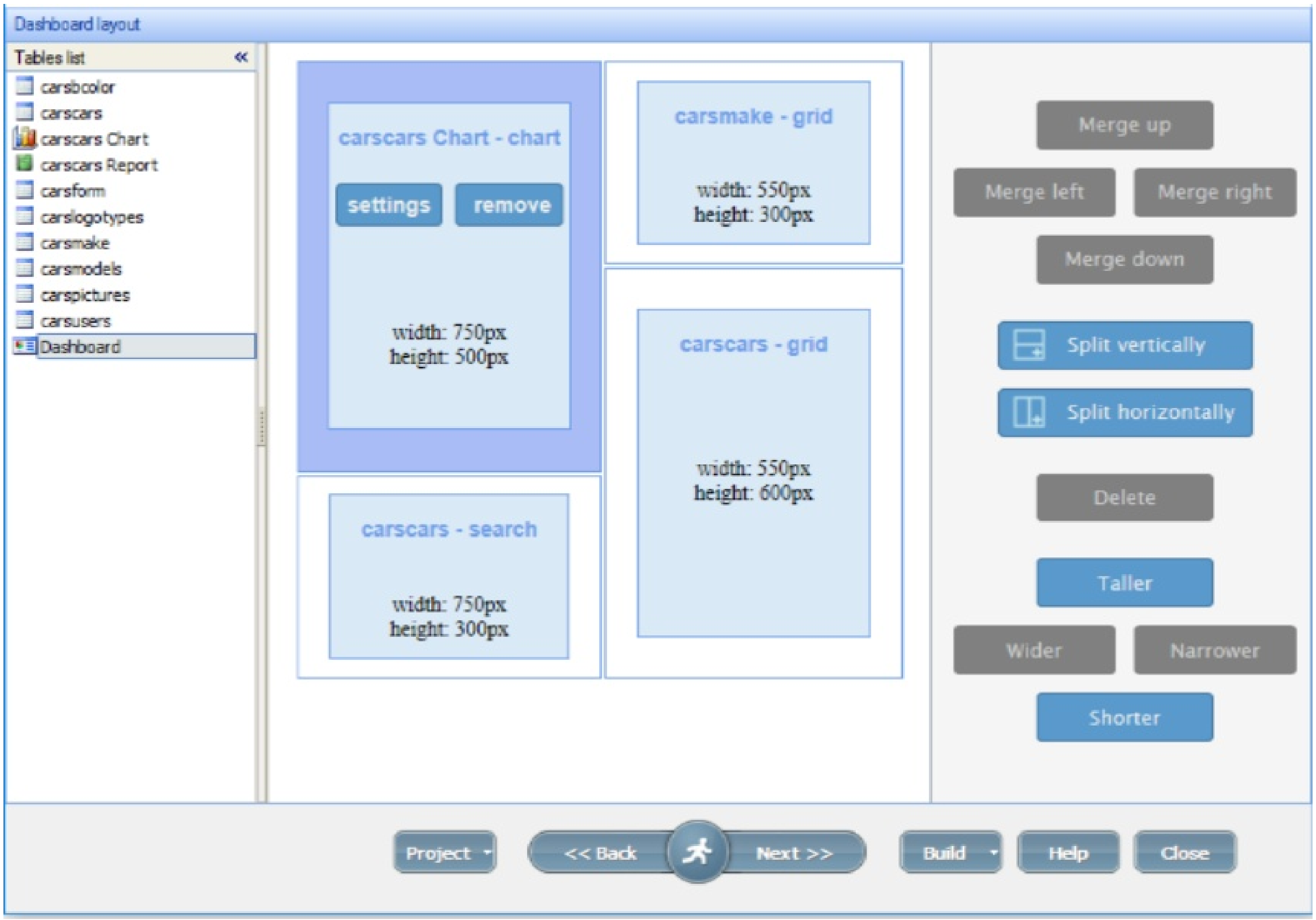
□ Reload element each seconds

□ Filter by master table

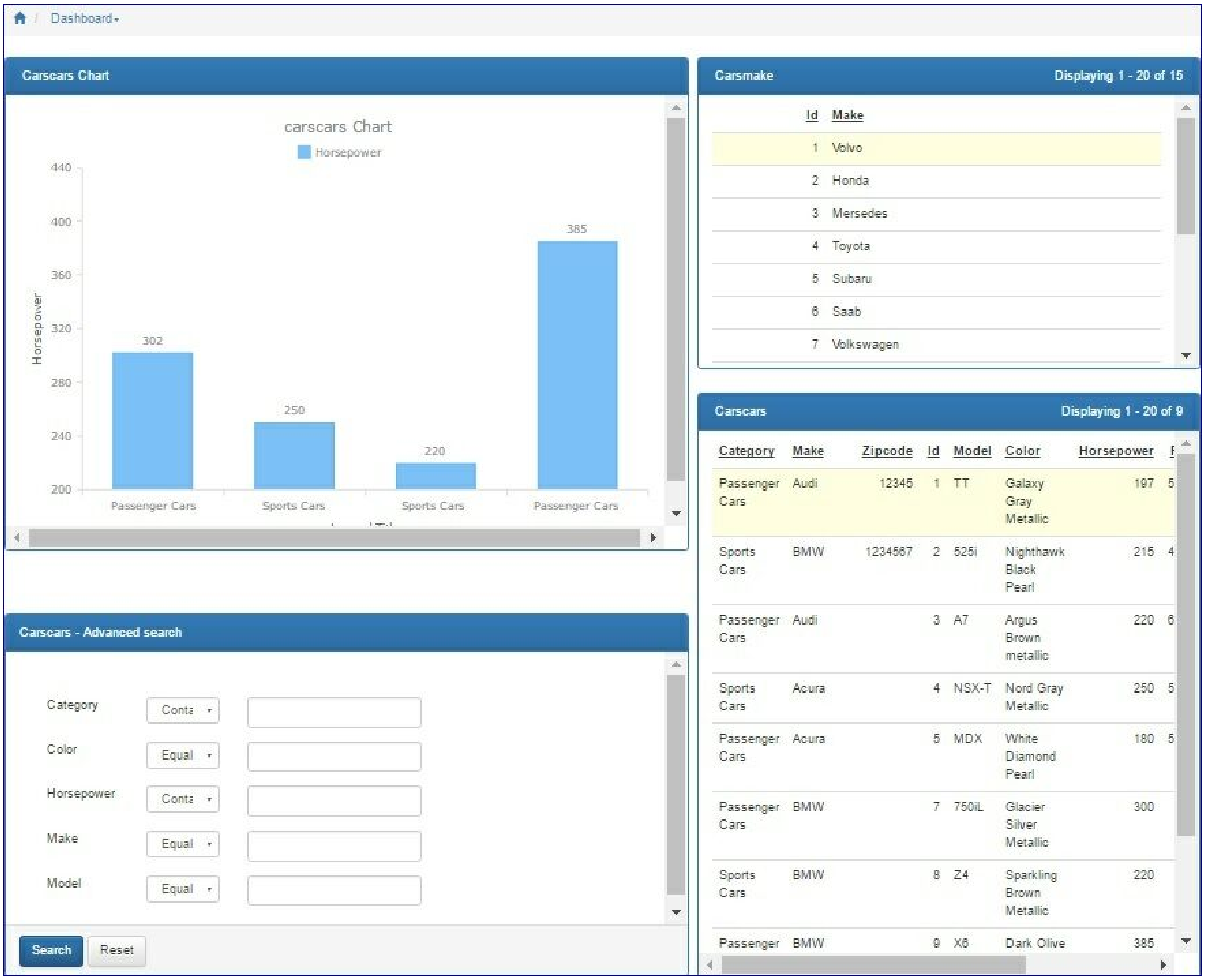
Grid features

□ Add
□ Edit
□ View
□ Inline add
□ Inline edit
□ Delete records

If a dashboard element has a strongly differing width or height, you may use the **Taller**/**Shorter** and **Wider**/**Narrower** buttons to change the cells' dimensions. In the example below, we make the *orders-record* element shorter.



How it looks in a browser:



Carsmake — Displaying 1 - 20 of 15

Id	Make
1	Volvo
2	Honda
3	Mersedes
4	Toyota
5	Subaru
6	Saab
7	Volkswagen

Carscars — Displaying 1 - 20 of 9

Category	Make	Zipcode	Id	Model	Color	Horsepower	F
Passenger Cars	Audi	12345	1	TT	Galaxy Gray Metallic	197	5
Sports Cars	BMW	1234567	2	525i	Nighthawk Black Pearl	215	4
Passenger Cars	Audi		3	A7	Argus Brown metallic	220	6
Sports Cars	Acura		4	NSX-T	Nord Gray Metallic	250	5
Passenger Cars	Acura		5	MDX	White Diamond Pearl	180	5
Passenger Cars	BMW		7	750iL	Glacier Silver Metallic	300	
Sports Cars	BMW		8	Z4	Sparkling Brown Metallic	220	
Passenger	BMW		9	X6	Dark Olive	385	

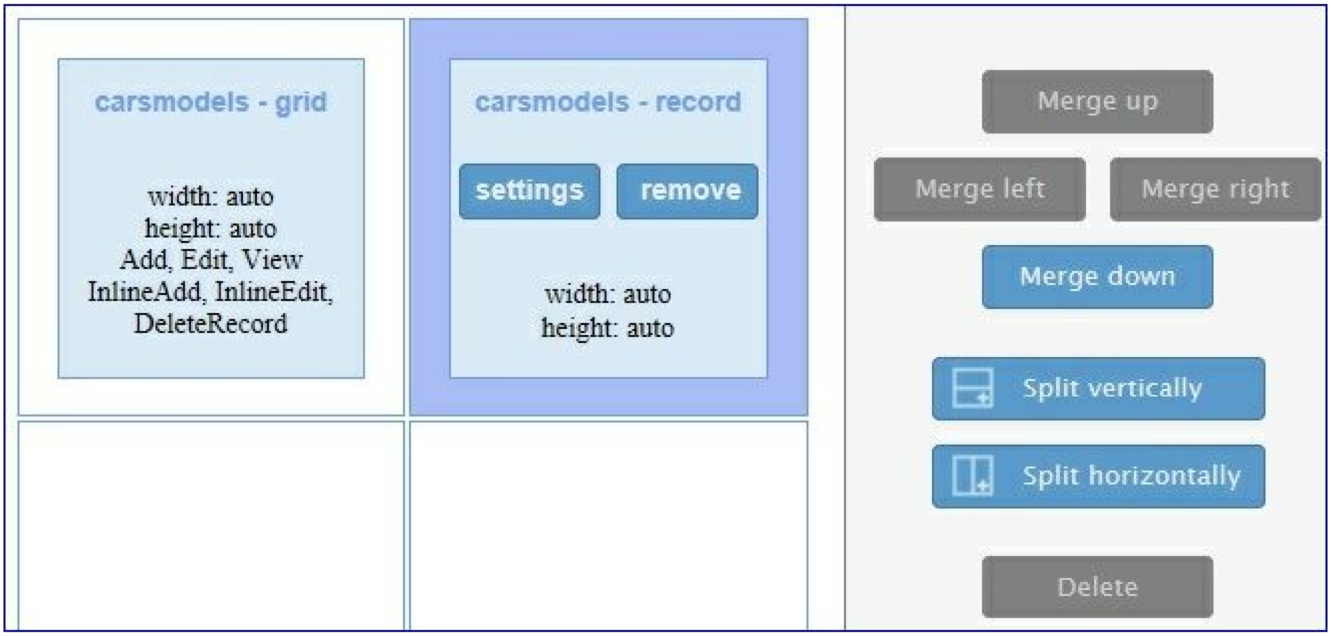
Carscars - Advanced search

Category	Conta	
Color	Equal	
Horsepower	Conta	
Make	Equal	
Model	Equal	

Dashboard with data grid and single record view

To create a dashboard in which a selection in a data grid updates the single record view:

1. Add two tables to a dashboard: one to use as a data grid and one as a single record.



carsmodels - grid

width: auto
height: auto
Add, Edit, View
InlineAdd, InlineEdit,
DeleteRecord

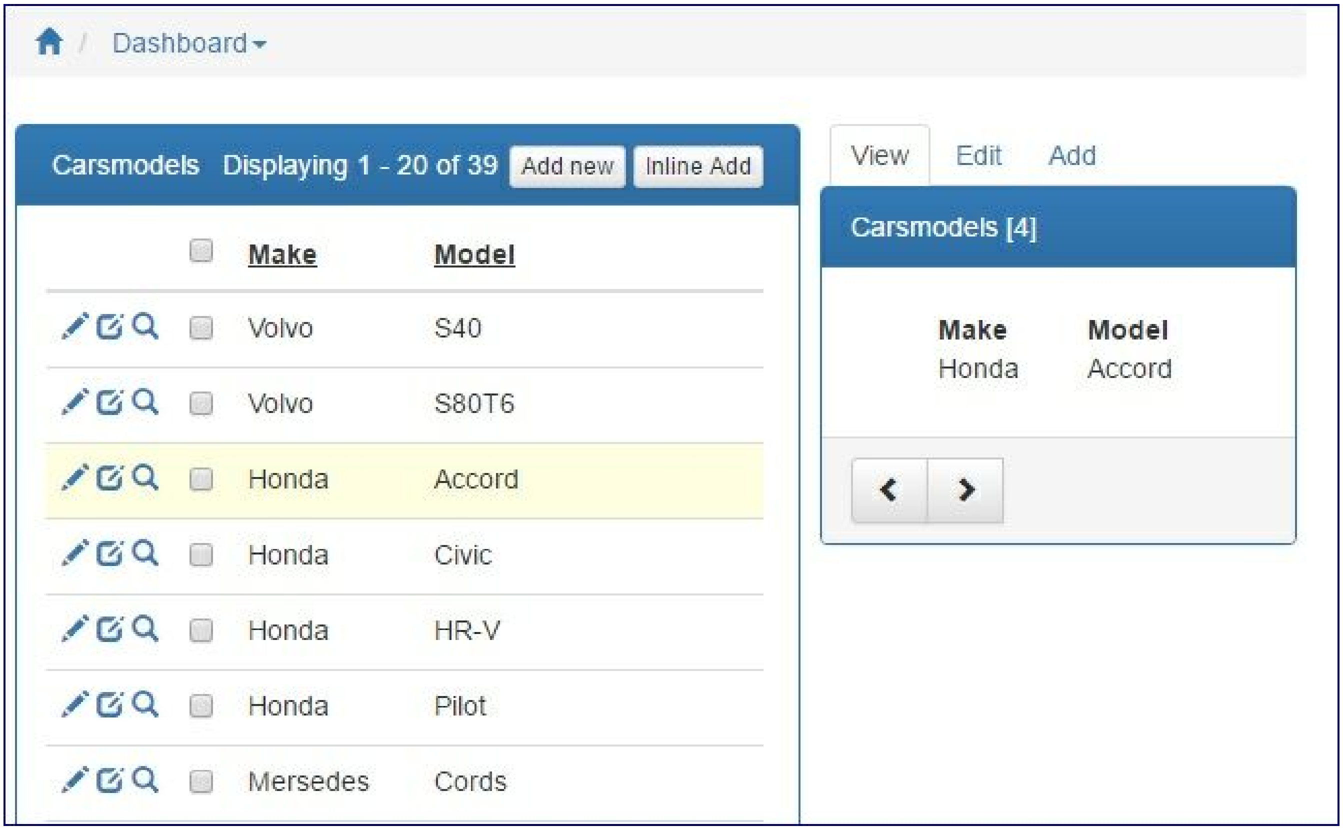
carsmodels - record

width: auto
height: auto

2. Click **settings** on the grid element and select grid features to be displayed in a browser (Add/Edit/View/Delete record buttons).

3. Click **settings** on the record element and select tabs to be displayed on the single record view (View/Add/Edit tabs).

Dashboard page in a browser:



Make	Model
Volvo	S40
Volvo	S80T6
Honda	Accord
Honda	Civic
Honda	HR-V
Honda	Pilot
Mersedes	Cords

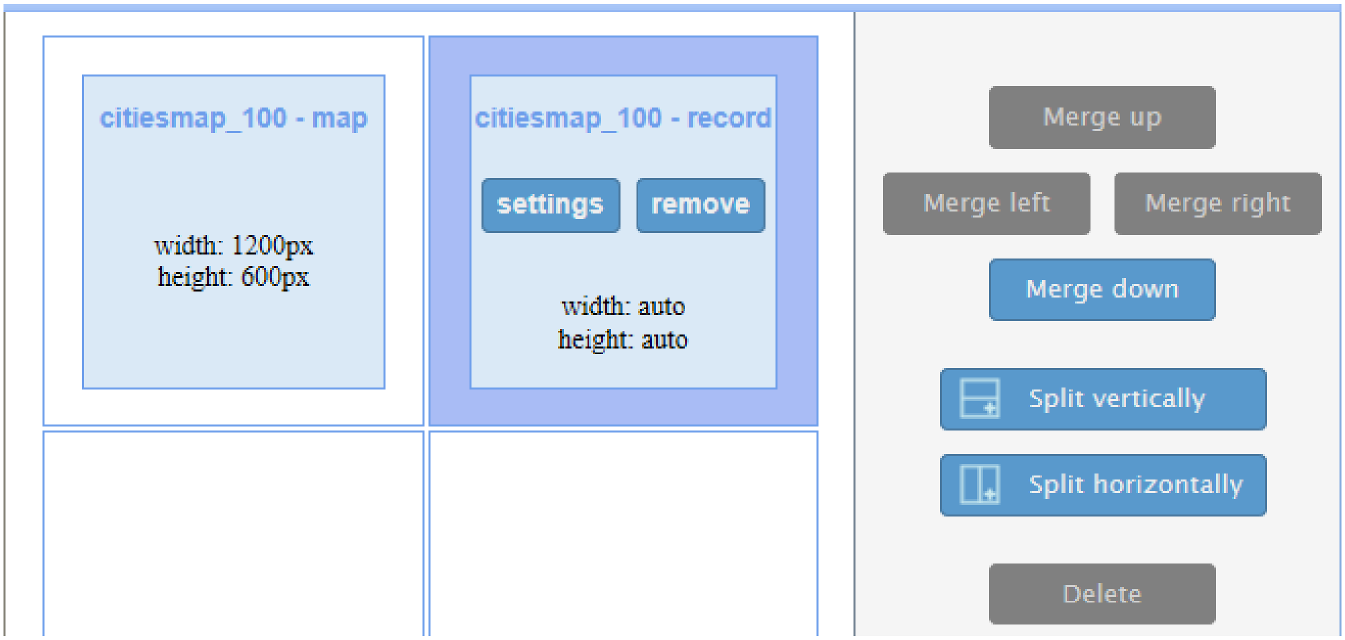
Make	Model
Honda	Accord

Dashboard with map and single record view

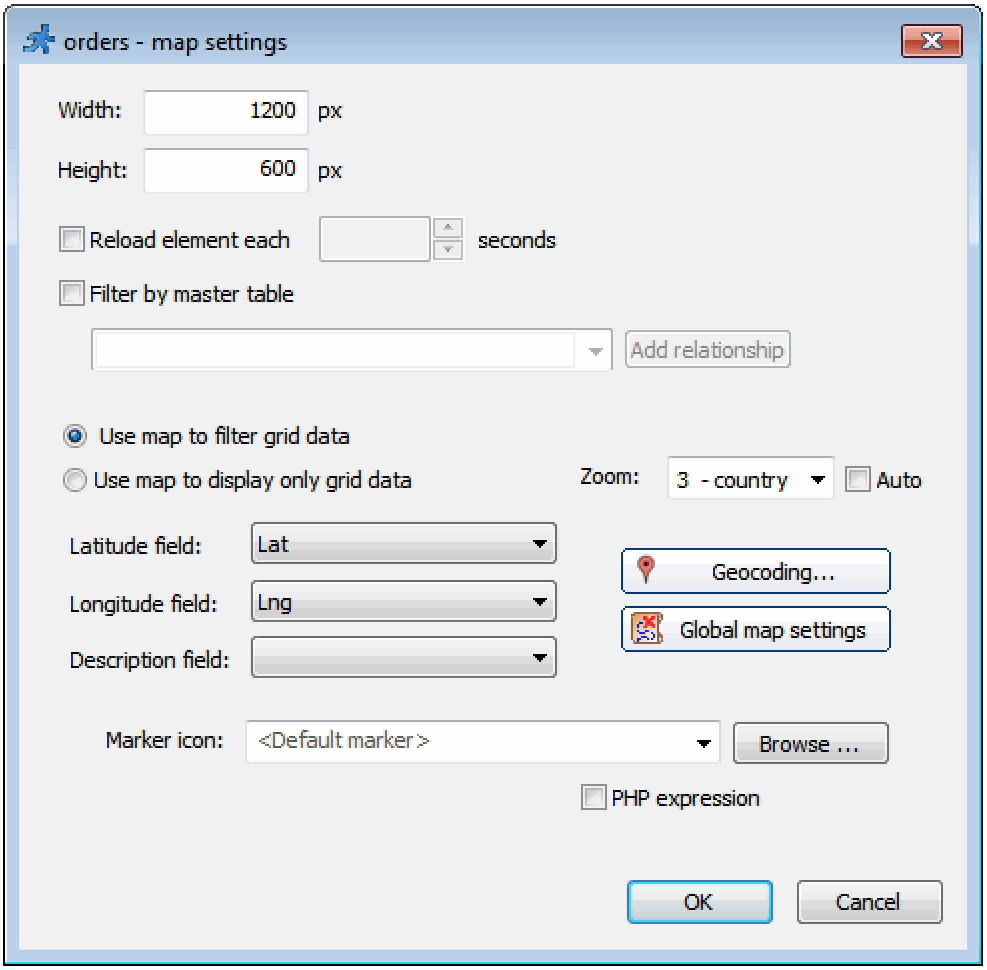
Clicking on a map marker located on a map within a dashboard in a browser will highlight the marker.
If you would like to see the record's data in a fashion similar to the View page after clicking on the marker you need to also add a single record of the table that is used as the map source.

To create a dashboard, where clicking on a map marker updates the single record view:

1. Add two tables to a dashboard: one to use as a map and one as a single record.



2. Click **settings** on the map element and select Latitude and Longitude fields.



orders - map settings

Width: 1200 px

Height: 600 px

□ Reload element each seconds

□ Filter by master table

● Use map to filter grid data

○ Use map to display only grid data

Zoom: 3 - country □ Auto

Latitude field: Lat

Longitude field: Lng

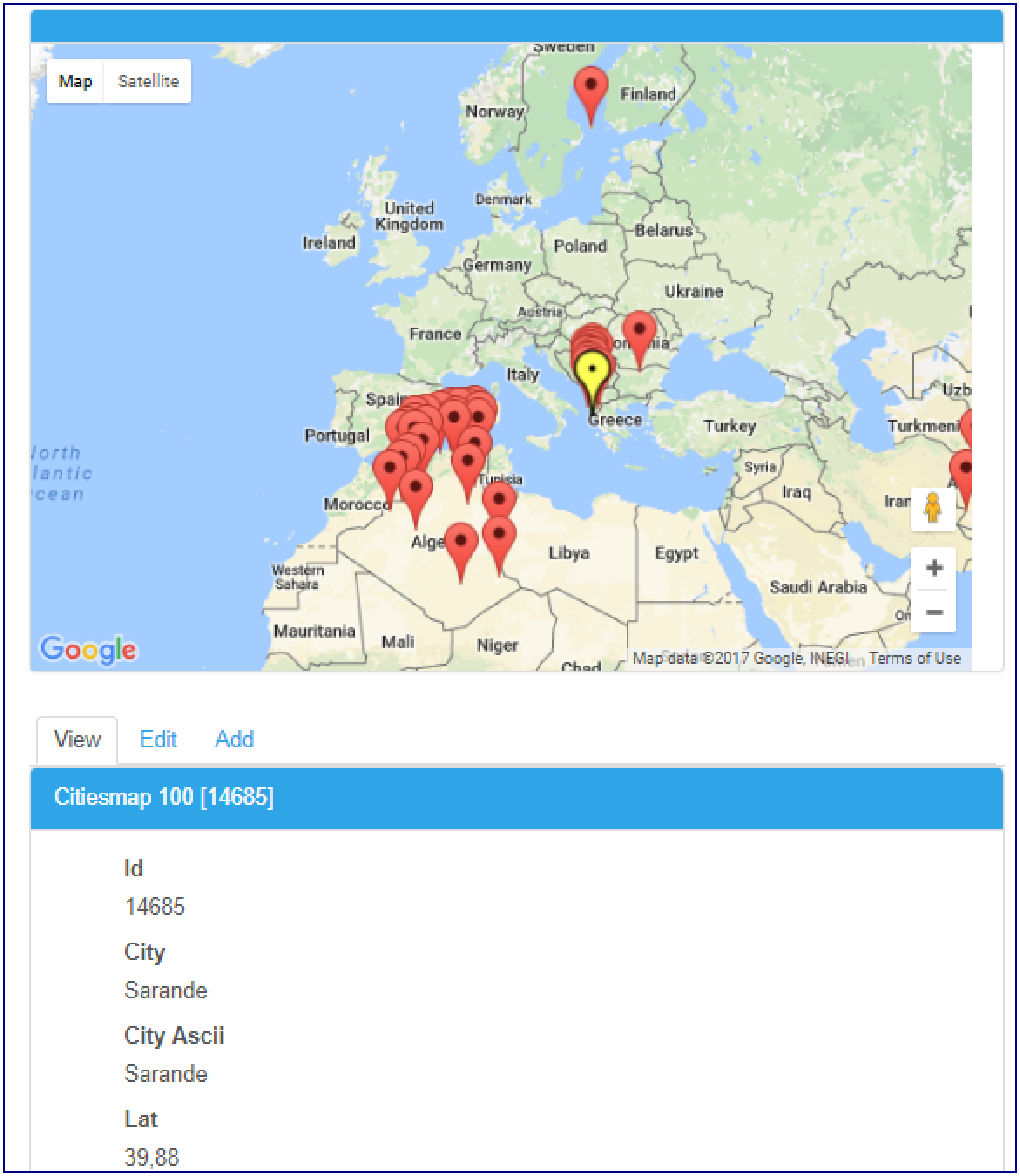
Description field:

Marker icon: <Default marker>

□ PHP expression

3. Click **settings** on the record element and select tabs to be displayed on the single record view (View/Add/Edit tabs).

Dashboard page in a browser:



View Edit Add

Citiesmap 100 [14685]

Id
14685

City
Sarande

City Ascii
Sarande

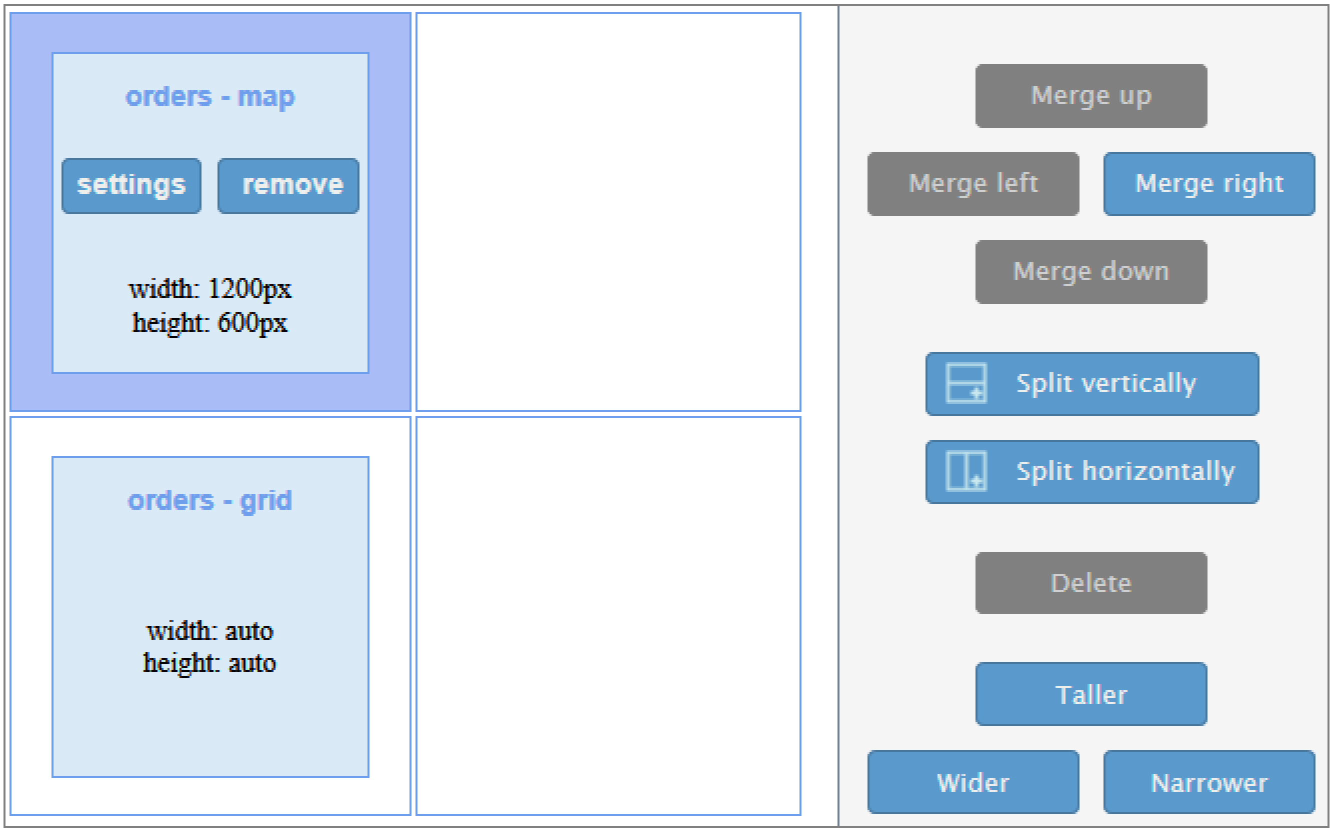
Lat
39,88

Dashboard with data grid and map

There are two different primary uses for a map. A map can serve as master, allowing you to zoom in into any area on the map, and the related grid will only show records in the specified region. In other words, you may use map a to filter grid contents. In second mode the map simply shows all the same records that appear in the grid.

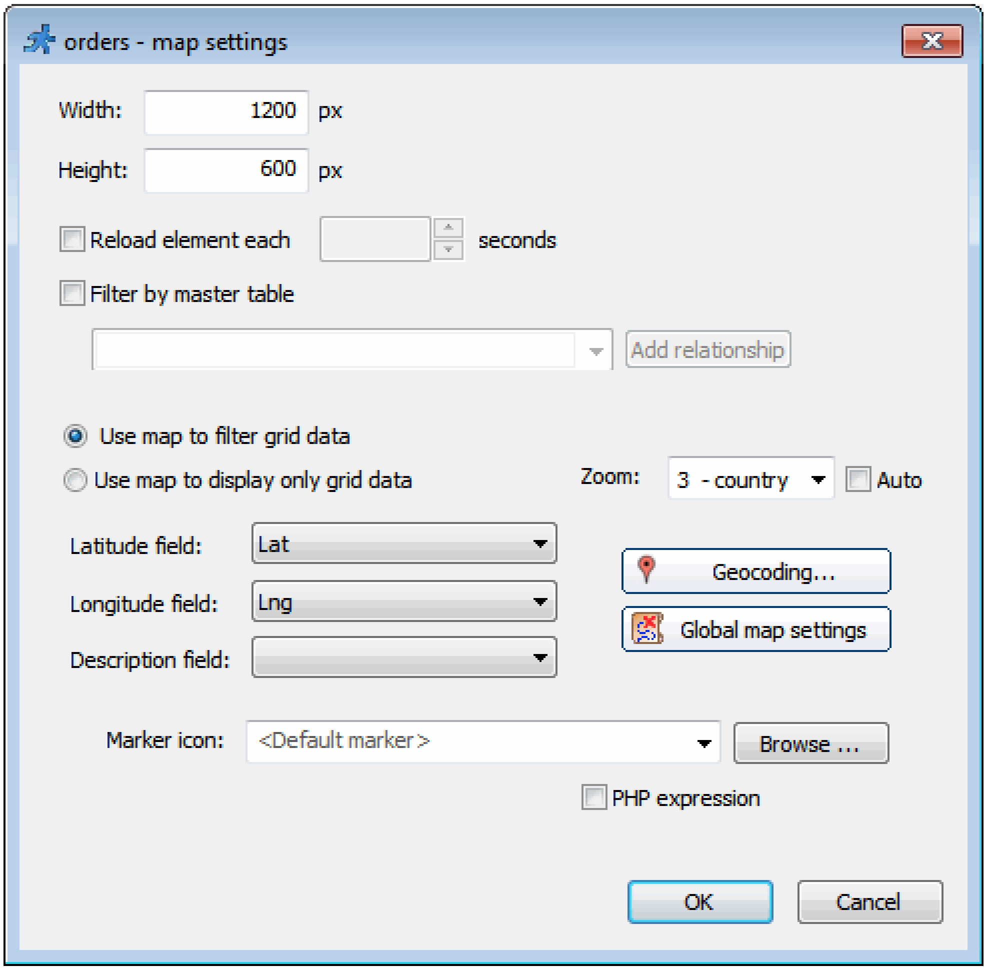
To create a dashboard where grid data is updated when you change the zoom level of the map:

1. Add two tables to a dashboard: one to use as a map and one as a data grid. Note that you need to use Latitude and Longitude fields to make the map function properly.



2. Click **settings** on the grid element and select grid features to be displayed in a browser (Add/Edit/View/Delete record buttons).

3. Click **settings** on the map element and select Latitude and Longitude fields. Select **Use map to filter grid data** option.



orders - map settings

Width: 1200 px

Height: 600 px

□ Reload element each [ ] seconds

□ Filter by master table

[ ] Add relationship

● Use map to filter grid data

○ Use map to display only grid data

Zoom: 3 - country □ Auto

Latitude field: Lat

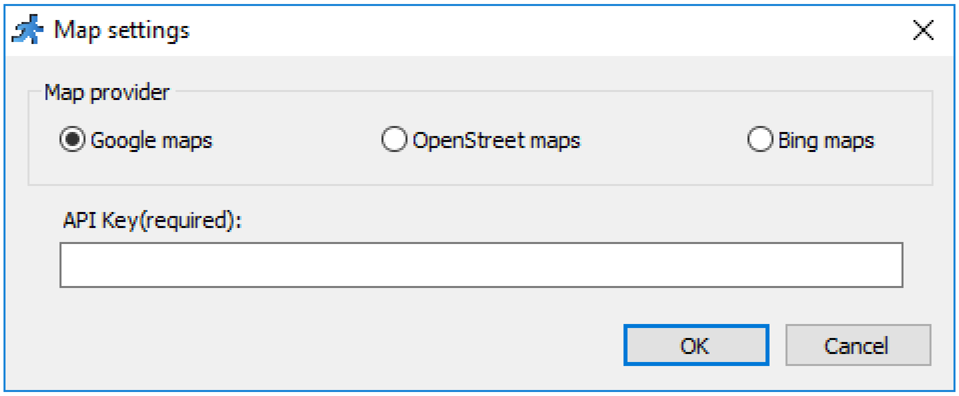
Longitude field: Lng

Description field:

Marker icon: <Default marker>

□ PHP expression

Click **Global map settings** to select a map provider. Enter API Key (if required).
You can obtain API keys for development purposes for free at https://developers.google.com (for Google maps) or at bingmapsportal.com (for Bing maps).



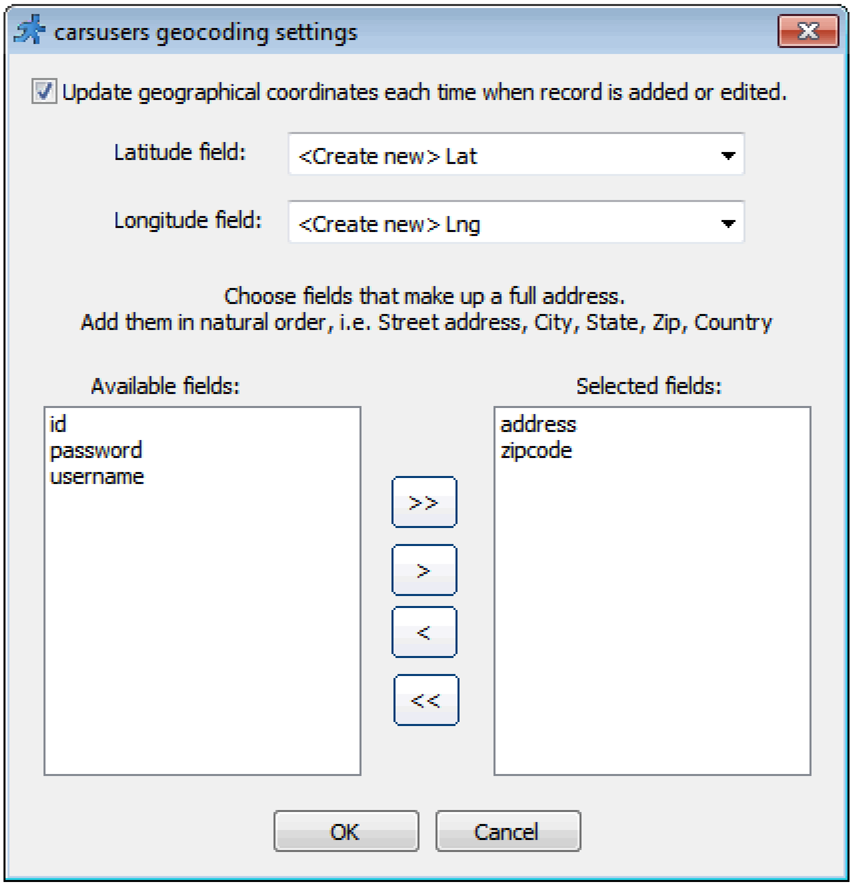
Map settings

Map provider

● Google maps ○ OpenStreet maps ○ Bing maps

API Key(required):

You may use the **Geocoding** option to update latitude/longitude information each time a record is added or updated. You may either select existing fields for storing latitude and longitude data or create new ones. Then select address field(s) to be used during geocoding.



☑ Update geographical coordinates each time when record is added or edited.

Latitude field: <Create new> Lat

Longitude field: <Create new> Lng

Choose fields that make up a full address.
Add them in natural order, i.e. Street address, City, State, Zip, Country

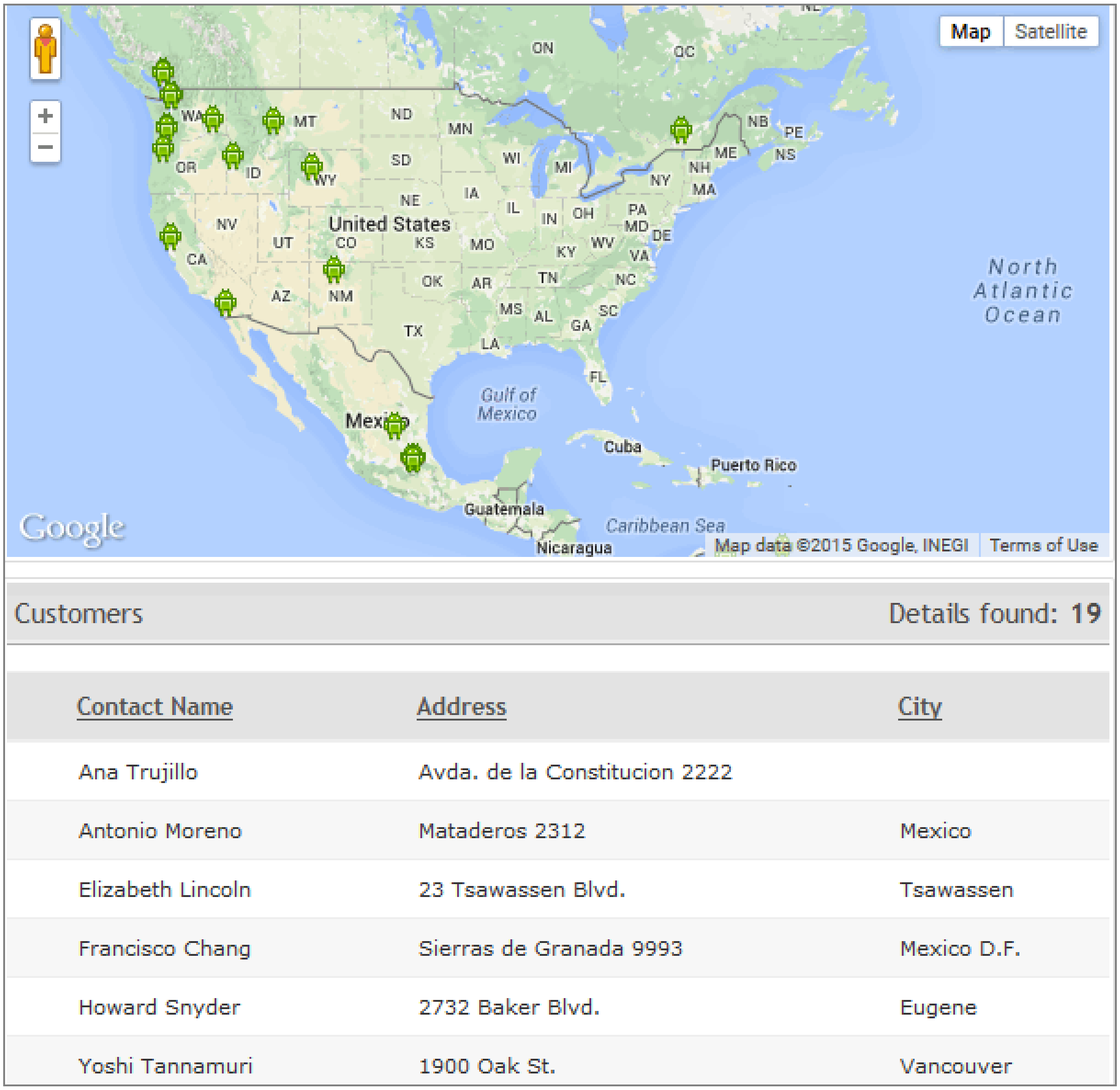
Available fields:

id
password
username

Selected fields:

address
zipcode

Dashboard page in a browser:



Customers Details found: 19

Contact Name	Address	City
Ana Trujillo	Avda. de la Constitucion 2222	
Antonio Moreno	Mataderos 2312	Mexico
Elizabeth Lincoln	23 Tsawassen Blvd.	Tsawassen
Francisco Chang	Sierras de Granada 9993	Mexico D.F.
Howard Snyder	2732 Baker Blvd.	Eugene
Yoshi Tannamuri	1900 Oak St.	Vancouver

2.10.2 Master-details dashboard

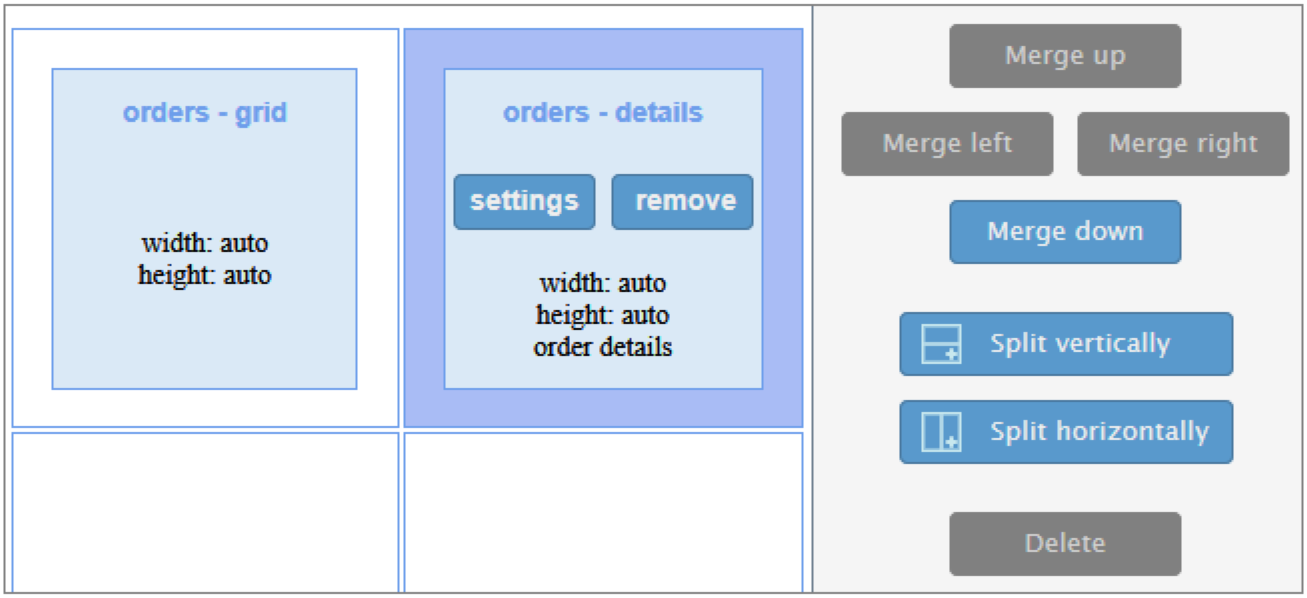
In master-details dashboard selection in a master table filters records in details table. Make sure you set master-details relationship between two or more tables before creating a dashboard. You have two ways to organize your master-details dashboard:

- Add master table to a dashboard twice: as data grid and then as details. In such dashboard if master table has several details tables, you can display all details (each on separate tab) or select which of them to display.

- Add master table and details table as data grids and enable the **Filter by master table** option for details table. Use this option to display nested master-details, such as customers-orders-order details. Also you can display master-details relationship not only between tables but other elements such as table and map.

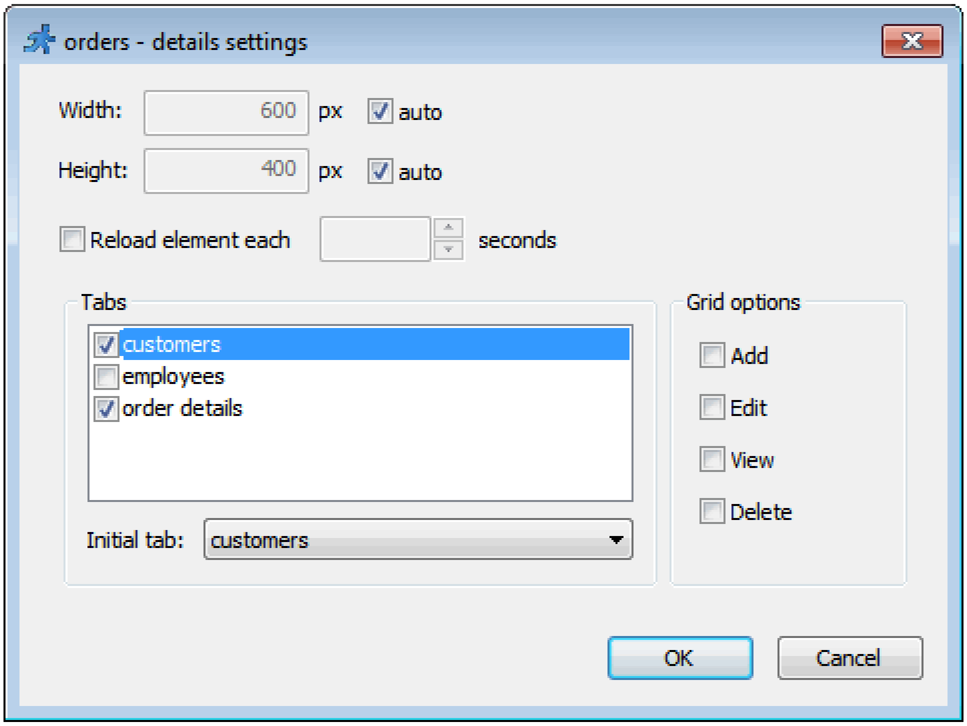
Dashboard where master table added as data grid and as details

1. Add master table to a dashboard as data grid and then as details.



2. Click **settings** on the grid element and select grid features to be displayed in a browser (Add/Edit/View/Delete record buttons).

3. If master table has several details tables, click **settings** on the details element and select details tables to display and set the initial one. Select grid features to be displayed for details table (Add/Edit/View/Delete record buttons).



orders - details settings

Width: 600 px ☑ auto

Height: 400 px ☑ auto

☐ Reload element each seconds

Tabs

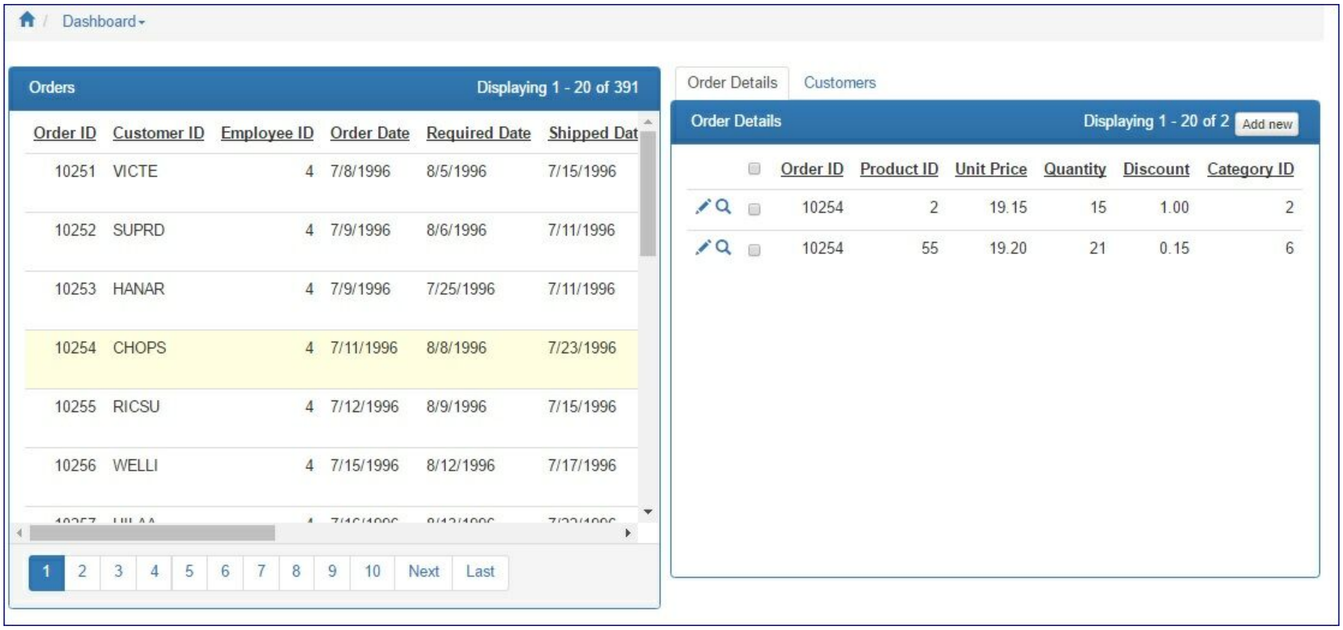
- ☑ customers
- ☐ employees
- ☑ order details

Initial tab: customers

Grid options

- ☐ Add
- ☐ Edit
- ☐ View
- ☐ Delete

When open the master-details dashboard in a browser, we will see the list of orders and order/customer details:



Orders Displaying 1 - 20 of 391

Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Dat
10251	VICTE	4	7/8/1996	8/5/1996	7/15/1996
10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996
10253	HANAR	4	7/9/1996	7/25/1996	7/11/1996
10254	CHOPS	4	7/11/1996	8/8/1996	7/23/1996
10255	RICSU	4	7/12/1996	8/9/1996	7/15/1996
10256	WELLI	4	7/15/1996	8/12/1996	7/17/1996

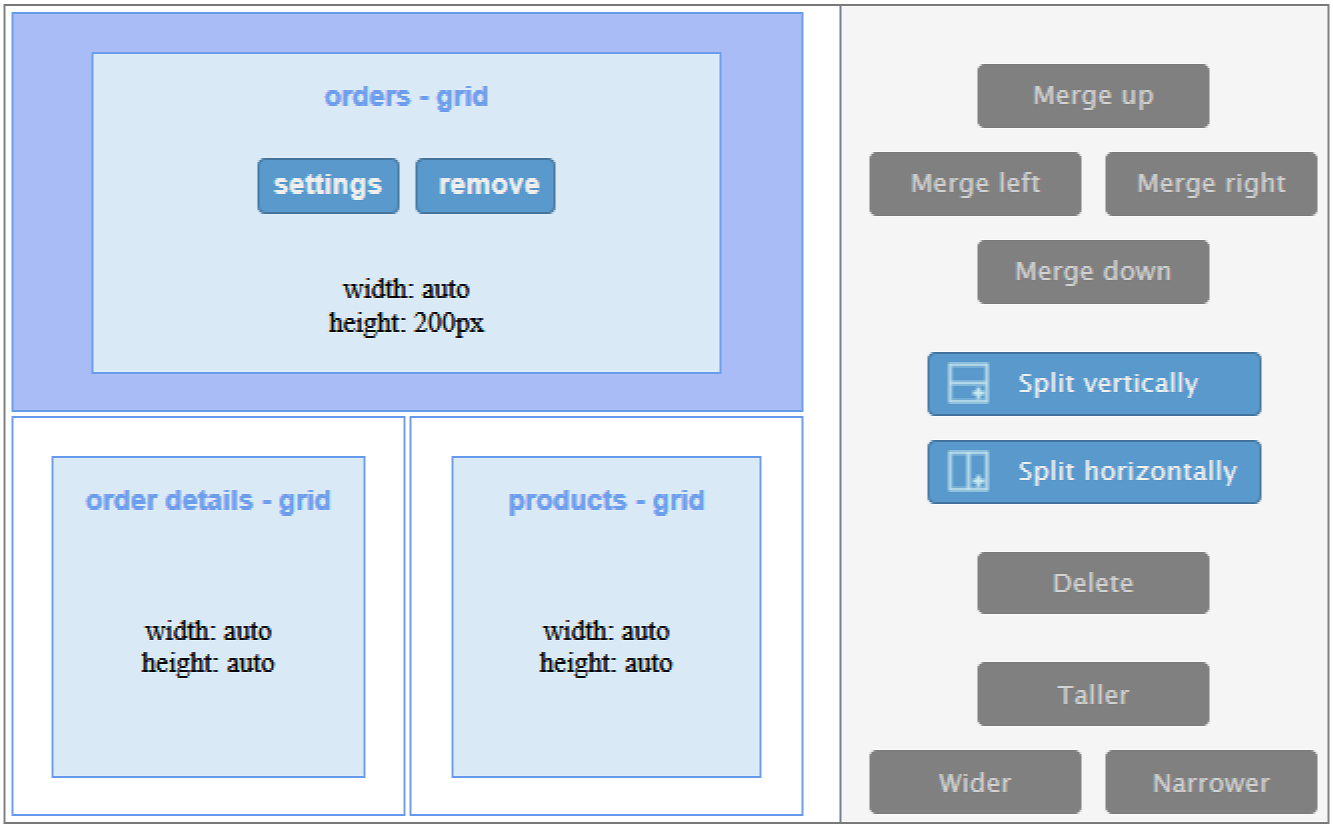
Order Details Displaying 1 - 20 of 2

	Order ID	Product ID	Unit Price	Quantity	Discount	Category ID
☐	10254	2	19.15	15	1.00	2
☐	10254	55	19.20	21	0.15	6

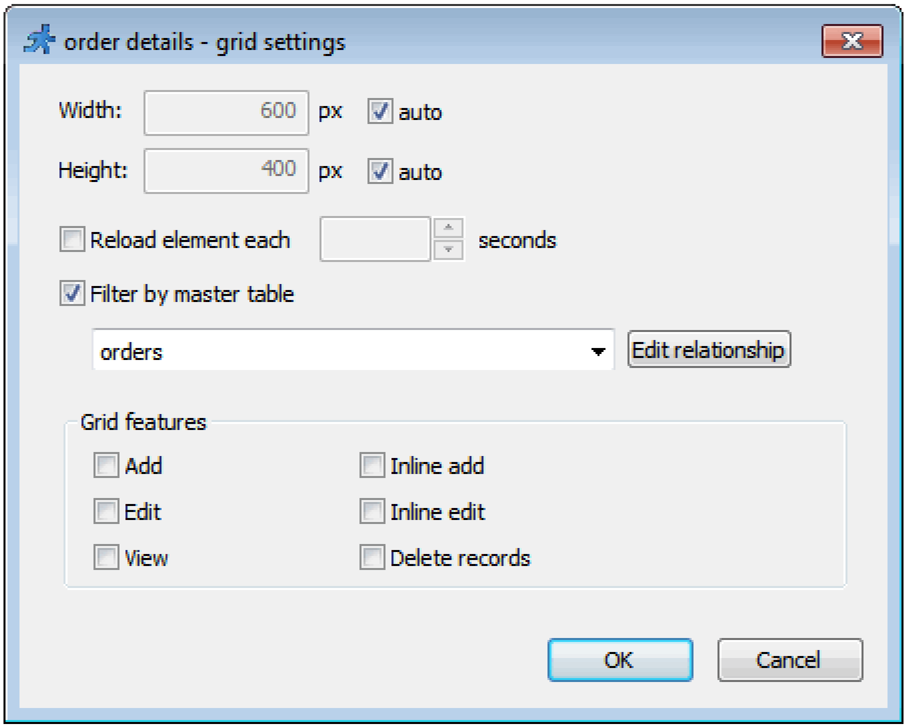
Dashboard where master table and details tables added as data grids

Let's create dashboard which displays data from nested master-details relationships Orders -> Order details -> Products, where selected order updates order details and selected record in order details grid updates products grid.

1. Add Order (master) table, Order details and Products (details) tables as data grids.



2. Click **settings** on the Order details grid element and enable the **Filter by master table** option. Select Orders as master table.



order details - grid settings

Width: 600 px ☑ auto

Height: 400 px ☑ auto

☐ Reload element each seconds

☑ Filter by master table

orders

Grid features

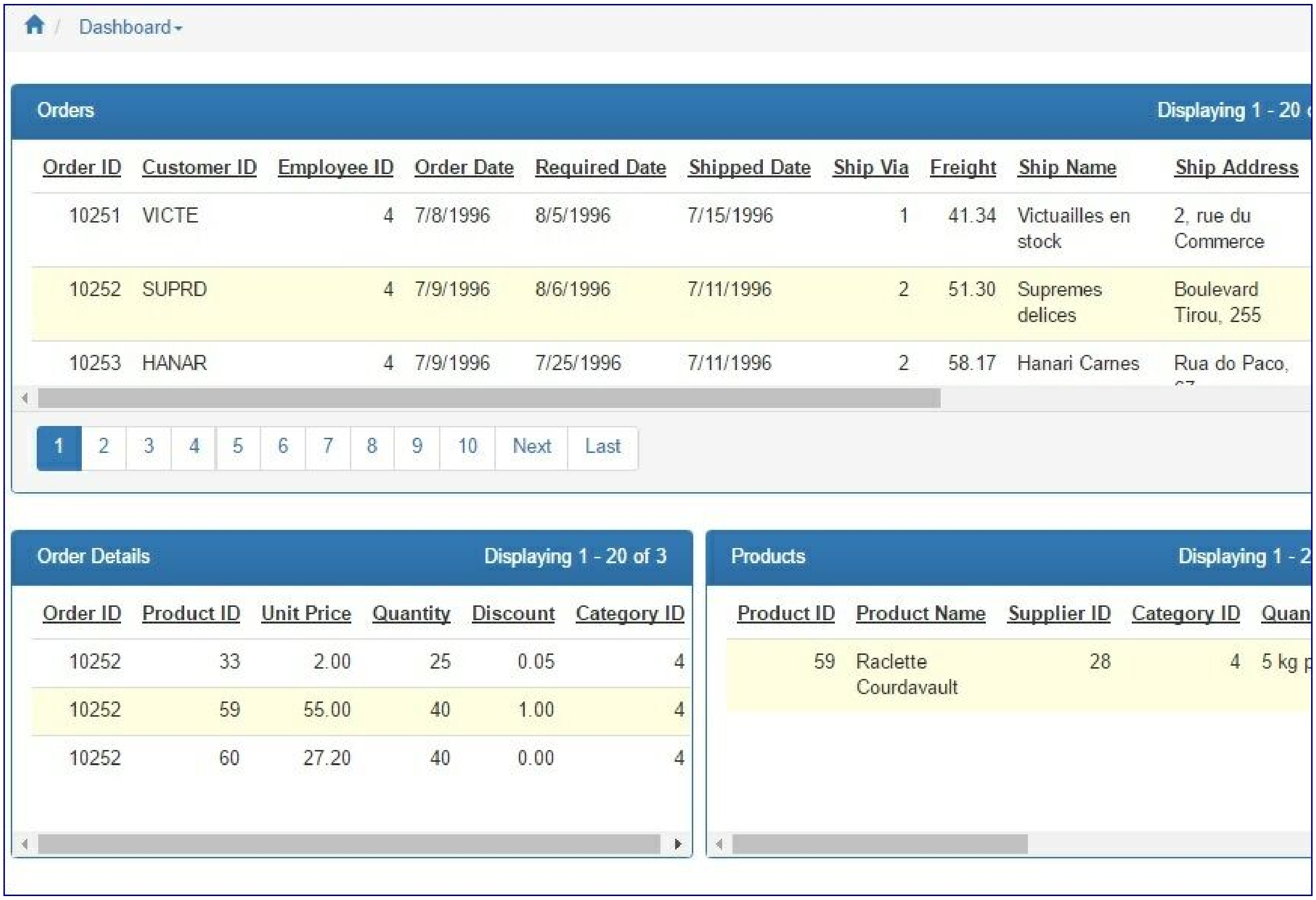
☐ Add ☐ Inline add

☐ Edit ☐ Inline edit

☐ View ☐ Delete records

3. Click **settings** on the Products grid element and enable the **Filter by master table** option. Select Order details as master table.

Dashboard page in a browser:



Orders — Displaying 1 - 20

Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date	Ship Via	Freight	Ship Name	Ship Address
10251	VICTE	4	7/8/1996	8/5/1996	7/15/1996	1	41.34	Victuailles en stock	2, rue du Commerce
10252	SUPRD	4	7/9/1996	8/6/1996	7/11/1996	2	51.30	Supremes delices	Boulevard Tirou, 255
10253	HANAR	4	7/9/1996	7/25/1996	7/11/1996	2	58.17	Hanari Carnes	Rua do Paco,

Order Details — Displaying 1 - 20 of 3

Order ID	Product ID	Unit Price	Quantity	Discount	Category ID
10252	33	2.00	25	0.05	4
10252	59	55.00	40	1.00	4
10252	60	27.20	40	0.00	4

Products — Displaying 1 - 2

Product ID	Product Name	Supplier ID	Category ID	Quan
59	Raclette Courdavault	28	4	5 kg p

Master-details relationship between chart and data grid

In the same way as with master and details tables grid in the dashboard you can create the dashboard with chart as a master and grid as details.

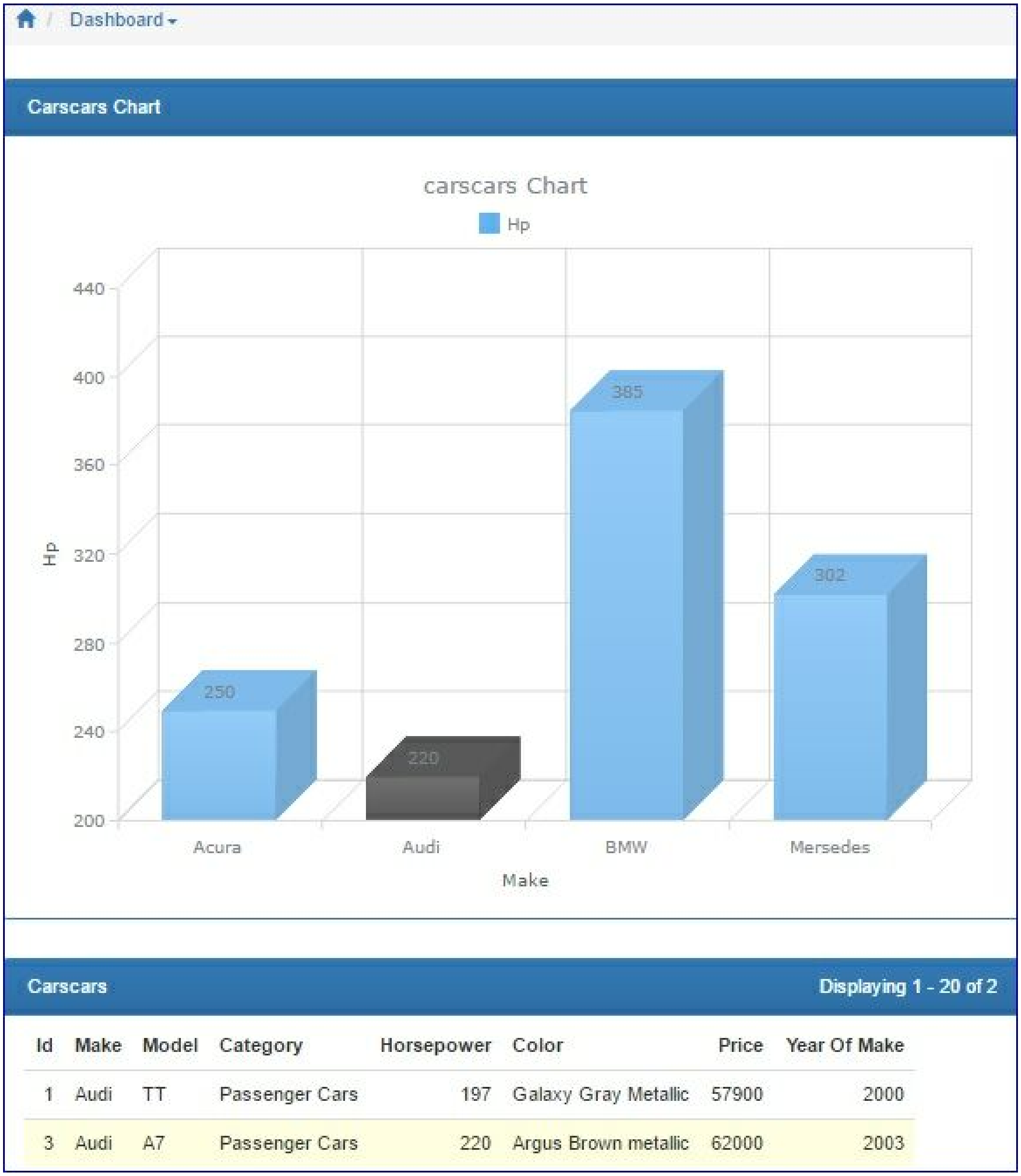
1. Create chart as a master and add details table to it as described in this article Charts and reports as master and details tables.

2. Create dashboard and add the chart to it.

3. Add the same chart to the dashboard but now choose **details** in the **Choose element to add** dialogue.

4. If necessary press **settings** and adjust chart or grid settings.

Dashboard page in a browser should look like this. Now when you click on the chart bar the details grid should be filtered.



Carscars Displaying 1 - 20 of 2

Id	Make	Model	Category	Horsepower	Color	Price	Year Of Make
1	Audi	TT	Passenger Cars	197	Galaxy Gray Metallic	57900	2000
3	Audi	A7	Passenger Cars	220	Argus Brown metallic	62000	2003

2.10.3 Dashboard search

Quick jump

Search across all dashboard items

Add a search page of any dashboard table as a dashboard item

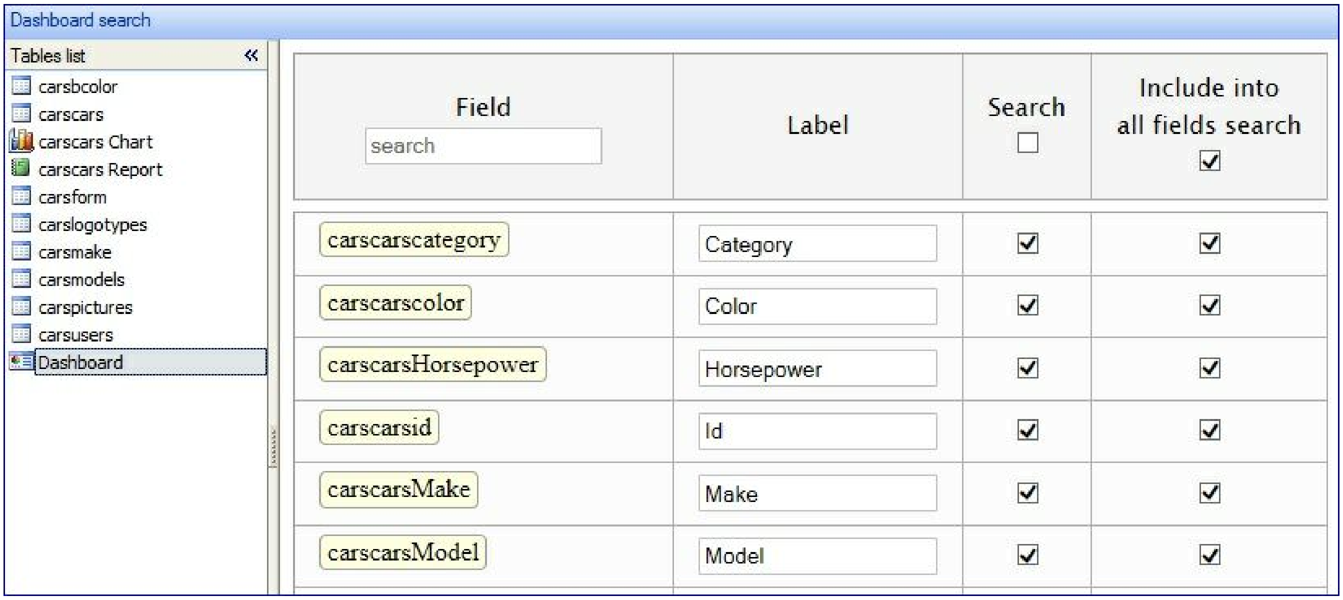
Dashboards feature two different types of search:

1. Search across all dashboard items. You can setup this on **Dashboard search** page choosing fields that will be searchable. This type of search will be available as a separate page in generated application and also as **Search all fields** search box on dashboard page.

2. Add a search page of any dashboard table as a dashboard item. When you add a new dashboard item select any table, view, chart or report and choose **Search** as an option. This option only makes sense if you also add this table/view/chart/report to the dashboard.

Search across all dashboard items

On **Dashboard search** screen define which fields will be searchable.


Dashboard search

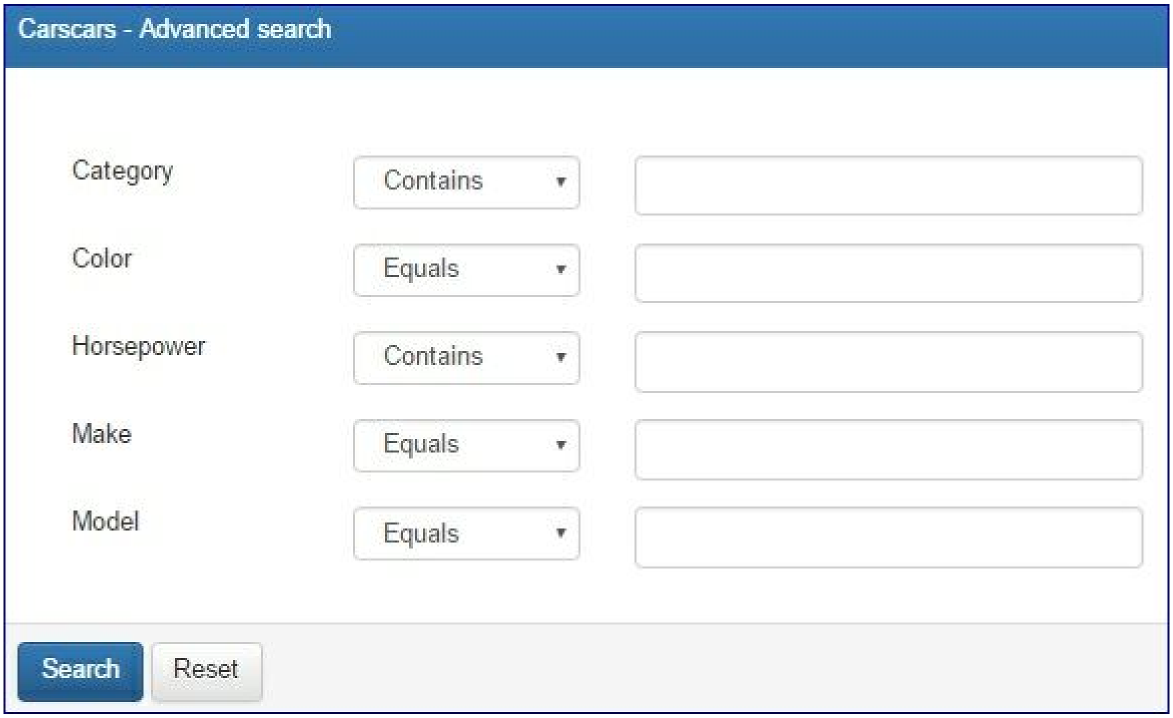
Tables list

- carsbcolor
- carscars
- carscars Chart
- carscars Report
- carsform
- carslogotypes
- carsmake
- carsmodels
- carspictures
- carsusers
- Dashboard

Field search	Label	Search ☐	Include into all fields search ☑
carscarscategory	Category	☑	☑
carscarscolor	Color	☑	☑
carscarsHorsepower	Horsepower	☑	☑
carscarsid	Id	☑	☑
carscarsMake	Make	☑	☑
carscarsModel	Model	☑	☑

You can drag fields from different tables/elements into a single cell. In this case search will be carried out through all the elements that belong to the selected fields. For example, you can search Orders table and Orders Chart at the same time.

Search option allows to add fields to the Advanced search page.



Carscars - Advanced search

Category Contains

Color Equals

Horsepower Contains

Make Equals

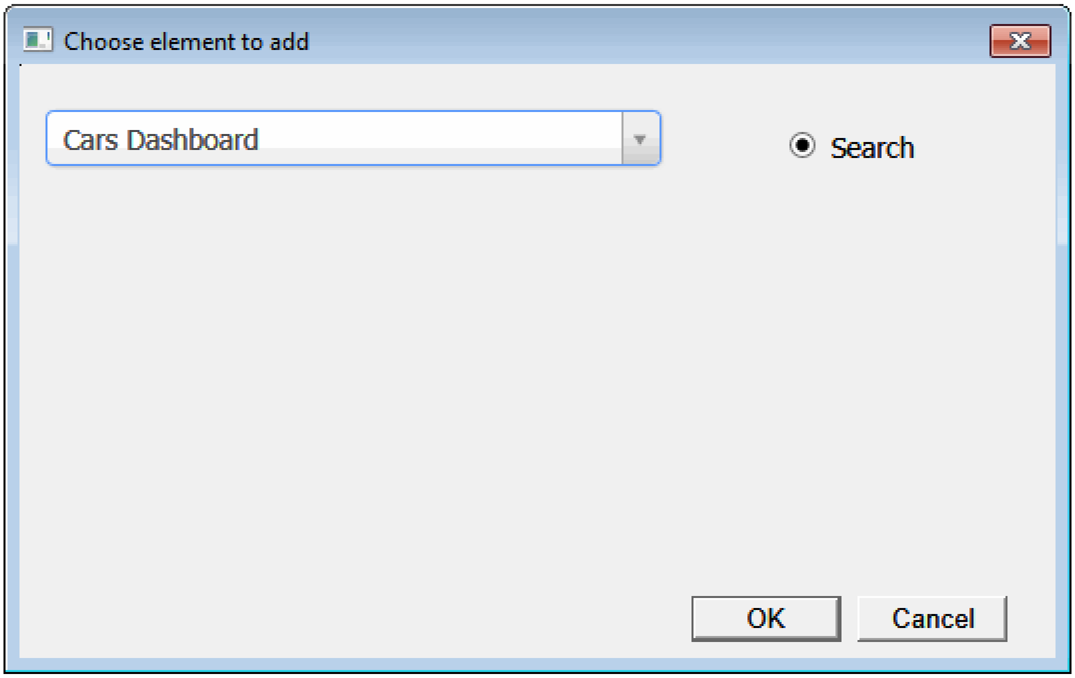
Model Equals

Include into all fields search option allows to add fields to the 'All fields' quick search.

To make some dashboard elements do not display the data initially, apply the **Hide data until search** option to them in Search and Filter settings on the **Choose fields** screen.

Add dashboard search page to a dashboard

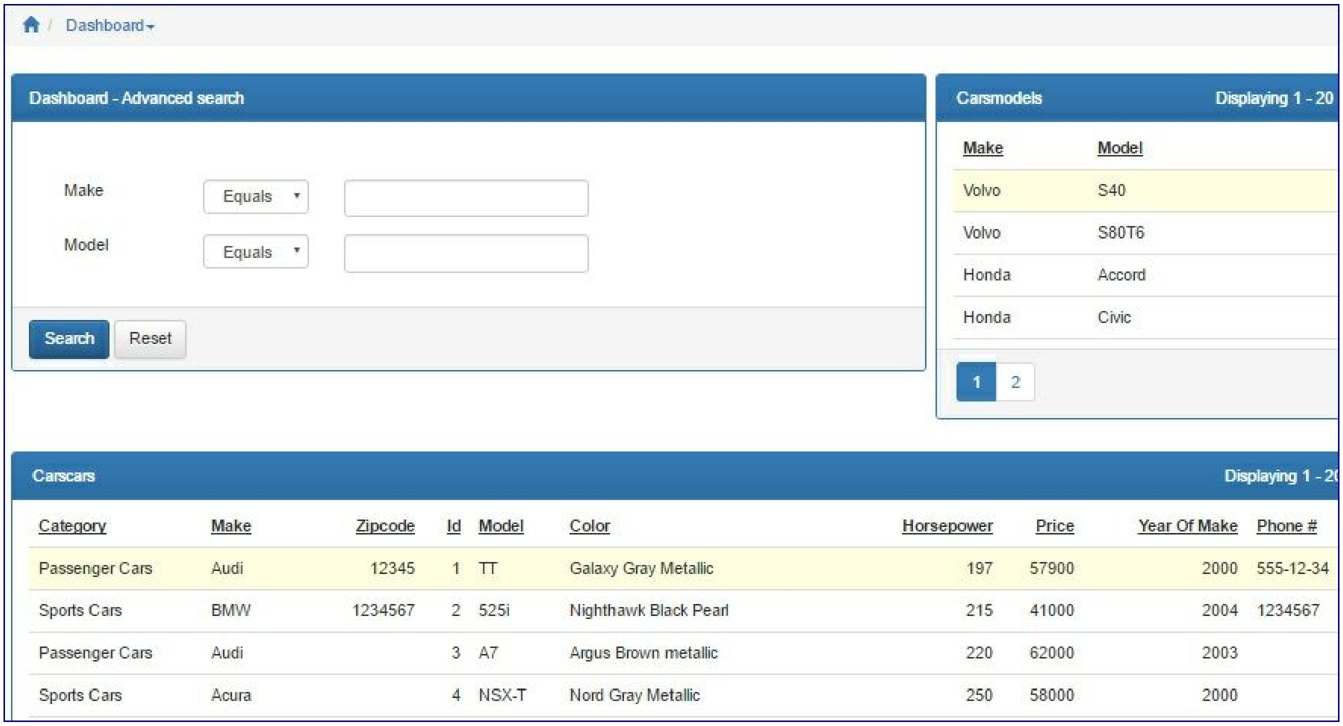
You can add dashboard advanced search page to a dashboard as one of the elements. On the **Dashboard layout** screen click **add** and select the dashboard itself.



Cars Dashboard

● Search

An example of the dashboard with search:



Dashboard - Advanced search

Make Equals

Model Equals

Carsmodels Displaying 1 - 20

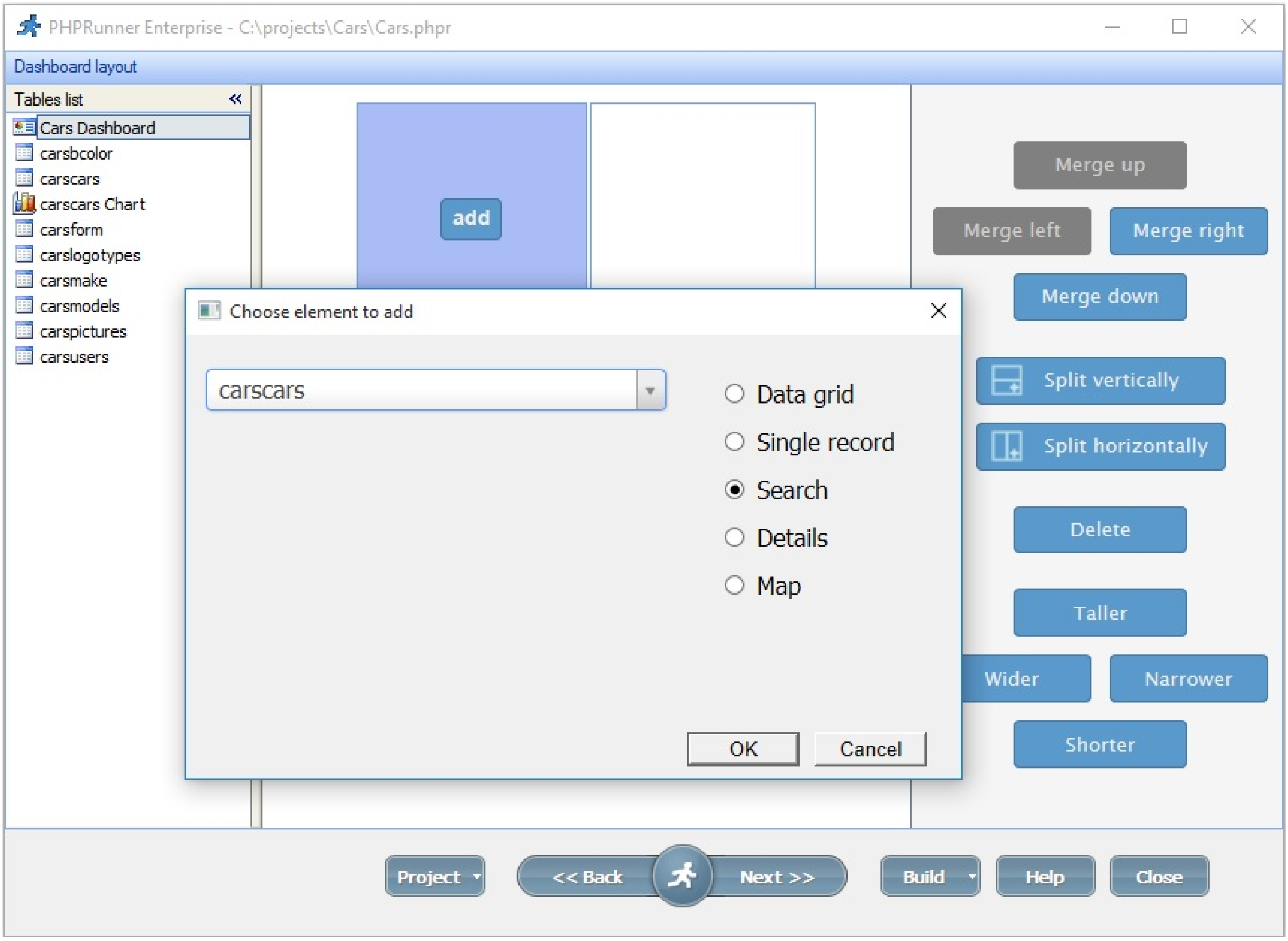
Make	Model
Volvo	S40
Volvo	S80T6
Honda	Accord
Honda	Civic

Carscars Displaying 1 - 20

Category	Make	Zipcode	Id	Model	Color	Horsepower	Price	Year Of Make	Phone #
Passenger Cars	Audi	12345	1	TT	Galaxy Gray Metallic	197	57900	2000	555-12-34
Sports Cars	BMW	1234567	2	525i	Nighthawk Black Pearl	215	41000	2004	1234567
Passenger Cars	Audi		3	A7	Argus Brown metallic	220	62000	2003	
Sports Cars	Acura		4	NSX-T	Nord Gray Metallic	250	58000	2000	

Add a search page of any dashboard table as a dashboard item

To add a search page to a dashboard, click **add**, select any table/view/chart/report and choose **Search** as an option.



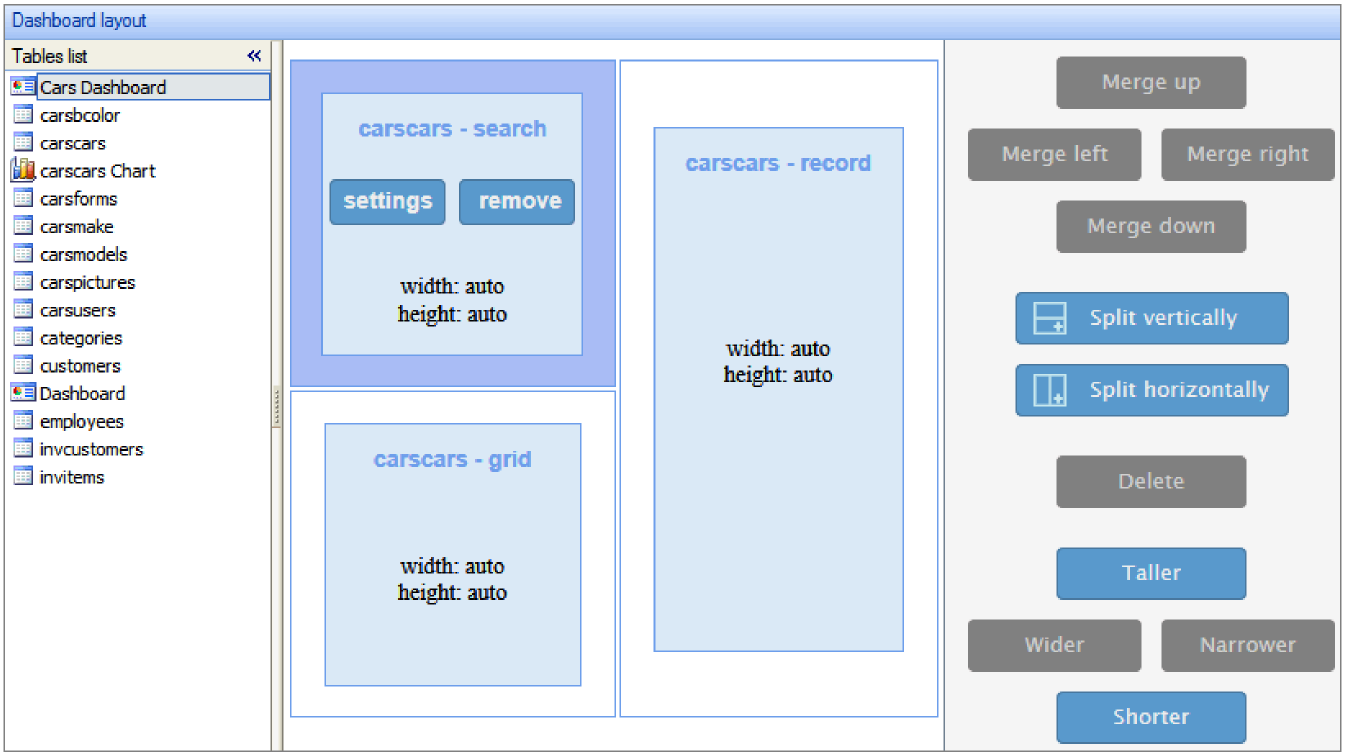
Choose element to add

carscars

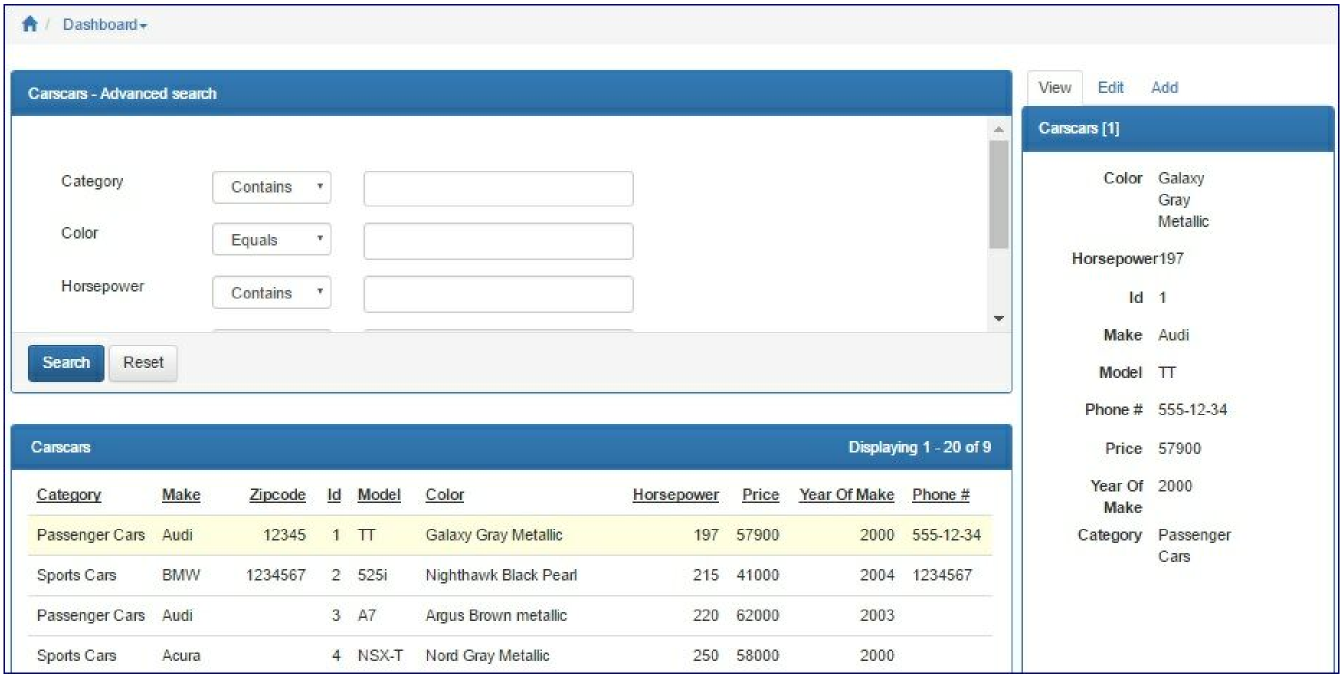
○ Data grid
○ Single record
● Search
○ Details
○ Map

OK Cancel

Make sure you've also added selected table/view/chart/report to the dashboard as data grid, single record, chart or report.



How it looks in a browser:



Carscars [1]

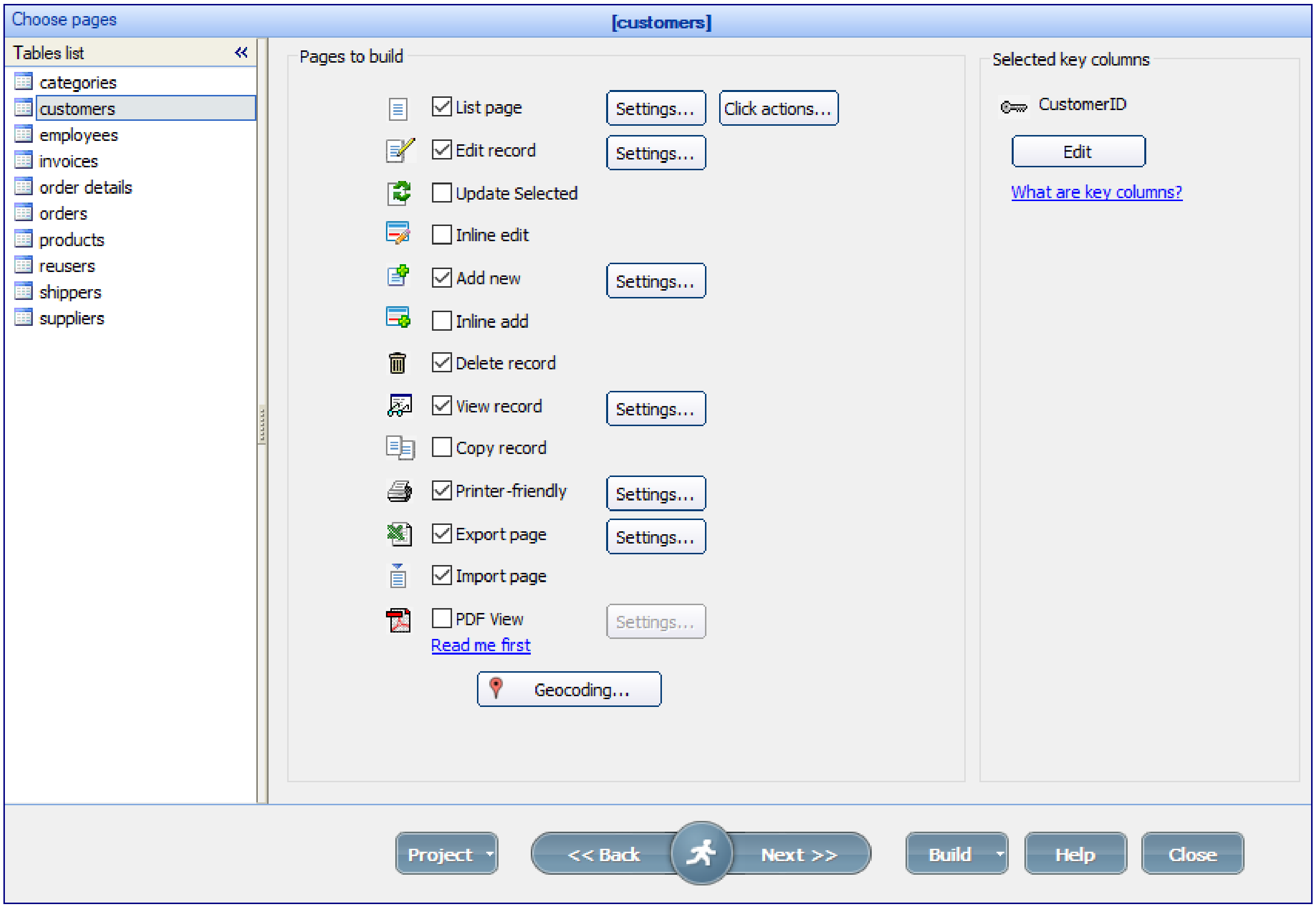
Color	Galaxy Gray Metallic
Horsepower	197
Id	1
Make	Audi
Model	TT
Phone #	555-12-34
Price	57900
Year Of Make	2000
Category	Passenger Cars

Carscars Displaying 1 - 20 of 9

Category	Make	Zipcode	Id	Model	Color	Horsepower	Price	Year Of Make	Phone #
Passenger Cars	Audi	12345	1	TT	Galaxy Gray Metallic	197	57900	2000	555-12-34
Sports Cars	BMW	1234567	2	525i	Nighthawk Black Pearl	215	41000	2004	1234567
Passenger Cars	Audi		3	A7	Argus Brown metallic	220	62000	2003	
Sports Cars	Acura		4	NSX-T	Nord Gray Metallic	250	58000	2000	

2.11 Choose pages

To create new pages select the corresponding check boxes under **Pages to build.** If check boxes near **View record**, **Edit record**, **Delete record**, **Copy record** options are disabled, you need to start by selecting key columns.



Choose pages [customers]

Tables list

- categories
- customers
- employees
- invoices
- order details
- orders
- products
- reusers
- shippers
- suppliers

Pages to build

☑ List page
☑ Edit record
☐ Update Selected
☐ Inline edit
☑ Add new
☐ Inline add
☑ Delete record
☑ View record
☐ Copy record
☑ Printer-friendly
☑ Export page
☑ Import page
☐ PDF View

Selected key columns

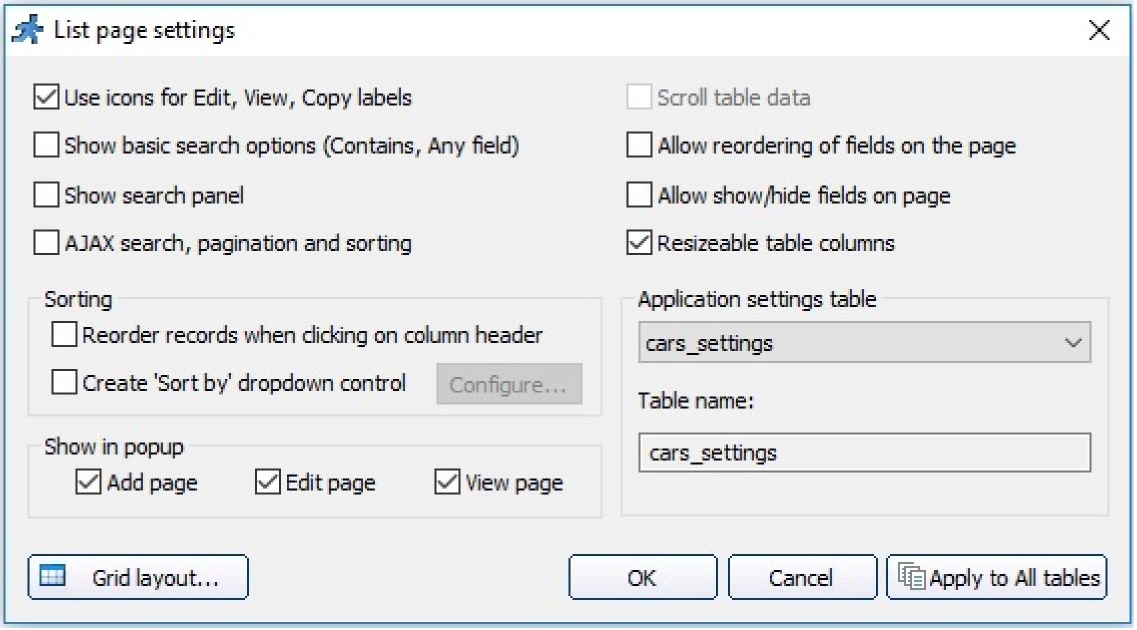
CustomerID

Inline add/**Inline edit** options allow you to add/edit multiple records without leaving the List page. Inline Add/Edit functions use the same **Edit as** types as the regular Add/Edit pages.

Click **Settings** next to the **List page**, **Edit record, Add new, View record**, **Printer-friendly** or **PDF view** check box to adjust page settings.

2.11.1 List page settings / Click actions

List page settings



☑ Use icons for Edit, View, Copy labels

☐ Show basic search options (Contains, Any field)

☐ Show search panel

☐ AJAX search, pagination and sorting

☐ Scroll table data

☐ Allow reordering of fields on the page

☐ Allow show/hide fields on page

☑ Resizeable table columns

Sorting

☐ Reorder records when clicking on column header

☐ Create 'Sort by' dropdown control

Show in popup

☑ Add page ☑ Edit page ☑ View page

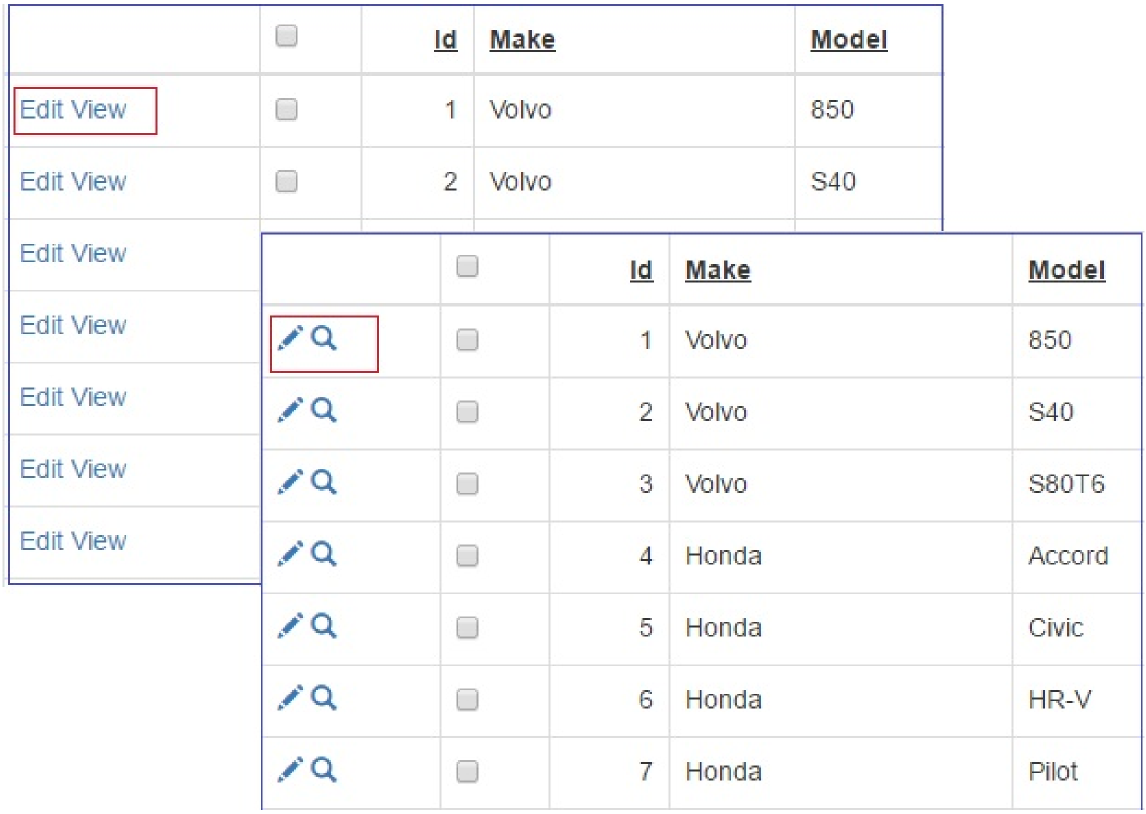
Application settings table

cars_settings

Table name: cars_settings

You can select the table to store application settings like order of columns, show/hide state, columns size and also saved searches. These settings are user specific.

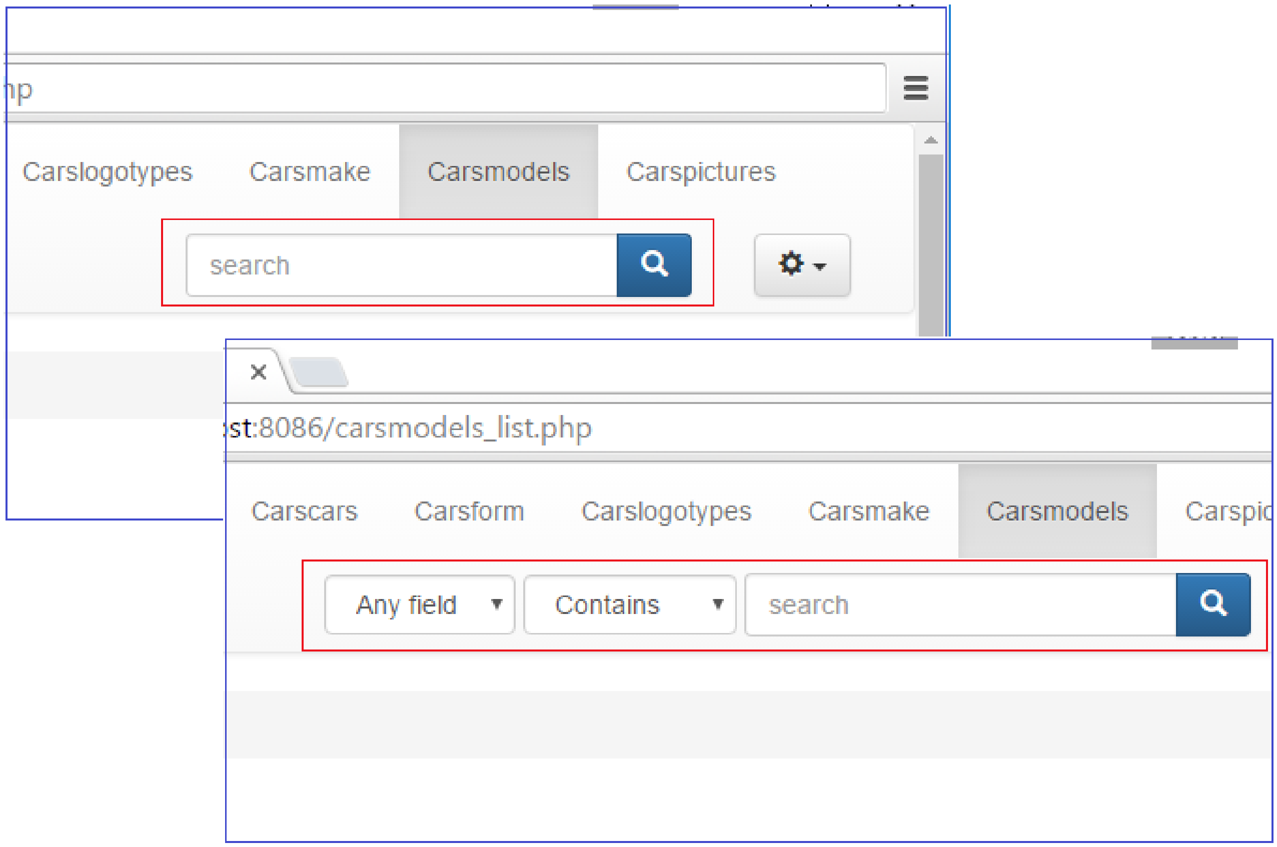
- **Use icons for Edit, View, Copy labels** - if this option is enabled, icons are displayed for *Edit, View, Copy* actions instead of text labels.



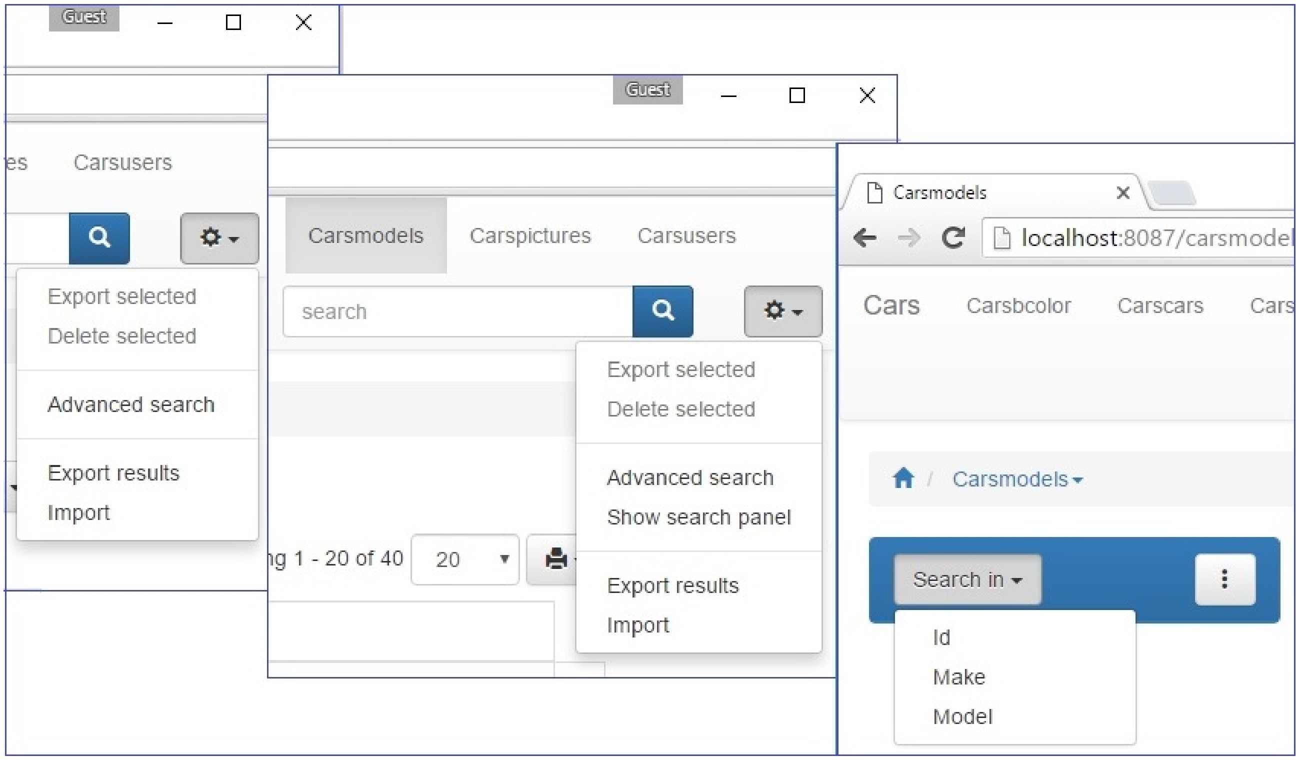
	☐	Id	Make	Model
Edit View	☐	1	Volvo	850
Edit View	☐	2	Volvo	S40
Edit View				
Edit View				
Edit View				
Edit View				
Edit View				

	☐	Id	Make	Model
✎ 🔍	☐	1	Volvo	850
✎ 🔍	☐	2	Volvo	S40
✎ 🔍	☐	3	Volvo	S80T6
✎ 🔍	☐	4	Honda	Accord
✎ 🔍	☐	5	Honda	Civic
✎ 🔍	☐	6	Honda	HR-V
✎ 🔍	☐	7	Honda	Pilot

- **Show basic search options** - if this option is enabled, basic search options (*Contains*, *Any* field) are displayed instead of just one *Search* field.



- **Show search panel** - if this option is enabled, search panel is displayed. Search panel allows you to manage the search criteria. For more information, see Web interface guide.



- **AJAX search, pagination and sorting** - this option enables AJAX search, pagination and sorting that allows updating data without loading the whole page again.

- **'Sort by' dropdown control**

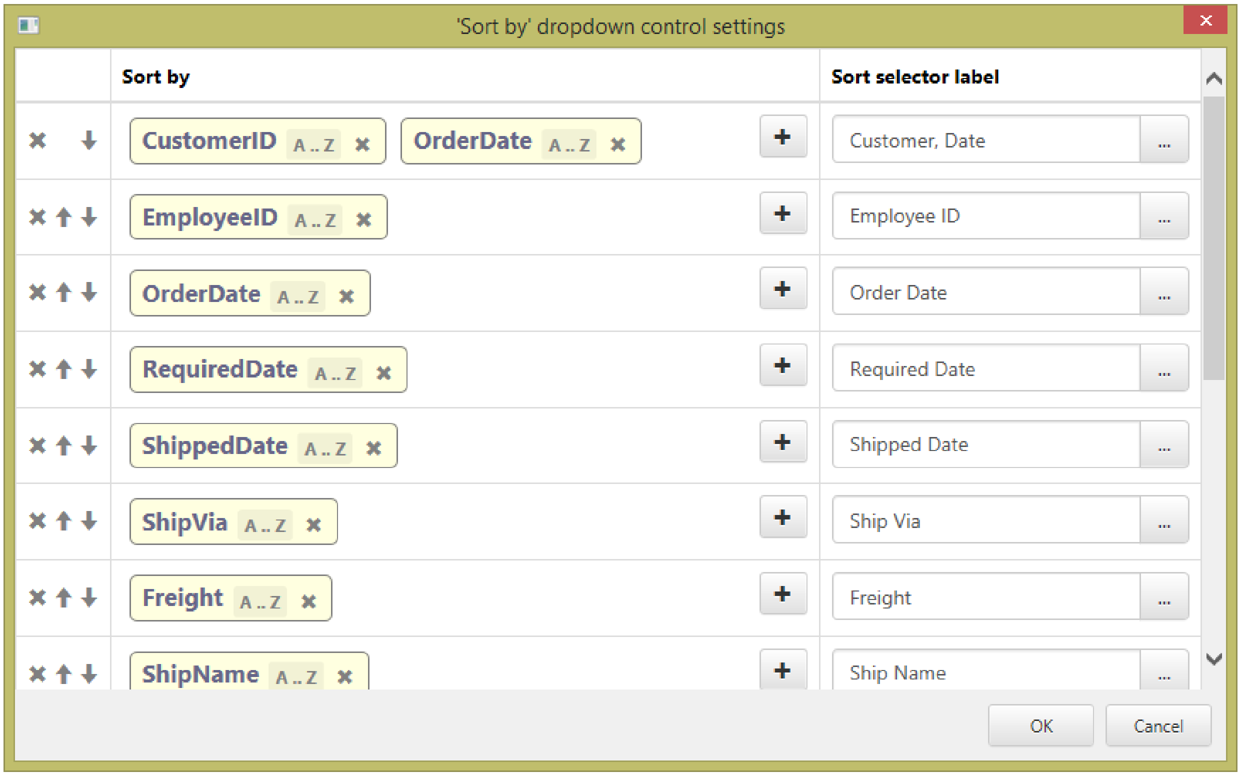
  This feature will only work with Bootstrap layout.
  When this option is useful:

  - Vertical or columns List page mode that previously didn't have sorting options

  - 'Sort by' control in mobile mode

  - Need to setup application specific sort modes

  You can also choose if users still can sort data by clicking on column headers.

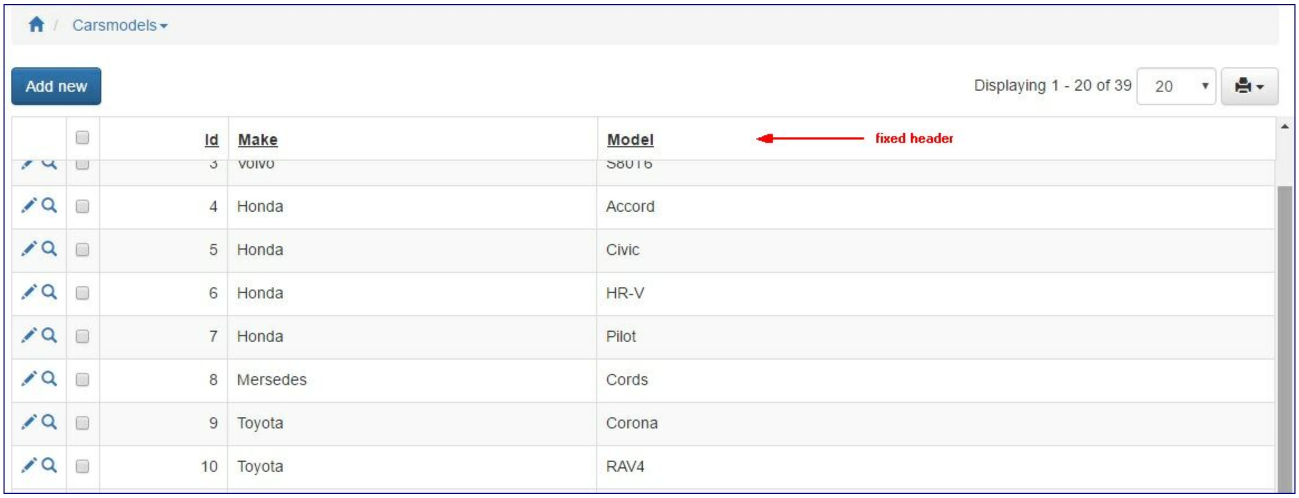


'Sort by' dropdown control settings

Sort by	Sort selector label
CustomerID A..Z, OrderDate A..Z	Customer, Date
EmployeeID A..Z	Employee ID
OrderDate A..Z	Order Date
RequiredDate A..Z	Required Date
ShippedDate A..Z	Shipped Date
ShipVia A..Z	Ship Via
Freight A..Z	Freight
ShipName A..Z	Ship Name

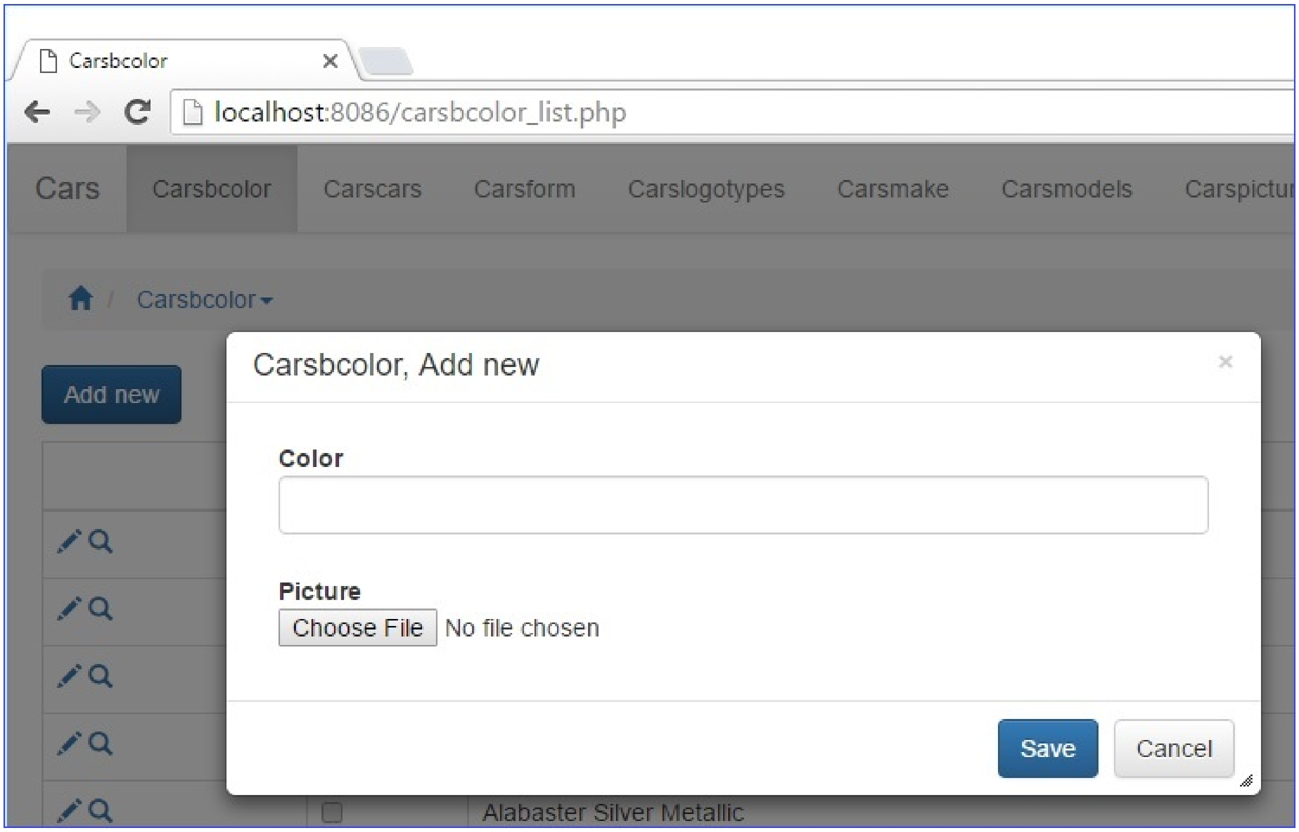
Here is how it looks in generated application.

- **Scroll table data** - if enabled, data records are displayed as scrollable table with fixed table header.



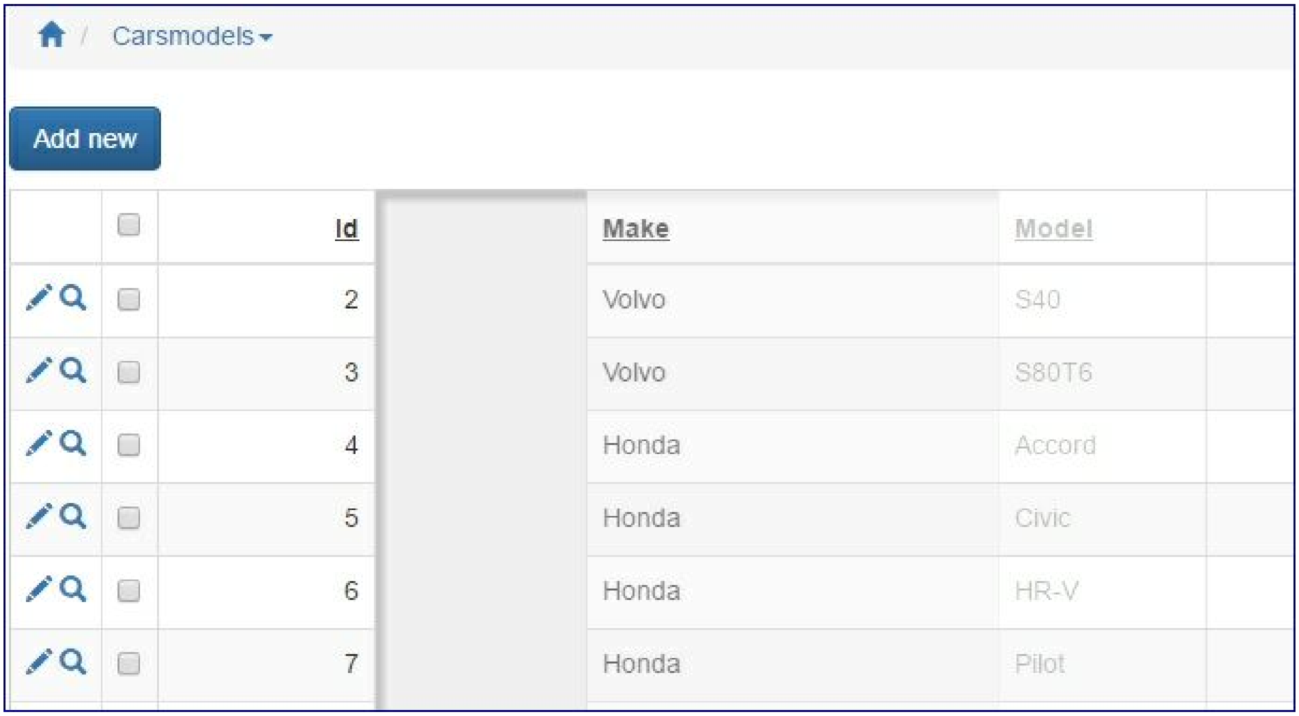
Id	Make	Model
3	Volvo	S80T6
4	Honda	Accord
5	Honda	Civic
6	Honda	HR-V
7	Honda	Pilot
8	Mersedes	Cords
9	Toyota	Corona
10	Toyota	RAV4

- **Show in popup** - this option allows to show Add/Edit/View pages in popup window.



- **Allow reordering of fields on the page.**

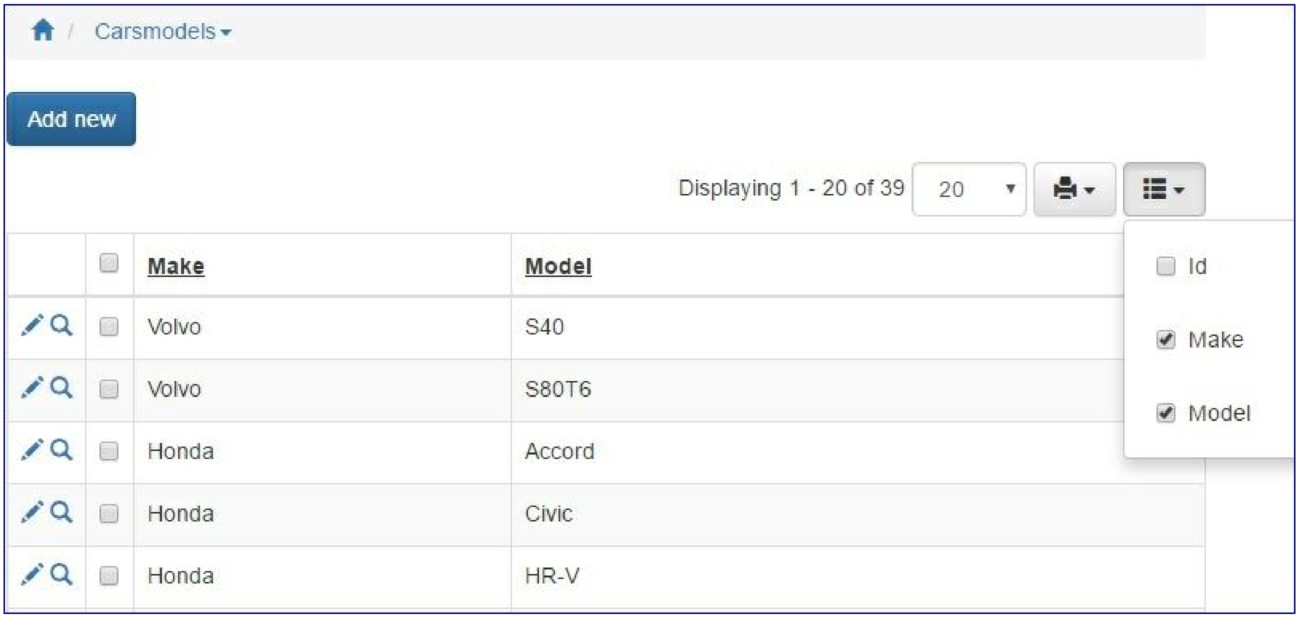
Drag-n-drop columns on the List page according to your needs. Your settings will be saved in the database and next time you open this page you will see the same layout. **This feature will only work with Bootstrap layout.**



	□	Id		Make	Model
	□	2		Volvo	S40
	□	3		Volvo	S80T6
	□	4		Honda	Accord
	□	5		Honda	Civic
	□	6		Honda	HR-V
	□	7		Honda	Pilot

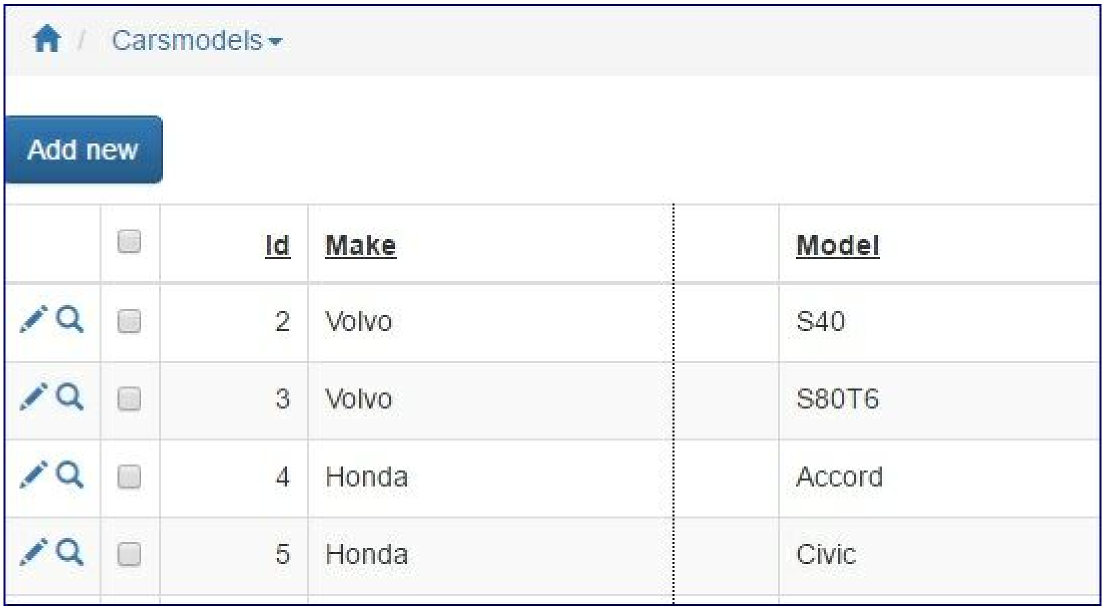
- **Allow show/hide fields on page**

Choose what columns to show on each page in run time. Settings are also saved in the database and preserved between sessions. Each user has its own set of settings. **This feature will only work with Bootstrap layout.**



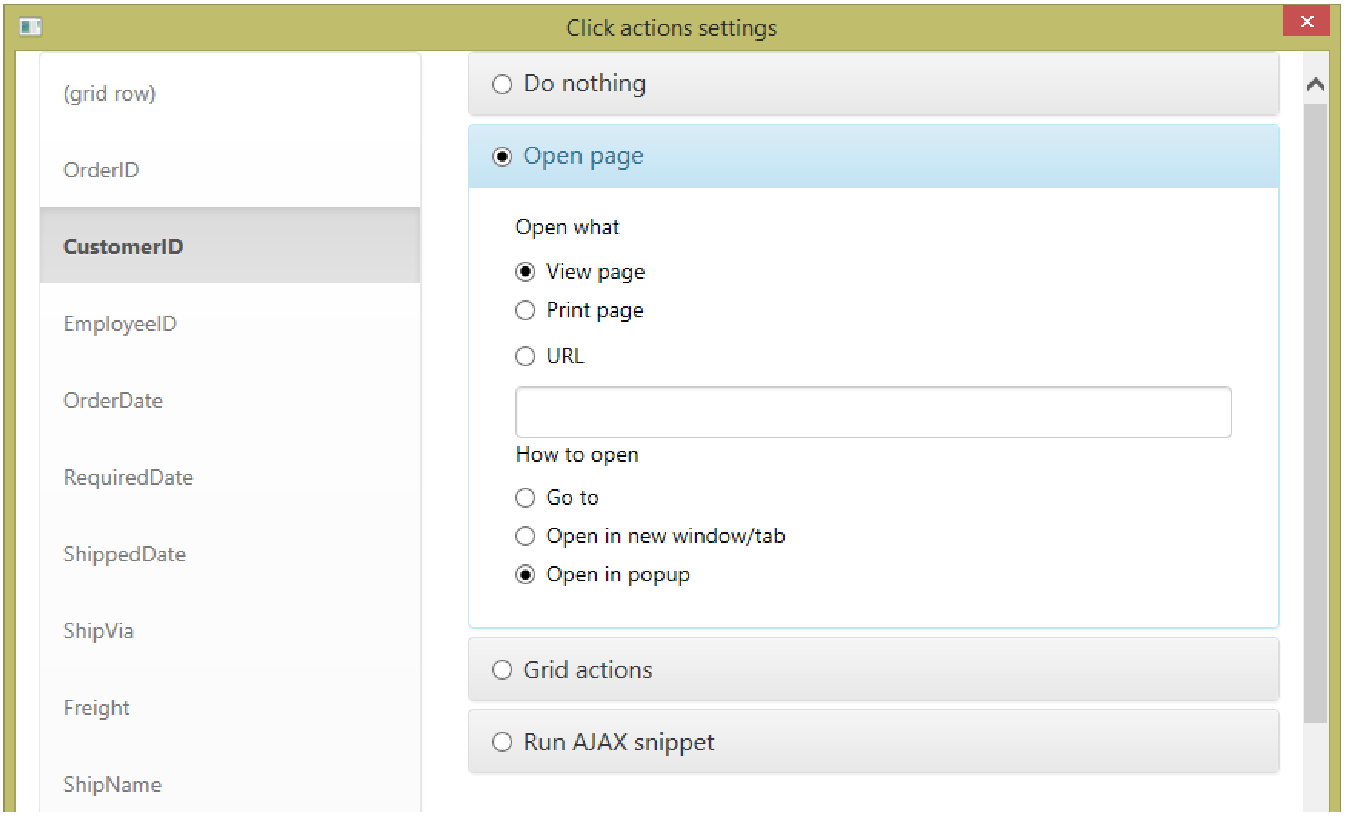
	□	Make	Model
	□	Volvo	S40
	□	Volvo	S80T6
	□	Honda	Accord
	□	Honda	Civic
	□	Honda	HR-V

- **Resizeable table columns**



	□	Id	Make		Model
	□	2	Volvo		S40
	□	3	Volvo		S80T6
	□	4	Honda		Accord
	□	5	Honda		Civic

List page Click actions



○ Do nothing

● Open page

Open what

● View page

○ Print page

○ URL

How to open

○ Go to

○ Open in new window/tab

● Open in popup

○ Grid actions

○ Run AJAX snippet

This one is pretty cool, you can assign actions like open a certain page, make record selected, expand/collapse details or run a custom code to row/cell click.

Check this live demo.

Click CustomerID cell to open Orders view page in popup.

Click OrderID field to retrieve current order total and display it in OrderID field.

Code for this example:

Client Before event

```
// pass OrderID to Server event
params["OrderID"] = row.getFieldValue("OrderID");
```

Server event

```
// run SQL Query to retrieve order total
$result["total"] = DBLookup("select sum(Quantity*UnitPrice) from
`Order Details` where OrderID=".$params["OrderID"]);
```

Client After event

```
// change cell background
row.fieldCell("OrderID").css('background', 'yellow' );
// add order total to OrderID cell content
row.fieldCell("OrderID").html(
   row.fieldCell("OrderID").html()+"<br>Total: "+result["total"]);
```

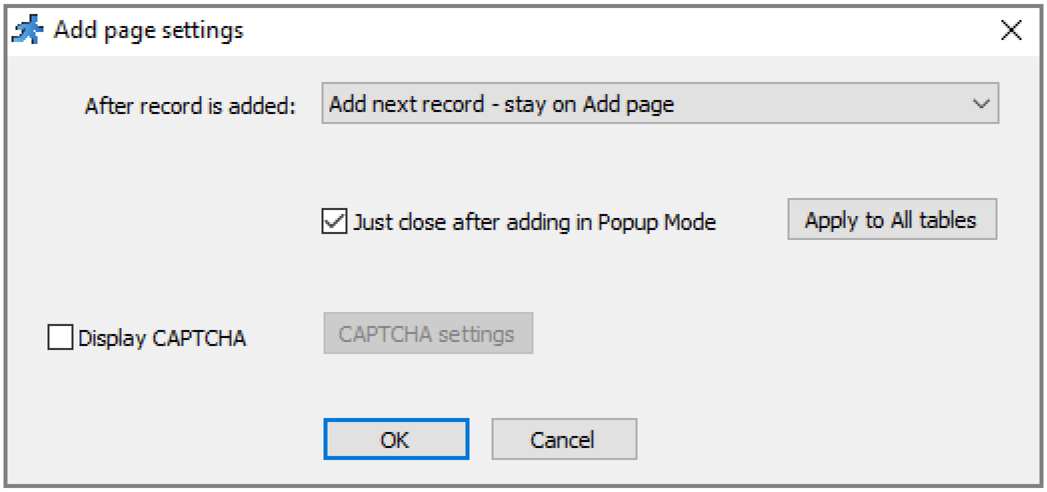
2.11.2 Add/Edit page settings

You can choose one of the action to be performed after the record is added or updated.

Add page settings

After record is added options:

- Return to the list page;
- Add next record - stay on Add page (the default action);
- Open the new record View page;
- Open the new record Edit page;



Add page settings

After record is added: Add next record - stay on Add page

☑ Just close after adding in Popup Mode

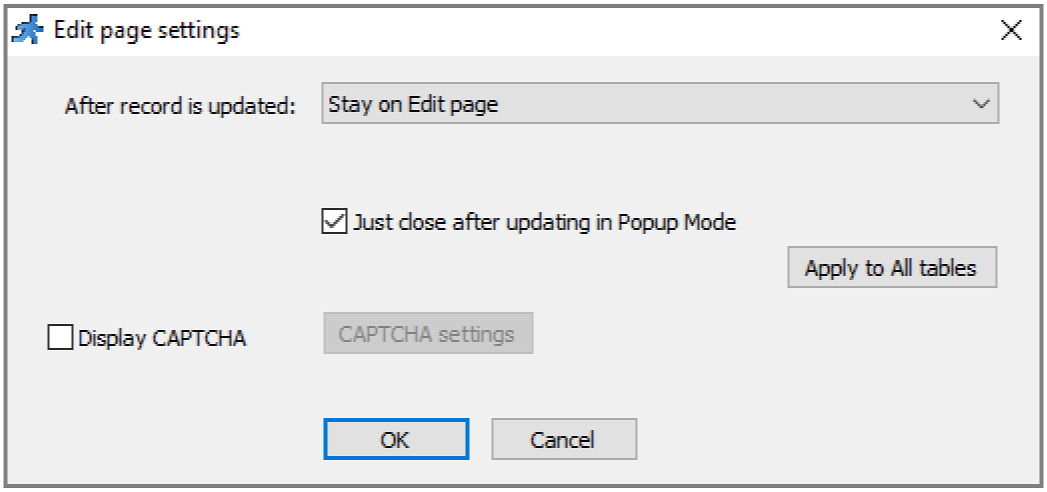
□ Display CAPTCHA

Edit page settings

After record is updated options:

- Return to the list page;
- Stay on Edit page (the default action);
- Go to the View page;
- Edit next record;

- Edit previous record;


Edit page settings

After record is updated: Stay on Edit page

☑ Just close after updating in Popup Mode

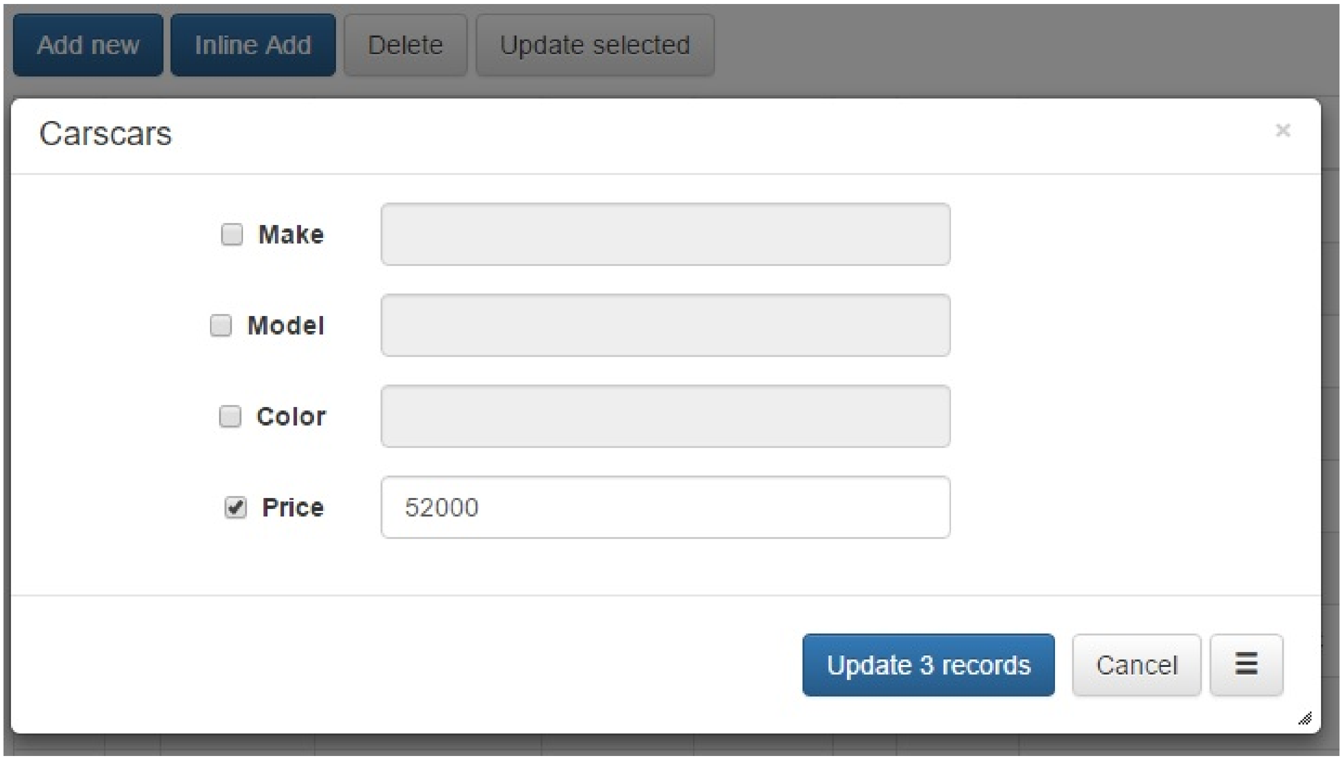
☐ Display CAPTCHA

Enable the **Display CAPTCHA** option and change its default settings if you want to use CAPTCHA on Add and Edit pages.

2.11.3 Update selected page

Update selected page settings

You can quickly make changes to multiple records. You can choose fields to appear on 'Update selected' dialog. Depending on Edit page settings, 'Update selected' page can be shown either in popup or as a separate page. **This feature will only work with Bootstrap layout.**



□ Make

□ Model

□ Color

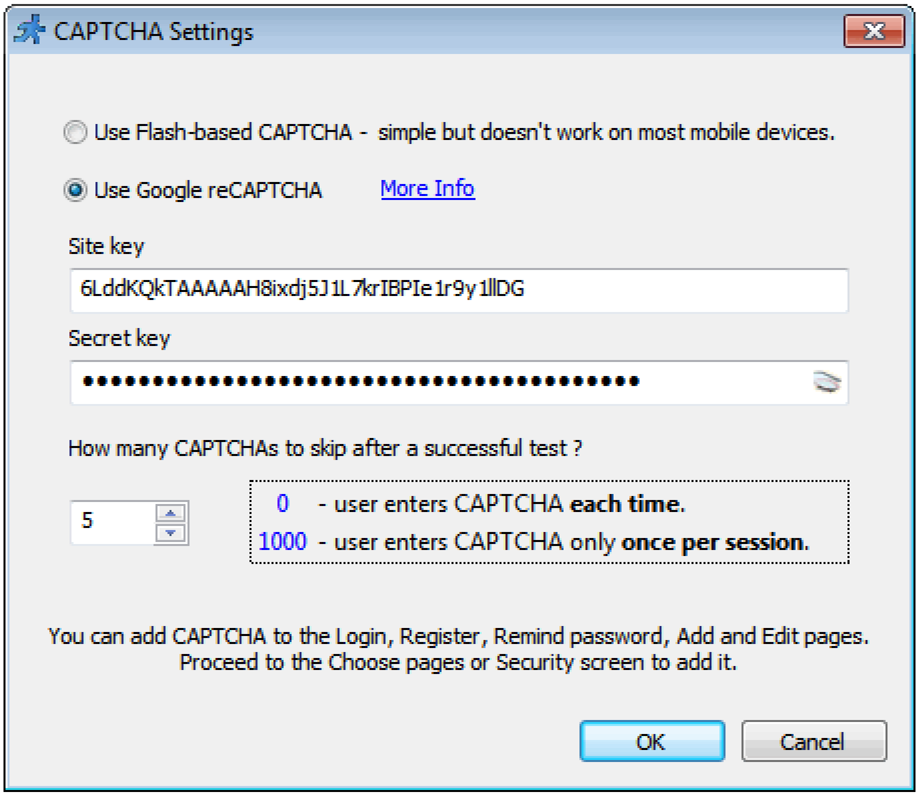
☑ Price 52000

2.11.4 CAPTCHA on Add/Edit page

Adding CAPTCHA

You can add CAPTCHA to Add and Edit pages (select the **Display CAPTCHA** checkbox). CAPTCHA is a simple test to determine if a user is a computer or a human. It is used to prevent spam abuse on the websites. Click **CAPTCHA settings** and choose what CAPTCHA to use: Flash-based CAPTCHA (it is simple but doesn't work on most mobile devices) or Google reCAPTCHA.

To use Google reCAPTCHA, register your web site at https://www.google.com/recaptcha/intro/index.html, copy site key and secret key there and paste them into the CAPTCHA settings.



○ Use Flash-based CAPTCHA - simple but doesn't work on most mobile devices.

● Use Google reCAPTCHA

Site key

6LddKQkTAAAAAH8ixdj5J1L7krIBPIe1r9y1llDG

Secret key

••

How many CAPTCHAs to skip after a successful test ?

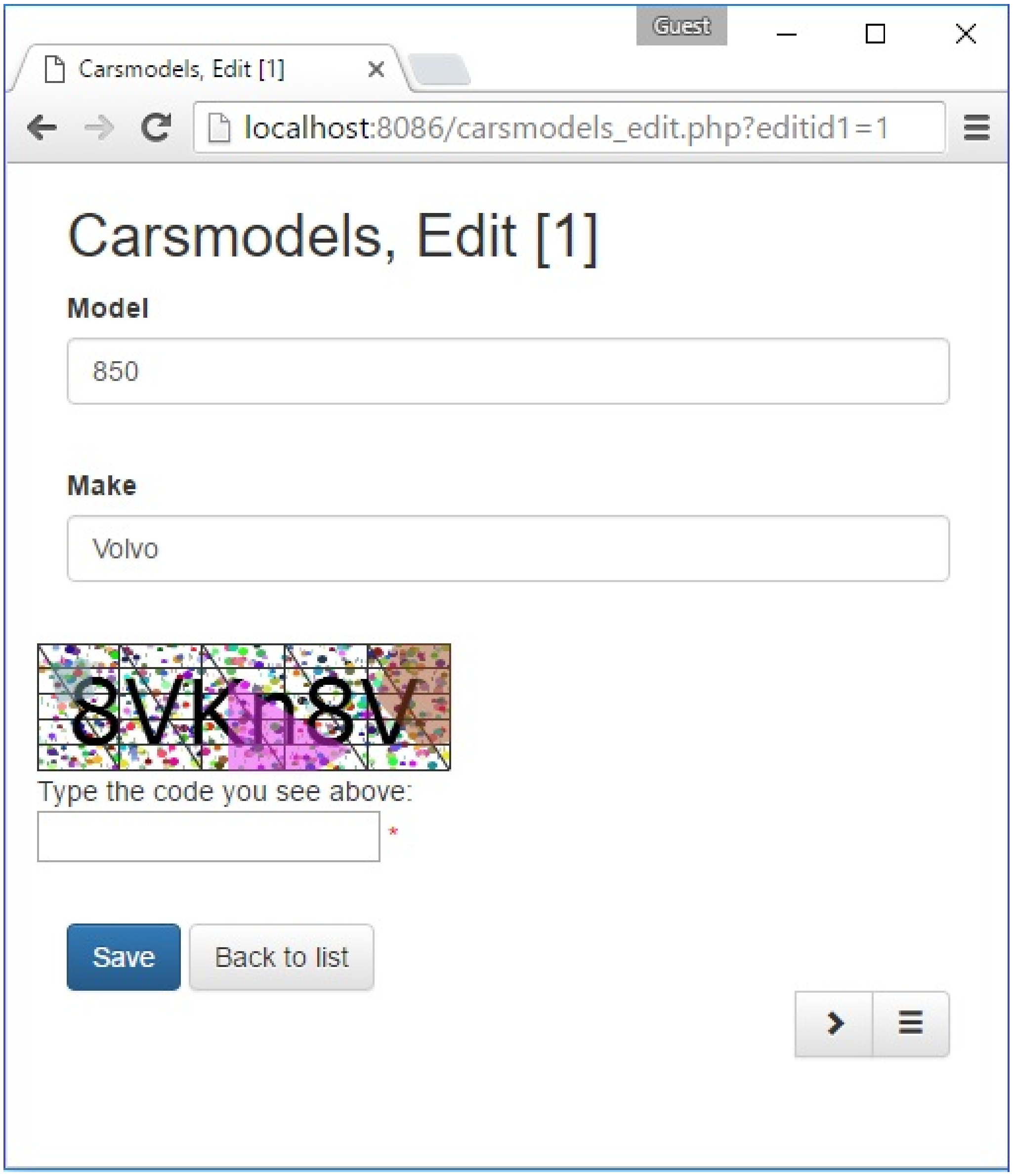
5

0 - user enters CAPTCHA **each time.**

1000 - user enters CAPTCHA only **once per session.**

You can add CAPTCHA to the Login, Register, Remind password, Add and Edit pages.
Proceed to the Choose pages or Security screen to add it.

Flash-based CAPTCHA:



Carsmodels, Edit [1]

Model

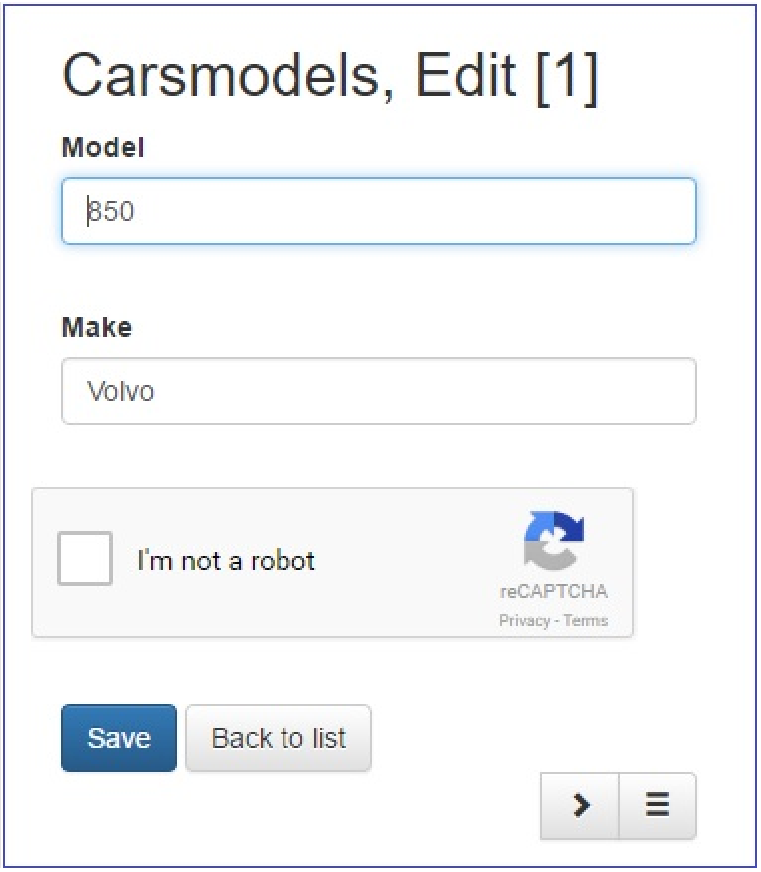
850

Make

Volvo

Type the code you see above:

Google reCAPTCHA:


Carsmodels, Edit [1]

Model: 850

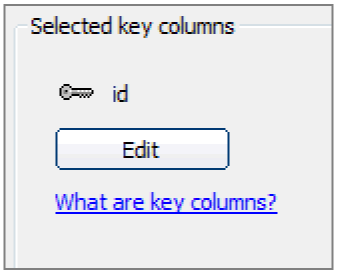
Make: Volvo

2.11.5 Key columns

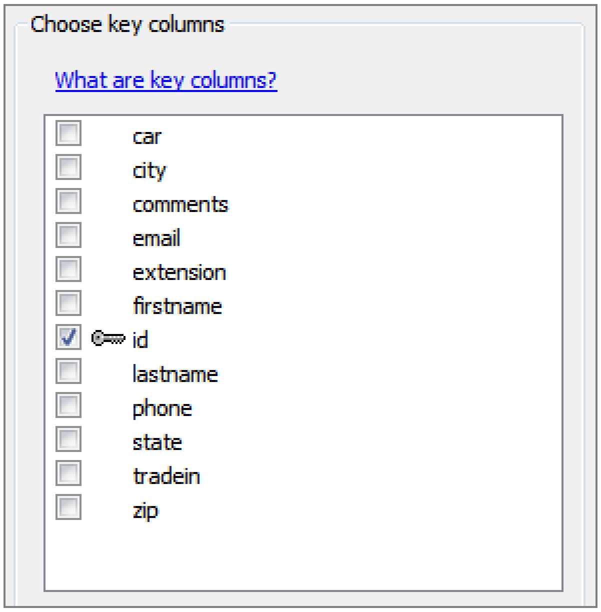
Key columns

Key column is the table field which lets you query each table row individually and modify each row without altering other rows in the same table. The values that compose a key column are unique; no two values are the same. You can specify any numbers of the columns that form a key. Generally, one key column is sufficient.

In your application primary key is required for those tables that need Edit, Delete or View functionality, need to work with images or provide functionality like Print/Export selected records. Do not to remove the key columns from the query.



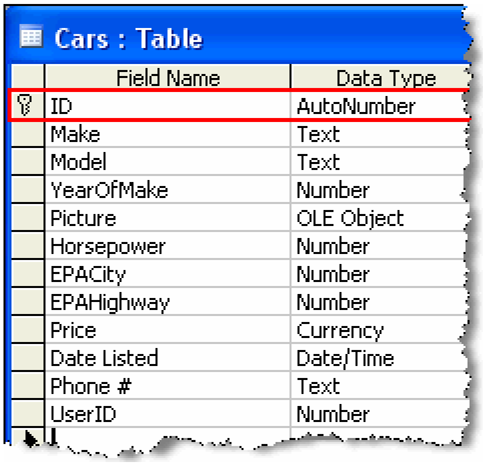
To change selected key columns, click **Edit**.



☐ car
☐ city
☐ comments
☐ email
☐ extension
☐ firstname
☑ id
☐ lastname
☐ phone
☐ state
☐ tradein
☐ zip

The best option is to use autoincrement field as a primary key. Your database will take of generating the proper key value every time you add a new record.

In MS Access the best choice for key column is auto number field.



Field Name	Data Type
ID	AutoNumber
Make	Text
Model	Text
YearOfMake	Number
Picture	OLE Object
Horsepower	Number
EPACity	Number
EPAHighway	Number
Price	Currency
Date Listed	Date/Time
Phone #	Text
UserID	Number

In SQL Server use INT IDENTITY field, in MySQL use INT AUTO_INCREMENT field.

2.11.6 Export/Import pages

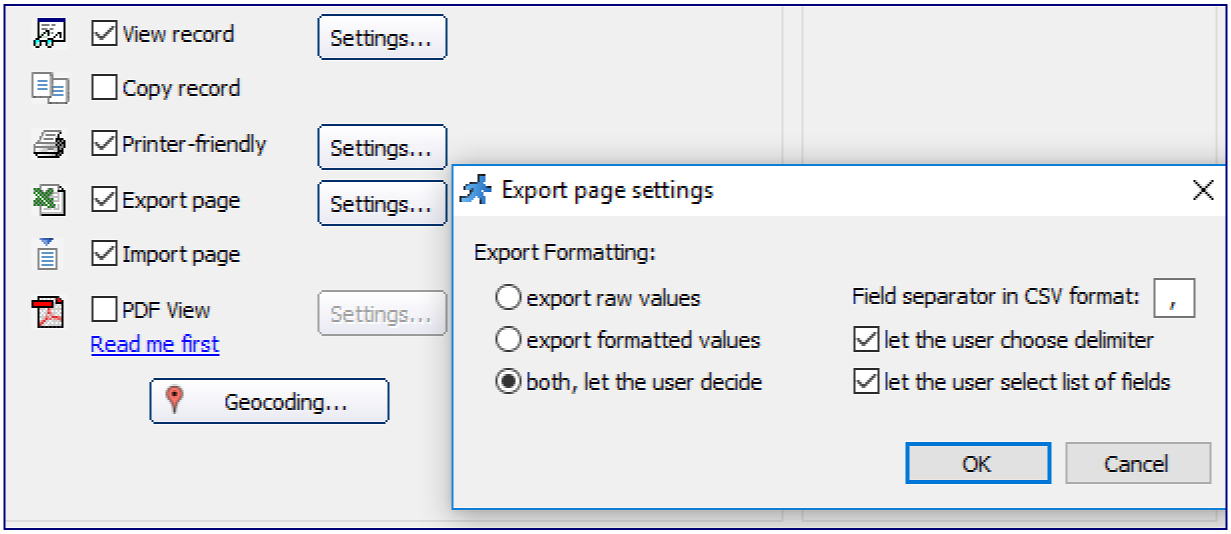
Export/Import pages

Export

Starting from version 9.8 onwards you are able to use new settings for the export function:

Export raw/formatted values option will work in both old (non-Bootstrap) and new layouts.

The Field Separator option and the new option to choose a list of fields are available in **Bootstrap layouts** only.



☑ View record — Settings...
□ Copy record
☑ Printer-friendly — Settings...
☑ Export page — Settings...
☑ Import page
□ PDF View — Settings...
Read me first
Geocoding...

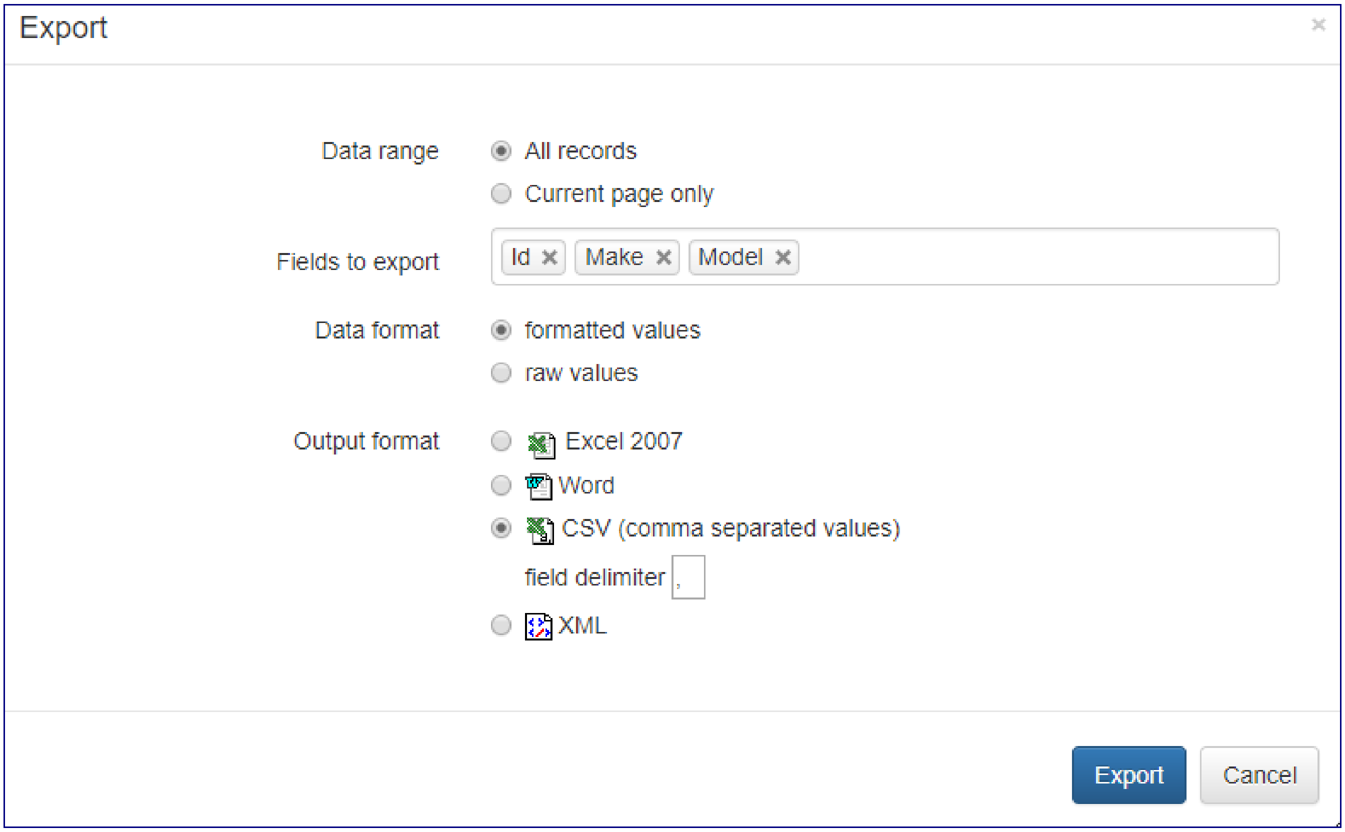
Export page settings

Export Formatting:
○ export raw values
○ export formatted values
● both, let the user decide

Field separator in CSV format: ,
☑ let the user choose delimiter
☑ let the user select list of fields

OK — Cancel

The export page by default is opened in popup. This is how it will look on the generated pages in Bootstrap layout.



Export

Data range ● All records ○ Current page only

Fields to export: Id × Make × Model ×

Data format ● formatted values ○ raw values

Output format ○ Excel 2007 ○ Word ● CSV (comma separated values) field delimiter , ○ XML

Supported file formats are:

- Excel 2007 (.xlsx)
- Word
- CSV (comma separated values)
- XML

Labels are used as headings in Excel/Word files. Field names are used as headings in XML/CSV files. This is done in order for CSV files that is exported using PHPRunner to be imported into other software.

Import

Supported file formats are:

- CSV
- Excel 2007 (.xlsx)
- Excel 97-2003 (.xls)

Field names or labels can be used as headers in Excel files. In CSV files field names should be used as headings.

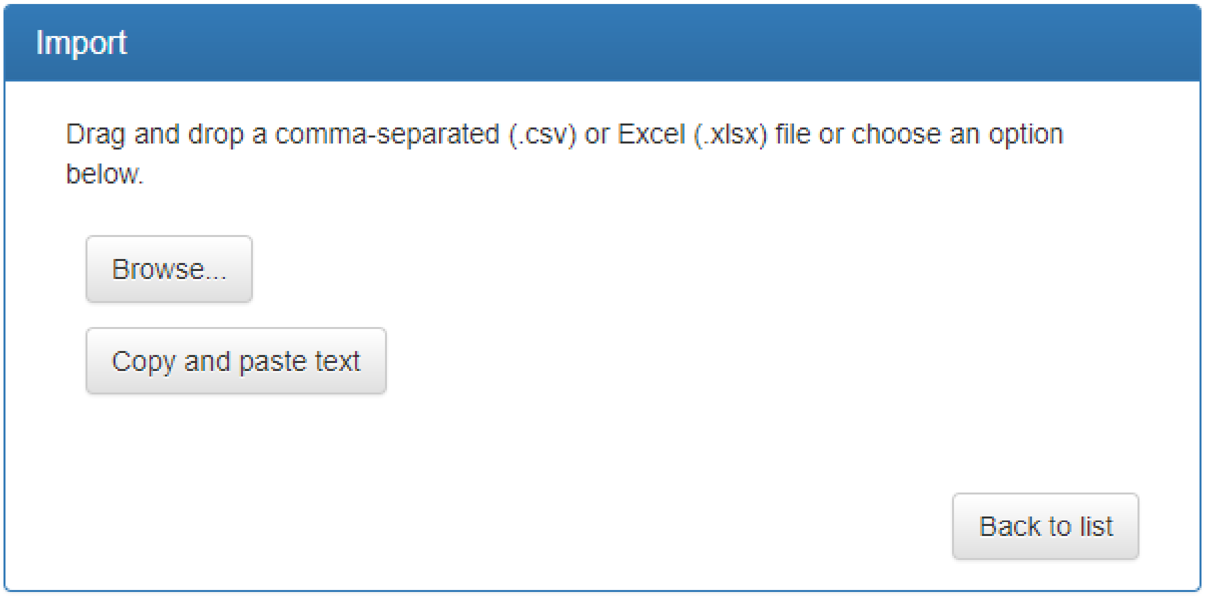
PHPRunner adds the new records or updates the existing ones while importing.

During import PHPRunner will try to insert a new record first. If the insert fails for any reason like duplicate primary key it will try to locate and update this record with new data.

It will update the existing records instead of adding them in the following cases:

1. A primary key is defined for the table in question.
2. Key fields selected on the 'Choose pages' screen match the primary key in the database.
3. Primary key column or columns exist in the file being imported.

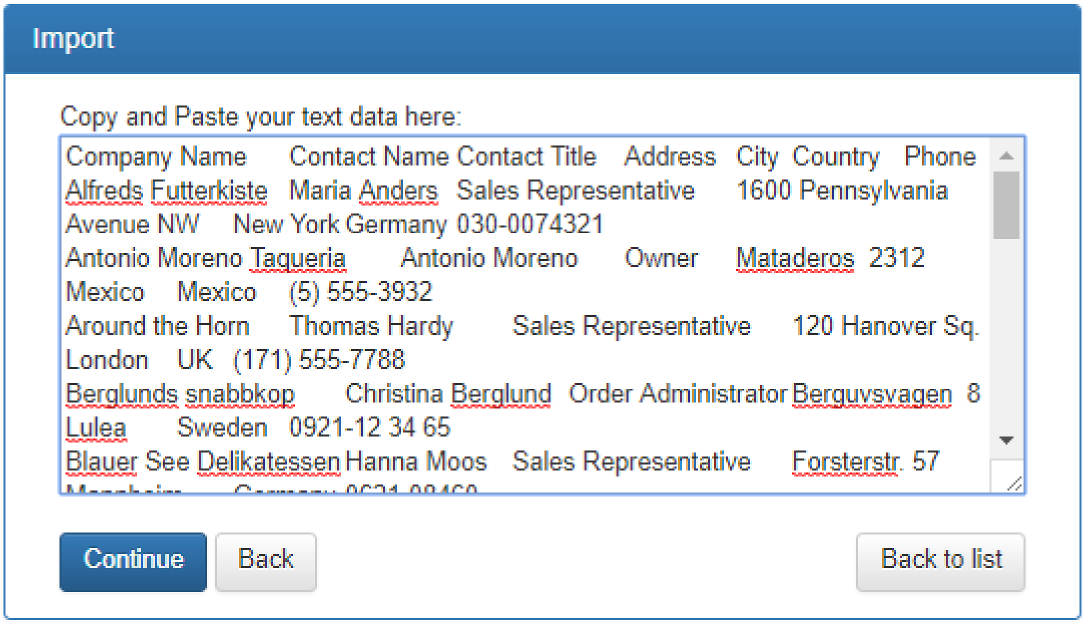
You can copy and paste import data instead of uploading the whole file.



Import

Drag and drop a comma-separated (.csv) or Excel (.xlsx) file or choose an option below.

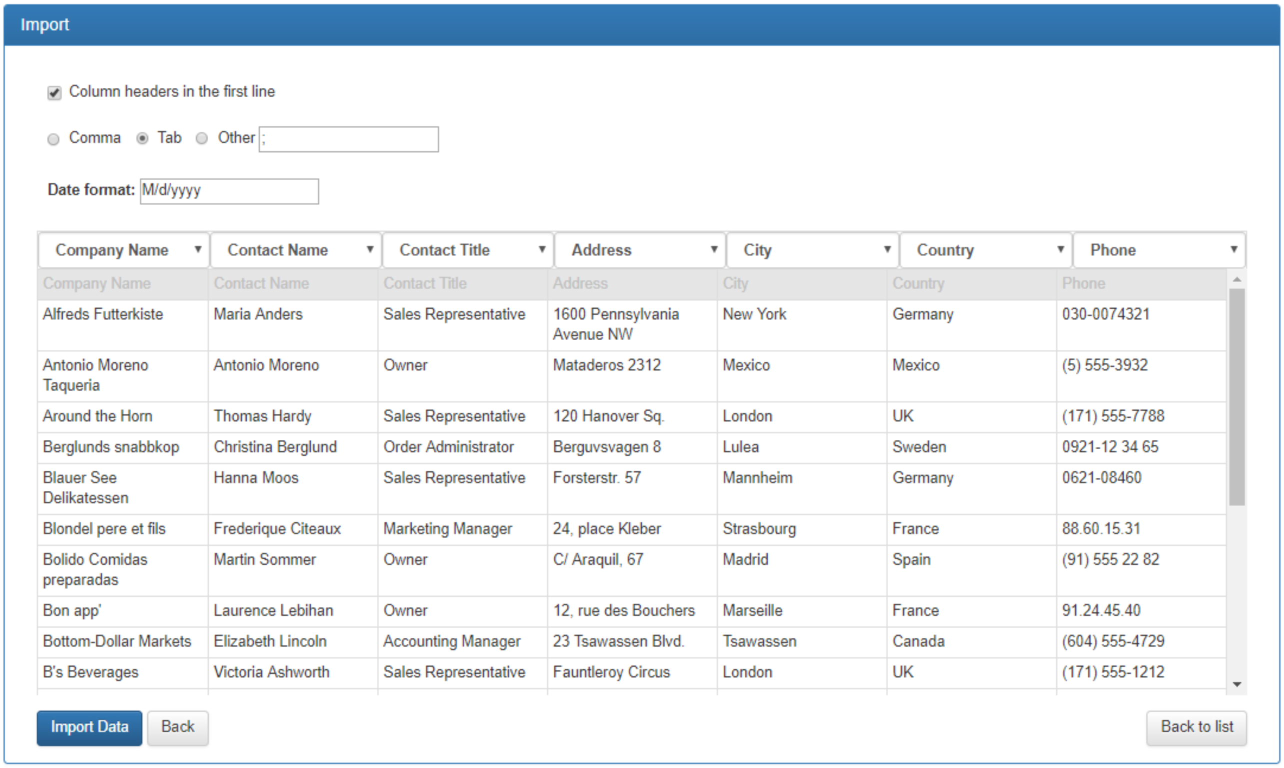
At first copy a few lines of data from Excel. You do not have to include column headers. Then paste it into the import page.



```
Company Name      Contact Name Contact Title    Address    City  Country    Phone
Alfreds Futterkiste   Maria Anders   Sales Representative      1600 Pennsylvania
Avenue NW     New York Germany 030-0074321
Antonio Moreno Taqueria        Antonio Moreno       Owner      Mataderos  2312
Mexico     Mexico     (5) 555-3932
Around the Horn       Thomas Hardy        Sales Representative       120 Hanover Sq.
London     UK   (171) 555-7788
Berglunds snabbkop       Christina Berglund   Order Administrator Berguvsvagen  8
Lulea      Sweden    0921-12 34 65
Blauer See Delikatessen Hanna Moos   Sales Representative       Forsterstr. 57
Mannheim      Germany 0621-08460
```

Map columns define date format and import the data. Examples of supported date formats:

- dd.mm.yyyy
- mm/dd/yyyy
- yyyy-mm-dd
- dd/mm/yyyy



Import

☑ Column headers in the first line

○ Comma ● Tab ○ Other ;

Date format: M/d/yyyy

Company Name ▾	Contact Name ▾	Contact Title ▾	Address ▾	City ▾	Country ▾	Phone ▾
Company Name	Contact Name	Contact Title	Address	City	Country	Phone
Alfreds Futterkiste	Maria Anders	Sales Representative	1600 Pennsylvania Avenue NW	New York	Germany	030-0074321
Antonio Moreno Taqueria	Antonio Moreno	Owner	Mataderos 2312	Mexico	Mexico	(5) 555-3932
Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	UK	(171) 555-7788
Berglunds snabbkop	Christina Berglund	Order Administrator	Berguvsvagen 8	Lulea	Sweden	0921-12 34 65
Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	Germany	0621-08460
Blondel pere et fils	Frederique Citeaux	Marketing Manager	24, place Kleber	Strasbourg	France	88.60.15.31
Bolido Comidas preparadas	Martin Sommer	Owner	C/ Araquil, 67	Madrid	Spain	(91) 555 22 82
Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	France	91.24.45.40
Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	Canada	(604) 555-4729
B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	UK	(171) 555-1212

If you wish to combine the import with any extra actions, use BeforeInsert Event. For example, if you want to specify the file creation date or fill in an OwnerID field.

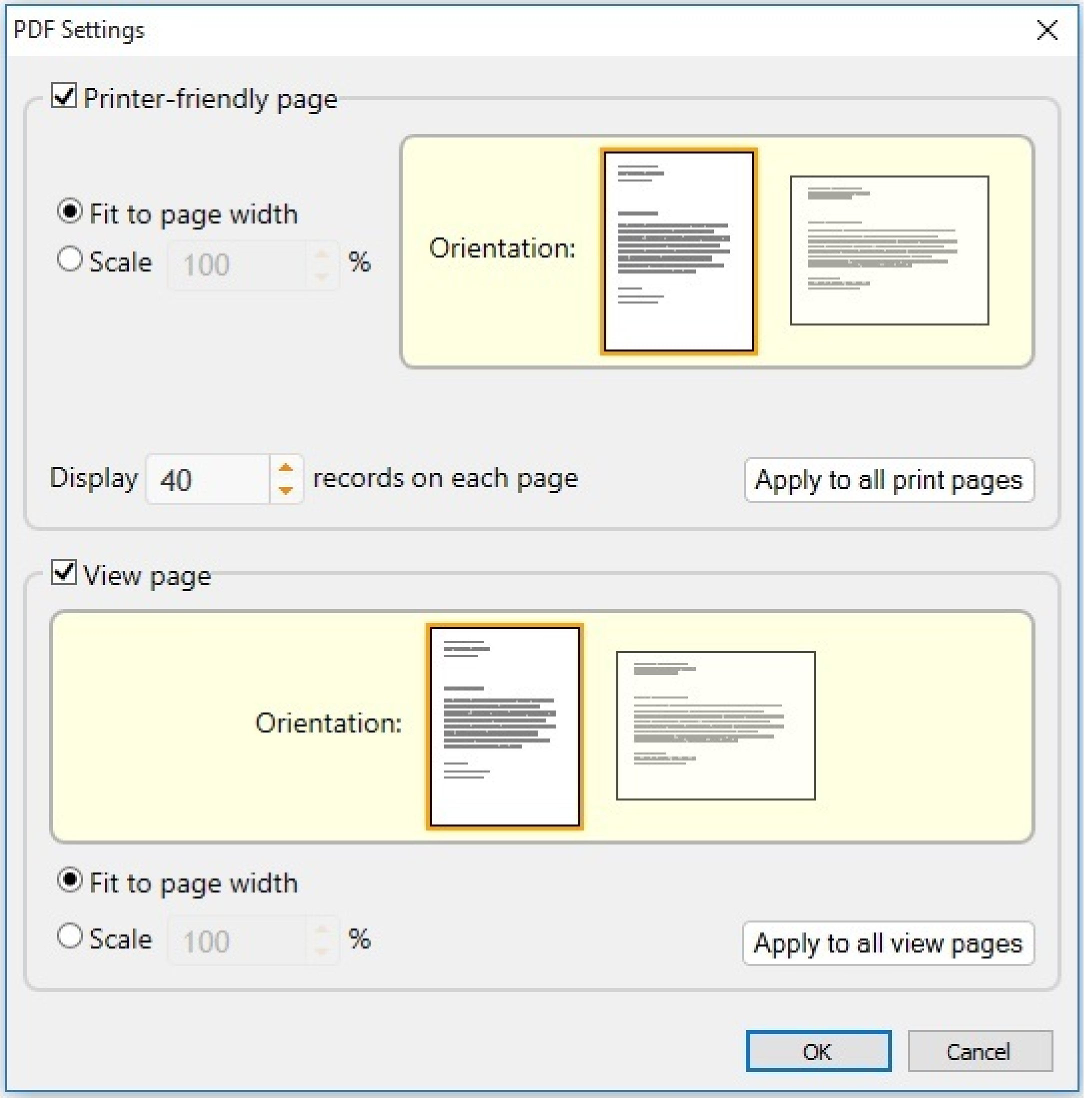
Note: since PHPRunner creates temporary files used for import preview in *templates_c* folder under *output* directory, web server user needs to have editing permissions on this folder.

2.11.7 Printer page/PDF view settings

Printer page/PDF view settings

To learn how to enable **PDF view** option, see PDF View settings.

Choose between portrait and landscape orientation, tune PDF export by setting the scale.



PDF Settings

☑ Printer-friendly page

● Fit to page width

○ Scale 100 %

Orientation:

Display 40 records on each page

☑ View page

Orientation:

● Fit to page width

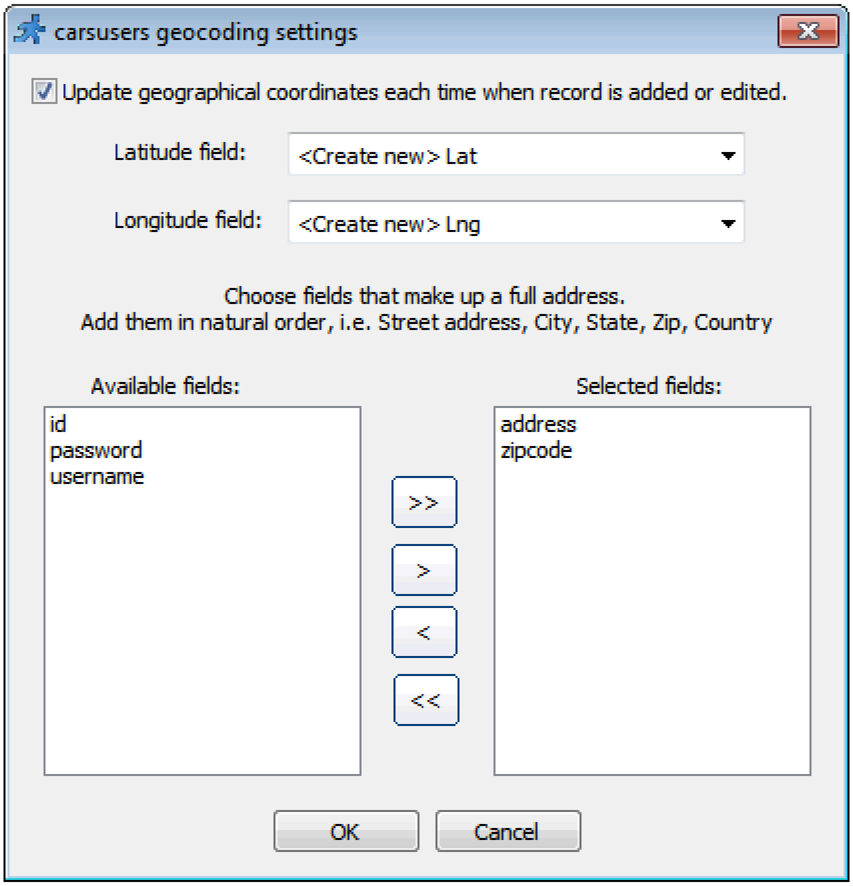
○ Scale 100 %

2.11.8 Geocoding

Geocoding

Geocoding is used for tables where location data is displayed as maps. It is recommended to use Latitude and Longitude fields and update them each time when a record is added or updated. Select existing fields for storing latitude and longitude data or create new ones. Then select address field(s) to be used during geocoding.

For more information about displaying data as maps, see Insert map.



carsusers geocoding settings

☑ Update geographical coordinates each time when record is added or edited.

Latitude field: <Create new> Lat

Longitude field: <Create new> Lng

Choose fields that make up a full address.
Add them in natural order, i.e. Street address, City, State, Zip, Country

Available fields: id, password, username

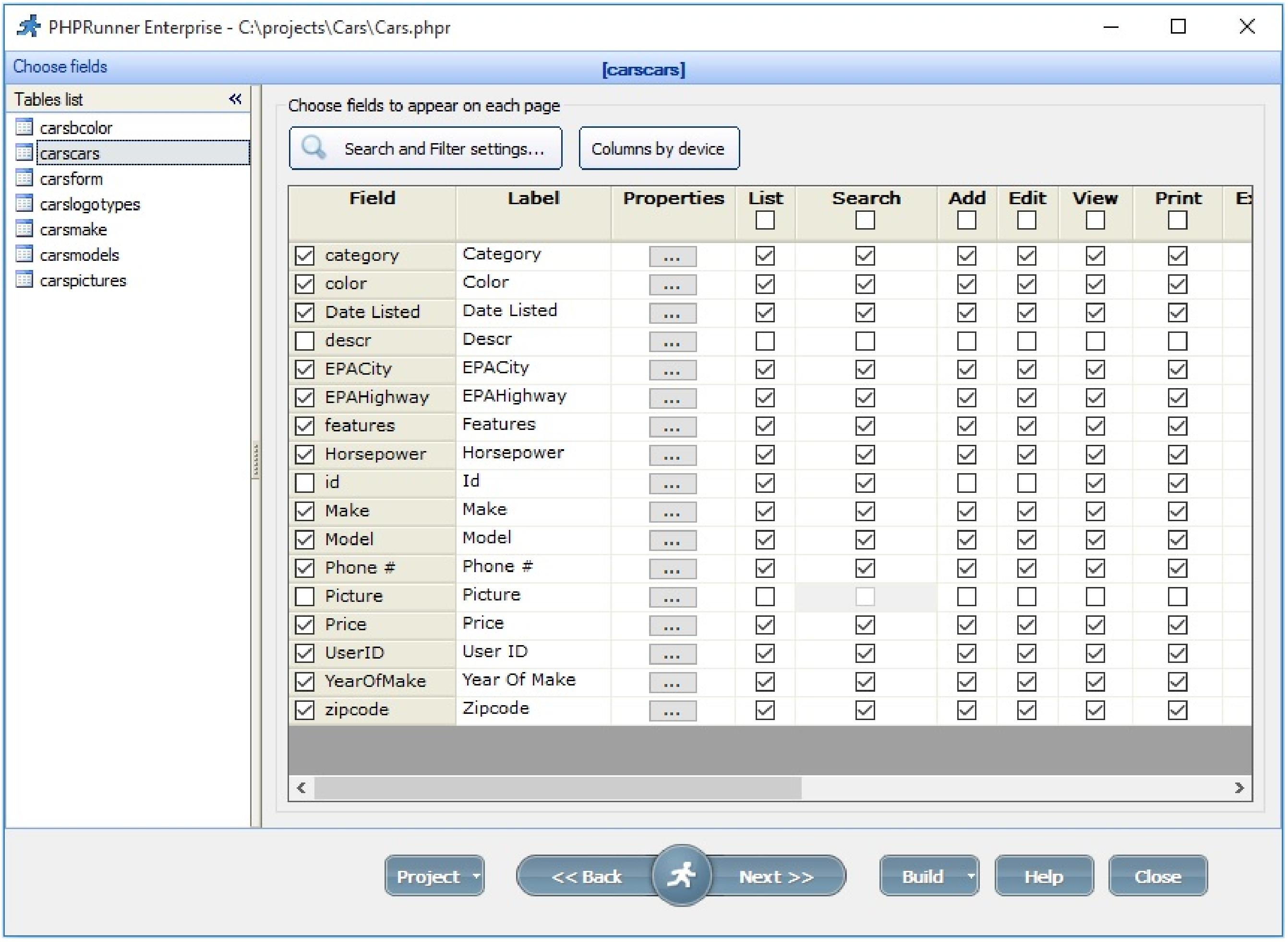
Selected fields: address, zipcode

2.12 Choose fields

Quick jump

Search and Filter settings

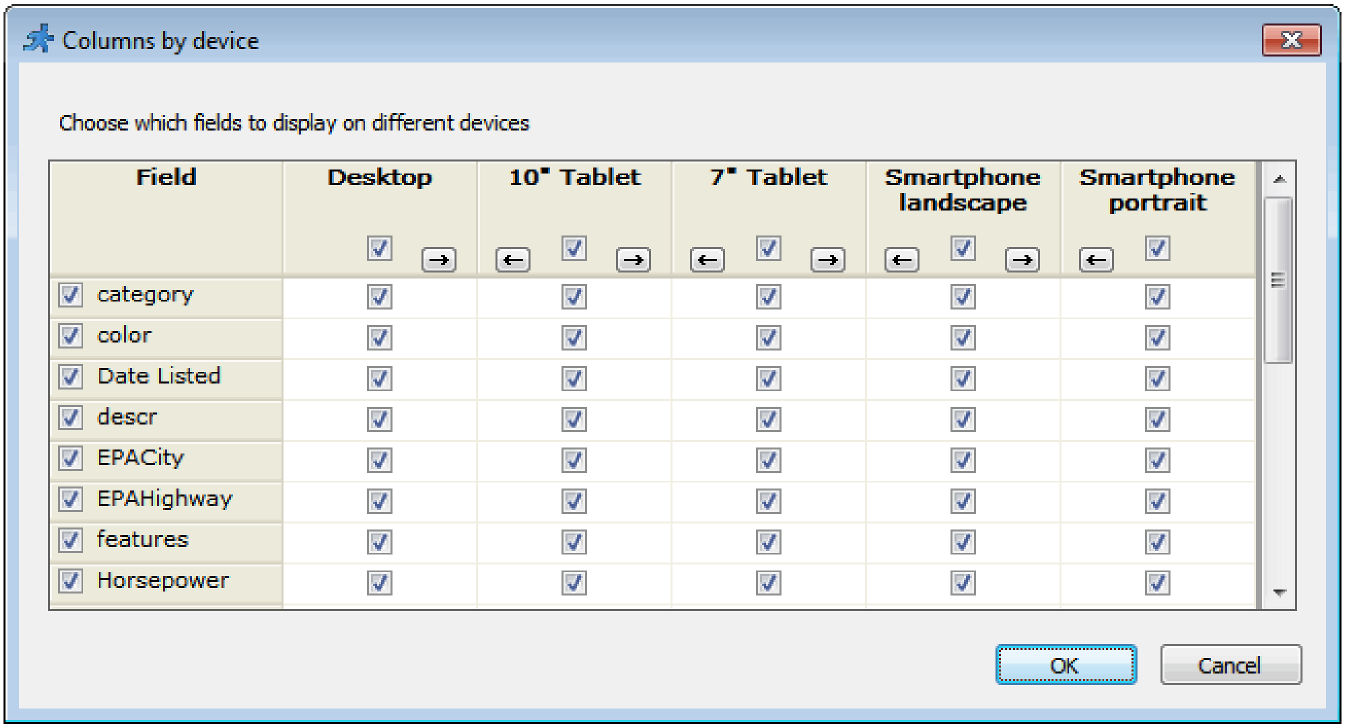
On the **Choose fields** page choose fields to appear on each page by selecting/clearing the corresponding check boxes. Note that you can display different fields on different devices. Also you can adjust search and filter settings here.



Field	Label	Properties	List ☐	Search ☐	Add ☐	Edit ☐	View ☐	Print ☐	E
☑ category	Category	...	☑	☑	☑	☑	☑	☑	
☑ color	Color	...	☑	☑	☑	☑	☑	☑	
☑ Date Listed	Date Listed	...	☑	☑	☑	☑	☑	☑	
☐ descr	Descr	...	☐	☐	☐	☐	☐	☐	
☑ EPACity	EPACity	...	☑	☑	☑	☑	☑	☑	
☑ EPAHighway	EPAHighway	...	☑	☑	☑	☑	☑	☑	
☑ features	Features	...	☑	☑	☑	☑	☑	☑	
☑ Horsepower	Horsepower	...	☑	☑	☑	☑	☑	☑	
☐ id	Id	...	☑	☑	☐	☐	☑	☑	
☑ Make	Make	...	☑	☑	☑	☑	☑	☑	
☑ Model	Model	...	☑	☑	☑	☑	☑	☑	
☑ Phone #	Phone #	...	☑	☑	☑	☑	☑	☑	
☐ Picture	Picture	...	☐	☐	☐	☐	☐	☐	
☑ Price	Price	...	☑	☑	☑	☑	☑	☑	
☑ UserID	User ID	...	☑	☑	☑	☑	☑	☑	
☑ YearOfMake	Year Of Make	...	☑	☑	☑	☑	☑	☑	
☑ zipcode	Zipcode	...	☑	☑	☑	☑	☑	☑	

Columns by device

You can select different fields to display on different devices. This feature helps you customize your list pages hiding certain columns on devices with smaller screens. You can easily copy selected fields from one column (device) to another using buttons with left/right arrow.



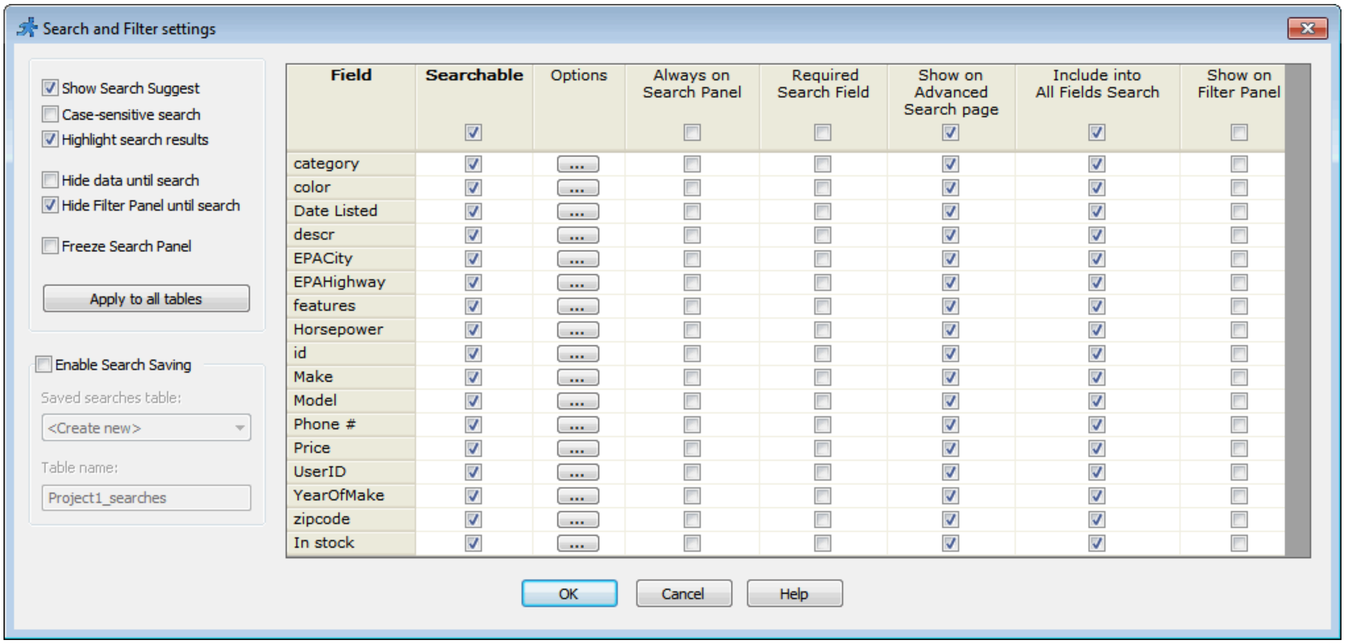
Choose which fields to display on different devices

Field	Desktop	10" Tablet	7" Tablet	Smartphone landscape	Smartphone portrait
	☑ →	← ☑ →	← ☑ →	← ☑ →	← ☑
☑ category	☑	☑	☑	☑	☑
☑ color	☑	☑	☑	☑	☑
☑ Date Listed	☑	☑	☑	☑	☑
☑ descr	☑	☑	☑	☑	☑
☑ EPACity	☑	☑	☑	☑	☑
☑ EPAHighway	☑	☑	☑	☑	☑
☑ features	☑	☑	☑	☑	☑
☑ Horsepower	☑	☑	☑	☑	☑

Search and Filter settings

Users can search by each word independently or by the whole phrase if double quoted (Google-like search).

Click **Apply to all tables** button to apply the selected above options to all tables.



☑ Show Search Suggest
☐ Case-sensitive search
☑ Highlight search results

☐ Hide data until search
☑ Hide Filter Panel until search

☐ Freeze Search Panel

☐ Enable Search Saving

Saved searches table: <Create new>

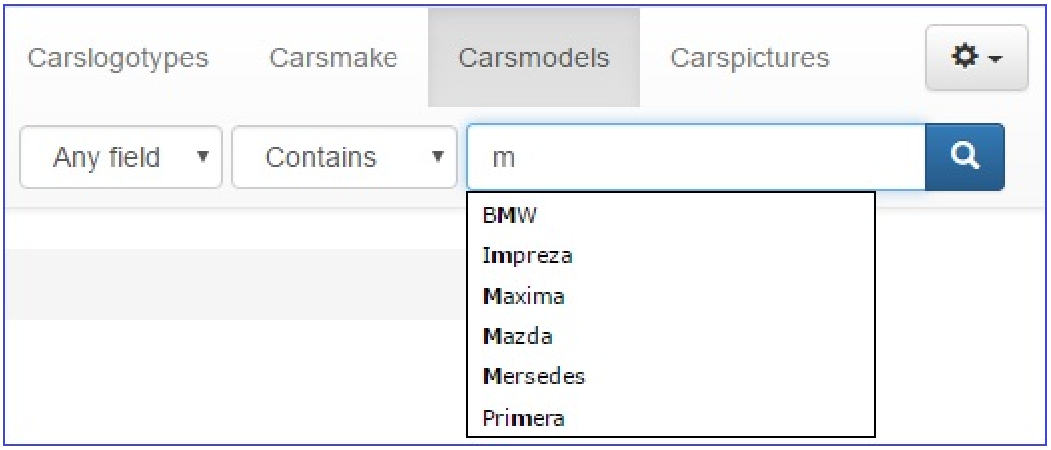
Table name: Project1_searches

Field	Searchable	Options	Always on Search Panel	Required Search Field	Show on Advanced Search page	Include into All Fields Search	Show on Filter Panel
	☑		☐	☐	☑	☑	☐
category	☑	...	☐	☐	☑	☑	☐
color	☑	...	☐	☐	☑	☑	☐
Date Listed	☑	...	☐	☐	☑	☑	☐
descr	☑	...	☐	☐	☑	☑	☐
EPACity	☑	...	☐	☐	☑	☑	☐
EPAHighway	☑	...	☐	☐	☑	☑	☐
features	☑	...	☐	☐	☑	☑	☐
Horsepower	☑	...	☐	☐	☑	☑	☐
id	☑	...	☐	☐	☑	☑	☐
Make	☑	...	☐	☐	☑	☑	☐
Model	☑	...	☐	☐	☑	☑	☐
Phone #	☑	...	☐	☐	☑	☑	☐
Price	☑	...	☐	☐	☑	☑	☐
UserID	☑	...	☐	☐	☑	☑	☐
YearOfMake	☑	...	☐	☐	☑	☑	☐
zipcode	☑	...	☐	☐	☑	☑	☐
In stock	☑	...	☐	☐	☑	☑	☐

Show Search suggest

When you start typing in the search box, AJAX popup displays search suggestions. For more information, see AJAX-based Functionality: AJAX Auto Suggest.

Example of the search suggest feature on the basic search page:

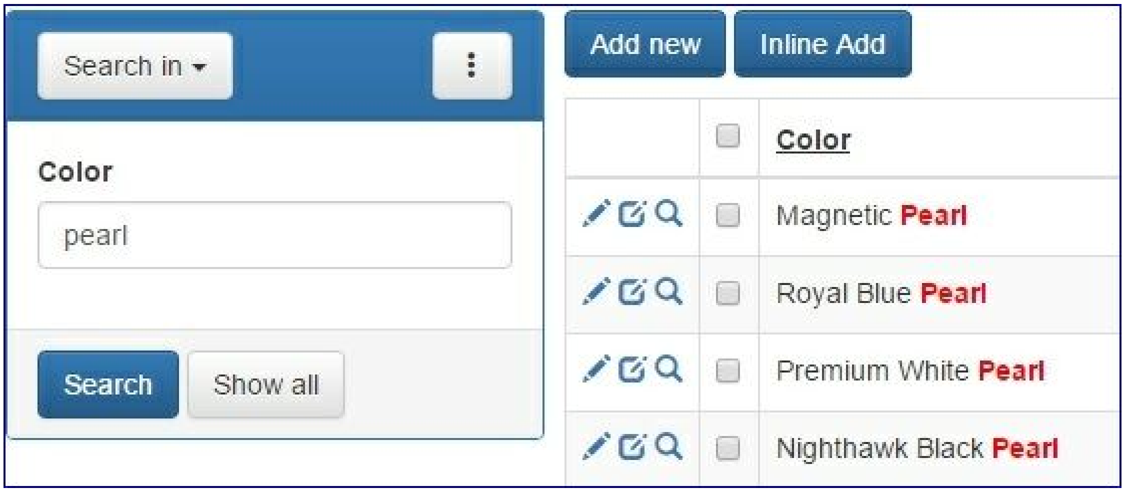


Case-sensitive search

Use this option to filter out search results that match the case of your search query. If this option is not selected, the search is case insensitive.

Highlight search results

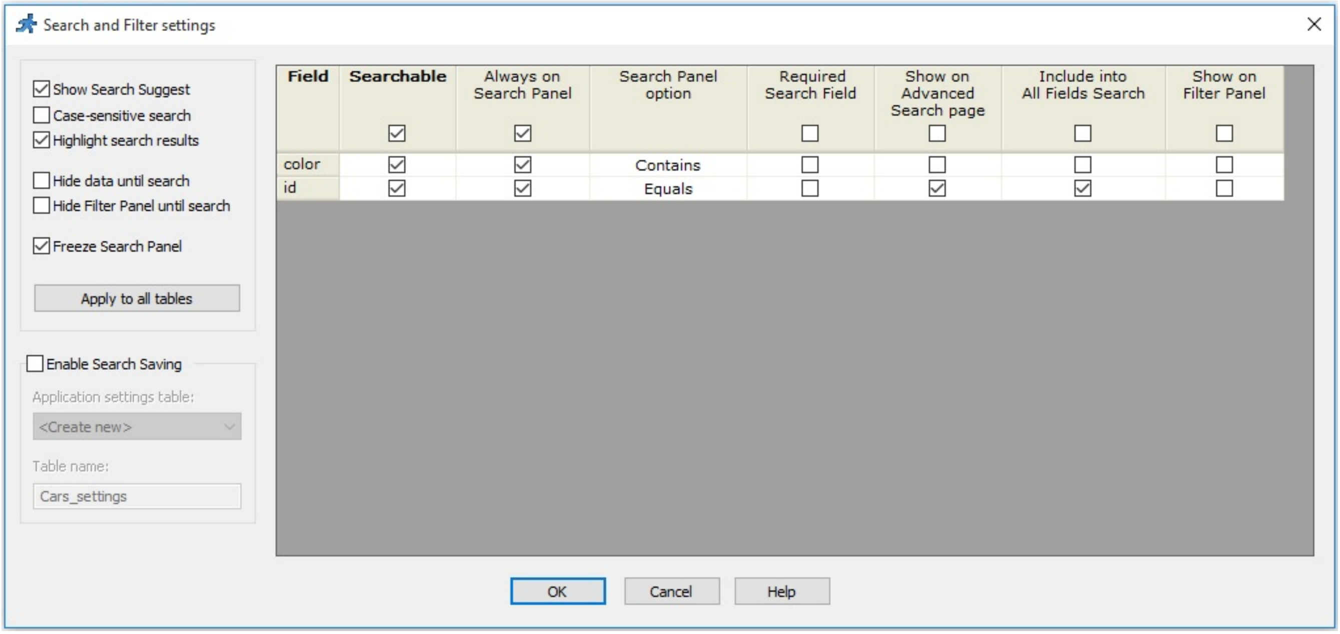
Use this option to highlight search results in the database grid.



	□	Color
	□	Magnetic **Pearl**
	□	Royal Blue **Pearl**
	□	Premium White **Pearl**
	□	Nighthawk Black **Pearl**

Freeze Search Panel

You can define default search option (Equals, Contains etc.) for each field on search panel.



☑ Show Search Suggest
☐ Case-sensitive search
☑ Highlight search results

☐ Hide data until search
☐ Hide Filter Panel until search

☑ Freeze Search Panel

☐ Enable Search Saving

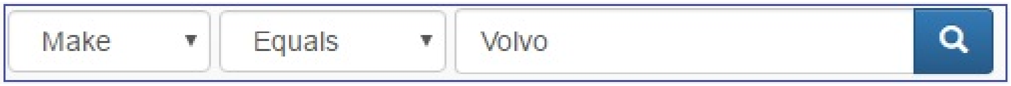
Application settings table: <Create new>

Table name: Cars_settings

Field	Searchable	Always on Search Panel	Search Panel option	Required Search Field	Show on Advanced Search page	Include into All Fields Search	Show on Filter Panel
	☑	☑		☐	☐	☐	☐
color	☑	☑	Contains	☐	☐	☐	☐
id	☑	☑	Equals	☐	☑	☑	☐

Searchable fields

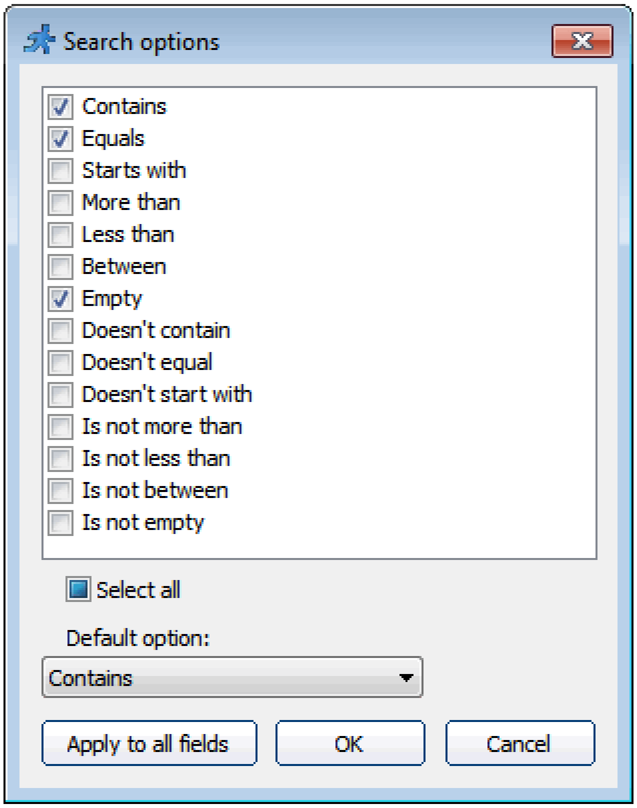
When you make some field searchable, you allow searching by this field in the **Quick search** box and also you can define other search options such as Search panel, Advanced search and 'All fields' search for this field. To be able to select field for quick search, enable the option **Show basic search options** for the List page on the Choose pages page.



Make ▾ Equals ▾ Volvo

Options

You can select what search options will be available for each field on search panel. Also you can define the default search option for each field.



Search options

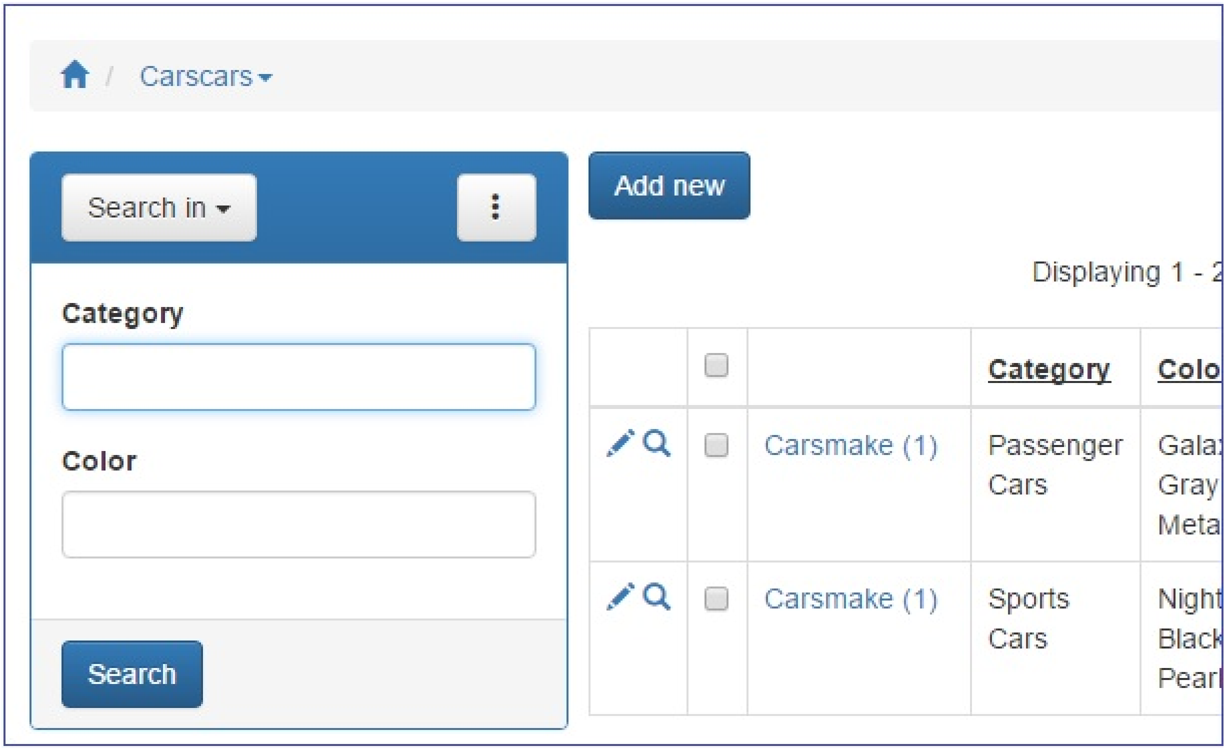
☑ Contains
☑ Equals
☐ Starts with
☐ More than
☐ Less than
☐ Between
☑ Empty
☐ Doesn't contain
☐ Doesn't equal
☐ Doesn't start with
☐ Is not more than
☐ Is not less than
☐ Is not between
☐ Is not empty

■ Select all

Default option: Contains

Always on Search panel

Use this option to add fields to the search panel on the List page. Search panel allows you to define the search criteria. For more information, see Web interface guide: Search panel on List page.



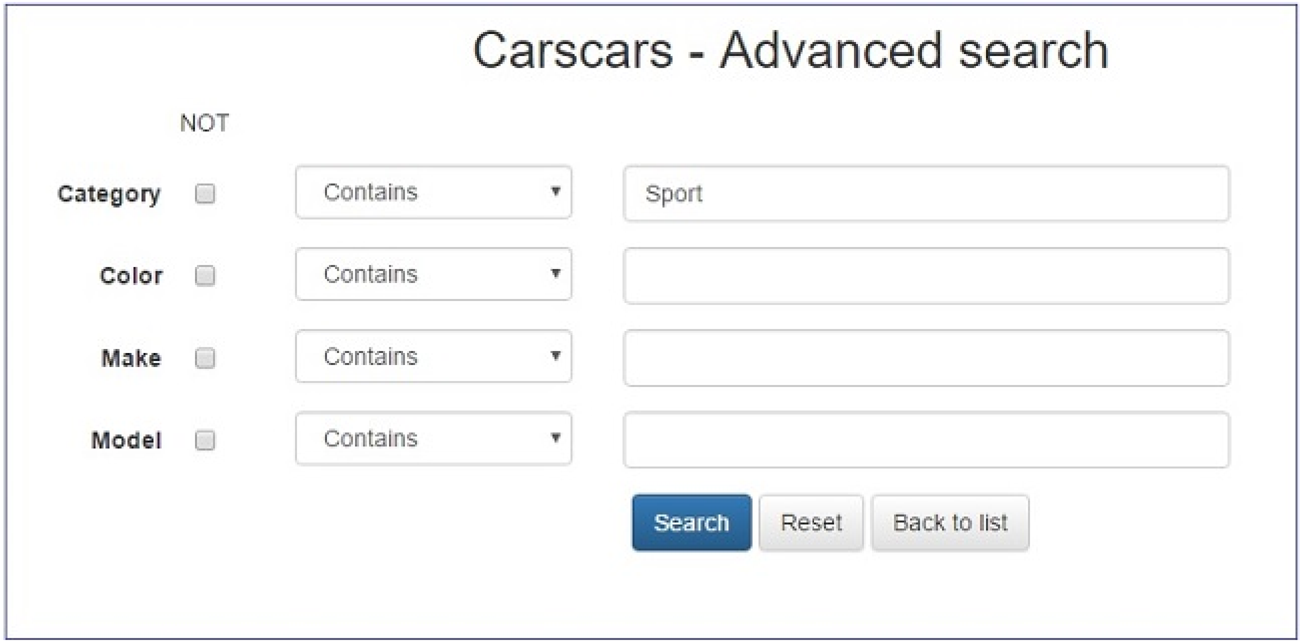
	□		Category	Colo
	□	Carsmake (1)	Passenger Cars	Gala Gray Meta
	□	Carsmake (1)	Sports Cars	Night Black Pearl

Required Search field

Users will not be able to search until they specify required search fields. For example, homes for sale database do not allow to show any data until zip code is selected.

Show on Advanced Search page

Allows to display fields on the Advanced search page.

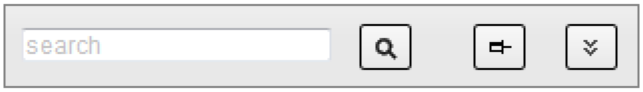


Carscars - Advanced search

	NOT		
Category	□	Contains	Sport
Color	□	Contains	
Make	□	Contains	
Model	□	Contains	

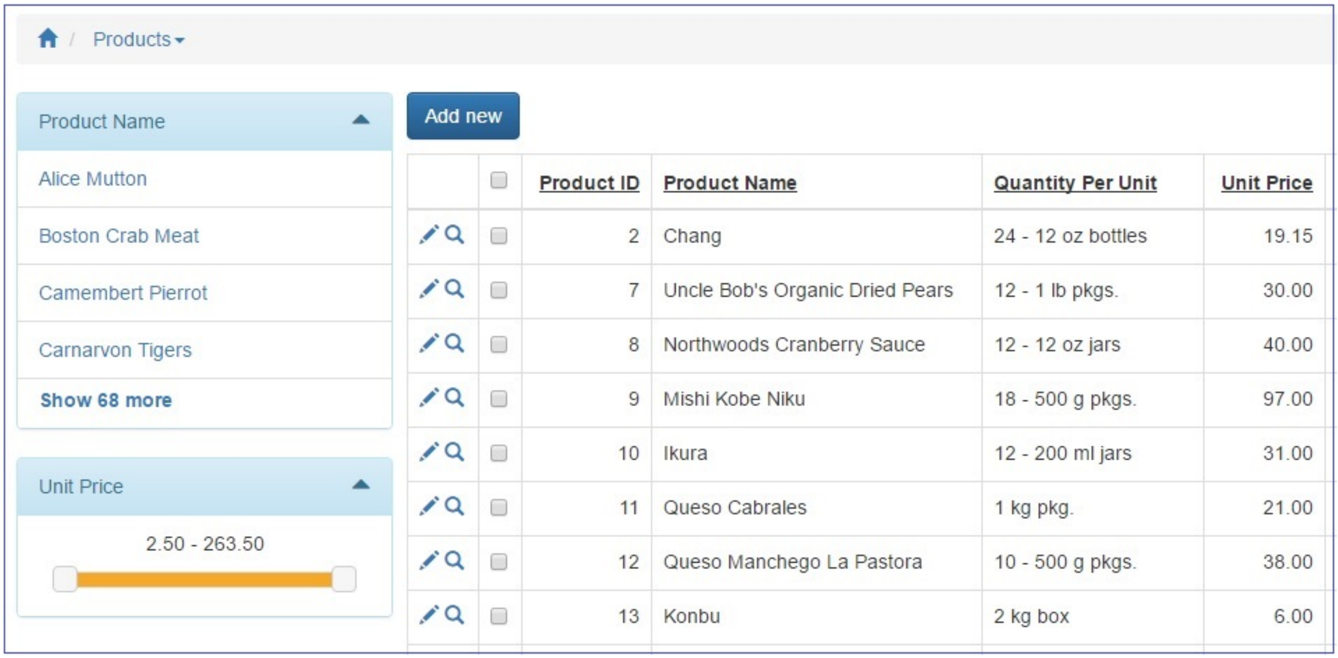
'All fields' search

Use this option to add fields to the 'All fields' quick search.



Show on Filter Panel

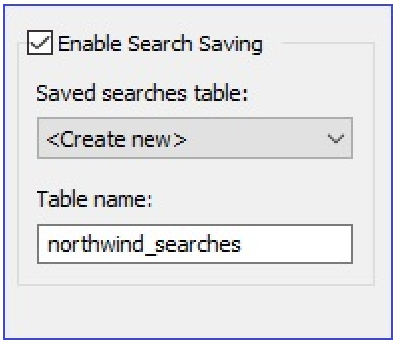
Allows to display fields on the Filter Panel. Filters provide instant feedback on numbers of records matching each criteria. You can select multiple criteria or multiple intervals for each filter. Filter panel is located on the left side. For more information about filters' settings, see "Filter as" settings.



		Product ID	Product Name	Quantity Per Unit	Unit Price
	□	2	Chang	24 - 12 oz bottles	19.15
	□	7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	30.00
	□	8	Northwoods Cranberry Sauce	12 - 12 oz jars	40.00
	□	9	Mishi Kobe Niku	18 - 500 g pkgs.	97.00
	□	10	Ikura	12 - 200 ml jars	31.00
	□	11	Queso Cabrales	1 kg pkg.	21.00
	□	12	Queso Manchego La Pastora	10 - 500 g pkgs.	38.00
	□	13	Konbu	2 kg box	6.00

Enable Search Saving

While searching for a data using Search panel, you can save searches in the database for later retrieval (use **Save search** button in the browser). Select *<project_name>_searches* as table for saving searches or create new one.

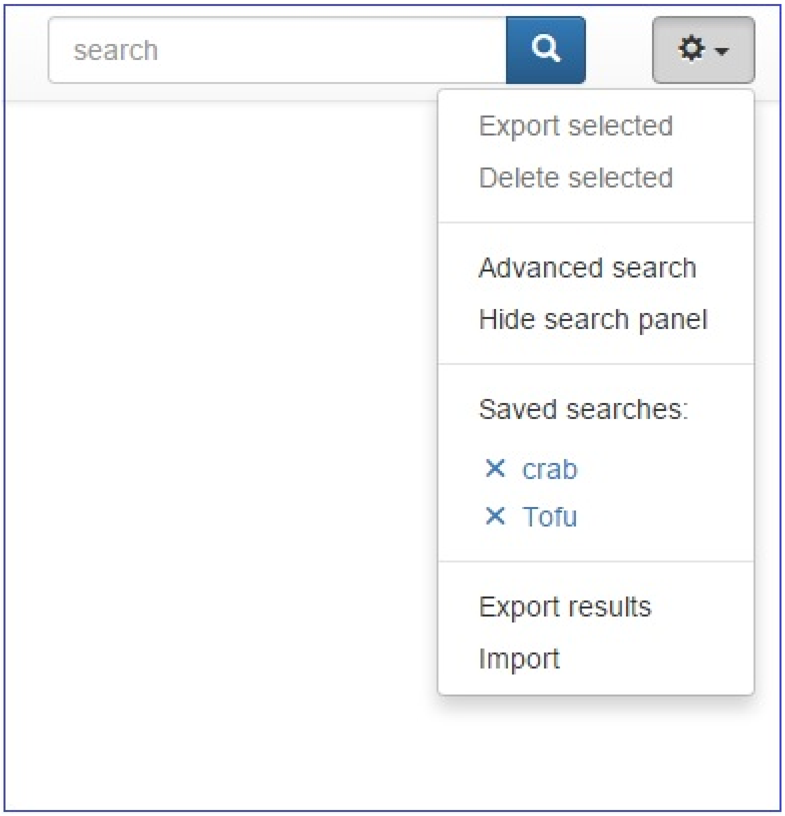


☑ Enable Search Saving

Saved searches table: <Create new>

Table name: northwind_searches

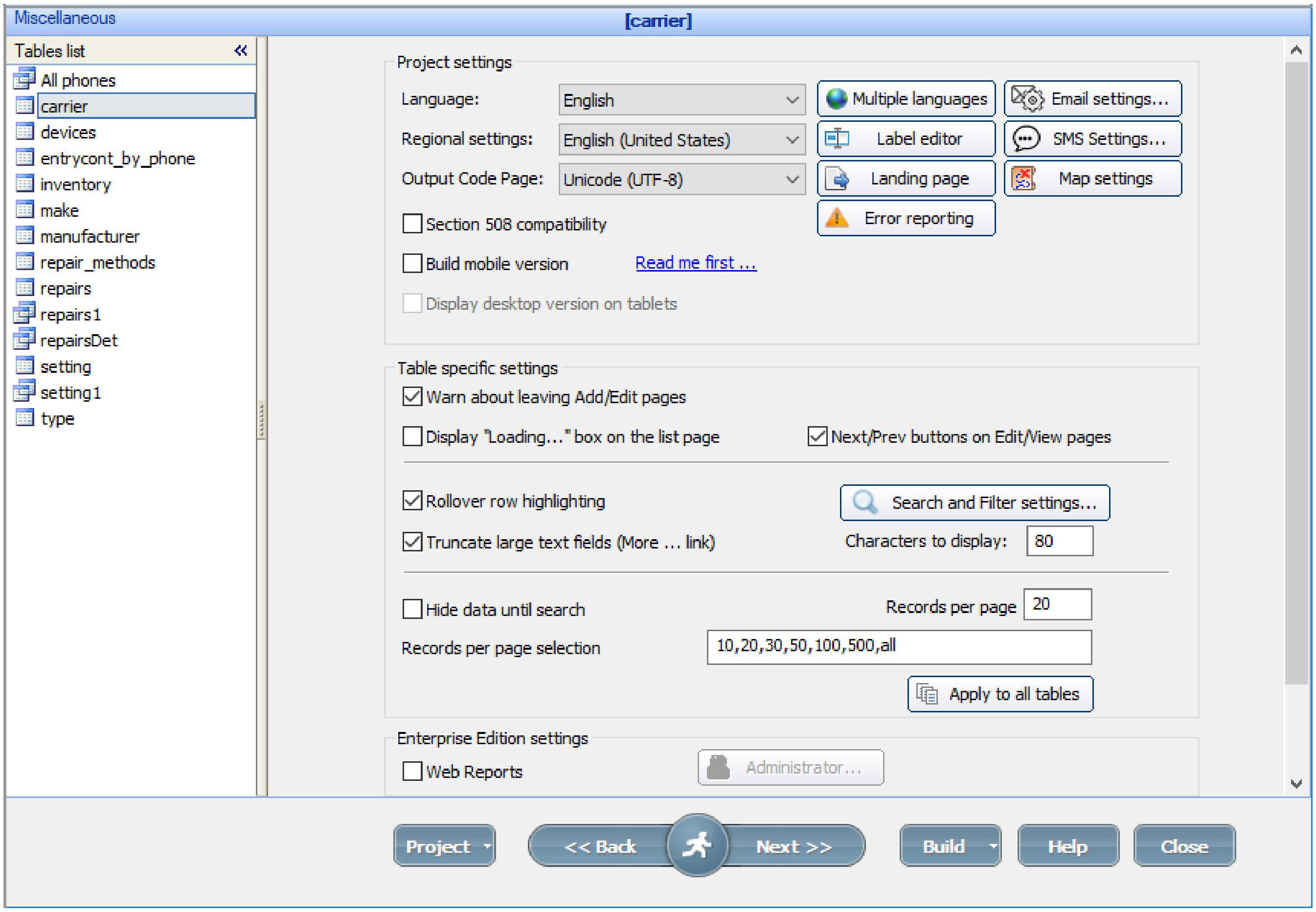
To view saved searches click **Saved searches** on Search panel.



2.13 Miscellaneous settings

Quick jump	
Language	Error reporting
Label Editor	Section 508 compatibility
Email settings	Table specific settings
Landing page	Enterprise Edition settings
Map settings	SMS Settings

The **Miscellaneous** page allows to set project, table specific and Enterprise Edition settings.



Project settings

Language: English

Regional settings: English (United States)

Output Code Page: Unicode (UTF-8)

☐ Section 508 compatibility

☐ Build mobile version

☐ Display desktop version on tablets

Table specific settings

☑ Warn about leaving Add/Edit pages

☐ Display "Loading..." box on the list page

☑ Next/Prev buttons on Edit/View pages

☑ Rollover row highlighting

☑ Truncate large text fields (More ... link)

Characters to display: 80

☐ Hide data until search

Records per page 20

Records per page selection 10,20,30,50,100,500,all

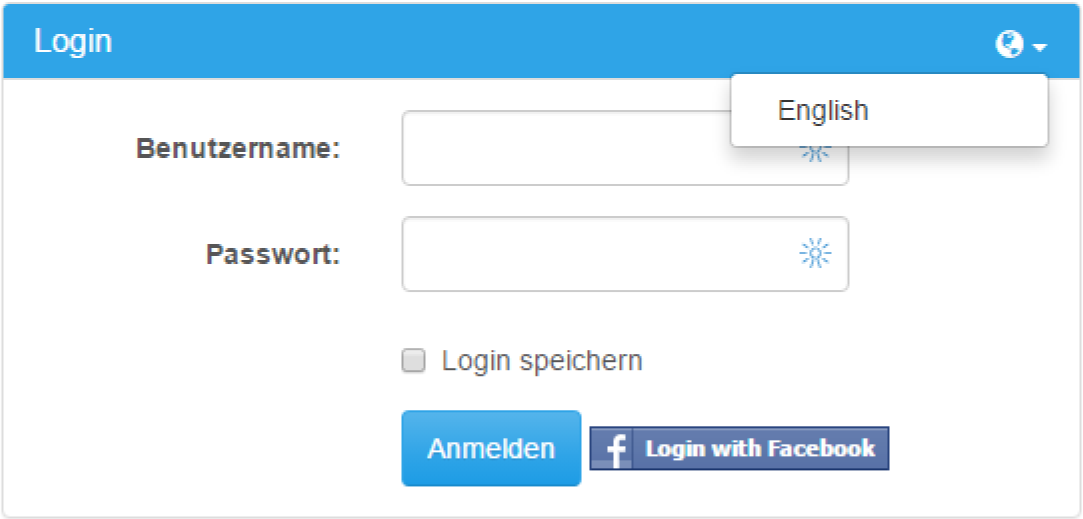
Enterprise Edition settings

☐ Web Reports

Project settings

Language

Use the **Language** drop-down box to choose the Project language. By clicking **Multiple languages** button you can select several Project languages and give the user ability to choose language while logging in.



Regional settings

Choose required country from drop-down box.

Label Editor

Label Editor allows you to edit table and field labels (**Table labels** tab), custom labels (**Custom labels** tab), add tooltips to the Edit forms (**Edit form tooltips** tab) and edit web page titles (**Page titles** tab). You can translate labels into several languages. Click **Choose languages** to add more languages to the project.

You can export multilanguage labels to CSV file (use **Export to CSV** button), then edit labels in the external editor and import the file back (use **Import from CSV** button). If you use languages other than English in the project, we recommend to edit CSV files using Google Docs or OpenOffice. Due to problems with character encoding we do not recommend using Microsoft Excel for projects with multiple languages.

Table labels

Modify table and field labels to make them more good-looking to users. Filter labels by table or field to limit the data displayed. Use <br> to insert a line break in field label.

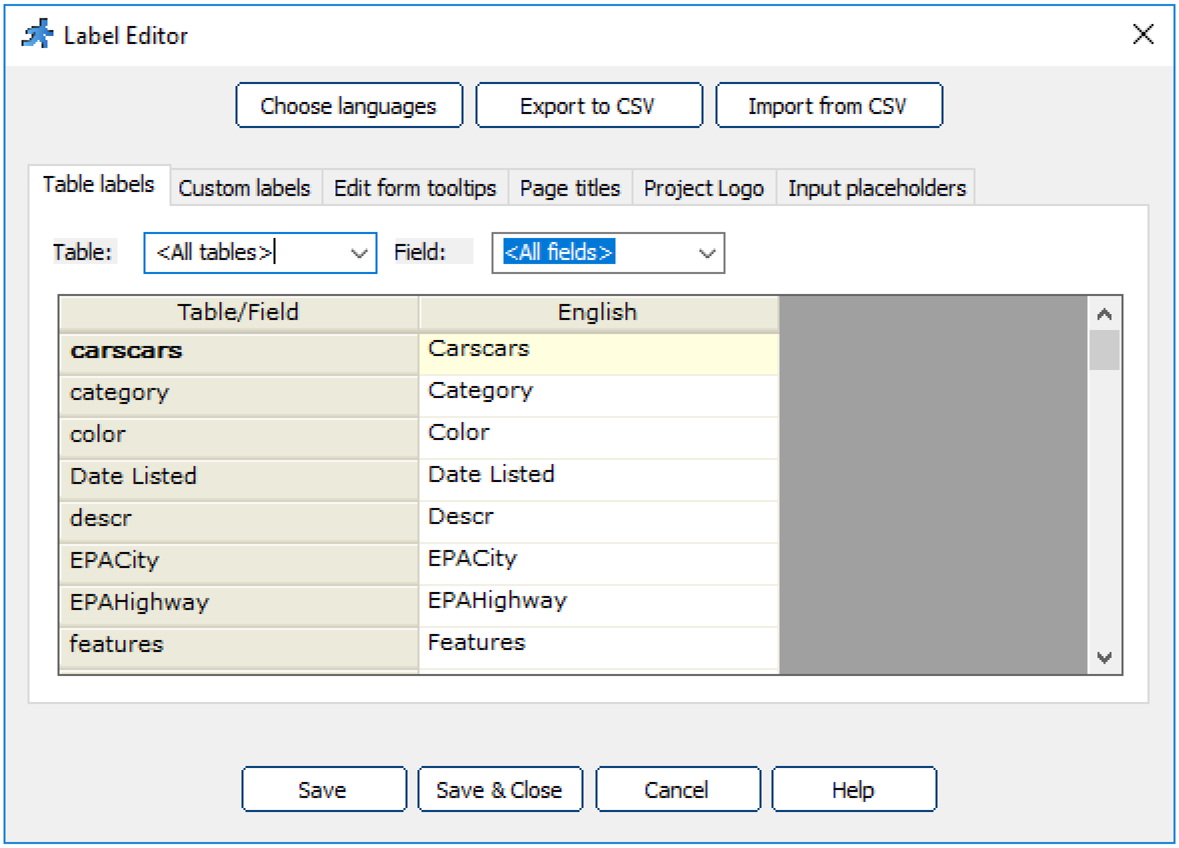


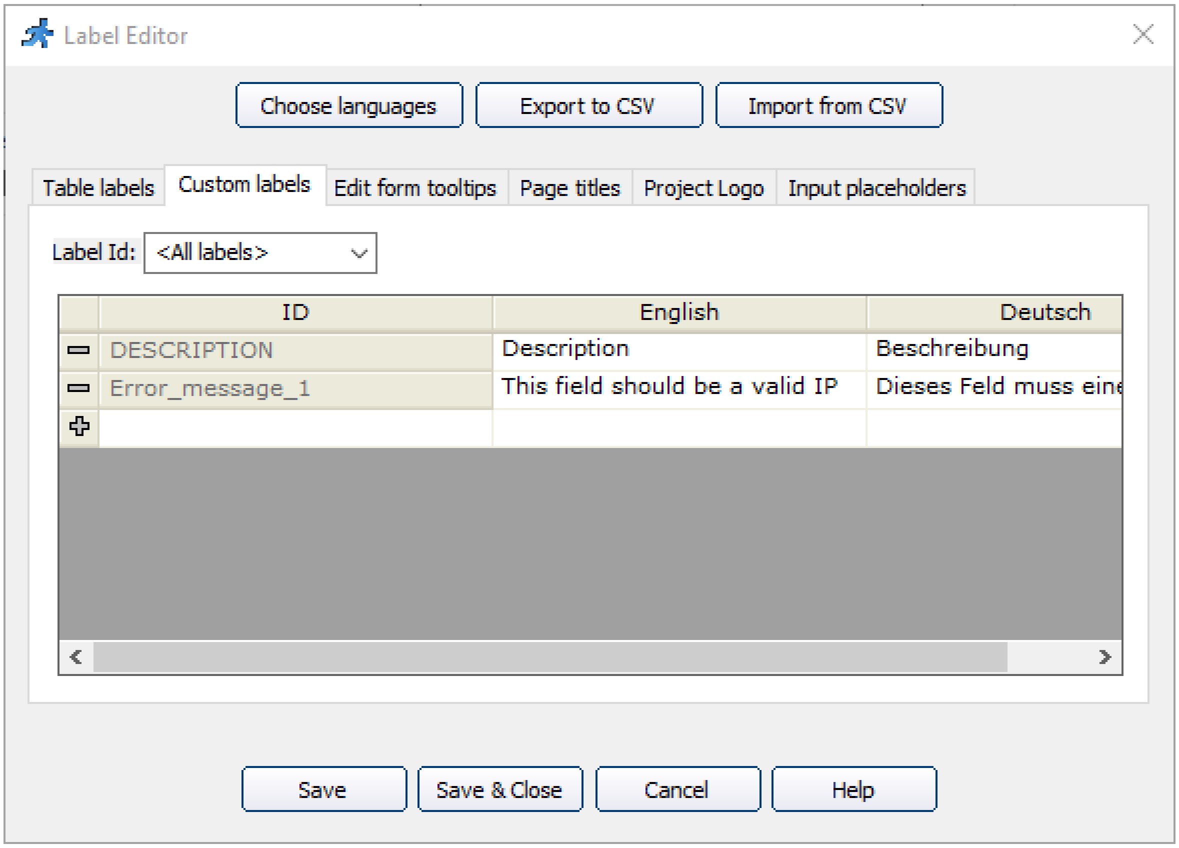
Table: <All tables>

Field: <All fields>

Table/Field	English
carscars	Carscars
category	Category
color	Color
Date Listed	Date Listed
descr	Descr
EPACity	EPACity
EPAHighway	EPAHighway
features	Features

Custom labels

Use custom labels to translate error messages in regular expressions and your own validation plugins. Also you can create your own custom labels to display some messages to a user. Filter labels by label id to limit the data displayed.



ID	English	Deutsch
DESCRIPTION	Description	Beschreibung
Error_message_1	This field should be a valid IP	Dieses Feld muss eine

If you have created Custom labels and want to place some text on a page, which will vary depending on the selected language, you can add PHP code snippet to the page and use the following code in it:

```
echo GetCustomLabel ("CustomLabelID");
```

Don't forget to replace *CustomLabelID* with the correct custom label identifier.

Also you can use custom labels in page template. Open Visual Editor and switch to HTML mode. Then add the following code:

```
{$custom CustomLabelID}
```

And in Javascript code:

```
Runner.getCustomLabel('CustomLabelID')
```

While editing page title you can use text for all pages and field values, e.g. {%ID} and {%Make}, for View and Edit pages. For example:

```
Cars, Edit record [ID:{%ID}]
Customers, {%FirstName} {%LastName}
```

Also you can use the field values from the master table on the List page of details table. Here is the example of page title for Order details table:

```
Orders, ID: {%master.OrderID}
```

Edit form tooltips

Add a tooltip to a field to display some additional information about it on the Edit page and during inline editing on the List page. To create a multiline tooltip, use <br> as a line separator.

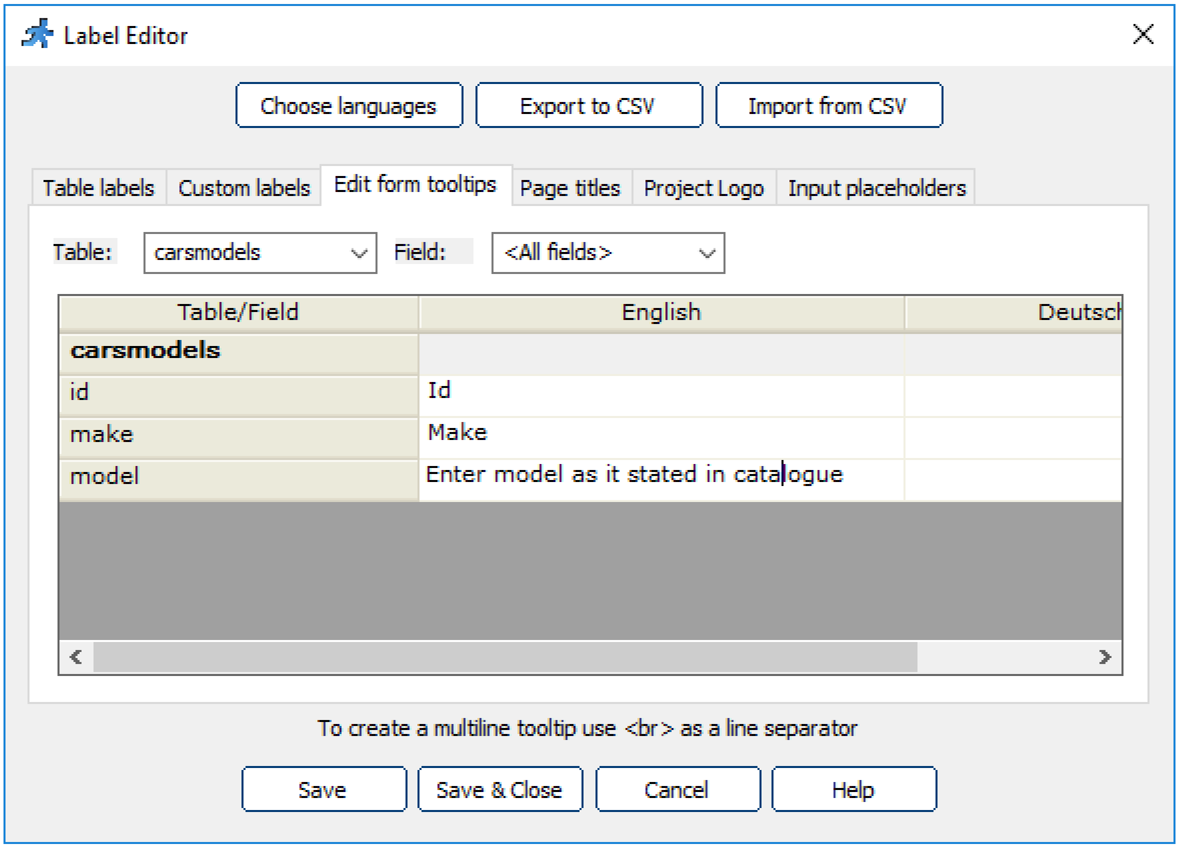
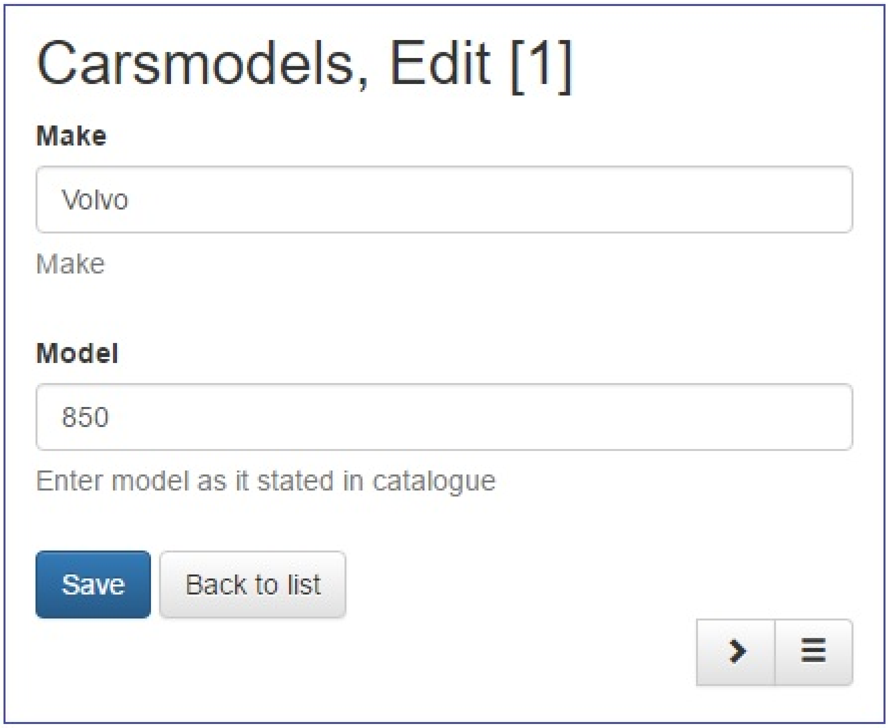


Table: carsmodels Field: <All fields>

Table/Field	English	Deutsch
carsmodels		
id	Id	
make	Make	
model	Enter model as it stated in catalogue	

To create a multiline tooltip use <br> as a line separator

How it looks in browser:



Carsmodels, Edit [1]

Make

Volvo

Make

Model

850

Enter model as it stated in catalogue

Page titles

To edit a web page title, deselect a Default checkbox next to it and make changes. Filter page titles by table or page to limit the data displayed.

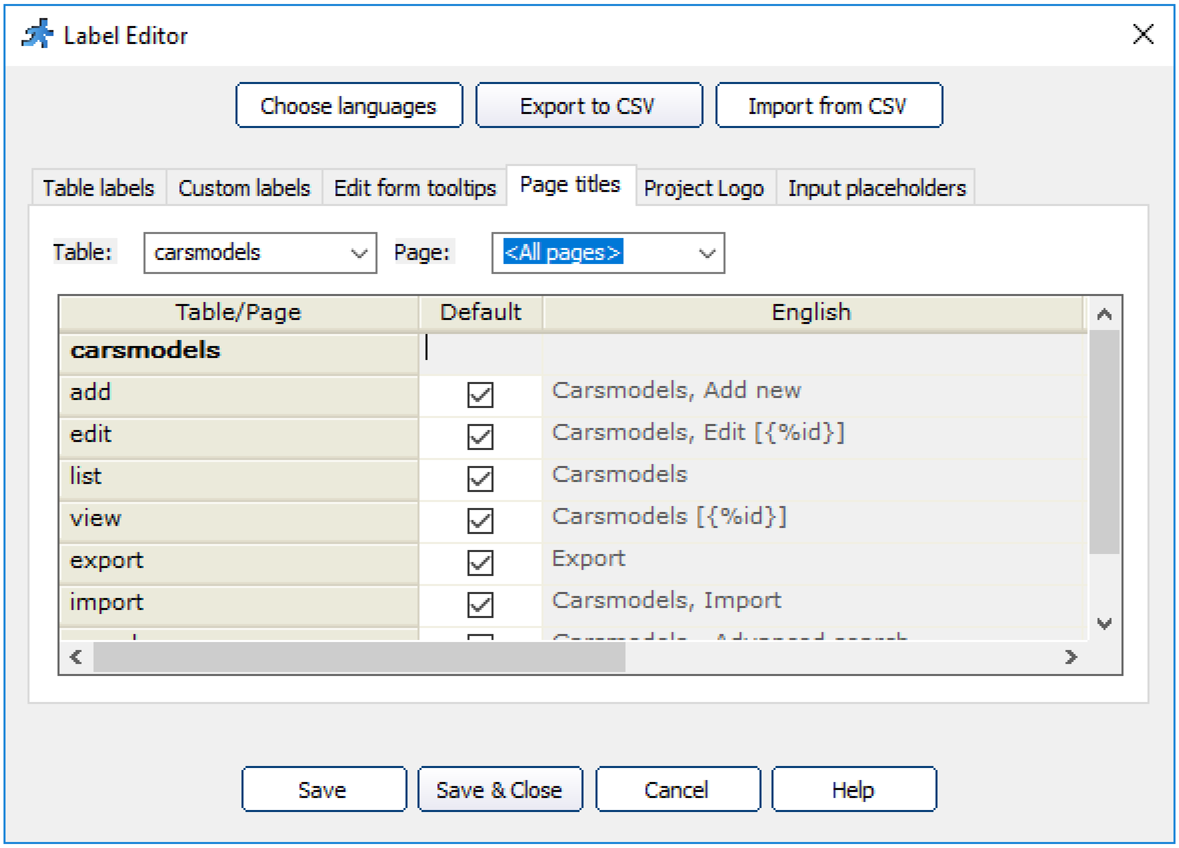
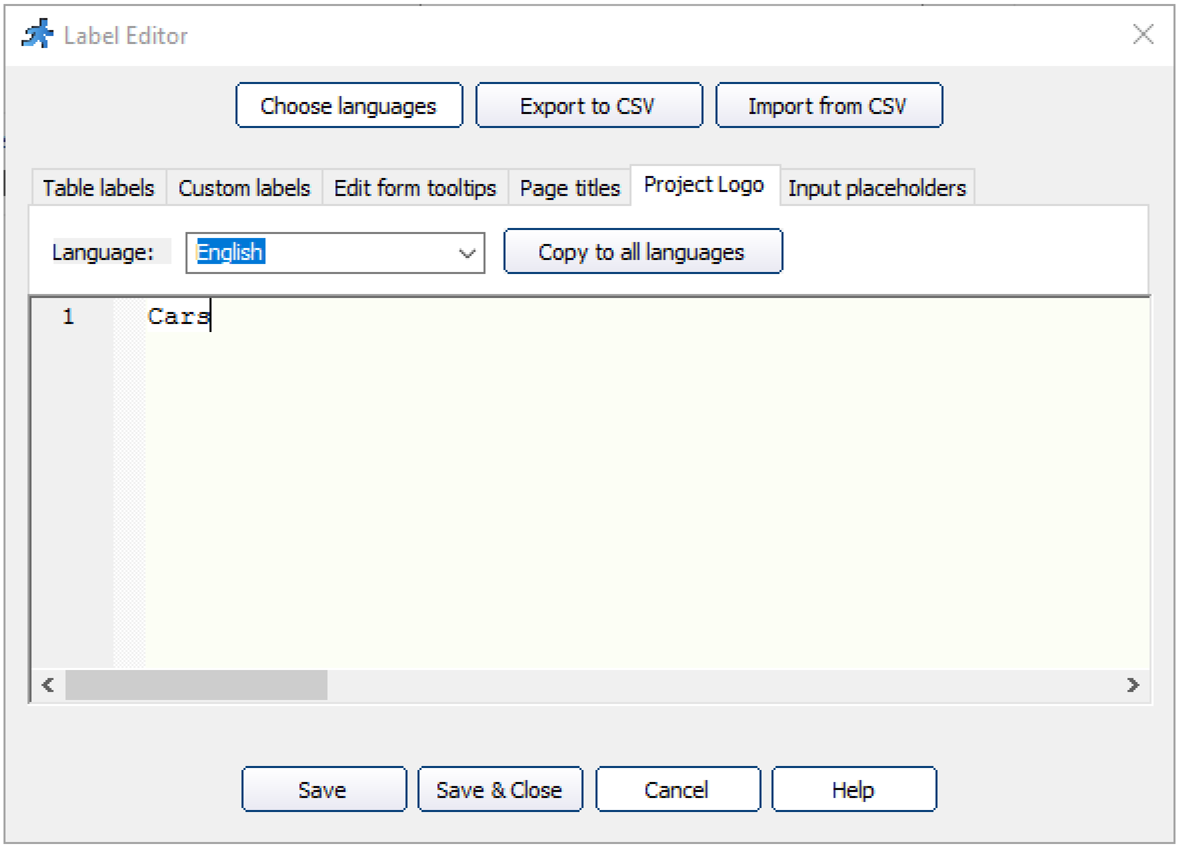


Table: carsmodels Page: <All pages>

Table/Page	Default	English
carsmodels		
add	☑	Carsmodels, Add new
edit	☑	Carsmodels, Edit [{%id}]
list	☑	Carsmodels
view	☑	Carsmodels [{%id}]
export	☑	Export
import	☑	Carsmodels, Import

Project logo



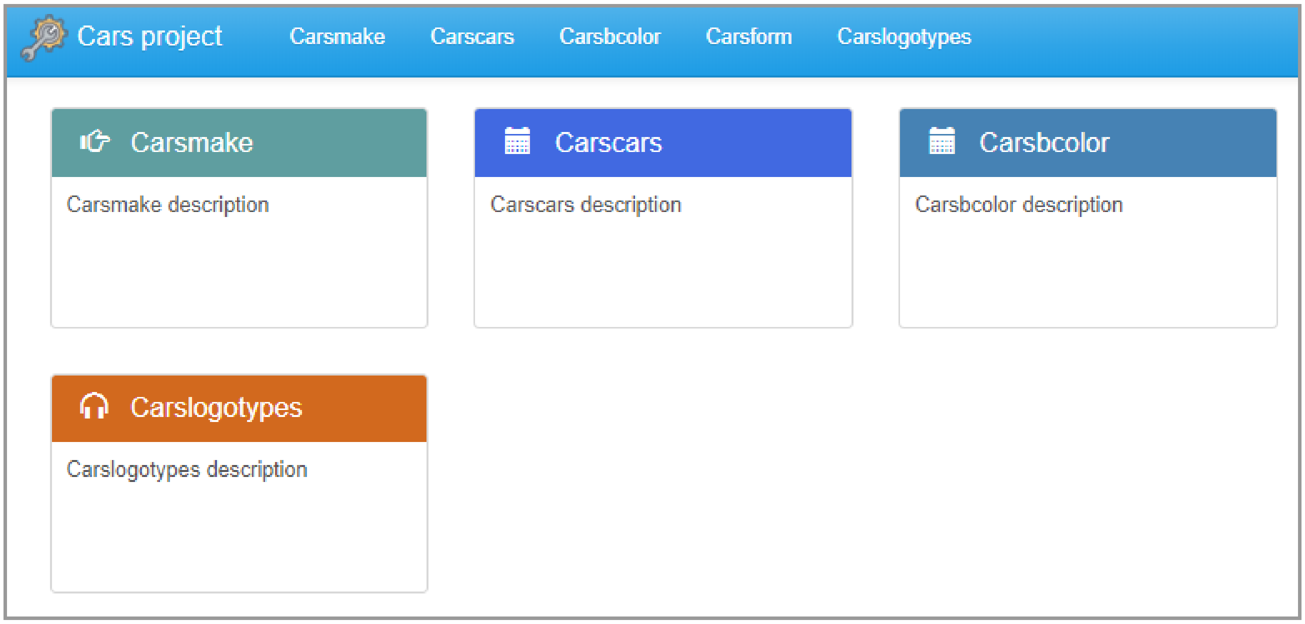
Language: English

```
Cars
```

To change your project logo, you can enter here text or your HTML-code.
Here is an example of the code that can be used to display logo icon and text:

```
<DIV style="position: relative; height: 40px; top: -6px; white-space: nowrap;">
      <IMG height="80%" id="navbarlogo" alt="Project logo"
src="http://mywebsite.com/images/logo.png">
      Cars project
</DIV>
```

That's how it will look on the generated page:



You can do even more with PHP code by using the function setProjectLogo from the Labels/Titles API.

In the following example you are adding logo and the current year to the project logo.

Simply put the logo.png to the images folder of your application and add the following code to AfterAppInit event.

```
$currentYear = date("Y");
Labels.setProjectLogo('
<DIV style="position: relative; height: 40px; top: -6px; white-space: nowrap;">
      <IMG height="80%" id="navbarlogo" alt="Project logo"
src="http://mywebsite.com/images/logo.png">
      Cars project'.$currentYear.'
</DIV>
');
```

See more about using Labels/Titles API here.

Form field placeholders

Placeholders are in-place tooltips that disappear as soon user starts typing something in that field.

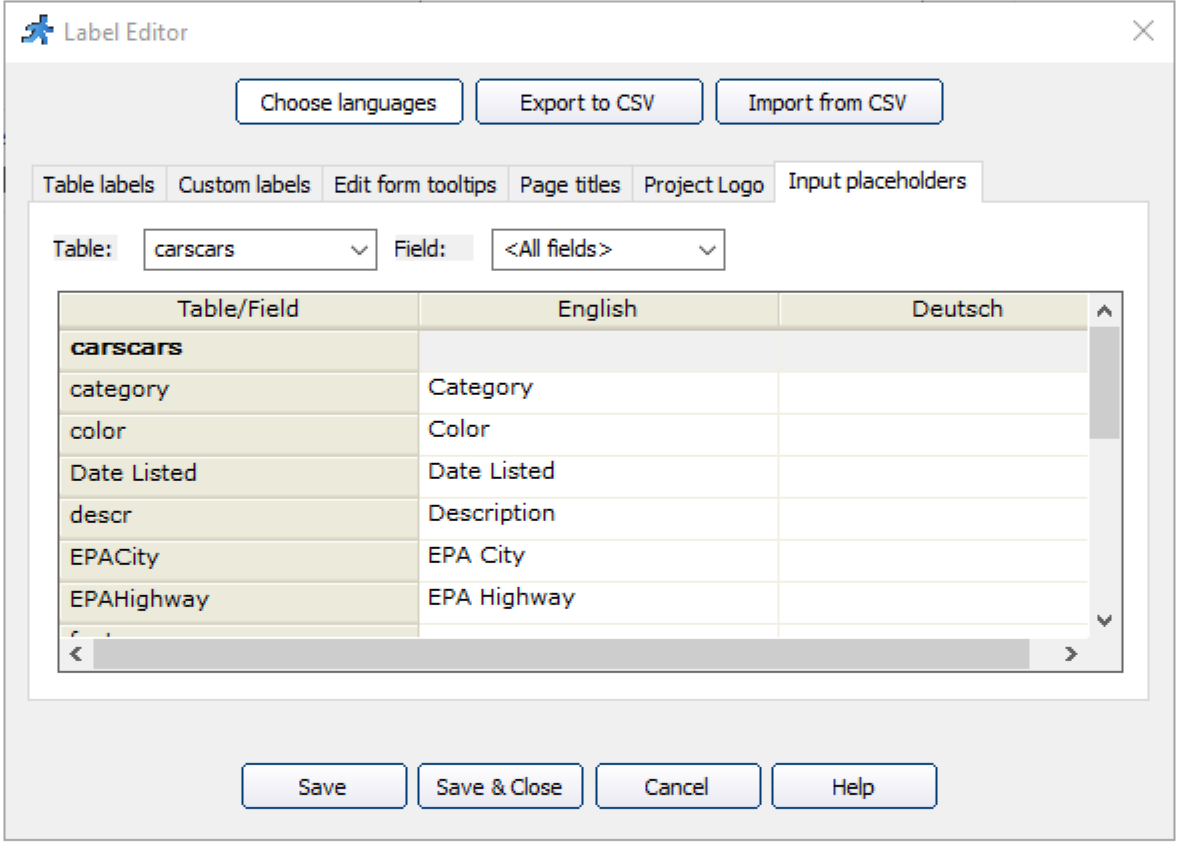


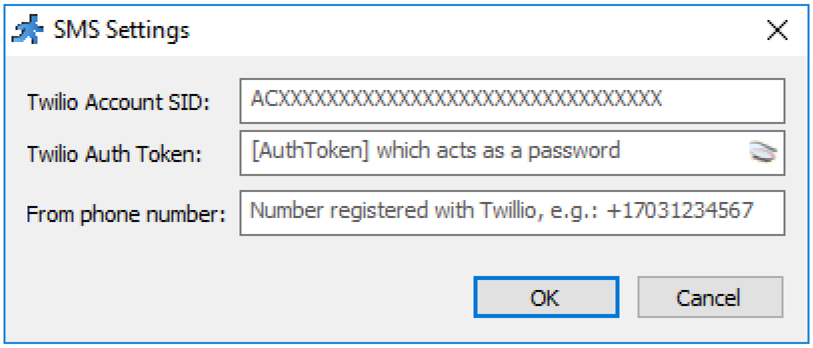
Table: carscars

Field: <All fields>

Table/Field	English	Deutsch
carscars		
category	Category	
color	Color	
Date Listed	Date Listed	
descr	Description	
EPACity	EPA City	
EPAHighway	EPA Highway	

Live Demo

SMS Settings



Twilio Account SID: ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Twilio Auth Token: [AuthToken] which acts as a password

From phone number: Number registered with Twillio, e.g.: +17031234567

Here you can set you Twilio Account that can be used used to send SMS data to users.
To use that functionality first you need to Open an account with https://www.twilio.com/

Note that Twilio is not free. Their rates depend on number of messages you going to send and many other things.

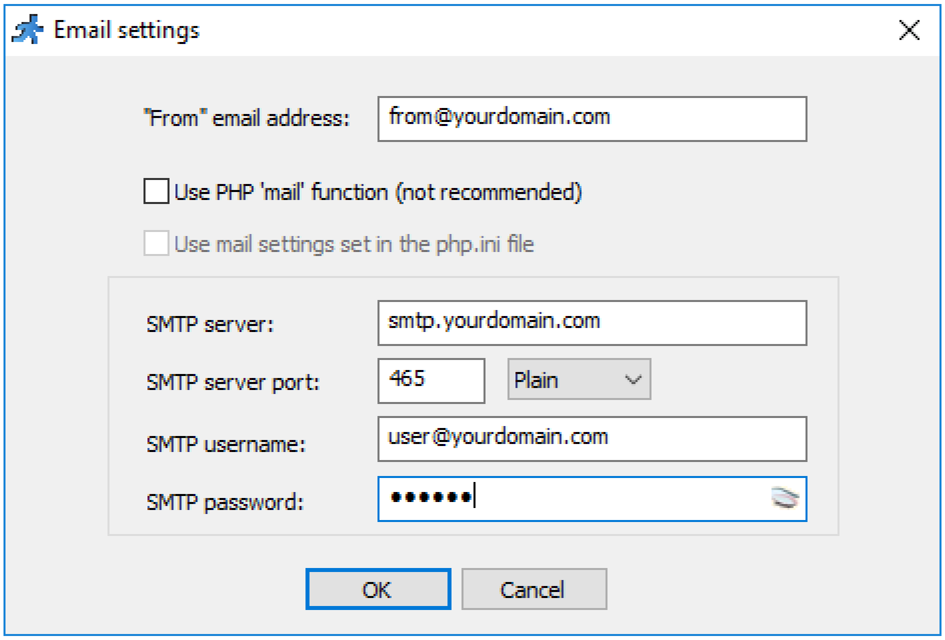
See more:

Two-factor authentication

Runner_sms function

Email settings

Here you can enter email which will be used to send emails to users and define settings of the custom mail server, if you do not use the built-in mail server. Note that SMTP server, SMTP server port as well as Secure connection (SSL) settings may differ from the stated in the example below. Contact your Mail Service Provider to get necessary information.



Email settings

"From" email address: from@yourdomain.com

☐ Use PHP 'mail' function (not recommended)

☐ Use mail settings set in the php.ini file

SMTP server: smtp.yourdomain.com

SMTP server port: 465 Plain

SMTP username: user@yourdomain.com

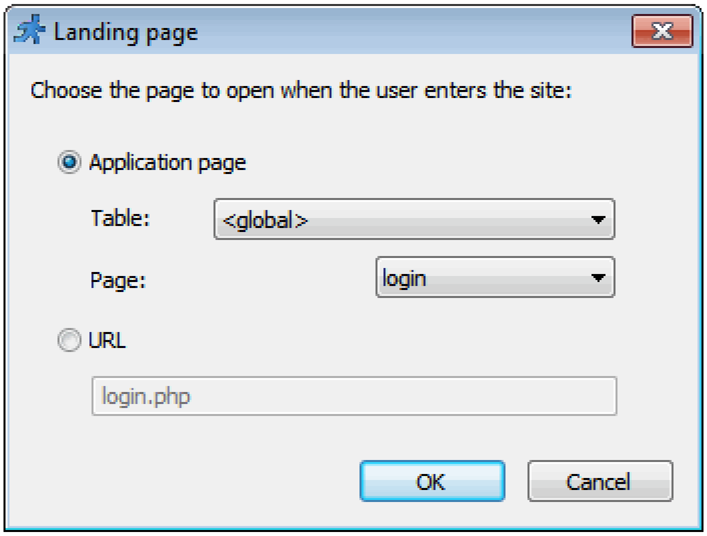
SMTP password: ••••••

To send emails via Gmail, use the following settings:

- **SMTP server**: smtp.gmail.com
- **SMTP server port**: 465
- **Secure connection (SSL):** On
- **SMTP username**: <your gmail address>
- **SMTP password**: <Gmail password>

Landing page

Select the page to open when the user enters the site (if login is disabled or login and guest user login are enabled) or just after login. You can enter custom URL, e.g URL with search options *carscars_list.php?qs=audi.*



Choose the page to open when the user enters the site:

● Application page

Table: <global>

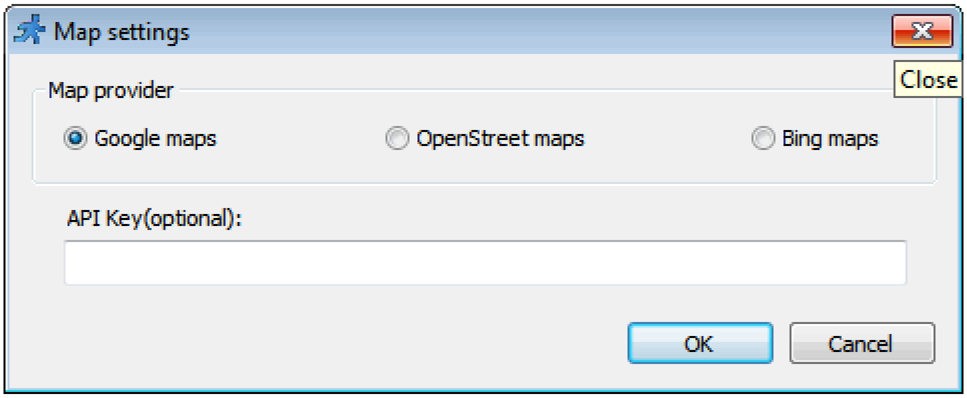
Page: login

○ URL

login.php

Map settings

Select map provider and enter API Key (if required). You can get Bing maps API key for development purposes for free at bingmapsportal.com.



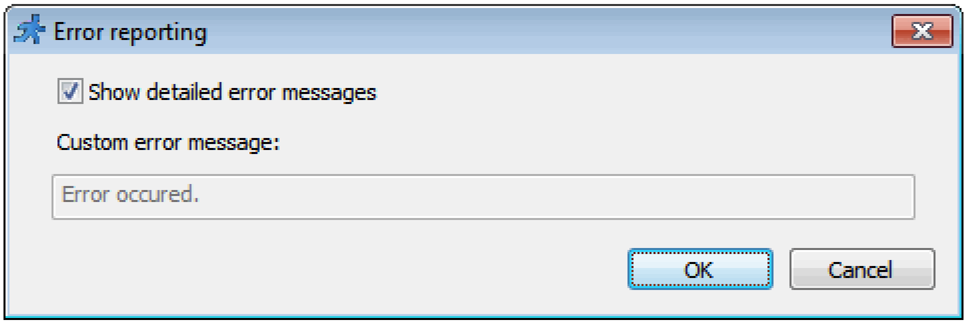
Map provider

● Google maps ○ OpenStreet maps ○ Bing maps

API Key(optional):

Error reporting

You can select to display detailed error messages or your custom error message.



☑ Show detailed error messages

Custom error message:

Error occured.

Section 508 compatibility

Select this check box to make online content accessible to a wide range of people with disabilities. For more information, visit http://www.section508.gov/.

Check keyboard-based navigation through table links on the List page (Arrow Up, Arrow Down, Tab, Shift Tab). The keyboard shortcuts available are described below:

- ALT+E - inline edit current record;
- ALT+S - save current record;
- ALT+C - cancel inline edit;
- ALT+F - go to the advanced search;
- ALT+M - jump to the menu;
- CTRL+left arrow - previous page;
- CTRL+right arrow - next page.

When using special internet browser readers, the **Hotkeys reference** link will be available.

Build Mobile version

Allows you to build a mobile version of your website automatically. For more information, see Mobile template.

Table specific settings

Miscellaneous settings that can vary from table to table. Use drop-down box with table names to switch between tables.

Warn about leaving Add/Edit pages

Displays a warning message before leaving Add/Edit pages with unsaved changes.

Move Next option

Displays Next/Prev buttons on Edit/View pages that allows to proceed to the next record without going back to the list page.

Rollover row highlighting

Select this check box to turn on row highlighting feature. It highlights current table row on the list page.

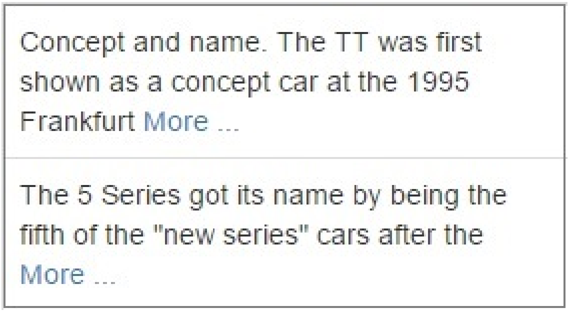
Resizable table columns

Allows resizing any column on the **List** page by dragging its edge.

Search and Filter settings

Allows adjusting search and filter settings. For more information, see Choose fields: Search and Filter settings.

Truncate large text fields



Concept and name. The TT was first shown as a concept car at the 1995 Frankfurt More ...

The 5 Series got its name by being the fifth of the "new series" cars after the More ...

This option adds "More ..." link to long text field while first XXX characters are displayed on the page. "More ..." link displays a popup with the whole field content.

Also you can enter number of records to appear on each list page.

Note: to make this option work correctly you must choose the key field for the selected table on the Choose pages screen.

No records on the first page

With this option turned on no records will be shown on initial page load. You will need to run a search in order to display results.

Records per page

The initial number of records to be displayed

Records per page selection

Defines what options to show in pagination control. "all" (without quotes) means show all records.

Enterprise Edition settings

These settings are available only in the Enterprise Edition of PHPRunner.

Web reports

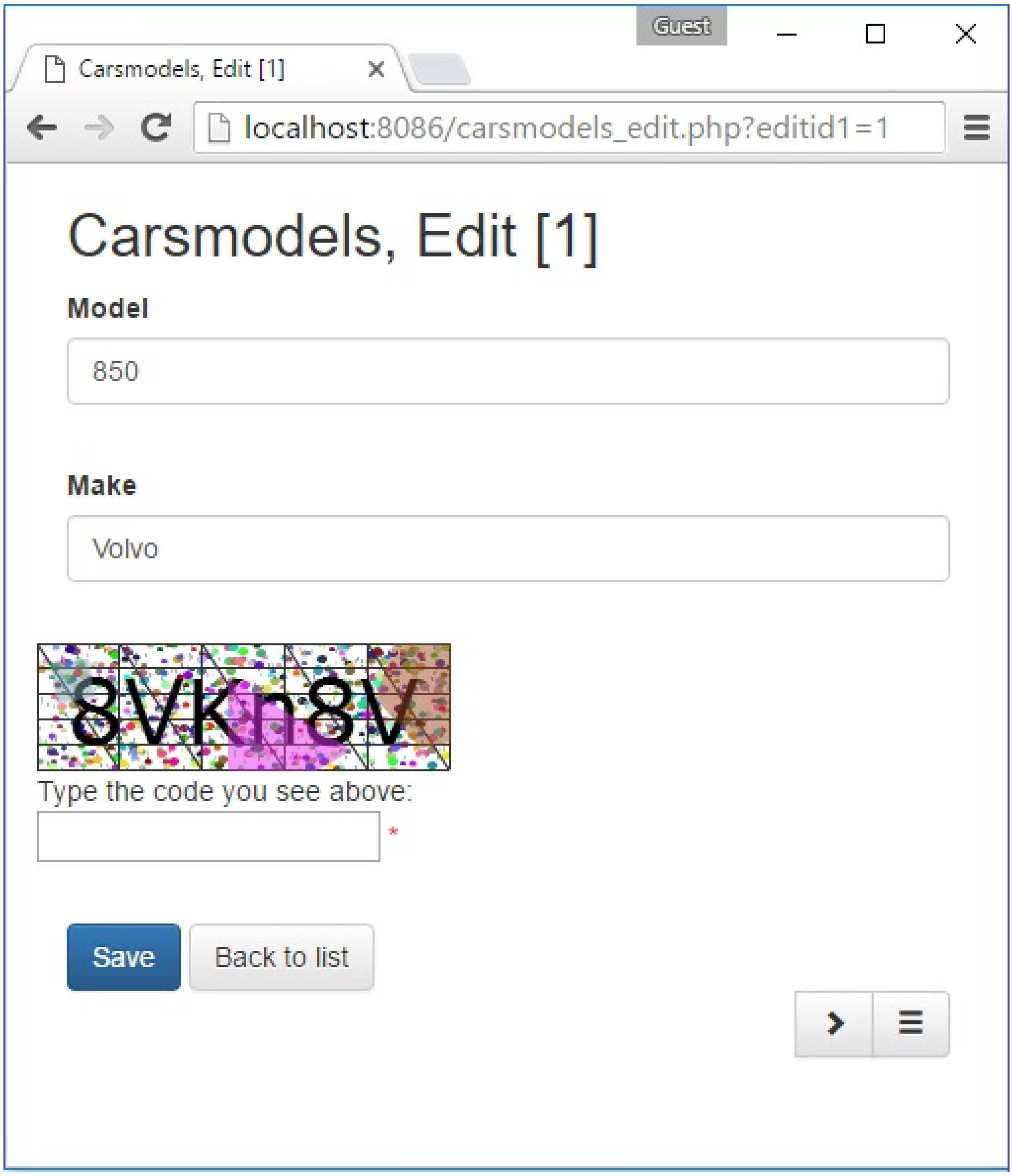
Enable online report/chart builder. Click **Administrator** button to specify the password to access the Web Reports and Charts admin area.

2.14 Adding CAPTCHA

CAPTCHA is a simple test to determine if a user is a computer or a human. It is used to prevent spam abuse on the websites. So if you use CAPTCHA in your application, this can help in stopping some bots and making life harder for other bots in accessing or using your project.

You can add CAPTCHA to the Add/Edit/Login/Register/Remind password pages. You can choose between Flash-based CAPTCHA (it is simple but doesn't work on most mobile devices) and Google reCAPTCHA.

Flash-based CAPTCHA on Edit page:



Carsmodels, Edit [1]

Model

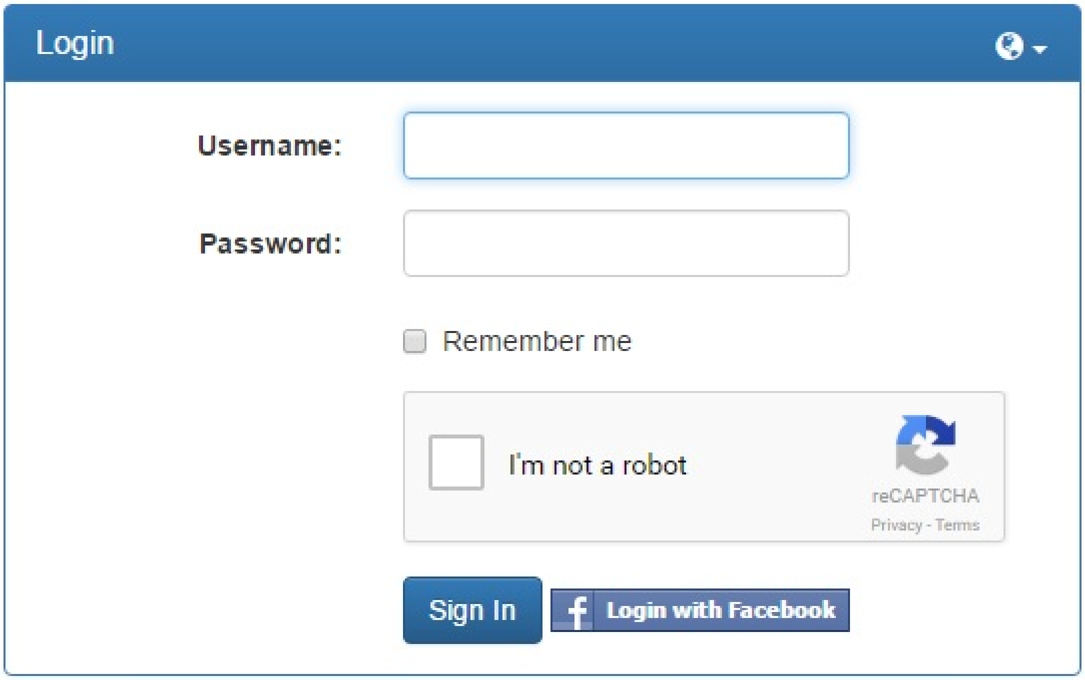
850

Make

Volvo

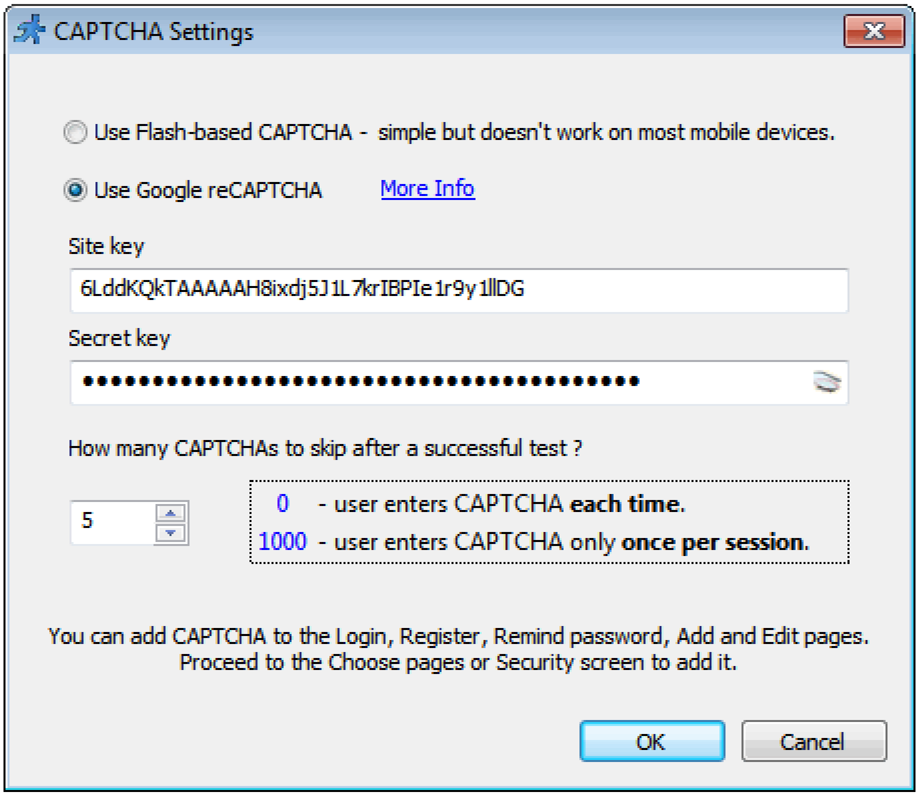
Type the code you see above:

Google reCAPTCHA on Login page:



You can add CAPTCHA to Add or Edit page on the Choose pages screen (click **Settings** near Edit record or Add new and select **Display CAPTCHA** checkbox). To add CAPTCHA to Login page, on the Security screen click Login form appearance and select **Display CAPTCHA** checkbox. To add CAPTCHA to Register and Remind password pages, on the Security screen click Registration and passwords and select **Display CAPTCHA** checkbox.

Then click **CAPTCHA settings** and choose what CAPTCHA to use: Flash-based CAPTCHA (it is simple but doesn't work on most mobile devices) or Google reCAPTCHA. To use Google reCAPTCHA, register your web site at https://www.google.com/recaptcha/intro/index.html, copy site key and secret key there and paste them into the CAPTCHA settings.



CAPTCHA Settings

○ Use Flash-based CAPTCHA - simple but doesn't work on most mobile devices.

● Use Google reCAPTCHA More Info

Site key

6LddKQkTAAAAAH8ixdj5J1L7krIBPIe1r9y1llDG

Secret key

•••

How many CAPTCHAs to skip after a successful test ?

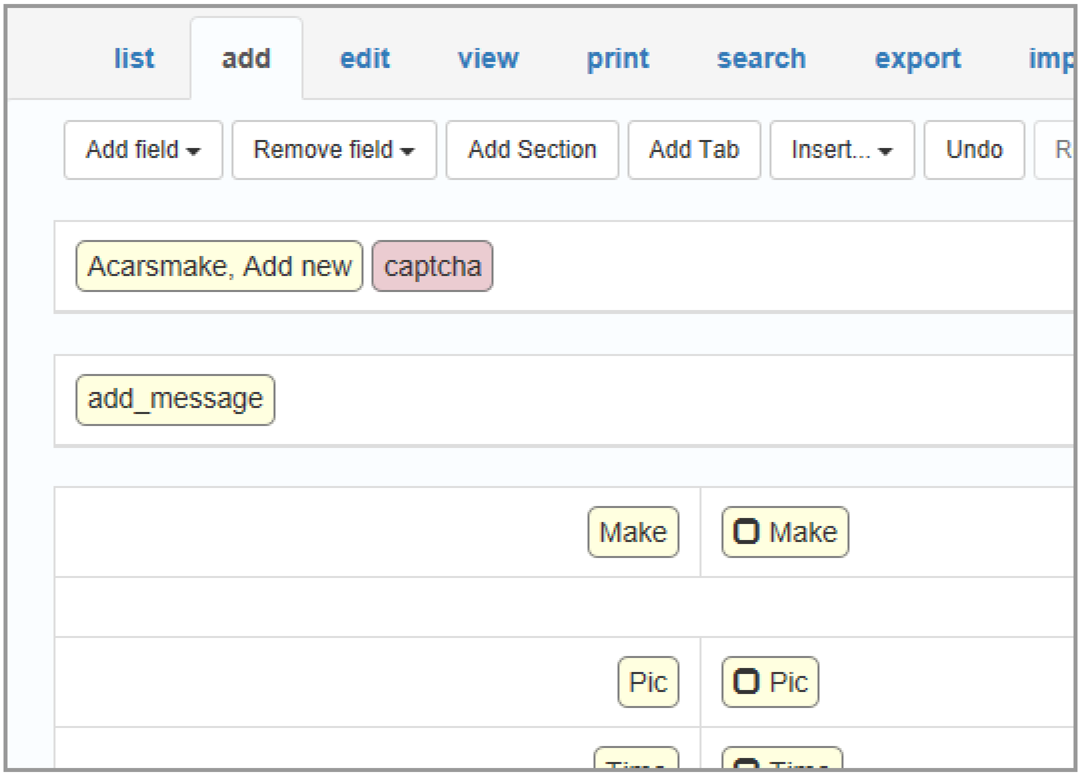
5

0 - user enters CAPTCHA **each time.**

1000 - user enters CAPTCHA only **once per session.**

You can add CAPTCHA to the Login, Register, Remind password, Add and Edit pages. Proceed to the Choose pages or Security screen to add it.

When you open a page with CAPTCHA you can see a page element labeled **captcha**. Drag this element to change its location.



2.15 Security

2.15.1 Login page

You can password-protect access to your database with PHPRunner. You can choose one the following options:

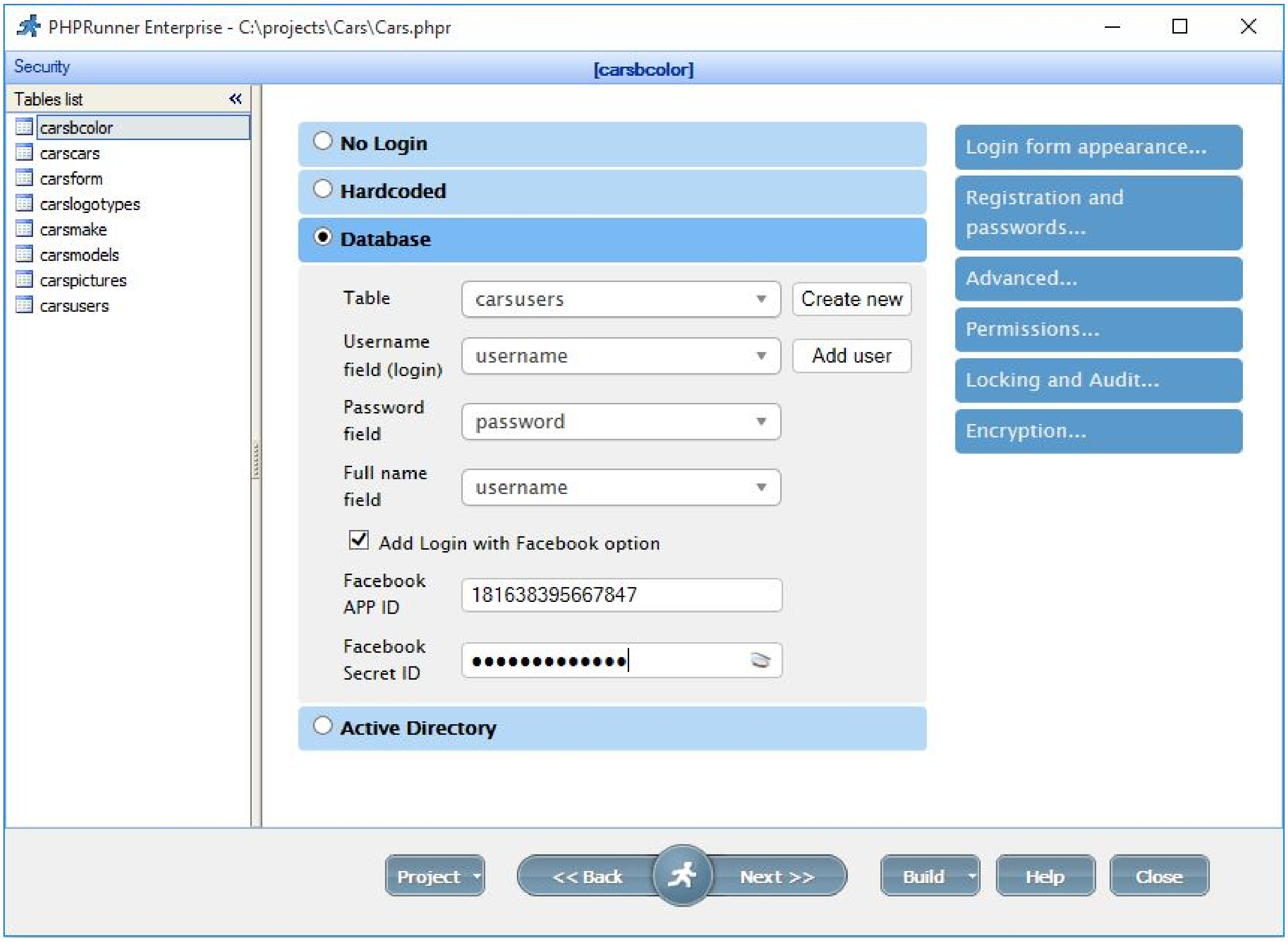
- **No Login -** there is no authentication.
- **Hardcoded** - all your users will use the same hardcoded login/password combination.
- **Database -** choose this method if you store username/password combinations in your database. In this case you need to choose database table with username/passwords data and fields that store usernames, passwords and user full name that will be displayed in the **Logged as <>** phrase. You can create new login table by clicking **Create new** and add new users to this table by clicking **Add user**.

  If it is necessary to display something more complex than just one field in the **Logged as <>** phrase, you can use AfterSuccessfulLogin event and, for example, the following code:

  ```
  $_SESSION["UserName"] = $data["FirstName"].' '.$data["LastName"]
  ```

  For more information about **Login with Facebook** option, see Facebook connect.

- **Active Directory** - Active Directory authentication. For more information, see Active Directory.



○ **No Login**

○ **Hardcoded**

● **Database**

Table: carsusers

Username field (login): username

Password field: password

Full name field: username

☑ Add Login with Facebook option

Facebook APP ID: 181638395667847

Facebook Secret ID: •••••••••••••

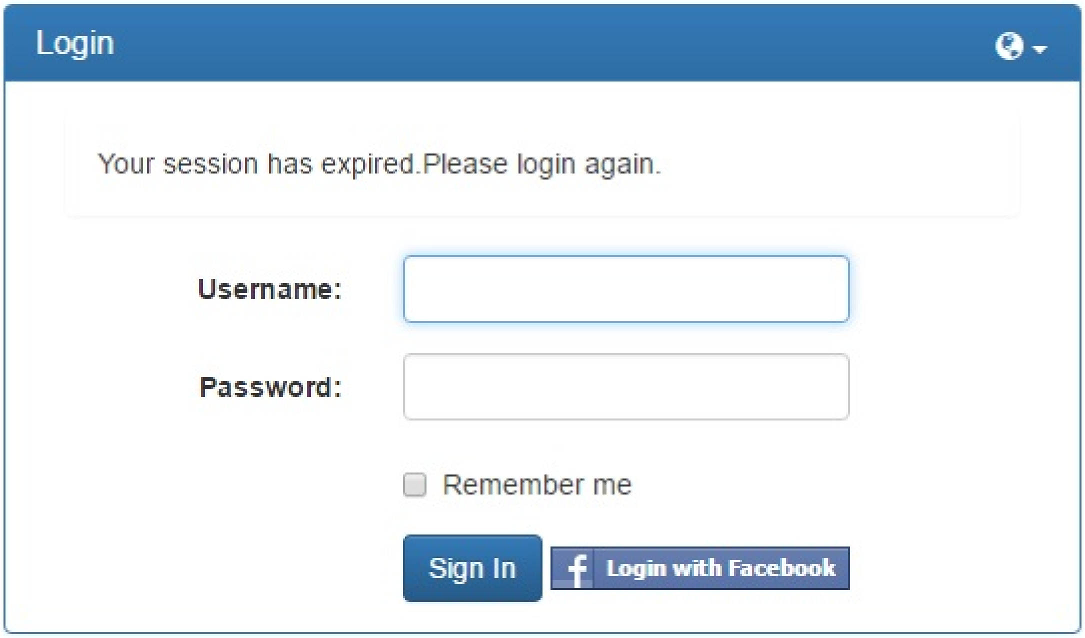
○ **Active Directory**

If using authentication PHPRunner will generate additional PHP page *login.php* intended for username and password validation.

Registration and passwords allows you to create user self-register page, password reminder, change password pages. Use Locking and Audit to set up record locking and user actions logging. Use Encryption to encrypt important data in the database. When using authentication, choose Login form appearance. If **Database** or **Active Directory** option is selected, you can set Advanced Security Settings and define Permissions.

If you need to customize email templates that will be send to the users/admins when a new user is registered use the Email templates dialog.

Example of Login page:



Login

Your session has expired.Please login again.

Username:

Password:

□ Remember me

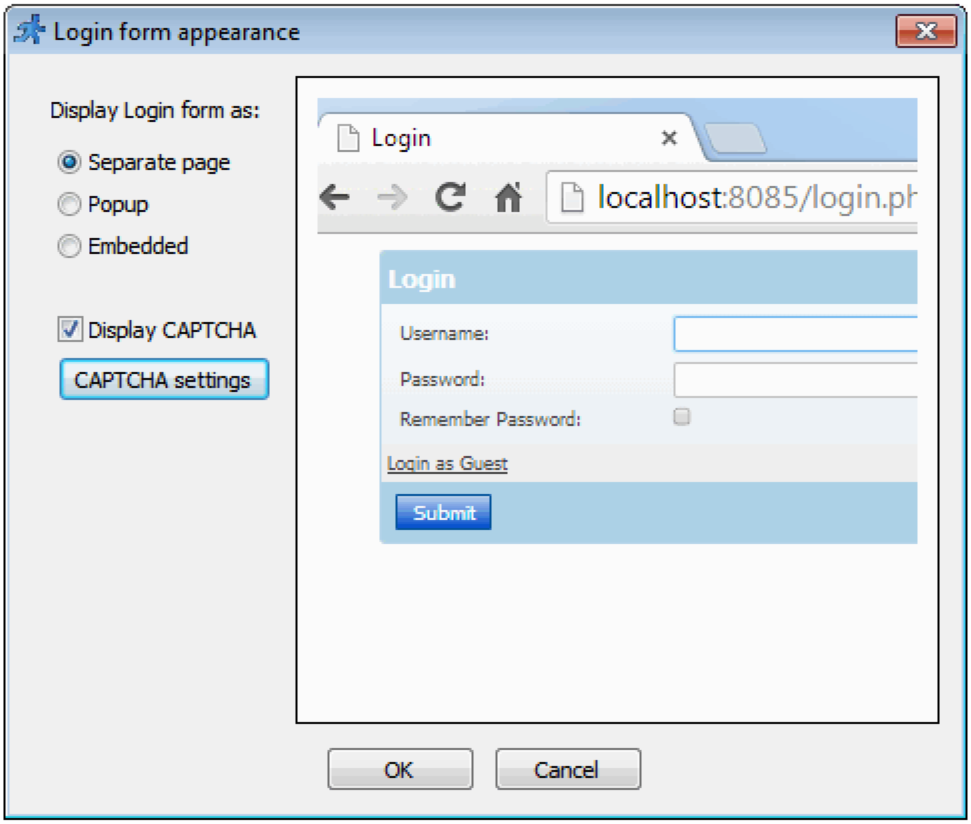
If you have selected the **Remember Password** check box and then opens one of the internal page of application, such as a Menu or List page, you will be logged in automatically.

Scenarios on how you can configure security options

- "Only one person or several persons with the same access rights have access to the site" - use the **Hardcoded username and password** option.
- "There are the site administrator (owner) and guests. The administrator has full access to the site, guests have read-only access to some pages/reports/charts" - use the **Username and password from database** option, enable guest login in the Advanced security settings, configure the access for guests in the User group permissions.
- "There are many users with different access levels and administrators with the full access" - use the **Username and password from database** option, configure Advanced security settings if you need to restrict the access within one table, configure User group permissions to assign table level permissions, configure Admin group for administrators.
- "If all user accounts and passwords are stored in the Active Directory" - use the **Active Directory** option.

2.15.2 Login form appearance

Choose between separate login page, login box in popup or embedded into the page. Now you have an option to logon right from the List page. But make sure that guests have access to that page. You can add CAPTCHA to the Login page.



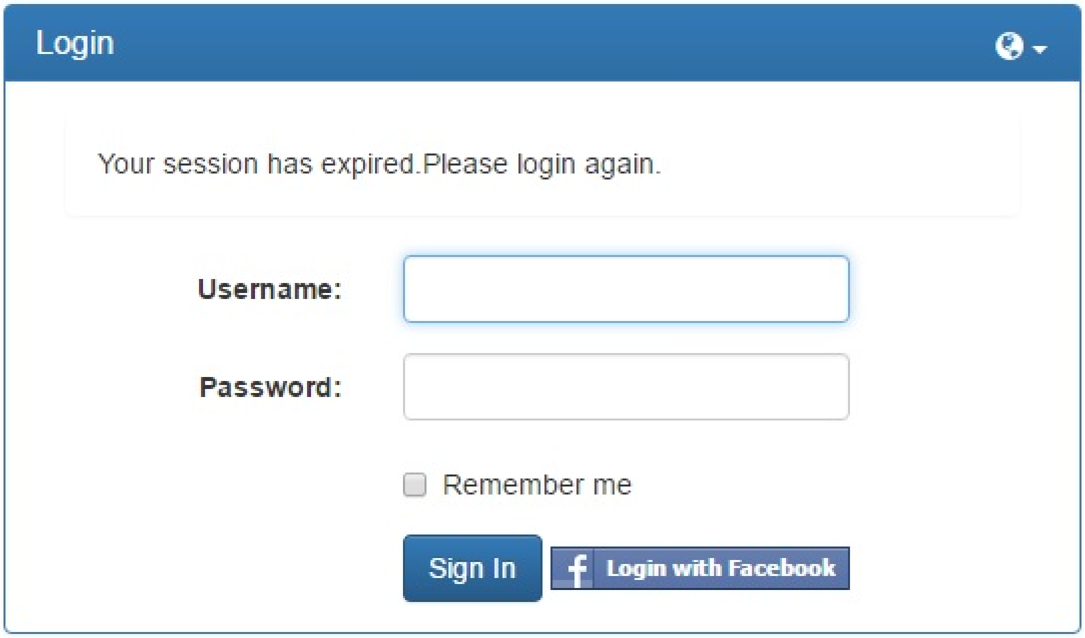
Login form appearance

Display Login form as:

● Separate page
○ Popup
○ Embedded

☑ Display CAPTCHA

Separate login page:



Login

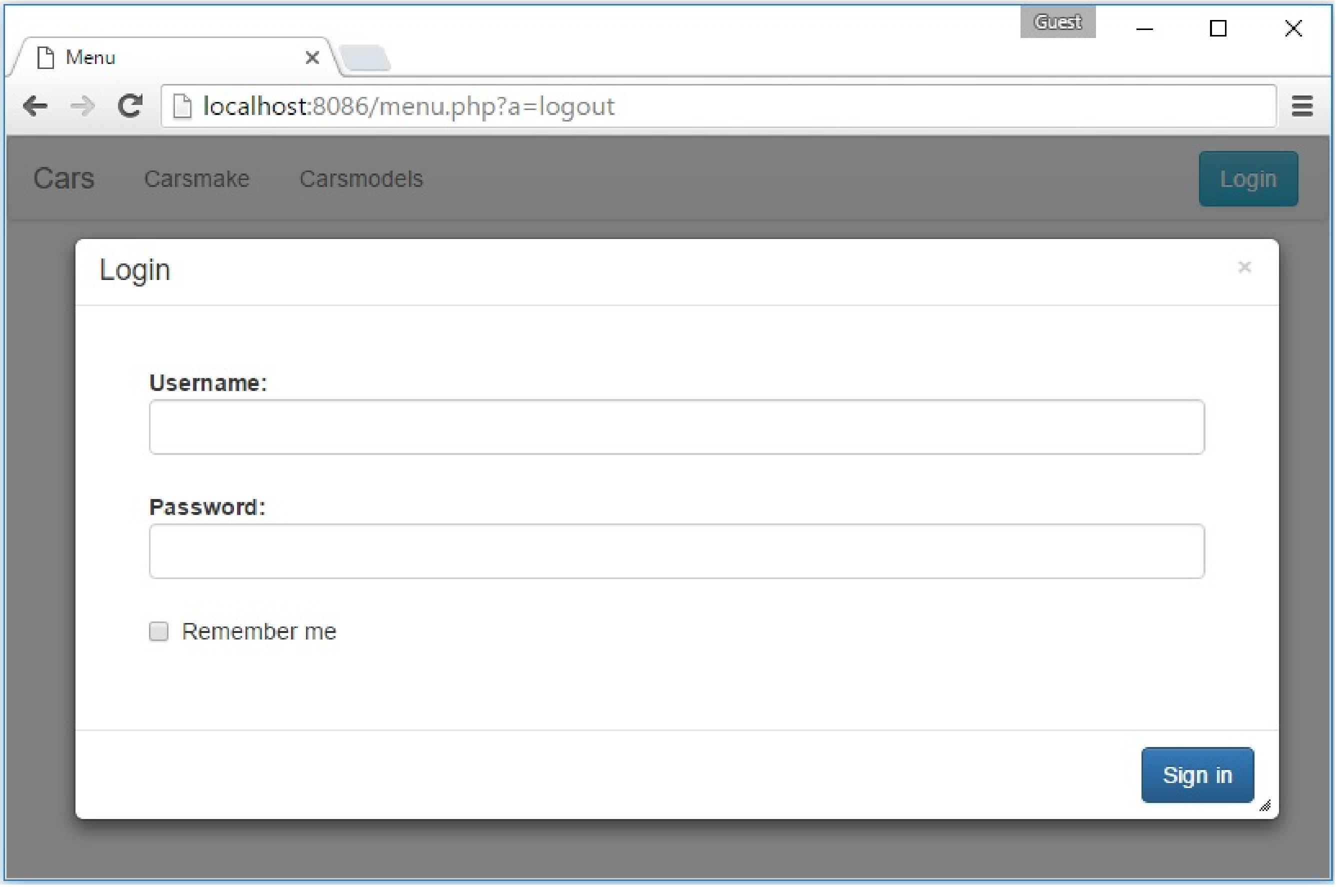
Your session has expired.Please login again.

Username:

Password:

□ Remember me

Login box in popup:



Login

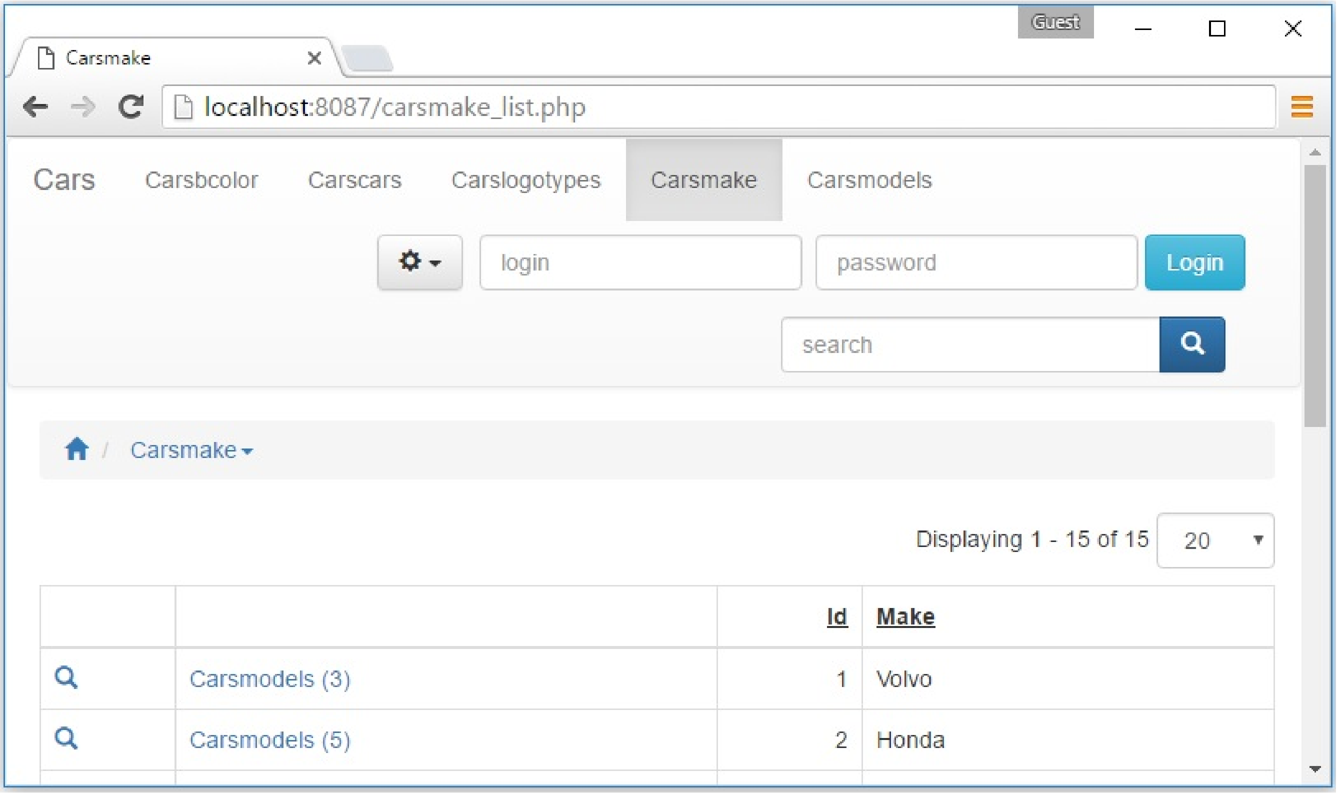
Username:

Password:

□ Remember me

Sign in

Embedded login box:

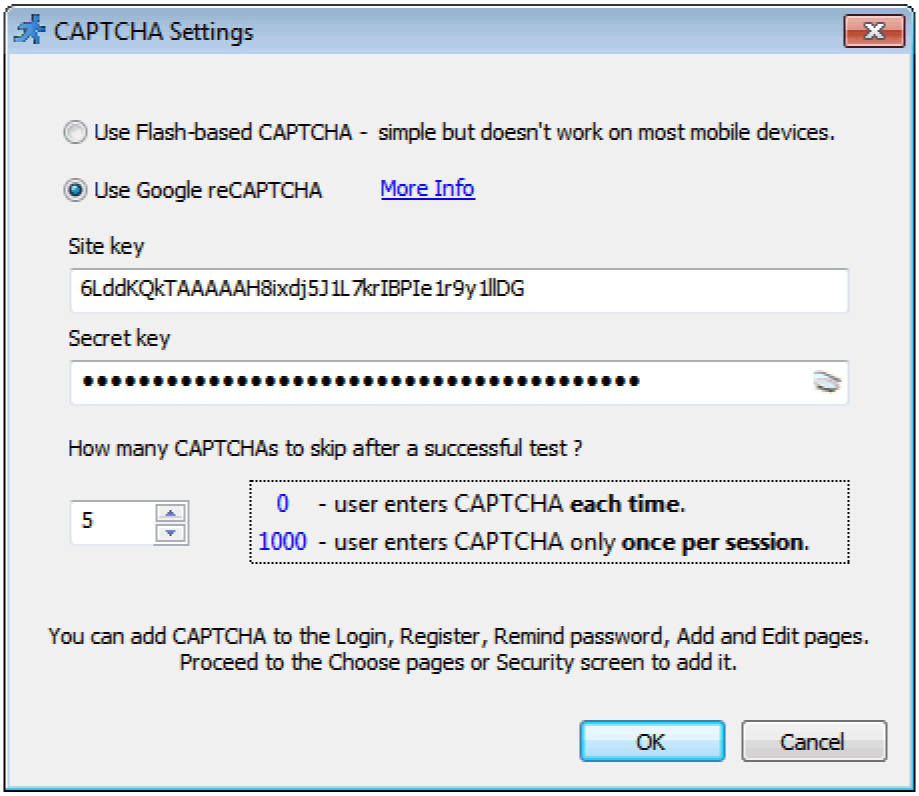


		Id	Make
	Carsmodels (3)	1	Volvo
	Carsmodels (5)	2	Honda

Adding CAPTCHA

To add CAPTCHA to the Login page select the **Display CAPTCHA** checkbox. CAPTCHA is a simple test to determine if a user is a computer or a human. It is used to prevent spam abuse on the websites. Click **CAPTCHA settings** and choose what CAPTCHA to use: Flash-based CAPTCHA (it is simple but doesn't work on most mobile devices) or Google reCAPTCHA.

To use Google reCAPTCHA, register your web site at https://www.google.com/recaptcha/intro/index.html, copy site key and secret key there and paste them into the CAPTCHA settings.



CAPTCHA Settings

○ Use Flash-based CAPTCHA - simple but doesn't work on most mobile devices.

● Use Google reCAPTCHA More Info

Site key

6LddKQkTAAAAAH8ixdj5J1L7krIBPIe1r9y1llDG

Secret key

●●

How many CAPTCHAs to skip after a successful test ?

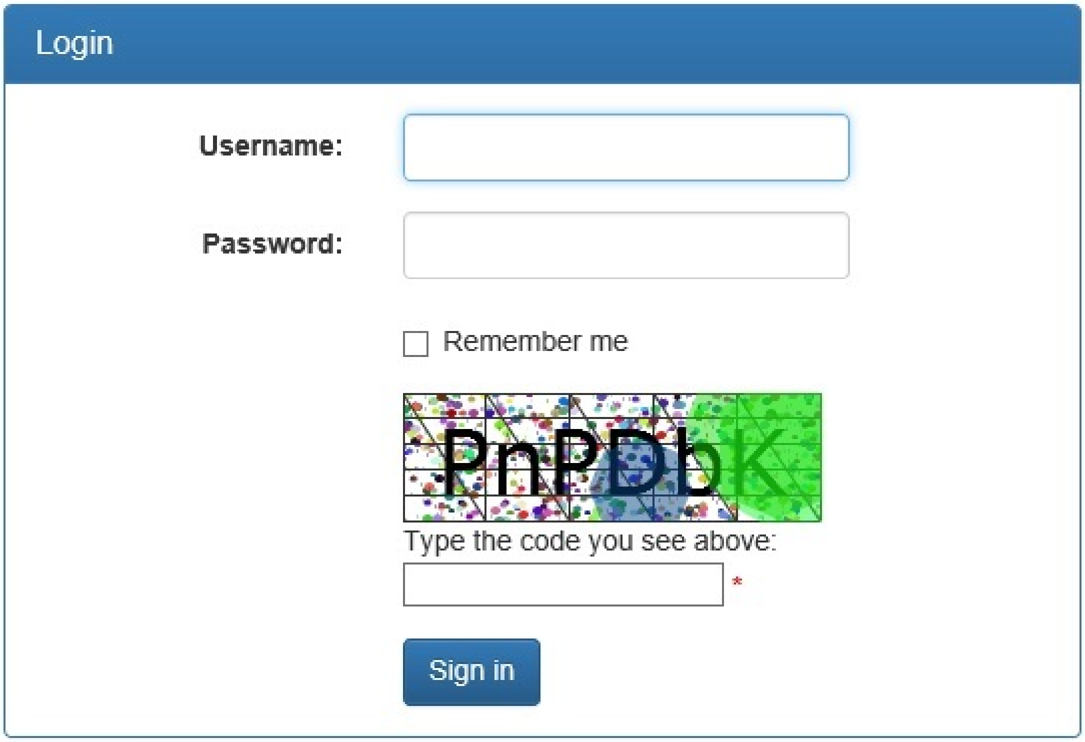
5

0 - user enters CAPTCHA **each time.**

1000 - user enters CAPTCHA only **once per session.**

You can add CAPTCHA to the Login, Register, Remind password, Add and Edit pages. Proceed to the Choose pages or Security screen to add it.

Flash-based CAPTCHA:



Login

Username:

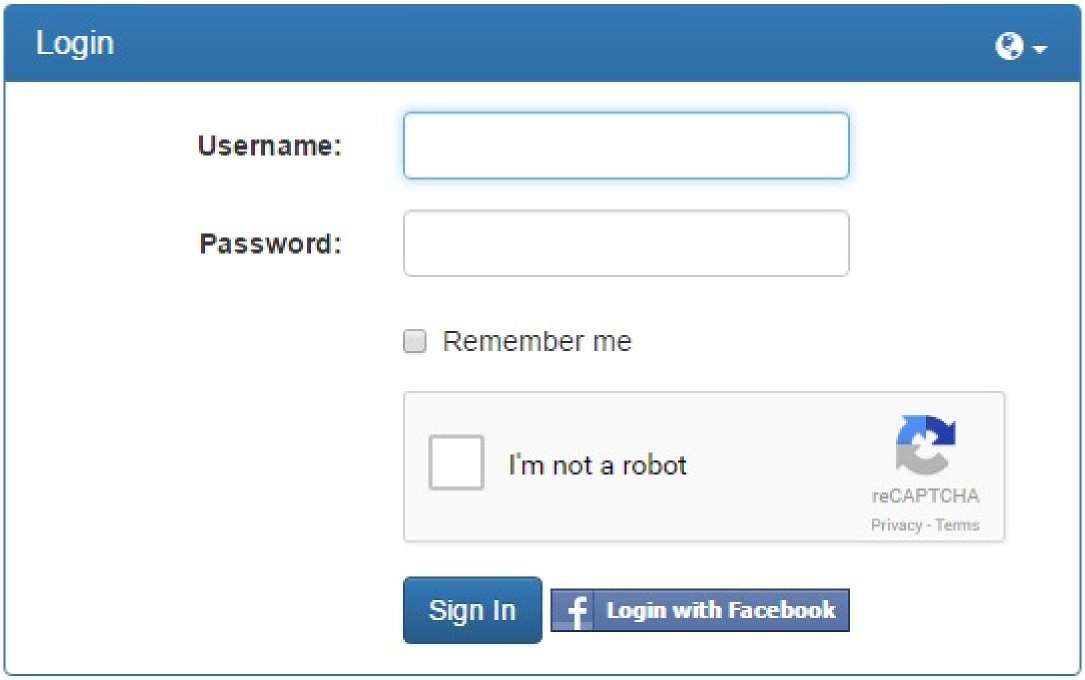
Password:

□ Remember me

Type the code you see above:

*

Google reCAPTCHA:



If you need to customize email templates that will be send to the users/admins when a new user is registered use the Email templates dialog.

2.15.3 Registration and passwords

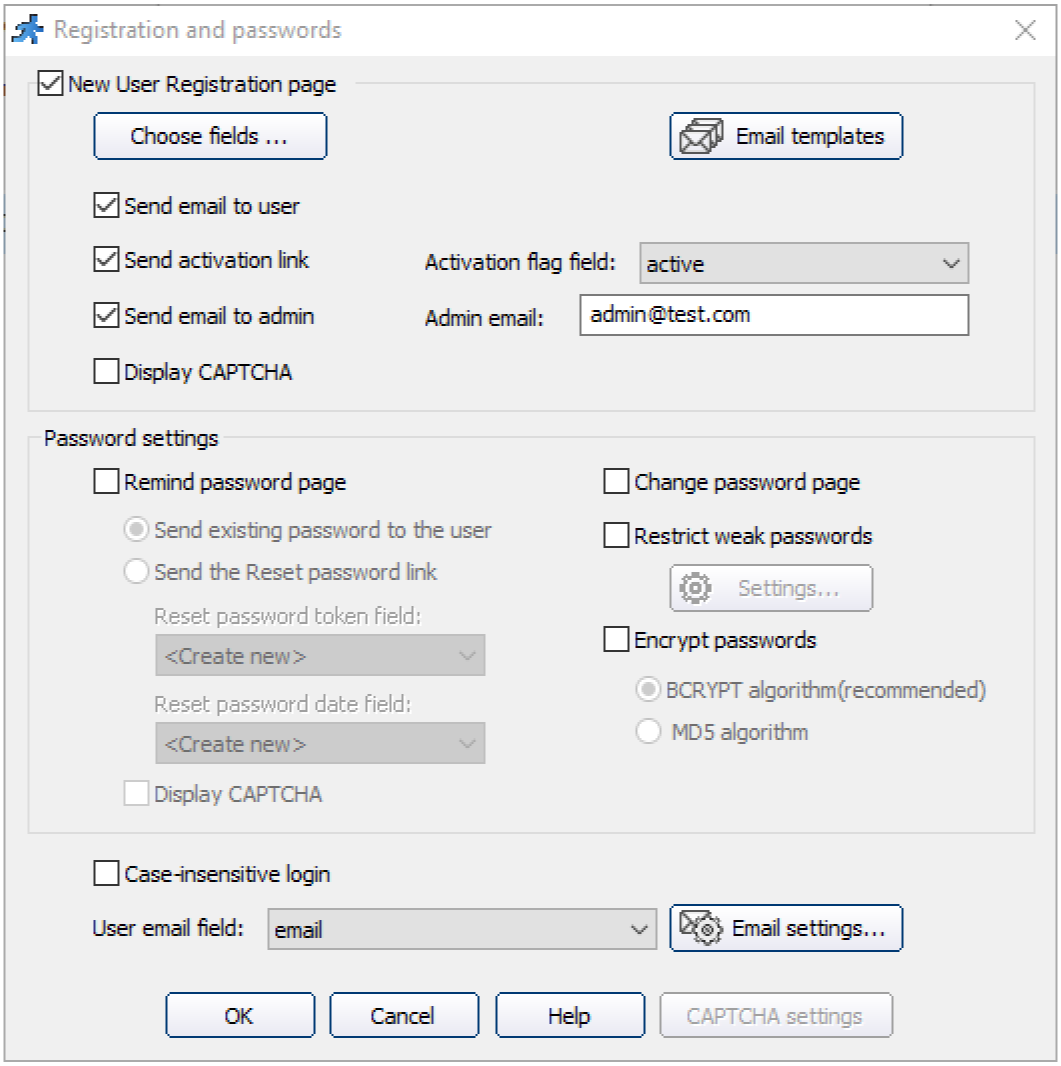
Quick jump

Adding CAPTCHA

Email settings

Email templates

On **Security** page click **Registration and passwords** to create user self-register page or password reminder, change password pages.



Registration and passwords

☑ New User Registration page

☑ Send email to user

☑ Send activation link — Activation flag field: active

☑ Send email to admin — Admin email: admin@test.com

□ Display CAPTCHA

Password settings

□ Remind password page

○ Send existing password to the user

○ Send the Reset password link

Reset password token field: <Create new>

Reset password date field: <Create new>

□ Display CAPTCHA

□ Change password page

□ Restrict weak passwords

□ Encrypt passwords

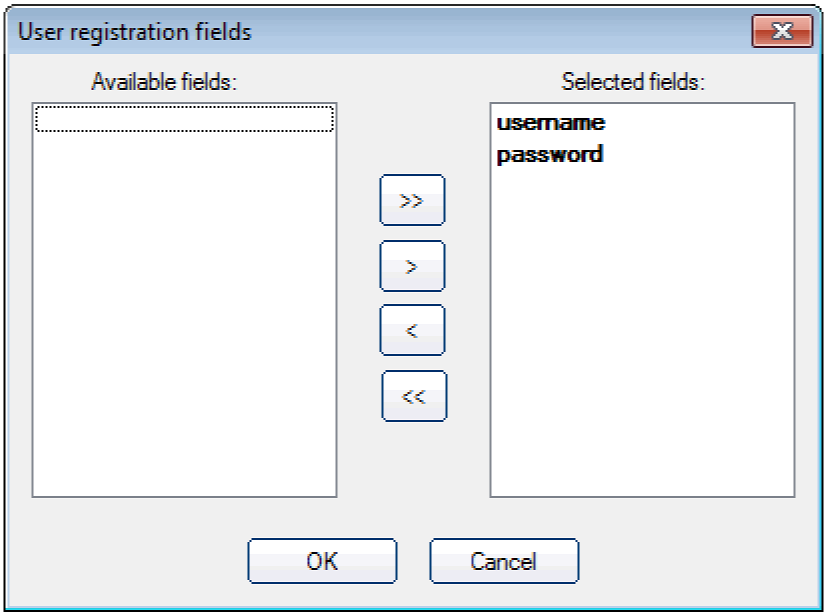
○ BCRYPT algorithm(recommended)

○ MD5 algorithm

□ Case-insensitive login

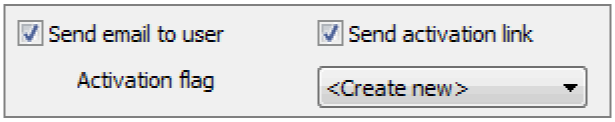
User email field: email

To create new user registration page, select the corresponding checkbox and click **Choose fields** to select fields that will appear on the registration page.



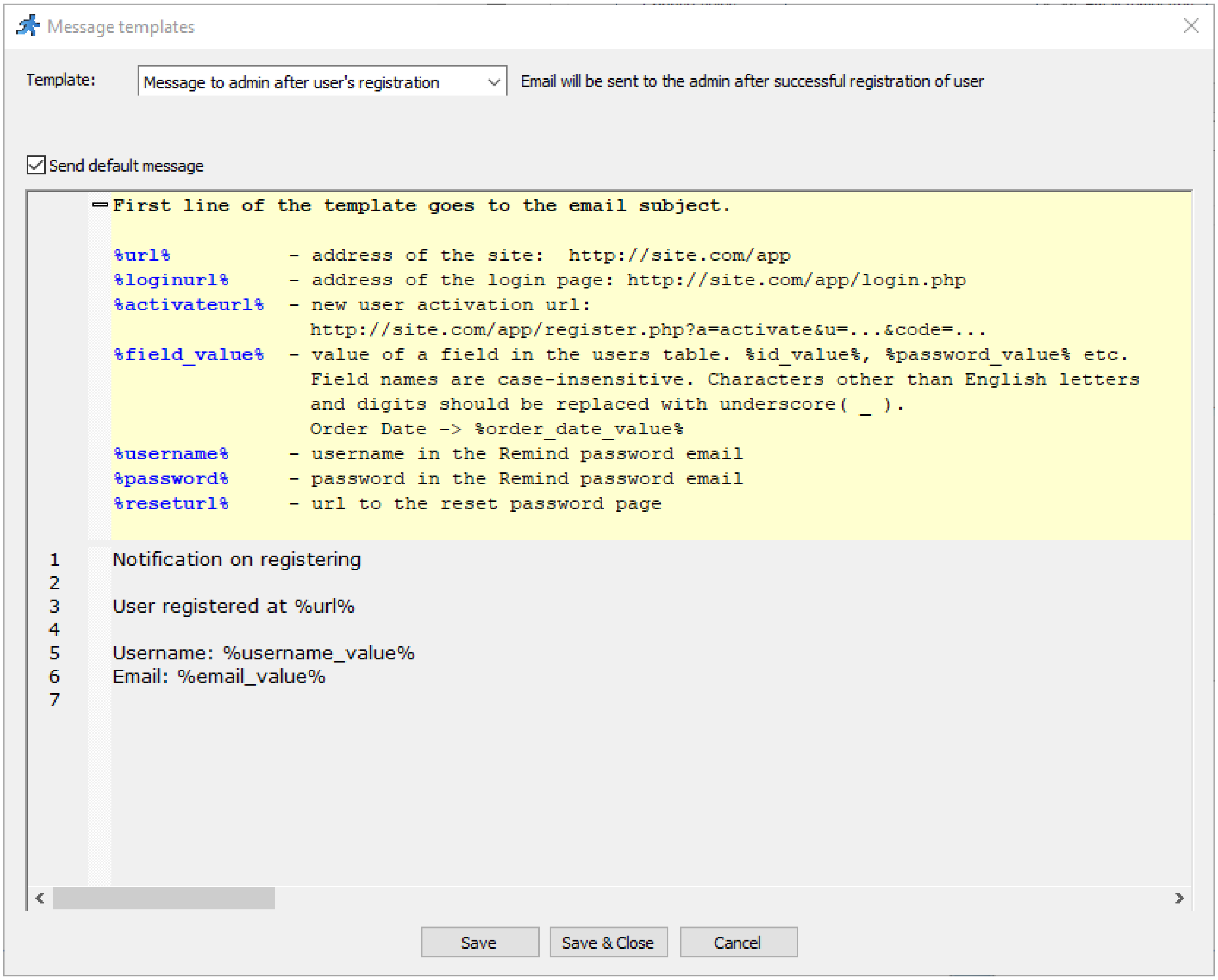
username
password

Select **Send activation link** checkbox to send email with activation link to a user after the registration. The purpose of the activation link is to ensure that the user signs up with a real, active email address that they own. This limits the number of times someone can sign up, and prevents people who have been booted off a site from signing up again multiple times with fake email addresses. User access is denied until activation link is open in the browser. Select one of the existing fields or create new one to store activation flag. This field needs to be numeric (Number, INT or TINYINT).



☑ Send email to user
☑ Send activation link
Activation flag <Create new>

If you need to customize email templates that will be send to the users/admins when a new user is registered use the Email templates dialog.



Template: Message to admin after user's registration

Email will be sent to the admin after successful registration of user

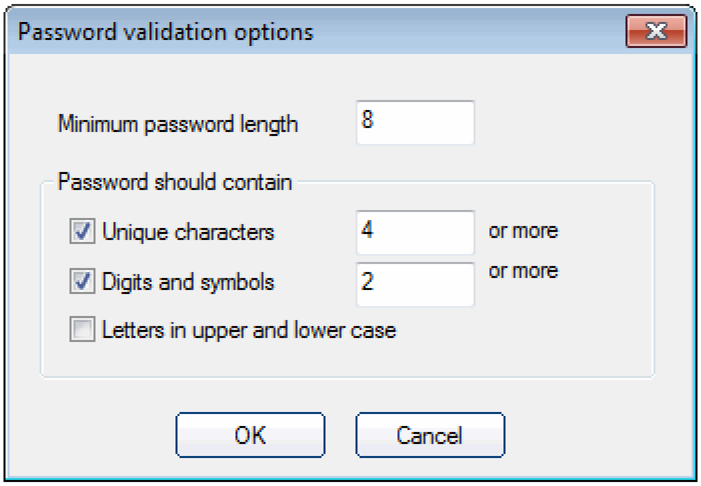
☑ Send default message

```
First line of the template goes to the email subject.

%url%          - address of the site:  http://site.com/app
%loginurl%     - address of the login page: http://site.com/app/login.php
%activateurl%  - new user activation url:
                 http://site.com/app/register.php?a=activate&u=...&code=...
%field_value%  - value of a field in the users table. %id_value%, %password_value% etc.
                 Field names are case-insensitive. Characters other than English letters
                 and digits should be replaced with underscore( _ ).
                 Order Date -> %order_date_value%
%username%     - username in the Remind password email
%password%     - password in the Remind password email
%reseturl%     - url to the reset password page
```

```
Notification on registering

User registered at %url%

Username: %username_value%
Email: %email_value%
```

If you are going to use **Send email to user** or **Remind password page** options, do not forget to choose a field where user email is stored.

Restrict weak password option allows to restrict using of weak passwords. You can select minimum password length, number of unique characters, digits or symbols.



Password validation options

Minimum password length 8

Password should contain

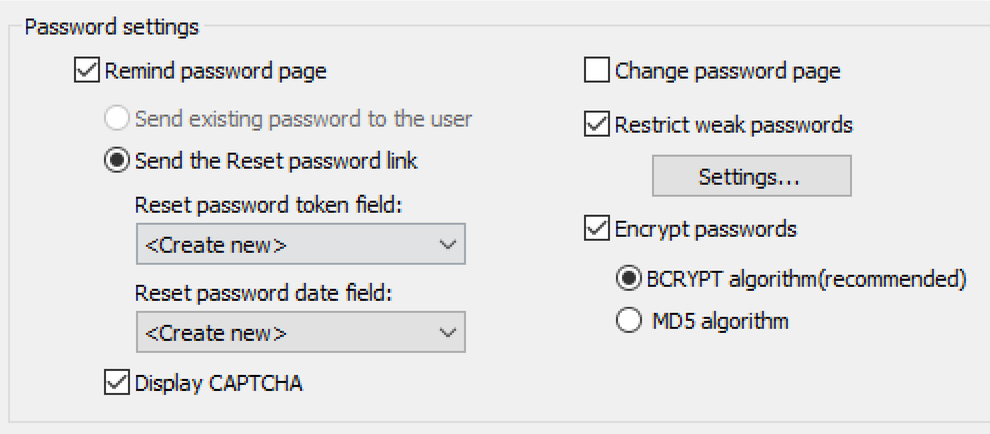
☑ Unique characters 4 or more

☑ Digits and symbols 2 or more

□ Letters in upper and lower case

Password hashing.

Option to use industry standard BCRYPT hashing algorithm. Send password reset link to user's email, link expires in 24 hours. BCRYPT requires PHP 5.5 or better.



Password settings

☑ Remind password page

○ Send existing password to the user

● Send the Reset password link

Reset password token field: <Create new>

Reset password date field: <Create new>

☑ Display CAPTCHA

□ Change password page

☑ Restrict weak passwords

☑ Encrypt passwords

● BCRYPT algorithm(recommended)

○ MD5 algorithm

You can also hash your passwords manually using events

For instance, you want to provide admin with direct access to the login table. To do so add the following code to BeforeAdd/BeforeEdit events of login table:

For BCRYPT:

```
$values["password"] = getPasswordHash( $values["password"] );
```

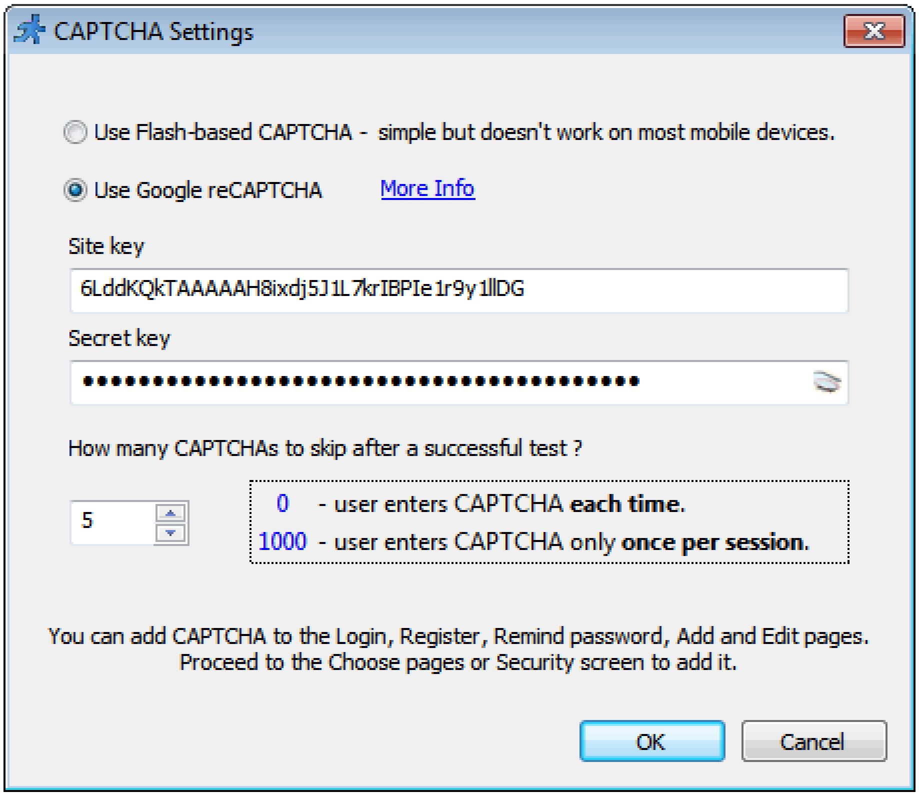
For MD5:

```
$values["password"] = md5( $values["password"] );
```

Adding CAPTCHA

You can add CAPTCHA to the remind password page and new user registration page (select the corresponding **Display CAPTCHA** checkbox). CAPTCHA is a simple test to determine if a user is a computer or a human. It is used to prevent spam abuse on the websites. Click **CAPTCHA settings** and choose what CAPTCHA to use: Flash-based CAPTCHA (it is simple but doesn't work on most mobile devices) or Google reCAPTCHA.

To use Google reCAPTCHA, register your web site at https://www.google.com/recaptcha/intro/index.html, copy site key and secret key there and paste them into the CAPTCHA settings.



CAPTCHA Settings

○ Use Flash-based CAPTCHA - simple but doesn't work on most mobile devices.

● Use Google reCAPTCHA More Info

Site key

6LddKQkTAAAAAH8ixdj5J1L7krIBPIe1r9y1llDG

Secret key

••

How many CAPTCHAs to skip after a successful test ?

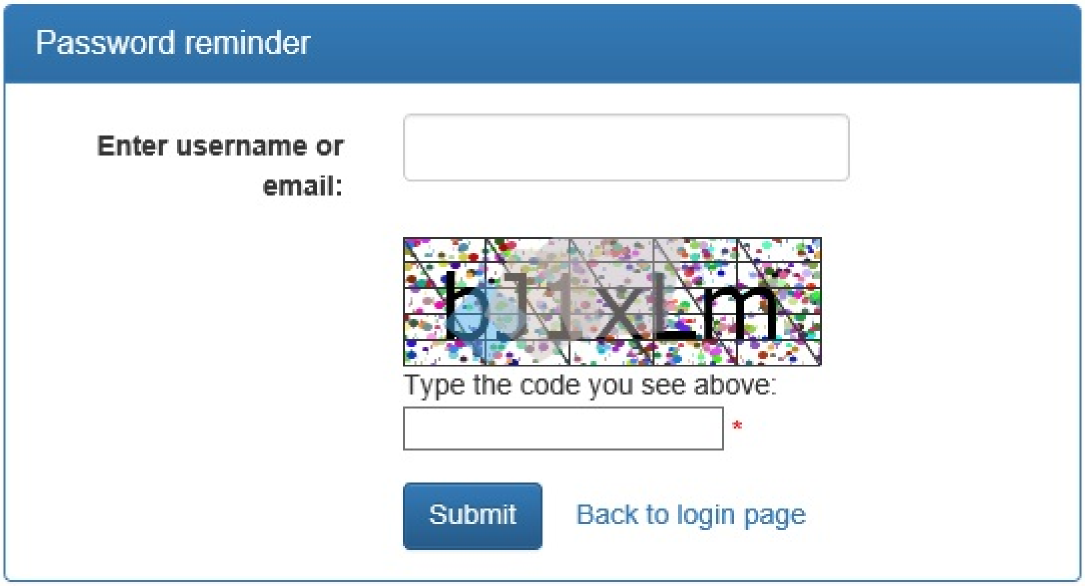
5

0 - user enters CAPTCHA **each time.**

1000 - user enters CAPTCHA only **once per session.**

You can add CAPTCHA to the Login, Register, Remind password, Add and Edit pages.
Proceed to the Choose pages or Security screen to add it.

Remind password page with Flash-based CAPTCHA:

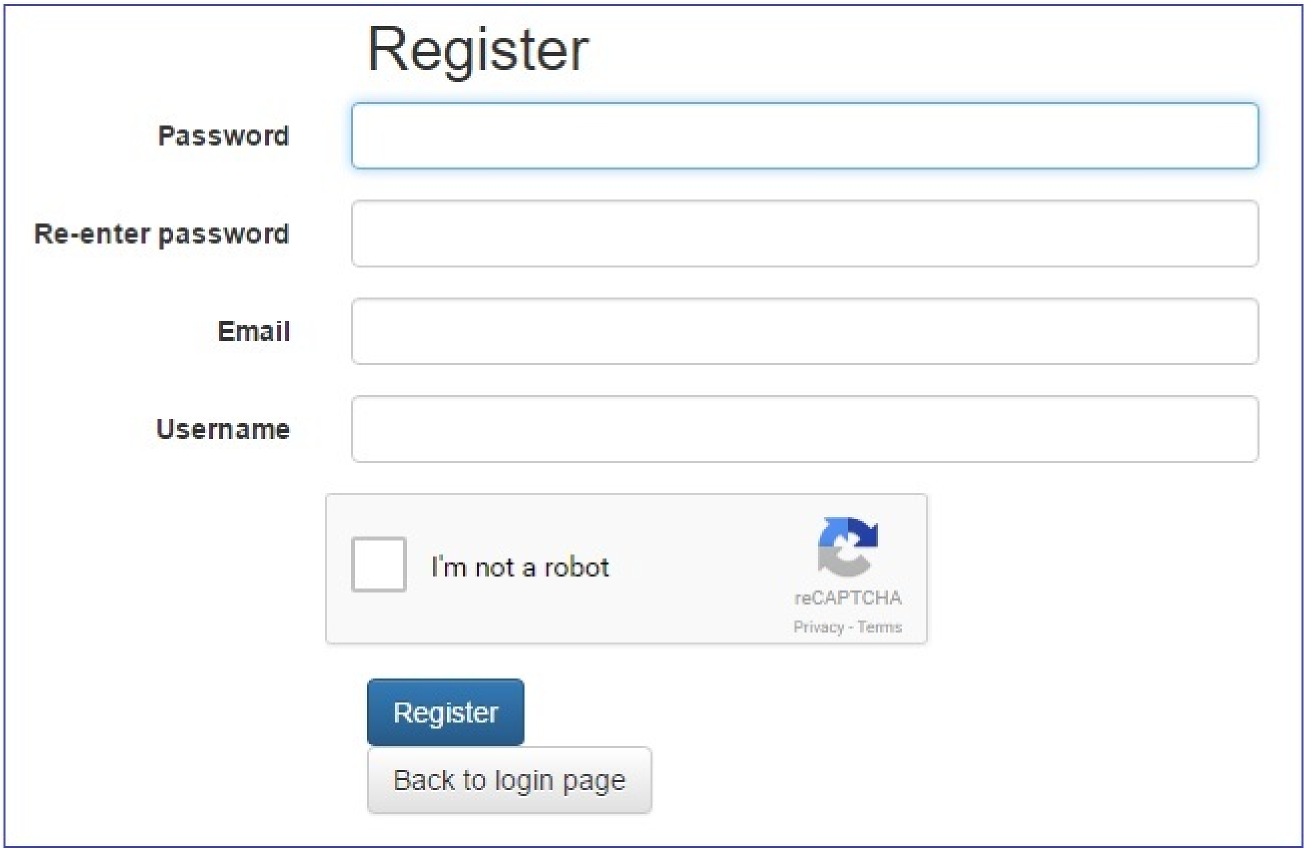


Password reminder

Enter username or email:

Type the code you see above:

New user registration page with Google reCAPTCHA:



Register

Password

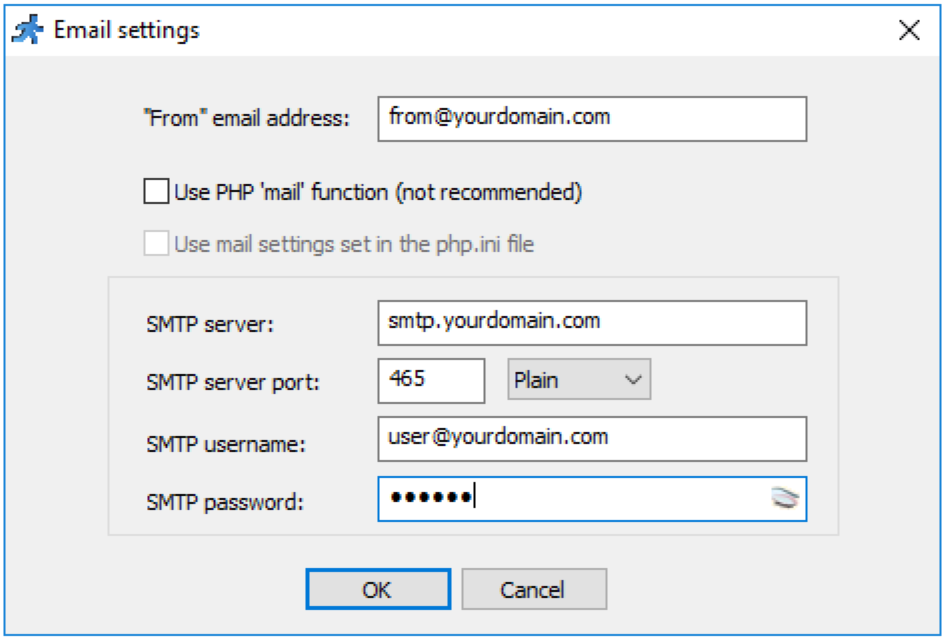
Re-enter password

Email

Username

Email settings

Here you can enter email which will be used to send emails to users and define settings of the custom mail server, if you do not use the built-in mail server. Note that SMTP server and SMTP server port settings may differ from the stated in the example below. Contact your Mail Service Provider to get necessary information.


Email settings

"From" email address: from@yourdomain.com

□ Use PHP 'mail' function (not recommended)

□ Use mail settings set in the php.ini file

SMTP server: smtp.yourdomain.com

SMTP server port: 465 Plain

SMTP username: user@yourdomain.com

SMTP password: ••••••

To send emails via Gmail, use the following settings:

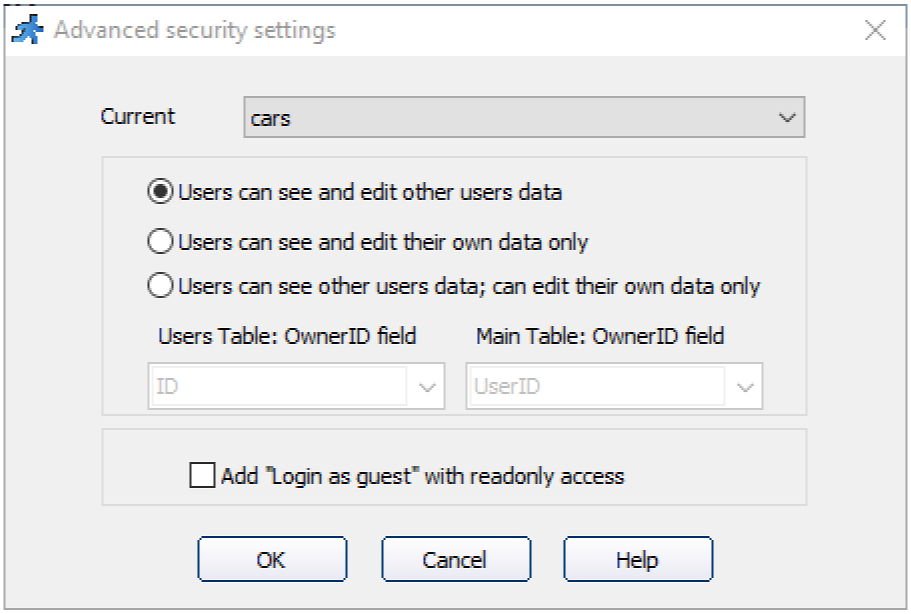
- **SMTP server**: smtp.gmail.com
- **SMTP server port**: 465, choose 'SSL' from dropdown list
- **SMTP username**: <your gmail address>
- **SMTP password**: <Gmail password>

If you use two-factor authentication for Gmail you need to generate and use an application specific password.
You can find details on the Gmail Accounts Help page.

2.15.4 Advanced security settings

1. Users can see and edit other users data

Every user who logged in successfully can view, edit, and delete other users data.



Current: cars

● Users can see and edit other users data

○ Users can see and edit their own data only

○ Users can see other users data; can edit their own data only

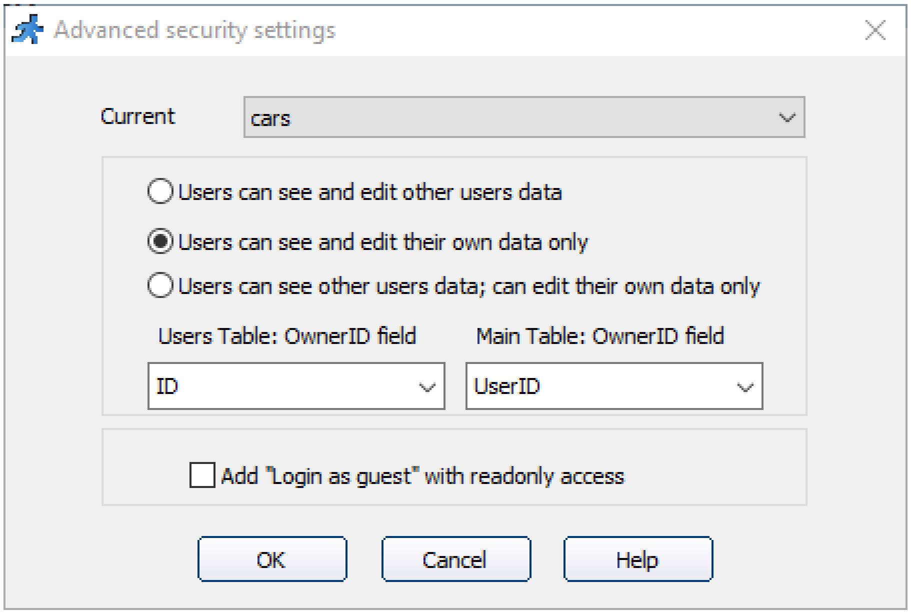
Users Table: OwnerID field: ID

Main Table: OwnerID field: UserID

□ Add "Login as guest" with readonly access

2. Users can see and edit their own data only

This option allows to restrict users to view and edit their own records only. To use this option for database-based login, you need to set a relationship between table that holds usernames and passwords and main data table. In case of Active Directory authentication select field to save username in.



Current: cars

○ Users can see and edit other users data

● Users can see and edit their own data only

○ Users can see other users data; can edit their own data only

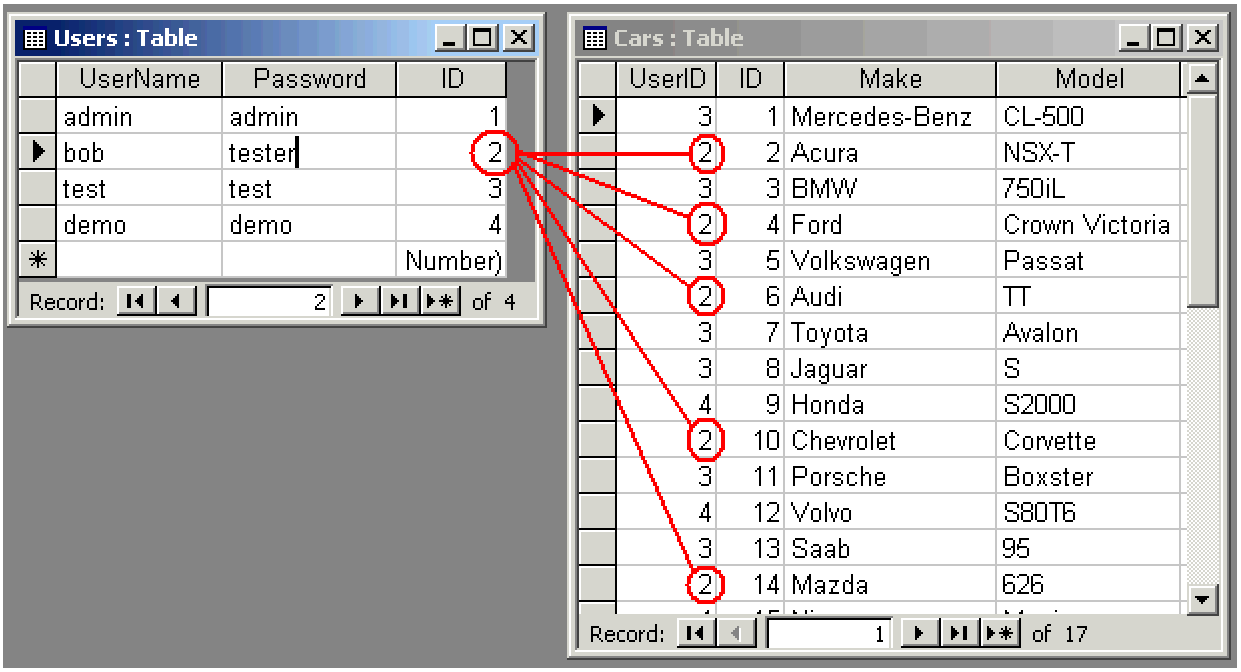
Users Table: OwnerID field: ID

Main Table: OwnerID field: UserID

□ Add "Login as guest" with readonly access

3. Users can see other users data, can edit their own data only

This option is similar to Option 2, however, all users can view other users data but cannot edit or delete them.

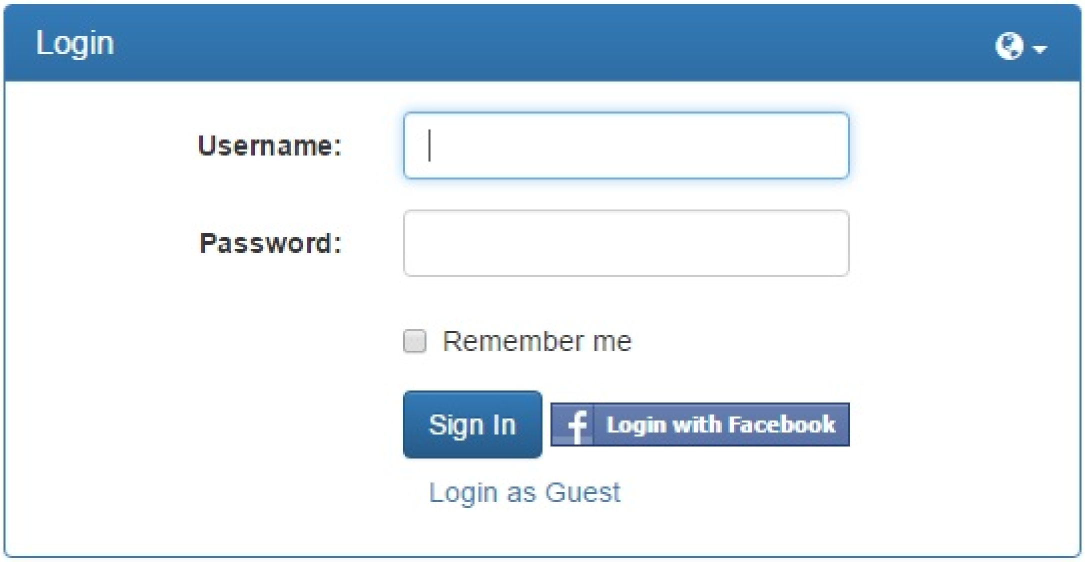


	UserName	Password	ID
	admin	admin	1
▶	bob	tester	2
	test	test	3
	demo	demo	4
*			Number)

	UserID	ID	Make	Model
▶	3	1	Mercedes-Benz	CL-500
	2	2	Acura	NSX-T
	3	3	BMW	750iL
	2	4	Ford	Crown Victoria
	3	5	Volkswagen	Passat
	2	6	Audi	TT
	3	7	Toyota	Avalon
	3	8	Jaguar	S
	4	9	Honda	S2000
	2	10	Chevrolet	Corvette
	3	11	Porsche	Boxster
	4	12	Volvo	S80T6
	3	13	Saab	95
	2	14	Mazda	626

Login as guest

This option adds guest read-only access to the database. If this option is selected, "Login as guest" link will be added to the login screen. Guest user cannot edit, delete or add new records to the database.



If you need to customize email templates that will be send to the users/admins when a new user is registered use the Email templates dialog.

2.15.5 User group permissions

This feature allows to assign table level permissions on database tables and views. I.e. user A can add data into table1 and edit data in table2, user B can edit and search data in table1 and can delete data in table2 etc.

To use this feature proceed to the **Security** page in PHPRunner, click the **Permissions** button.

There are two ways to define user group permissions:

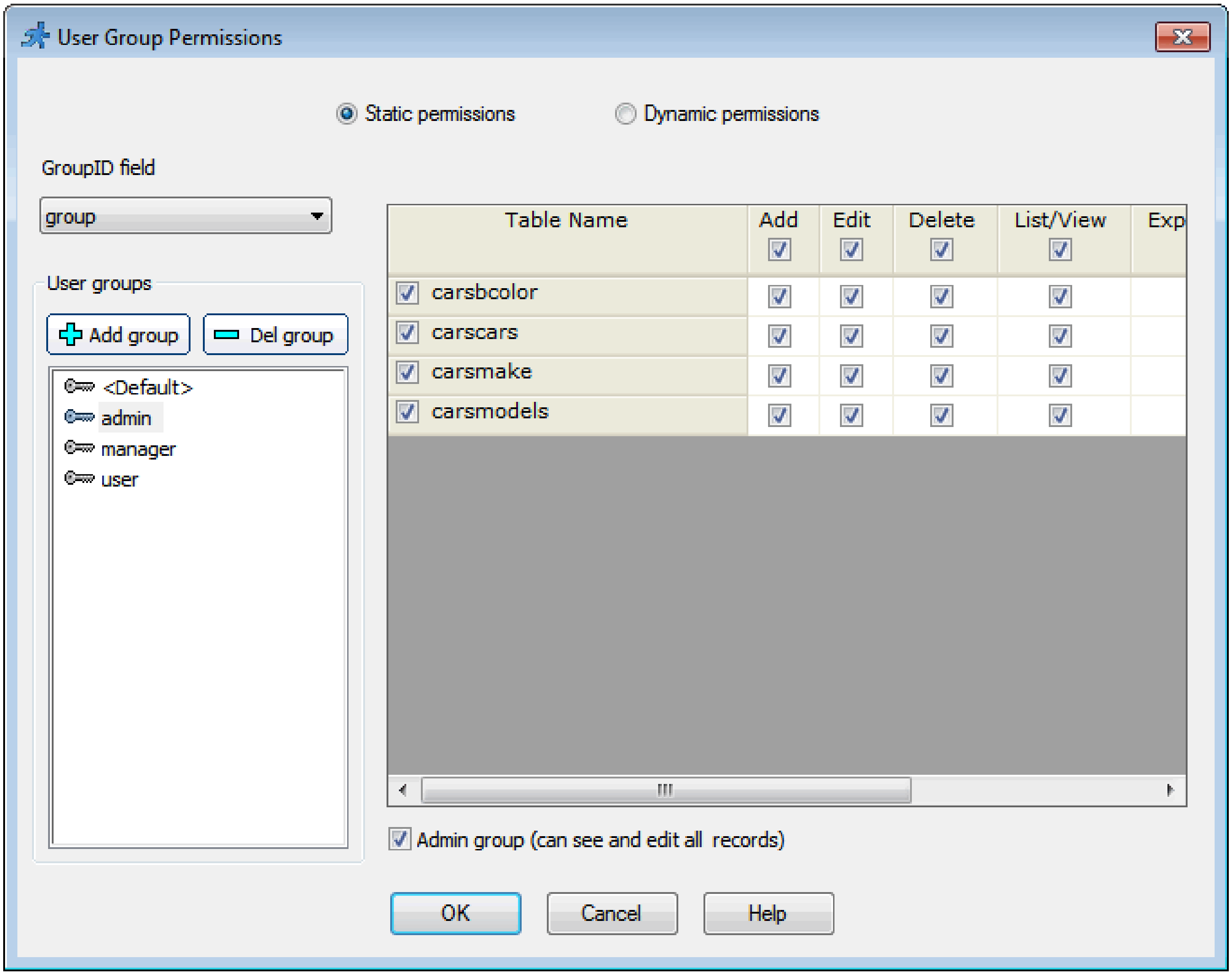
- in the PHPRunner directly before building project (**Static permissions** option). In this case if you want to change some security settings you'll need to open PHPRunner project, make some changes on the **User Group Permissions** dialog and rebuild project.

- on the generated pages (Dynamic permissions option). PHPRunner create new tables for user group permissions settings in the database and build admin area in the application. Administrators can update permissions online.

Static permissions

On this screen, you can create user groups with assigned set of permissions.

First of all choose **GroupID** field that stores group name or id.



● Static permissions ○ Dynamic permissions

GroupID field: group

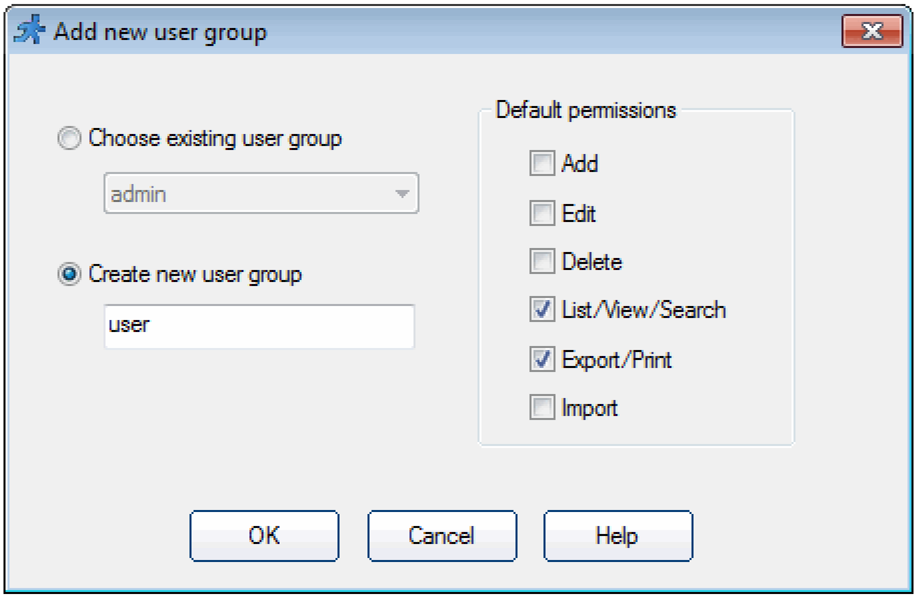
User groups:

- <Default>
- admin
- manager
- user

Table Name	Add	Edit	Delete	List/View	Exp
	☑	☑	☑	☑	
☑ carsbcolor	☑	☑	☑	☑	
☑ carscars	☑	☑	☑	☑	
☑ carsmake	☑	☑	☑	☑	
☑ carsmodels	☑	☑	☑	☑	

☑ Admin group (can see and edit all records)

To add a new group, click the **Add group** button. A dialog box will appear where you can choose username which defines this group and define initial set of permissions. Afterwards, you can change permissions for certain tables.



○ Choose existing user group

admin

● Create new user group

user

Default permissions

- □ Add
- □ Edit
- □ Delete
- ☑ List/View/Search
- ☑ Export/Print
- □ Import

You can create a user group based on non-existing **GroupID** field value (if you plan to add users to this group later).

Apart from this, you can create a default group with a default set of permissions. When a user that does not belong to any of specified groups is logged in, the default set of permissions mentioned above is assigned to him. For example, as shown on the screenshot above, the user not belonging to any of the groups, is able to edit and delete data, but is not able to add or search records in any table.

To delete selected group, click the **Del group** button. To overwrite existing group, click the **Add group** button and choose the same group id value.

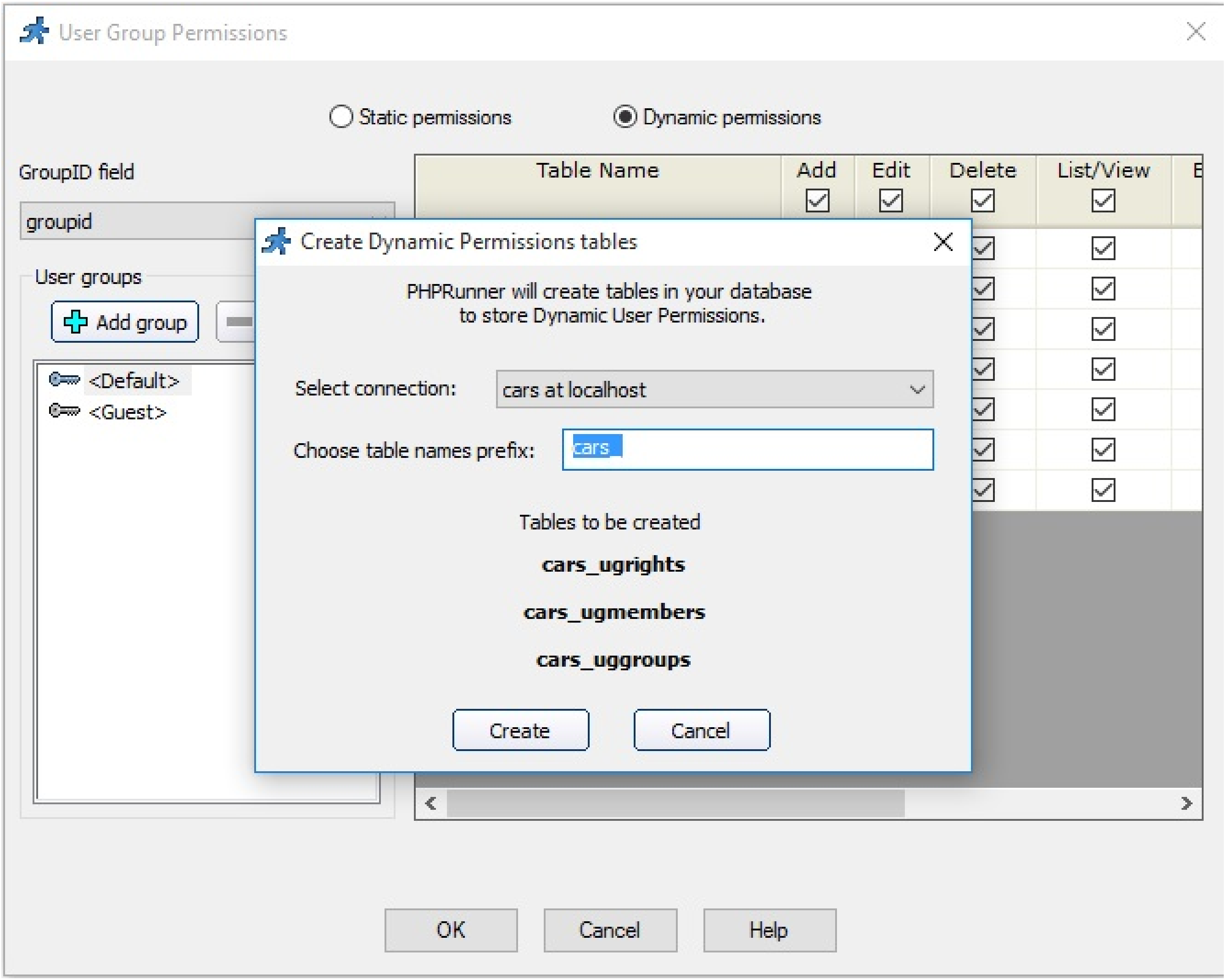
When Advanced Security options are in use, you can set any group to be an admin group (enable **Admin group** check box).

2.15.6 Dynamic Permissions

The main difference between static and dynamic permissions is that PHPRunner create new tables for user group permissions settings in the database and build admin area in the application. Admins can update permissions online.

Dynamic permissions require three database tables to store permissions, groups, and group members. PHPRunner allows you to create these tables or select existing ones. By adding a prefix in front of table names PHPRunner allows several projects to share the same database.

On this screen you should select existing tables or chose prefix for the new tables where user group permissions will be stored and add admin user on the **Set Initial Permissions** dialog.



PHPRunner will create tables in your database to store Dynamic User Permissions.

Select connection: cars at localhost

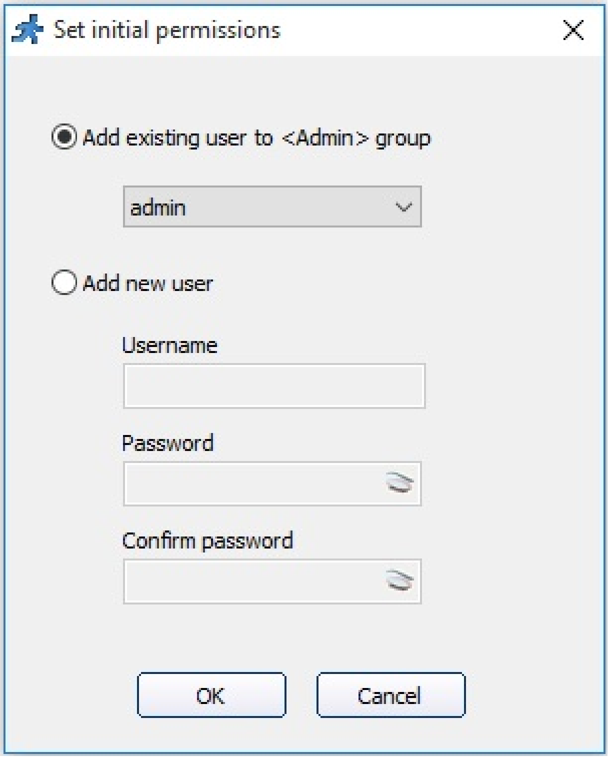
Choose table names prefix: cars_

Tables to be created

cars_ugrights

cars_ugmembers

cars_uggroups


Set initial permissions

● Add existing user to <Admin> group

admin

○ Add new user

Username

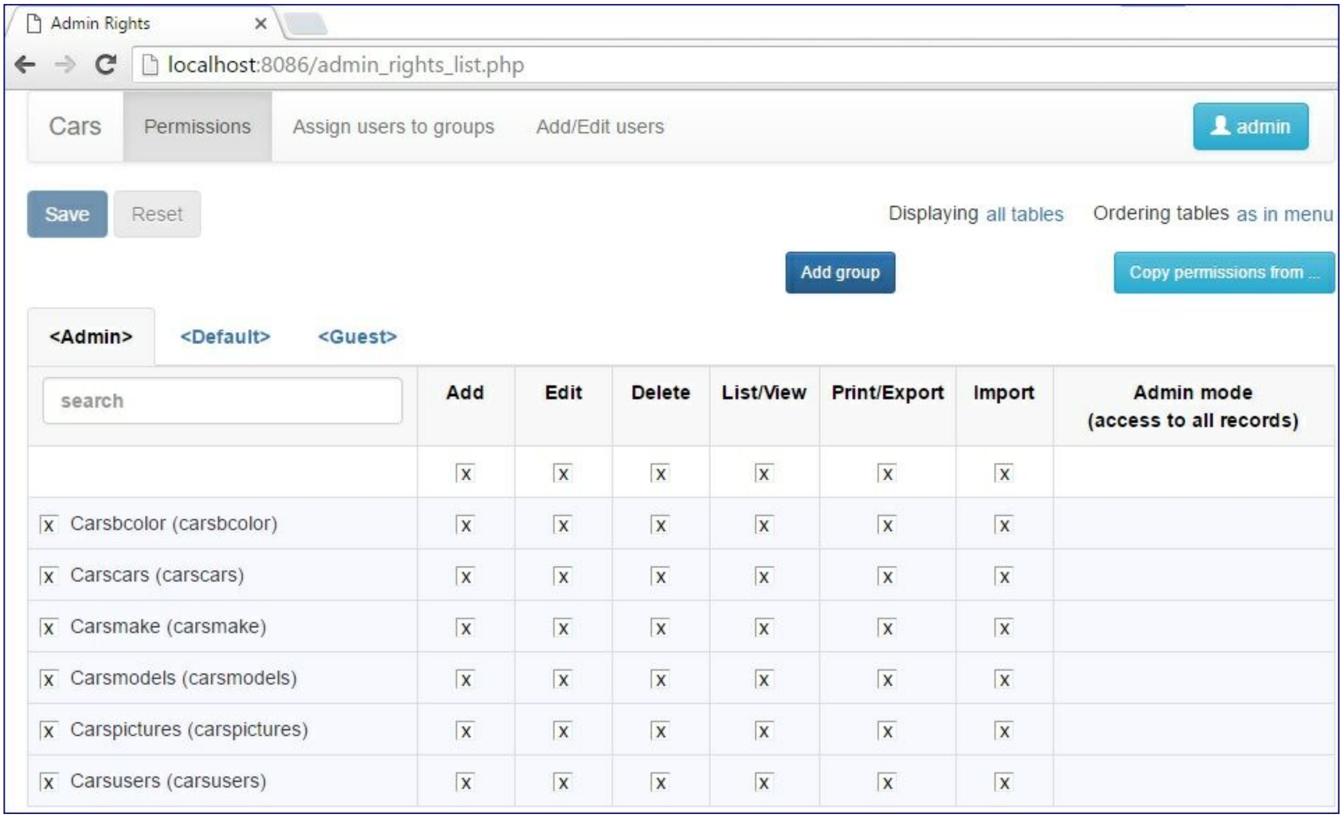
Password

Confirm password

After building the project you need to login as admin, add groups, set up permissions and assign users to created groups.

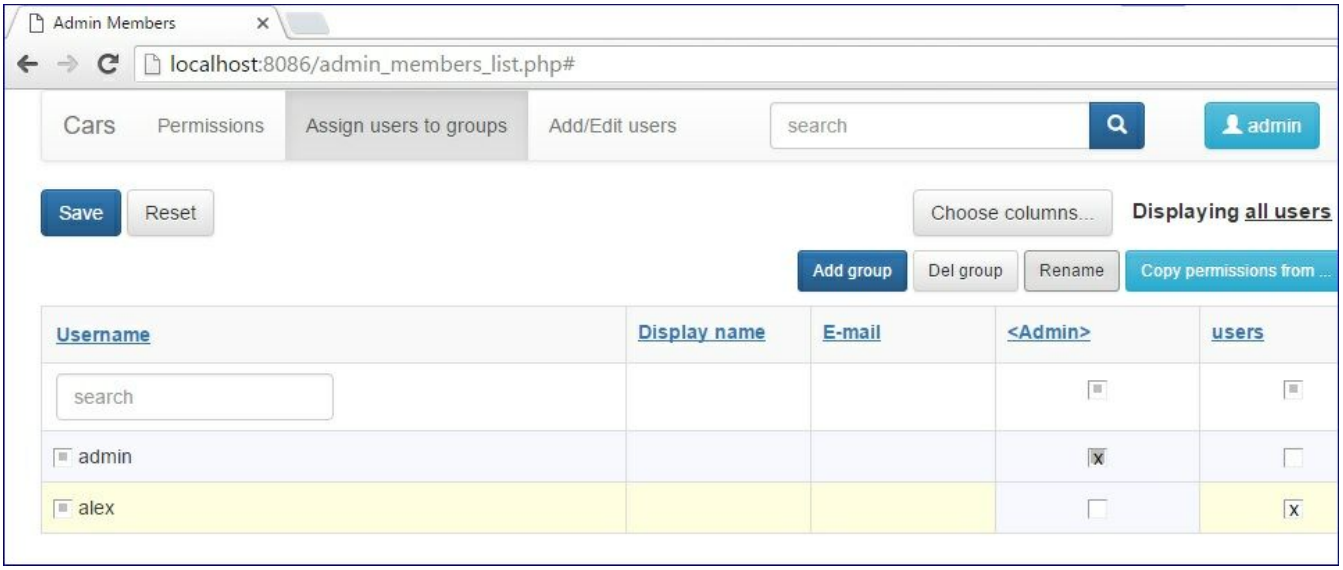
Admin area on the generated pages

Add new groups and assign permissions to the groups. Use live search for a table name to filter the tables. You can display all or modified tables only. Order tables alphabetically or the way they appear in the menu (with groups and subgroups). Click **Copy permissions from** to copy permissions from one user group to the currently selected.



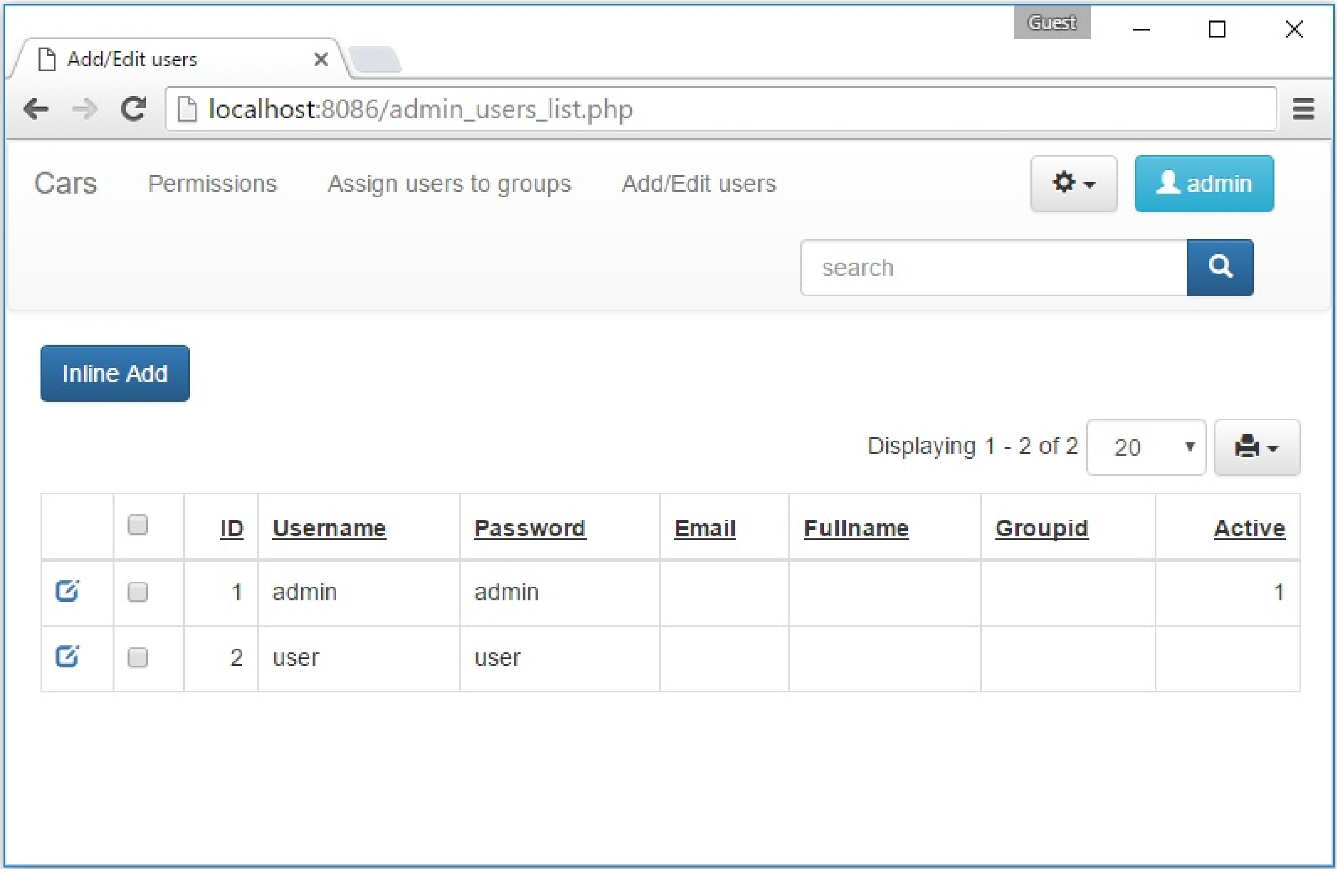
search	Add	Edit	Delete	List/View	Print/Export	Import	Admin mode (access to all records)
	☒	☒	☒	☒	☒	☒	
☒ Carsbcolor (carsbcolor)	☒	☒	☒	☒	☒	☒	
☒ Carscars (carscars)	☒	☒	☒	☒	☒	☒	
☒ Carsmake (carsmake)	☒	☒	☒	☒	☒	☒	
☒ Carsmodels (carsmodels)	☒	☒	☒	☒	☒	☒	
☒ Carspictures (carspictures)	☒	☒	☒	☒	☒	☒	
☒ Carsusers (carsusers)	☒	☒	☒	☒	☒	☒	

Assign existing users to the groups:



Username	Display name	E-mail	<Admin>	users
search			▣	▣
▣ admin			☒	□
▣ alex			□	☒

Add, edit or delete users:



	□	ID	Username	Password	Email	Fullname	Groupid	Active
	□	1	admin	admin				1
	□	2	user	user				

2.15.7 Audit and record locking

The **Audit trail** page allows you to define the audit settings and configure record locking.

⊞ Audit settings

You can record the user actions like login/logout/record editing/adding/deleting etc to a database or file. On default the action logging is disabled.

To enable logging:

1. Go to the **Security** page and click **Audit trail**.

2. Select where to log user actions:

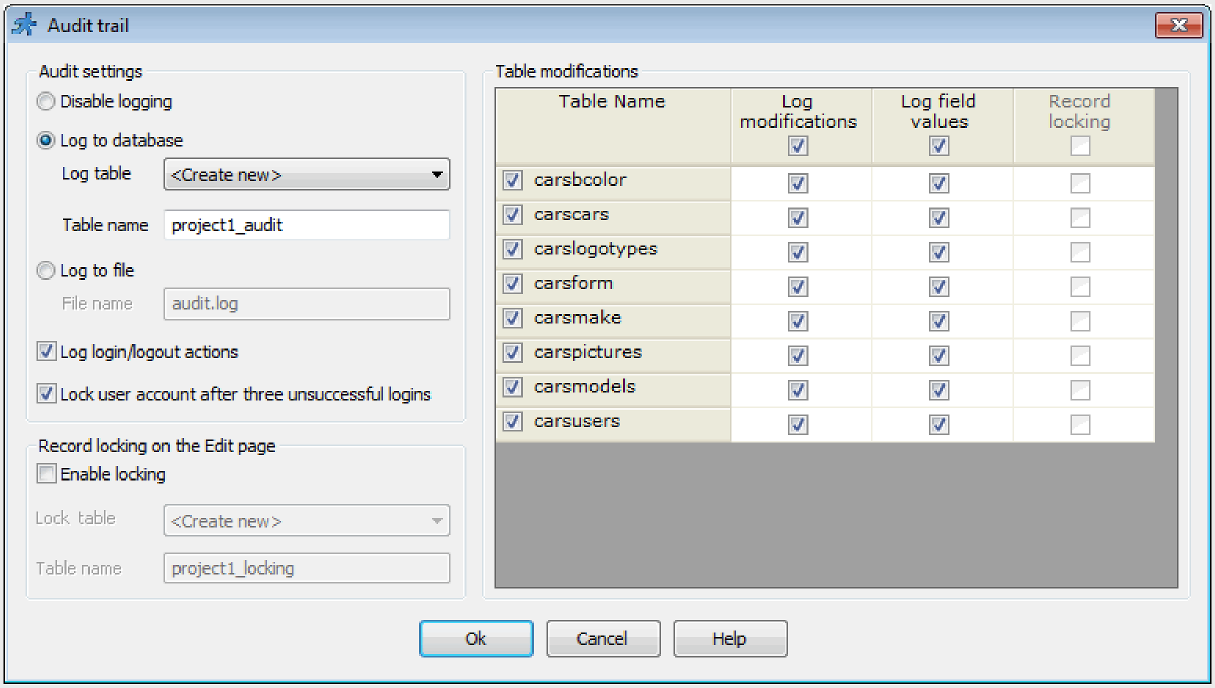
   - **Log to database**. Select a log table. To create new log table, select *<Create new>* in **Log table** list box and enter the table name to the text box below, for example *project1_audit*.

- **Log to file**. Enter the log file name, for example *audit.log*. The current date is added automatically to the file name and log file is stored as *audit_yyyymmdd.log*, for example *audit_20091222.log*. When specified as *action.log*, the log files will be stored into your project root folder. To change this enter the file name as *<folder>/<file name>.* For example, *log/audit.log*.

3. Select the **Log login/logout actions** and **Lock user account after three unsuccessful logins** check boxes to enable the corresponding actions. Note that **Lock user account after three unsuccessful logins** option is available only if you selected **Log to database**.

4. Select tables to which the user actions should be logged. You can log only modifications performed to the table records (e.g. action = add, key field = 10) or modifications and field values (e.g. action = add, key field = 10, field = color, new value = red). So select the corresponding check boxes in the **Table modifications** area.

5. Click **Ok**.



Audit settings

○ Disable logging

● Log to database

Log table: <Create new>

Table name: project1_audit

○ Log to file

File name: audit.log

☑ Log login/logout actions

☑ Lock user account after three unsuccessful logins

Record locking on the Edit page

☐ Enable locking

Lock table: <Create new>

Table name: project1_locking

Table modifications

Table Name	Log modifications ☑	Log field values ☑	Record locking ☐
☑ carsbcolor	☑	☑	☐
☑ carscars	☑	☑	☐
☑ carslogotypes	☑	☑	☐
☑ carsform	☑	☑	☐
☑ carsmake	☑	☑	☐
☑ carspictures	☑	☑	☐
☑ carsmodels	☑	☑	☐
☑ carsusers	☑	☑	☐

To analyze information from the log table, you can add report or chart to your project or use online report/chart builder.

Example of log file

```
Date Time IP User Table Action Key field Field Old value New value

Dec 23,2009 15:41:37 127.0.0.1 admin carsusers login
Dec 23,2009 15:41:48 127.0.0.1 admin carsbcolor edit 1 color Taffeta White1 Taffeta White

Dec 23,2009 15:42:26 127.0.0.1 admin carsmake delete 15 id 15

Dec 23,2009 15:42:26 127.0.0.1 admin carsmake delete 15 make Lexus
```

Record locking

You can use record locking to prevent simultaneous update of the same data on the **Edit** page. A locked record means it is not available for editing by other users. On default the record locking is disabled.

To enable record locking:

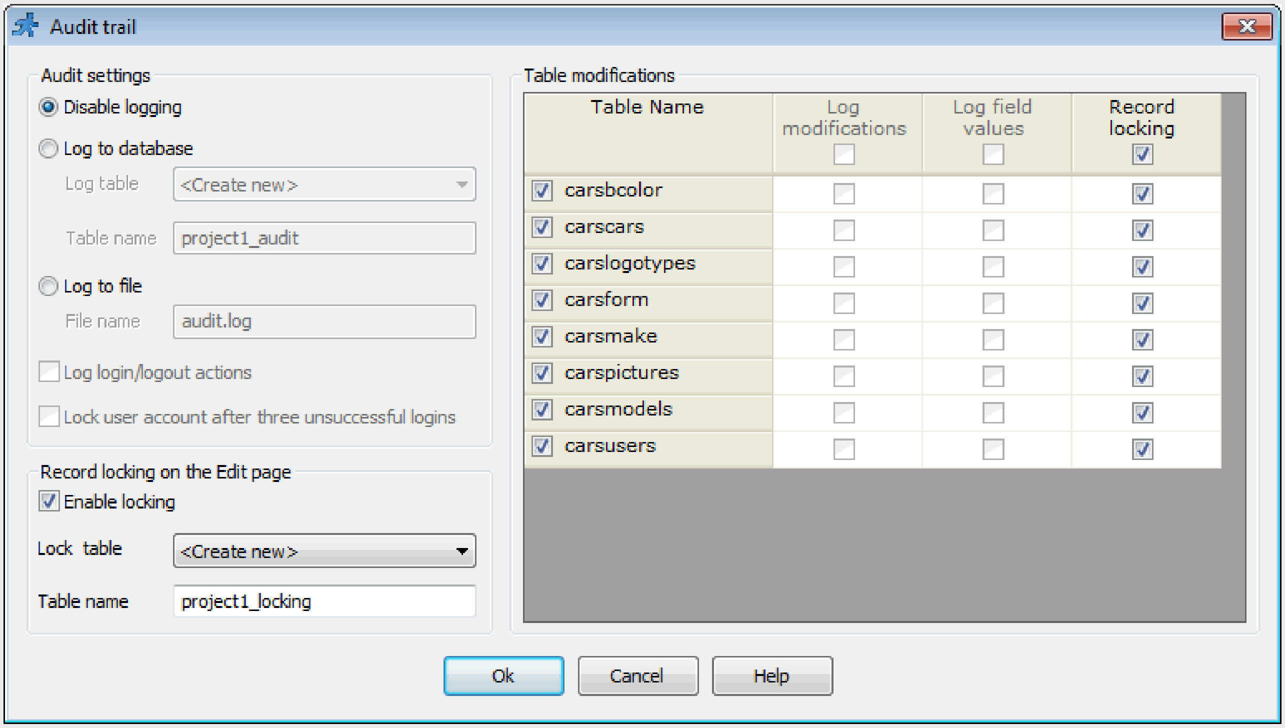
1. Go to the **Security** page and click **Audit trail**.

2. Select the **Enable locking** check box.

3. Select a lock table. The lock table is used to store record locks. To create new lock table, select *<Create new>* in **Lock table** list box and enter the table name to the text box below, for example *project1_locking*.

4. Select tables to which the record locking should be applied. You need to select the corresponding check boxes in the **Table modifications** area under **Record locking**.

5. Click **Ok**.



Audit settings

● Disable logging

○ Log to database

Log table <Create new>

Table name: project1_audit

○ Log to file

File name: audit.log

□ Log login/logout actions

□ Lock user account after three unsuccessful logins

Record locking on the Edit page

☑ Enable locking

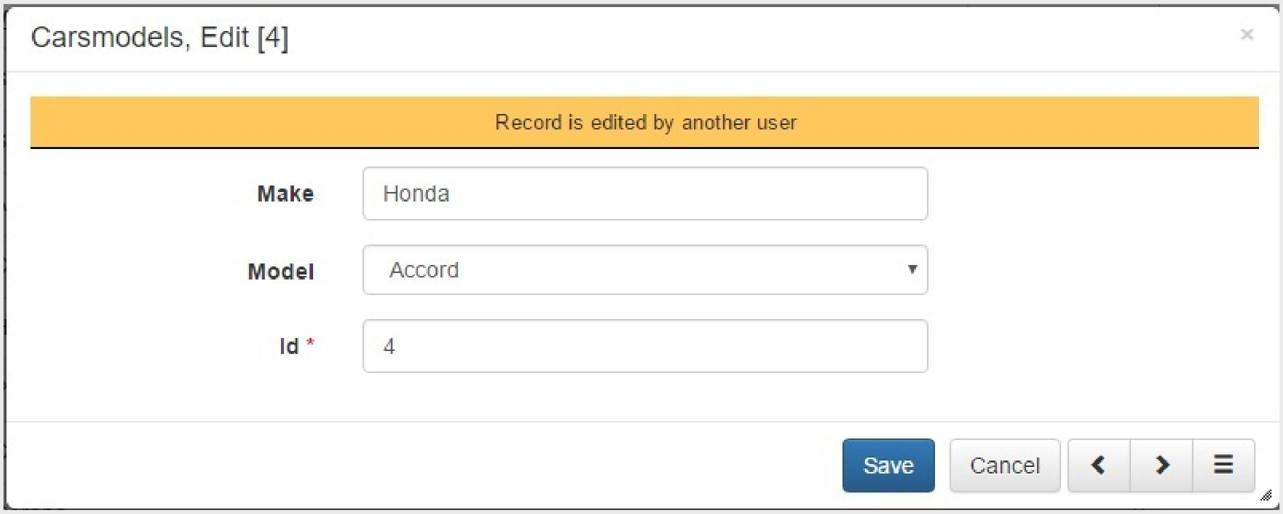
Lock table <Create new>

Table name: project1_locking

Table modifications

Table Name	Log modifications □	Log field values □	Record locking ☑
☑ carsbcolor	□	□	☑
☑ carscars	□	□	☑
☑ carslogotypes	□	□	☑
☑ carsform	□	□	☑
☑ carsmake	□	□	☑
☑ carspictures	□	□	☑
☑ carsmodels	□	□	☑
☑ carsusers	□	□	☑

User who tries to edit a locked record would be shown the message that the record is locked.


Carsmodels, Edit [4]

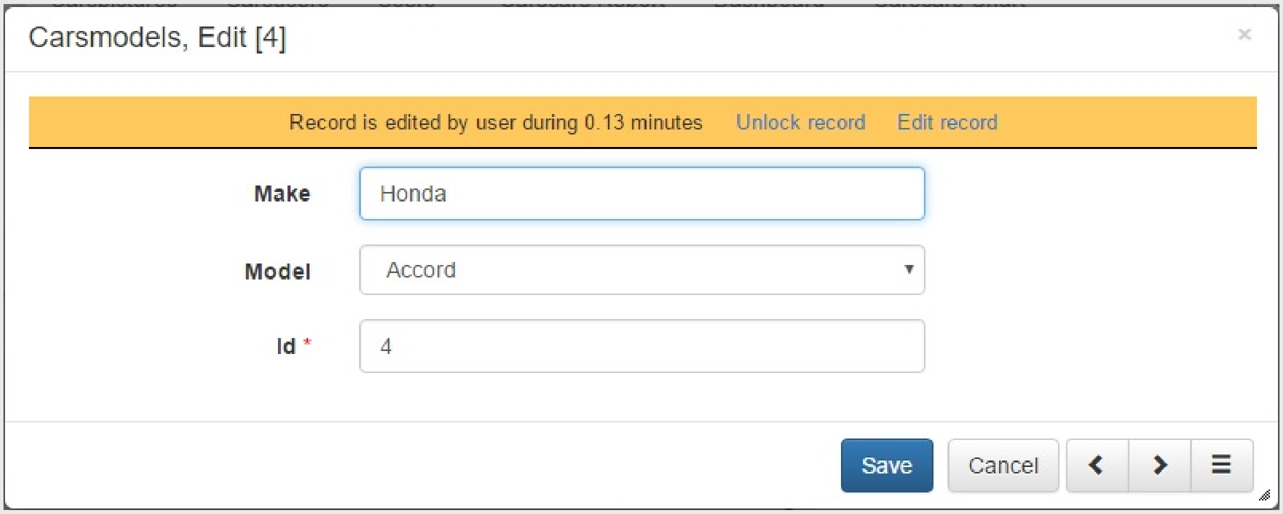
Record is edited by another user

Make: Honda

Model: Accord

Id *: 4

Note: admin user is able to unlock a record if it was blocked for too long or take ownership.


Carsmodels, Edit [4]

Record is edited by user during 0.13 minutes   Unlock record   Edit record

Make: Honda

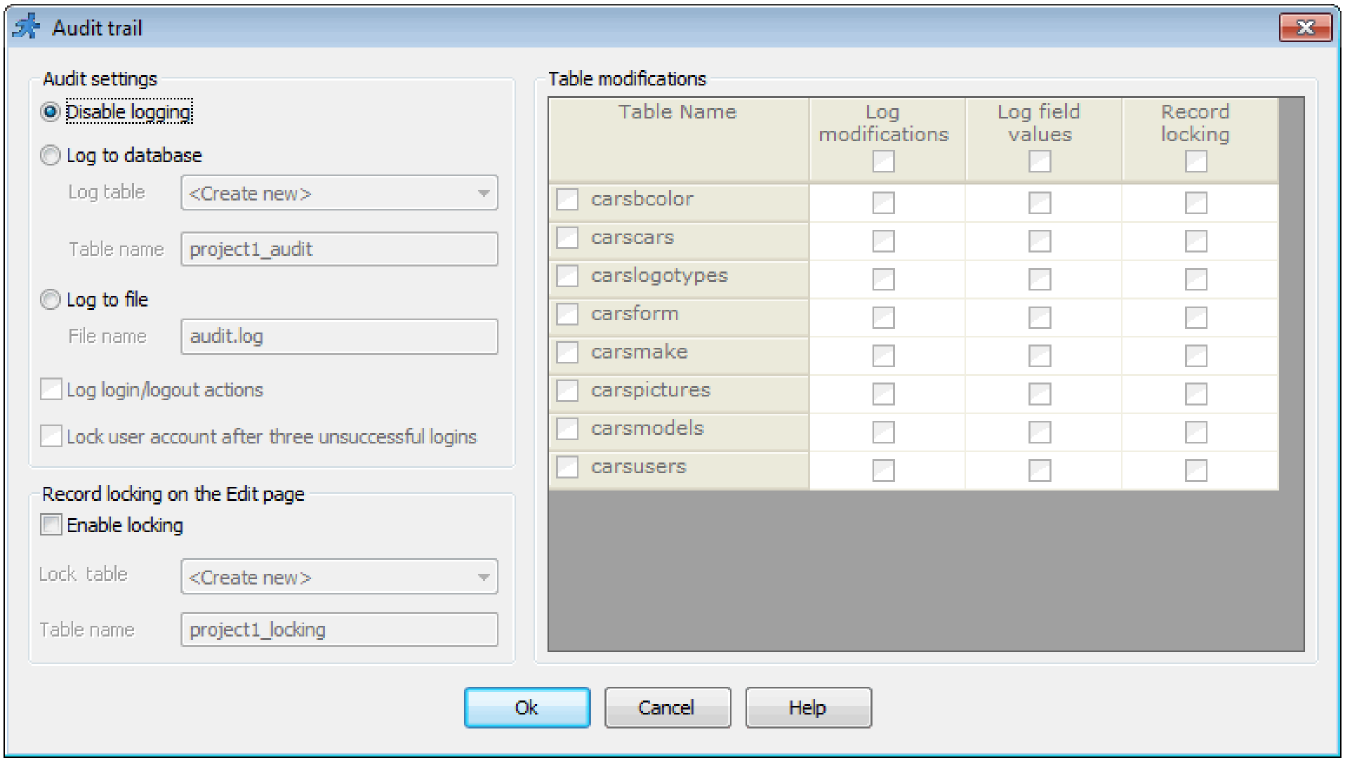
Model: Accord

Id *: 4

If user goes to the **Edit** page and then clicks **Back,** it turns out that the record remains blocked. To prevent this, automatic unlocking method is used. By default, the record will be unlocked after 300 seconds. You can change this value by editing the file *include/locking.php*:

```
var $ConfirmTime = 250;
var $UnlockTime = 300;
```

The variable *$ConfirmTime* determines how often the Edit page sends confirmation that the record is still locked. The variable *$UnlockTime* determines the time (in seconds) after the last confirmation when the record will be unlocked automatically.



Audit trail

Audit settings

● Disable logging

○ Log to database

Log table <Create new>

Table name project1_audit

○ Log to file

File name audit.log

□ Log login/logout actions

□ Lock user account after three unsuccessful logins

Record locking on the Edit page

□ Enable locking

Lock table <Create new>

Table name project1_locking

Table modifications

Table Name	Log modifications □	Log field values □	Record locking □
□ carsbcolor	□	□	□
□ carscars	□	□	□
□ carslogotypes	□	□	□
□ carsform	□	□	□
□ carsmake	□	□	□
□ carspictures	□	□	□
□ carsmodels	□	□	□
□ carsusers	□	□	□

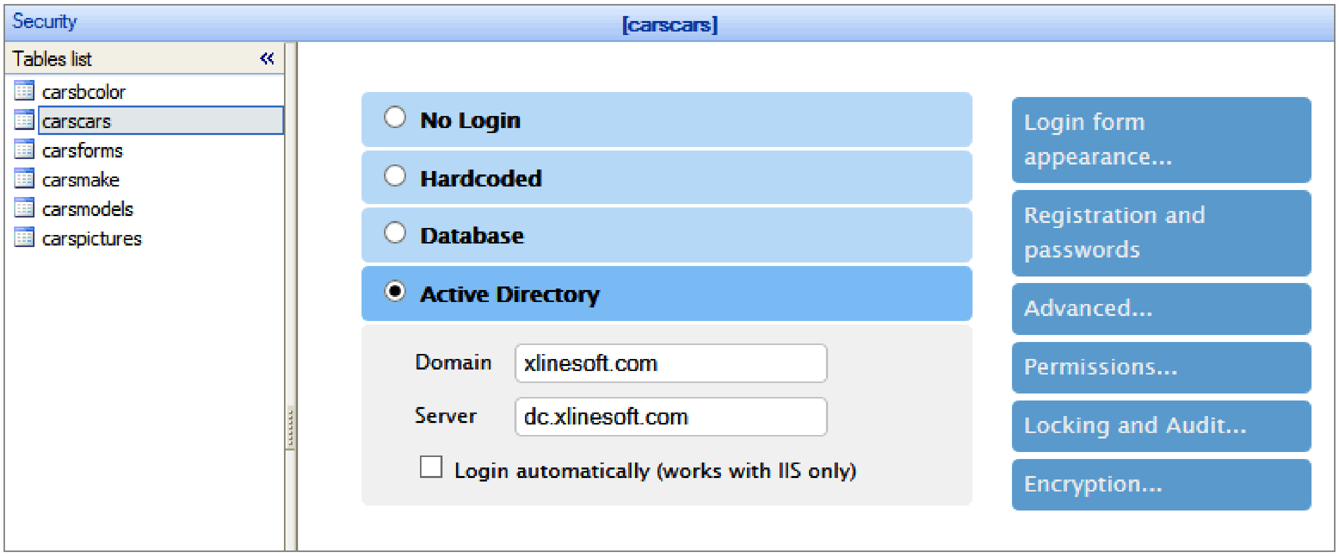
2.15.8 Active Directory

Quick jump

Active Directory authentication and Permissions

Active Directory authentication allows users to log in to the generated by PHPRunner applications if they have an account in an Active Directory domain. You need to define the Active Directory domain and server. When logging in the login name and password are checked against Active Directory. In the simplest use case no additional configuration is needed.

Note: Active Directory authentication feature is available only in the Enterprise Edition of PHPRunner. See Editions Comparison.



○ No Login

○ Hardcoded

○ Database

● Active Directory

Domain: xlinesoft.com

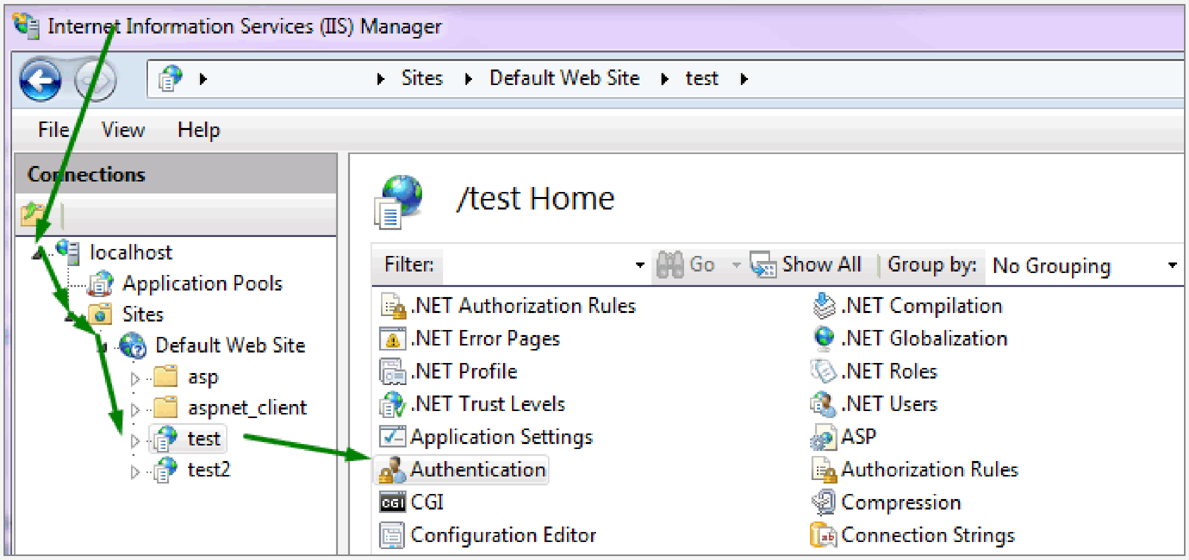
Server: dc.xlinesoft.com

□ Login automatically (works with IIS only)

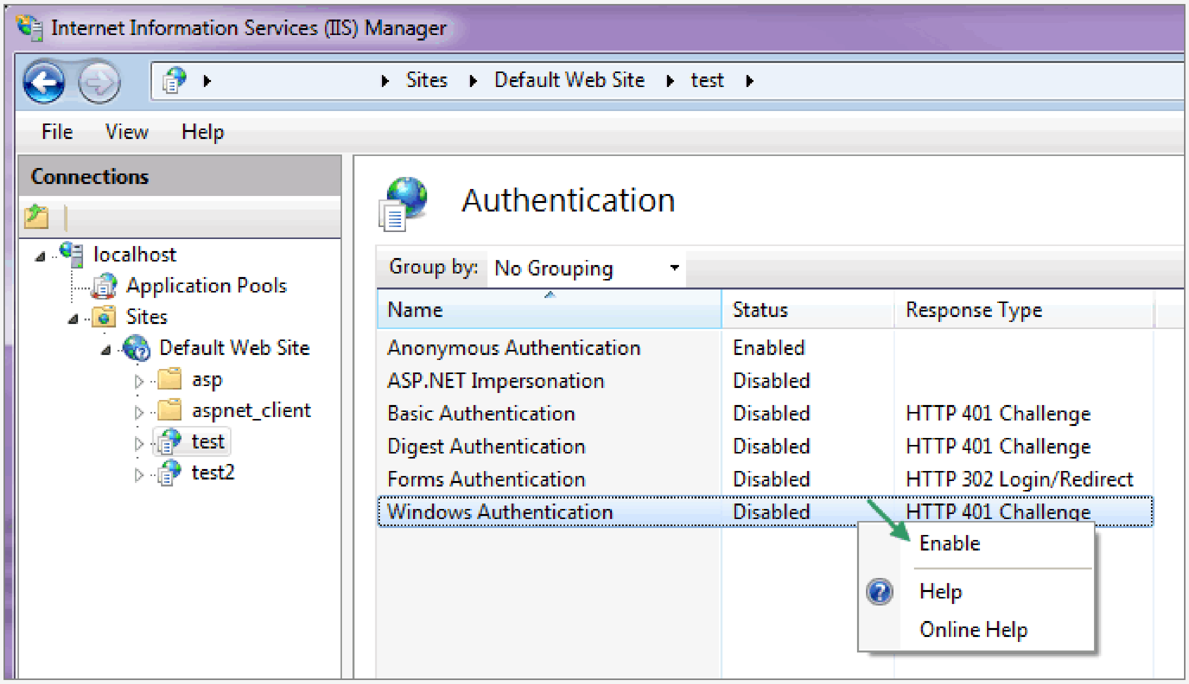
Login automatically check box enables Autologon functionality: if a person is already logged into Windows, he/she will be automatically logged into the generated application. To use Automatic logon, make sure Windows Authentication is enabled in Internet Information Services (IIS).

How to enable Windows Authentication

- Run IIS as administrator of the computer (Click **Start** -> **Control Panel** -> **Administrative Tools** -> **Internet Information Services (IIS) Manager**).
- Expand the server in **Connections** frame and choose the site, or click on server if you wish to apply settings for all sites.
- Double-click the **Authentication** icon in main frame.



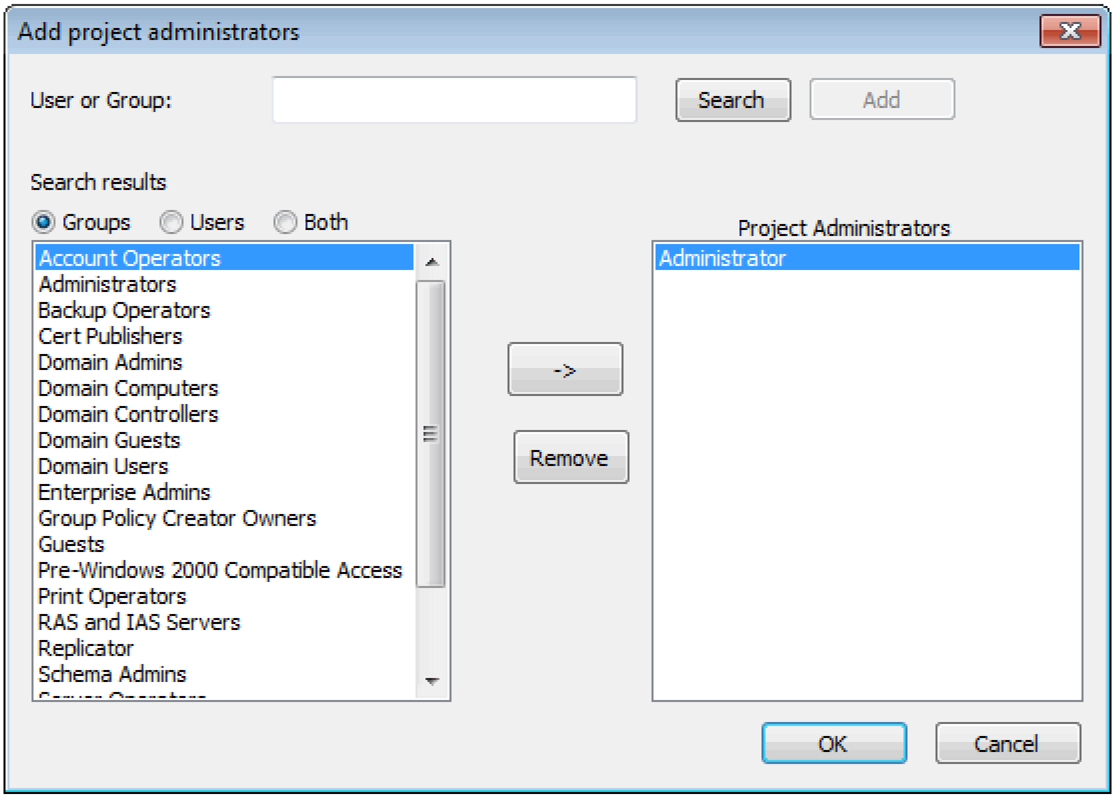
- Right-click **Windows Authentication** and choose **Enable**.



Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

Active Directory authentication and Permissions

You can use Permissions feature along with Active Directory authentication. Click on **Permissions** and enable the **Use Dynamic Permissions** check box. You need to choose tables which will store permissions and create admin user. To add admin user, click **Add admin user** and then **Search**. You will be asked username and password to connect to Active Directory and then you will be able to select a group or groups that will have admin access.



User or Group:

Search results

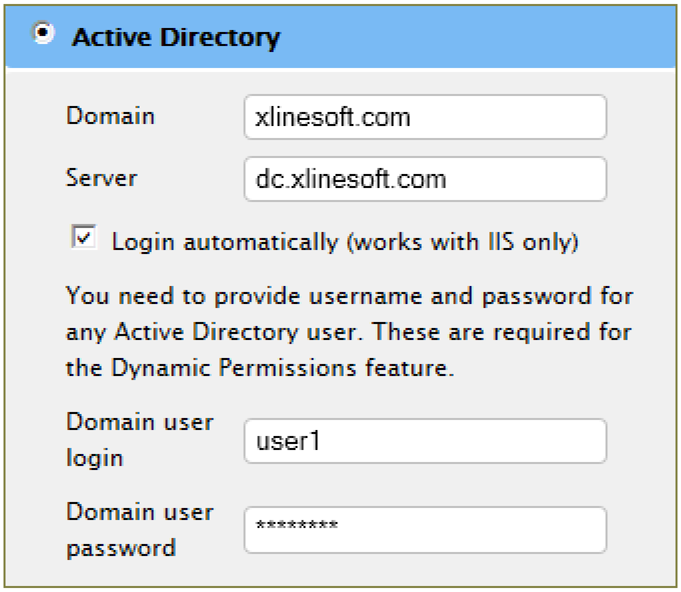
● Groups ○ Users ○ Both

Account Operators
Administrators
Backup Operators
Cert Publishers
Domain Admins
Domain Computers
Domain Controllers
Domain Guests
Domain Users
Enterprise Admins
Group Policy Creator Owners
Guests
Pre-Windows 2000 Compatible Access
Print Operators
RAS and IAS Servers
Replicator
Schema Admins

Project Administrators

Administrator

If you enabled **Login automatically** and **Use dynamic permissions** check boxes, you also need to specify domain user login and password.


● Active Directory

Domain: xlinesoft.com

Server: dc.xlinesoft.com

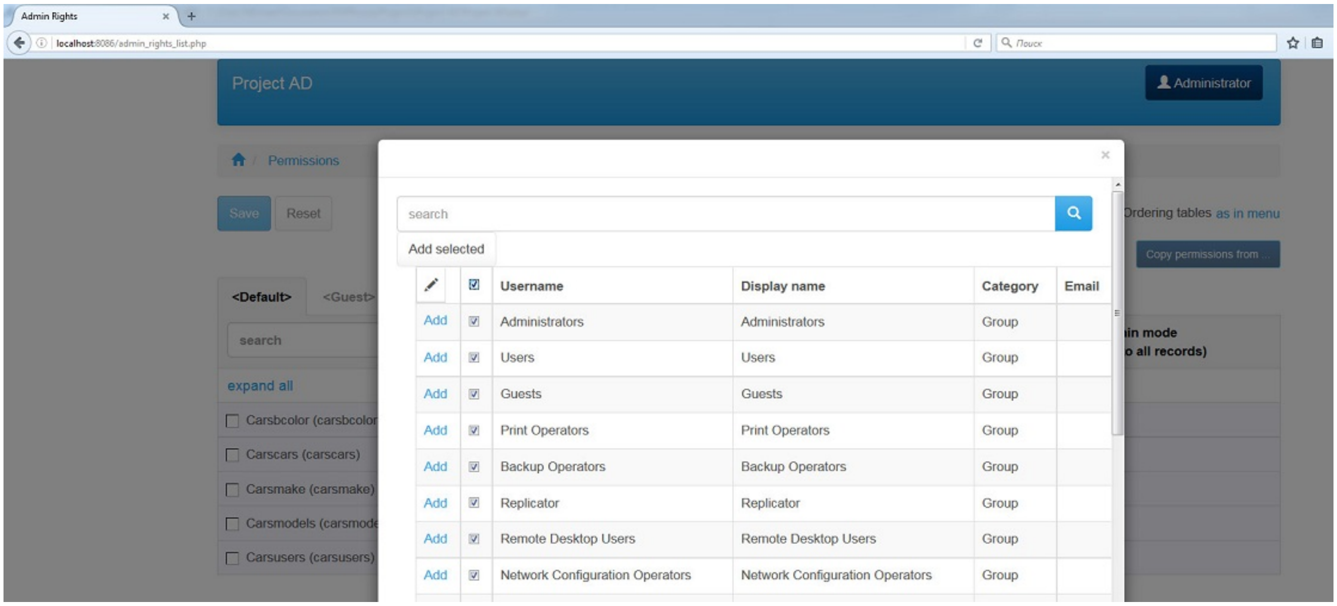
☑ Login automatically (works with IIS only)

You need to provide username and password for any Active Directory user. These are required for the Dynamic Permissions feature.

Domain user login: user1

Domain user password: *********

Build your project and login as admin to the generated application. In Admin Area on Admin Rights page you can add groups via **Add Group** and assign permissions to them. Note that you can not create groups manually since they are stored on the Active Directory server and should be modified there.

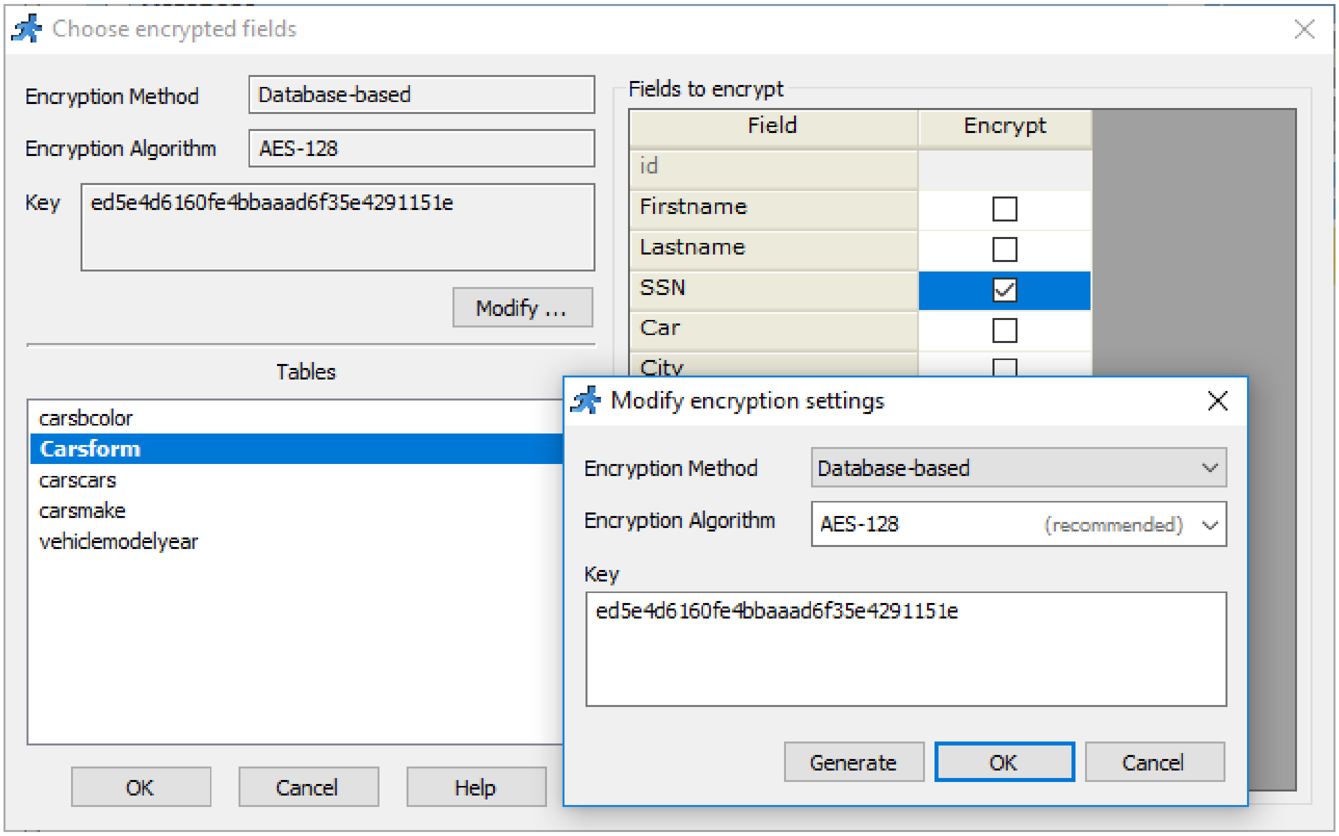


	☑	Username	Display name	Category	Email
Add	☑	Administrators	Administrators	Group	
Add	☑	Users	Users	Group	
Add	☑	Guests	Guests	Group	
Add	☑	Print Operators	Print Operators	Group	
Add	☑	Backup Operators	Backup Operators	Group	
Add	☑	Replicator	Replicator	Group	
Add	☑	Remote Desktop Users	Remote Desktop Users	Group	
Add	☑	Network Configuration Operators	Network Configuration Operators	Group	

2.15.9 Encryption

Encryption feature allows you to encrypt important data in the database such as credit card number or Social Security number. You need to select encryption method, enter encryption key and choose fields to be encrypted.

Note: Encryption feature is available only in the Enterprise Edition of PHPRunner. See Editions Comparison.



Encryption Method: Database-based

Encryption Algorithm: AES-128

Key: ed5e4d6160fe4bbaaad6f35e4291151e

Tables:
- carsbcolor
- Carsform
- carscars
- carsmake
- vehiclemodelyear

Field	Encrypt
id	
Firstname	□
Lastname	□
SSN	☑
Car	□
City	□

Modify encryption settings:

Encryption Method: Database-based

Encryption Algorithm: AES-128 (recommended)

Key: ed5e4d6160fe4bbaaad6f35e4291151e

You can select database-based or code-based encryption method. Note that database-based encryption method is available only for MySQL, Oracle, PostgreSQL and MS SQL Server databases. Database-based method is recommended since some features will not work in case of code-based method: encrypted fields are not involved in the sorting and grouping, search suggest and search with all search operators other than Equal will not work etc. So make sure you encrypt really important data.

Database-based encryption

To use database-based encryption method in PostgreSQL module *pgcrypto* must be installed; in Oracle a user must be given the rights to the package *SYS.DBMS_CRYPTO* and Oracle version shall be 10 or higher; MySQL should be configured to support SSL.

Code-based encryption

To use PHP encryption, *mcrypt* extension needs to be enabled in *php.ini* (in PHP 5.3 and higher it is included in the distribution by default).

When using PHP encryption encrypted fields are not involved in sorting and grouping, search suggest and search with all search operators other than Equal will not work.

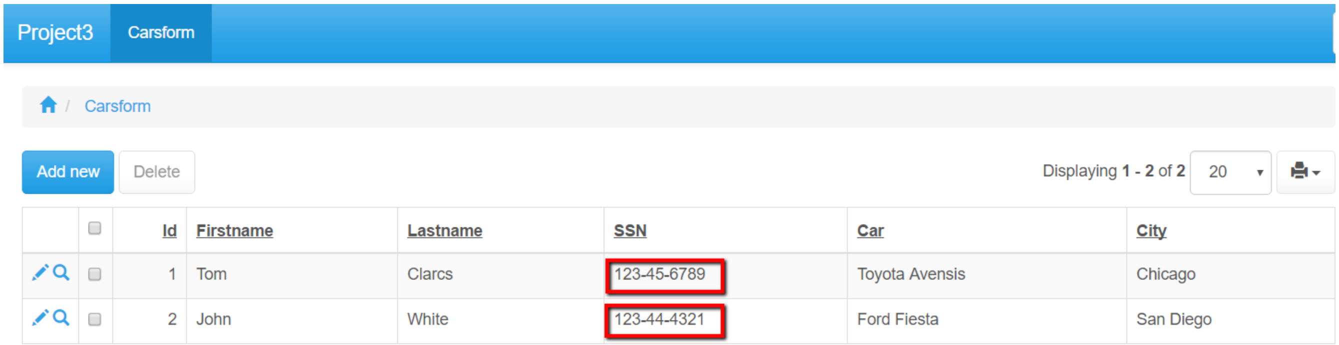
Encryption key

The encryption key length more than 10 characters is recommended. Use **Generate** button to generate a random key. You can encrypt only text fields. Since the encrypted value is much longer than source value (at least 2-3 times), you should choose fields with maximum length such as TEXT in MySQL or MEMO in MS Access.

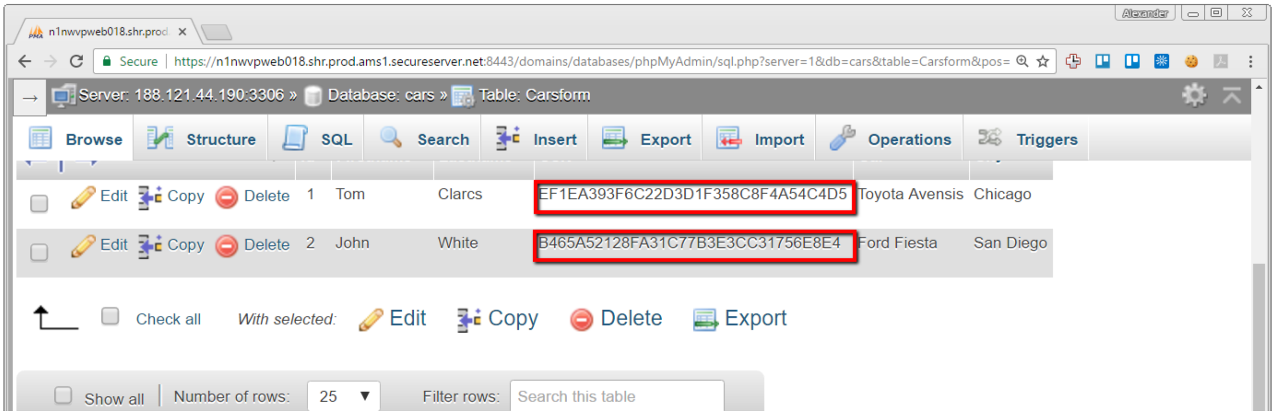
Note: PHPRunner does not encrypt existing data. Encryption will be applied only to the add/edit operations.

Note: Once encrypted data are stored in the database you should not change the encryption type or key, also you can not cancel the encryption otherwise data will remain encrypted.

Example of how encrypted data are stored in the database and displayed in the application



	□	Id	Firstname	Lastname	SSN	Car	City
	□	1	Tom	Clarcs	123-45-6789	Toyota Avensis	Chicago
	□	2	John	White	123-44-4321	Ford Fiesta	San Diego


n1nwvpweb018.shr.prod
Secure
https://n1nwvpweb018.shr.prod.ams1.secureserver.net:8443/domains/databases/phpMyAdmin/sql.php?server=1&db=cars&table=Carsform&pos=
Server: 188.121.44.190:3306 » Database: cars » Table: Carsform
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Triggers

□	Edit Copy Delete	1	Tom	Clarcs	EF1EA393F6C22D3D1F358C8F4A54C4D5	Toyota Avensis	Chicago
□	Edit Copy Delete	2	John	White	B465A52128FA31C77B3E3CC31756E8E4	Ford Fiesta	San Diego

□ Check all
With selected:
Edit
Copy
Delete
Export
□ Show all
Number of rows: 25
Filter rows: Search this table

Functions used for database-based encryption

Oracle:

- Encryption: DBMS_CRYPTO.ENCRYPT()
- Decryption: DBMS_CRYPTO.DECRYPT()

MS SQL Server:

- Encryption: EncryptByPassPhrase(), EncryptByKey()
- Decryption: DecryptByPassPhrase(), DecryptByKey()

MySQL:

- Encryption: DES_ENCRYPT(), AES_ENCRYPT()
- Decryption: DES_DECRYPT(), AES_DECRYPT()

PostgreSQL:

- Encryption: pgp_sym_encrypt()

- Decryption: pgp_sym_decrypt()

Functions used for code-based encryption:

PHPRunner uses DES or AES-128 encryption algorithm.

Encrypt existing values in the database

Before starting this procedure make a backup of the database. You may perform the encryption of existing values only once. Double encryption will cause problems and it is not possible to determine definitely whether the data was encrypted or not before the procedure.

To encrypt existing values in the database add following code to the List page: Before process event of your table:

```
include_once("ciphcoding.php");
```

Then run you List page that contains encrypted fields with *ciphcoding=1* parameter, eg:

```
mytable_list.php?ciphcoding=1
```

Once the data has been encrypted, it is necessary to delete the file *ciphcoding.php* in the output directory, remove the code from the **List page: Before process** event and re-upload application. We recommend to perform this procedure on the development machine or server without public access.

Decrypt custom query results

MySQL, AES encryption

$key variable should contain encryption key specified in PHPRunner on encryption screen.

```
//define encryption key
$key='09a308862fbe462095dd6eba33ab9dd21b8fd35b0d884b48819a34ce8636983b';

$sql = "SELECT cast(AES_DECRYPT(unhex(customer_name), '".$key."') as char) AS
custname FROM customers_table WHERE id = 'CUST123'";
$rs = CustomQuery($sql);
$data = db_fetch_array($rs);

echo $data["custname"];
```

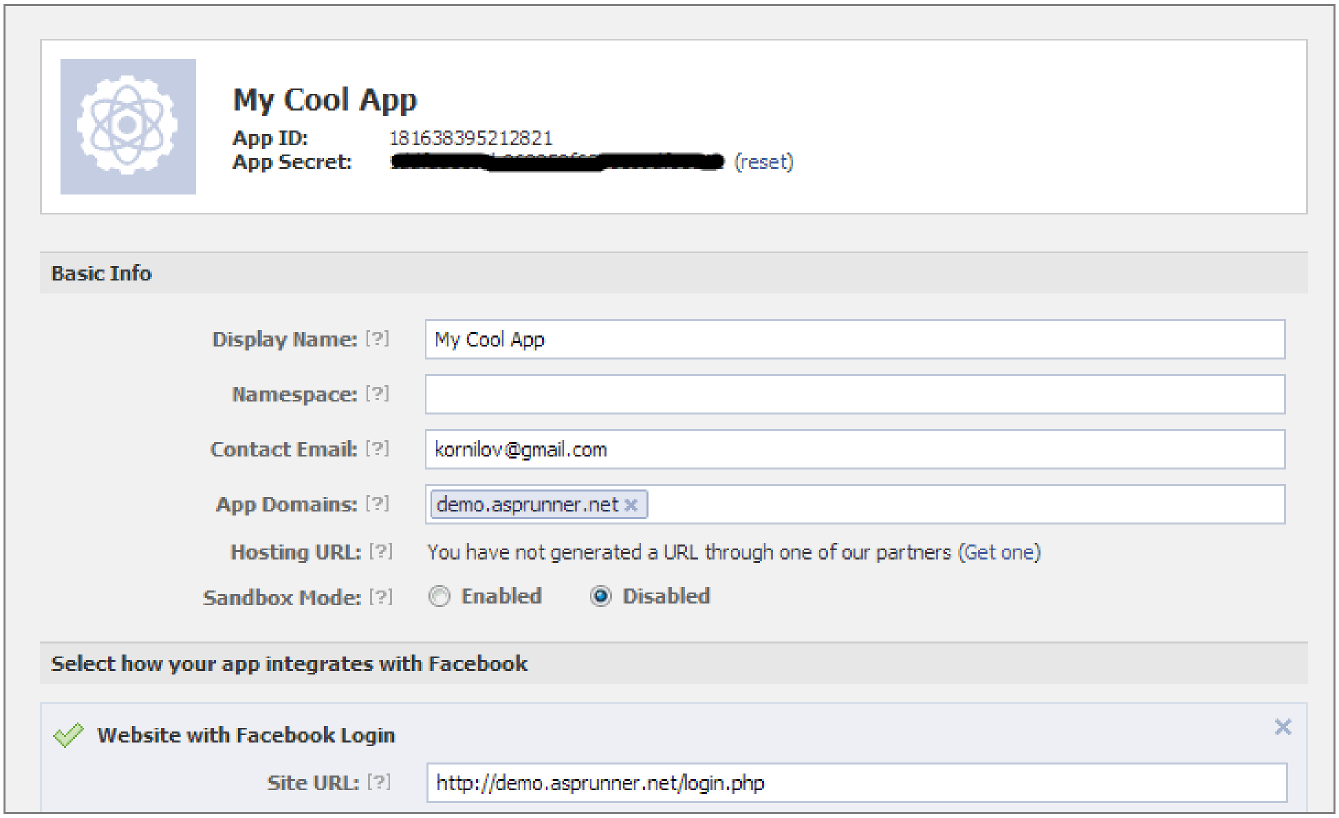
2.15.10 Facebook connect

Now you can integrate Facebook Connect into your existing site with PHPRunner. Facebook Connect is a set of APIs from Facebook that provides a simple way for people to sign in to Facebook-linked web sites, applications and mobile devices using their Facebook ID (**Login with Facebook** option). In addition to simplifying the registration process, Facebook Connect allows you to extend some social networking features onto your sites, thus making it easier to share content through Facebook.

How to configure Facebook Connect?

1. Register your website at http://developers.facebook.com/setup/. You create new Facebook application and configure it to point to your existing site's root URL. You should specify the URL of login page as Site URL e.g. *http://demo.asprunner.net/login.php*.

Then you can see your application settings (Facebook App ID and Secret ID) at http://www.facebook.com/developers/apps.php.



My Cool App

App ID: 181638395212821

App Secret: [redacted] (reset)

Basic Info

Display Name: [?] My Cool App

Namespace: [?]

Contact Email: [?] kornilov@gmail.com

App Domains: [?] demo.asprunner.net ×

Hosting URL: [?] You have not generated a URL through one of our partners (Get one)

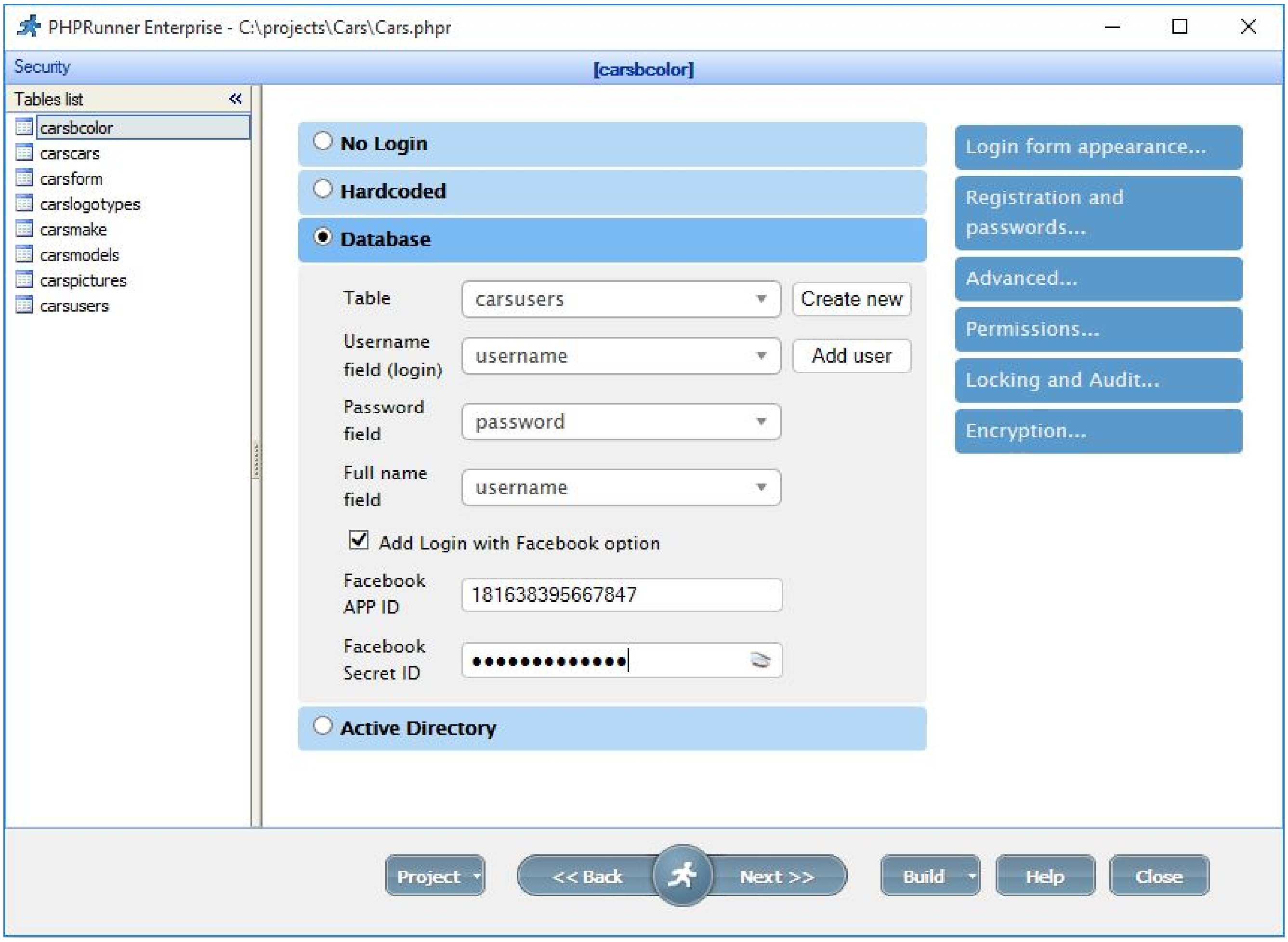
Sandbox Mode: [?] ○ Enabled ● Disabled

Select how your app integrates with Facebook

✓ Website with Facebook Login

Site URL: [?] http://demo.asprunner.net/login.php

2. On the **Security** page enable **Add login with Facebook** check box and enter your Facebook App ID and Secret ID.



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Security [carsbcolor]

Tables list: carsbcolor, carscars, carsform, carslogotypes, carsmake, carsmodels, carspictures, carsusers

○ No Login

○ Hardcoded

● Database

Table	carsusers	Create new
Username field (login)	username	Add user
Password field	password	
Full name field	username	

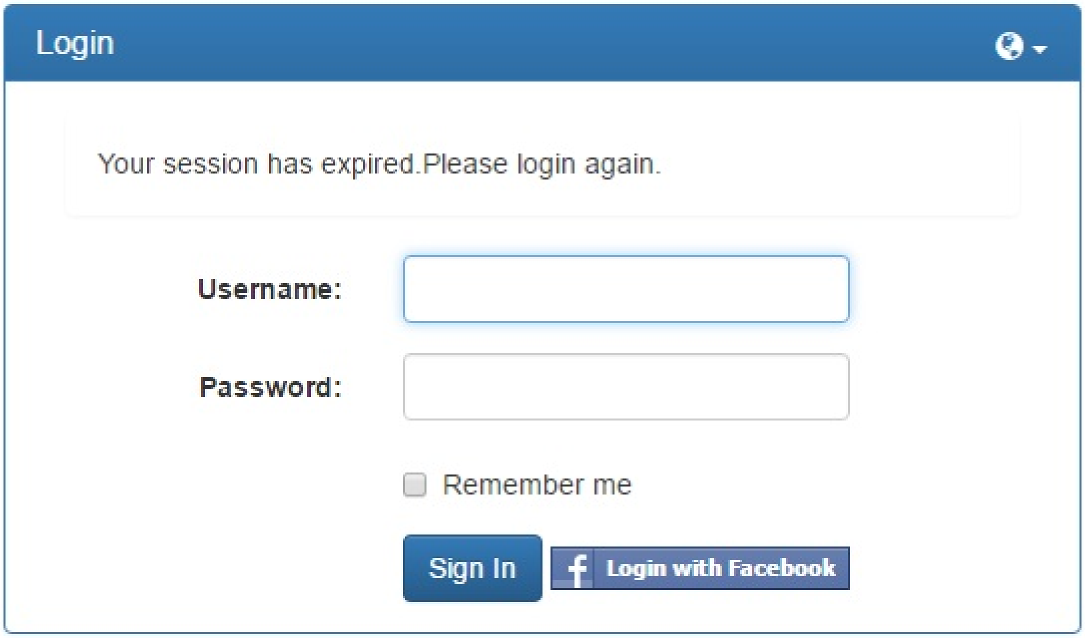
☑ Add Login with Facebook option

Facebook APP ID: 181638395667847

Facebook Secret ID: •••••••••••••

○ Active Directory

Now you can see **Login with Facebook** button on the login page of your application.



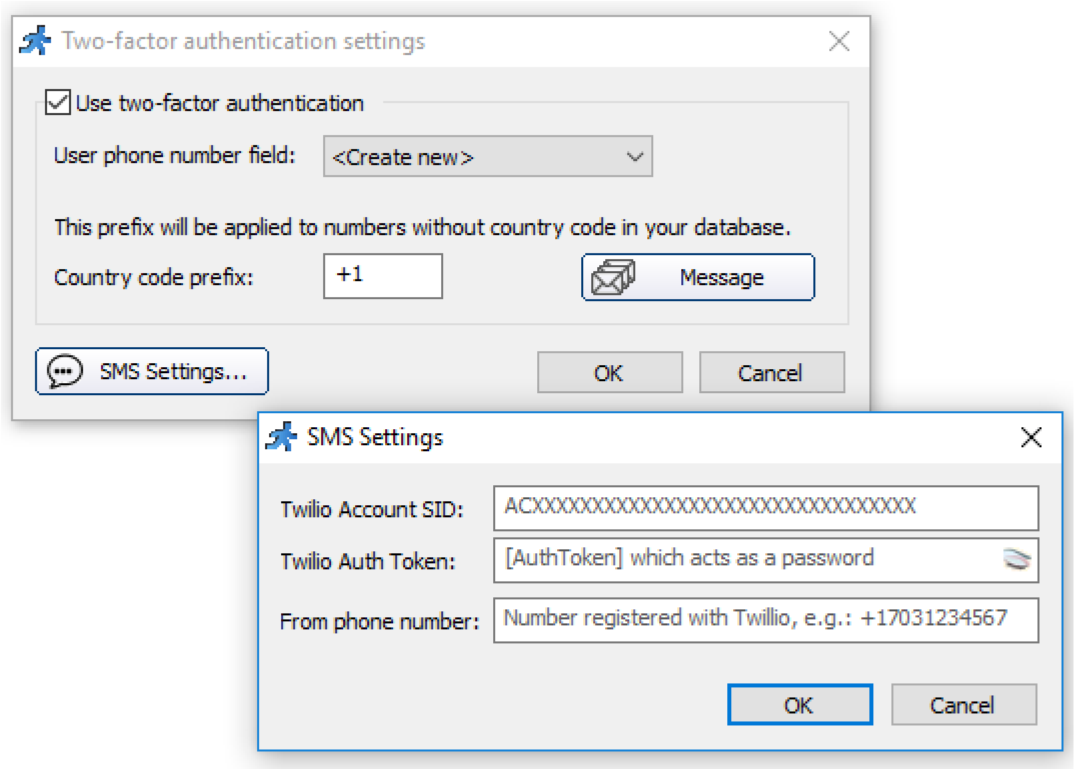
Your session has expired.Please login again.

If you are interested in additional integration capabilities check the article at http://xlinesoft.com/blog/...facebook-connect/.

If you have enabled Dynamic permissions, Facebook users will be assigned the rights of Default group: Facebook users -> Default users.

2.15.11 Two-factor authentication

You can use this feature if you need that during the login process the user should receive test message with verification code. You need to have the Twilio account to use this functionality. Use them for your SMS Settings in the application.
You can read more about the two-factor authentication on Twilio website



Two-factor authentication settings

☑ Use two-factor authentication

User phone number field: <Create new>

This prefix will be applied to numbers without country code in your database.

Country code prefix: +1

SMS Settings

Twilio Account SID: ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Twilio Auth Token: [AuthToken] which acts as a password

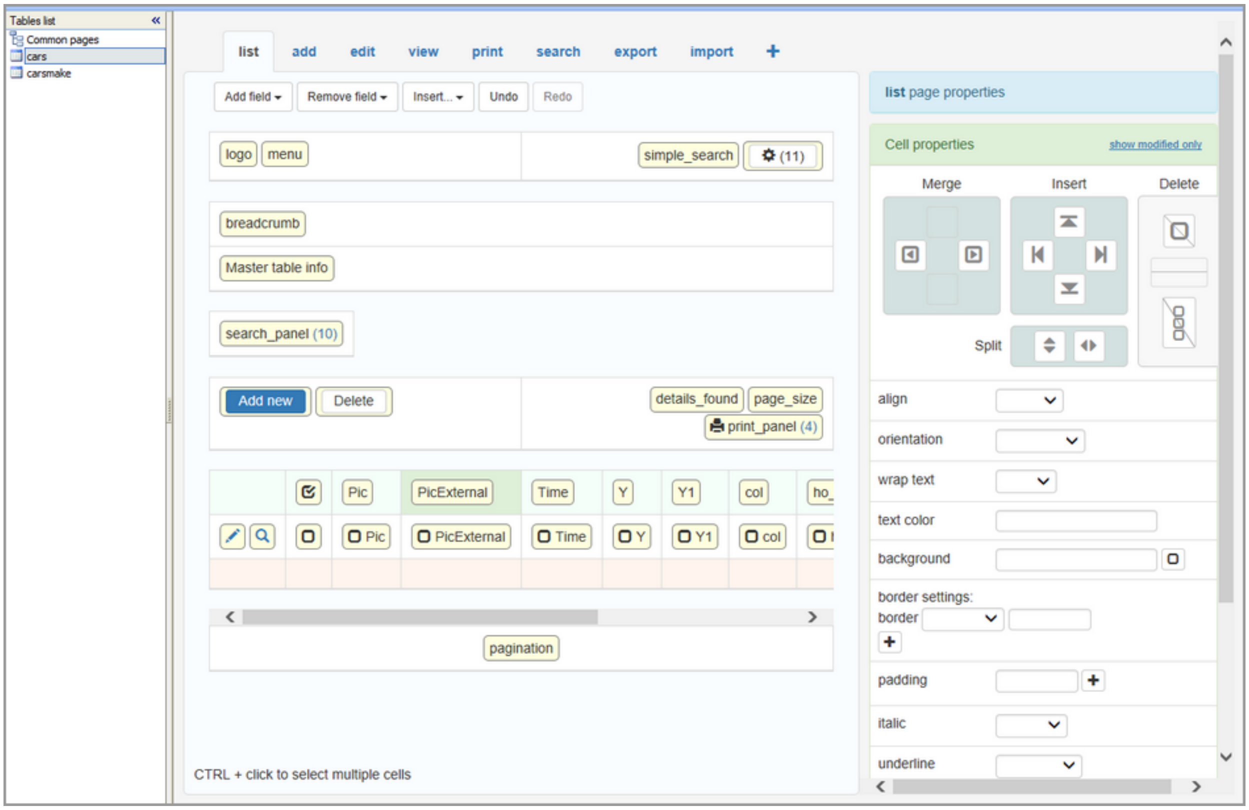
From phone number: Number registered with Twillio, e.g.: +17031234567

2.16 Page Designer

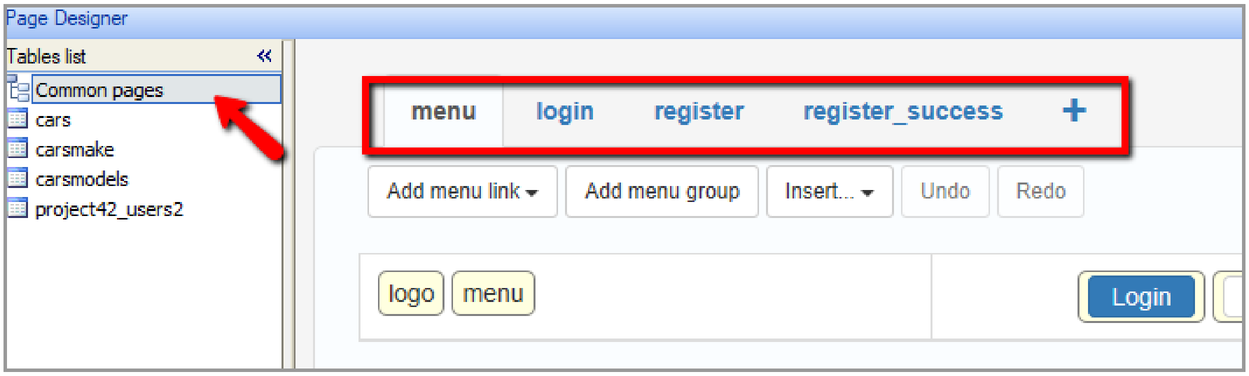
2.16.1 About Page Designer

Page Designer

Page Designer It is a grid-based design tool that adds more functionality to the software. You can modify appearance of all pages in your app. Drag-n-drop everything, insert buttons, code snippets, fields, tabs, sections etc.

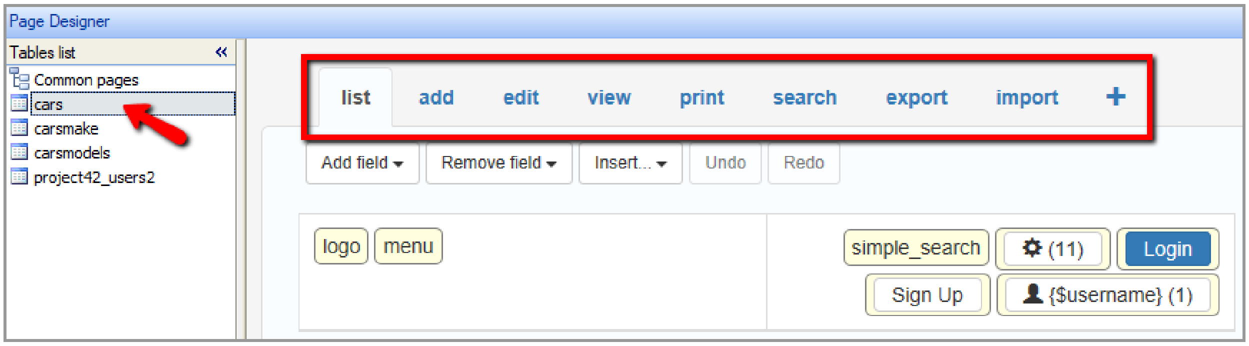


You can work with different types of objects on the Page designer screen: pages, cells and elements. Each type has different set of options and properties. To see functionality that is available for the current object you need to click on it on the Page designer tab. If you click on the Common pages in the tables list you'll see tabs like menu, login, register and so on. You can click on "+" tab to add page.

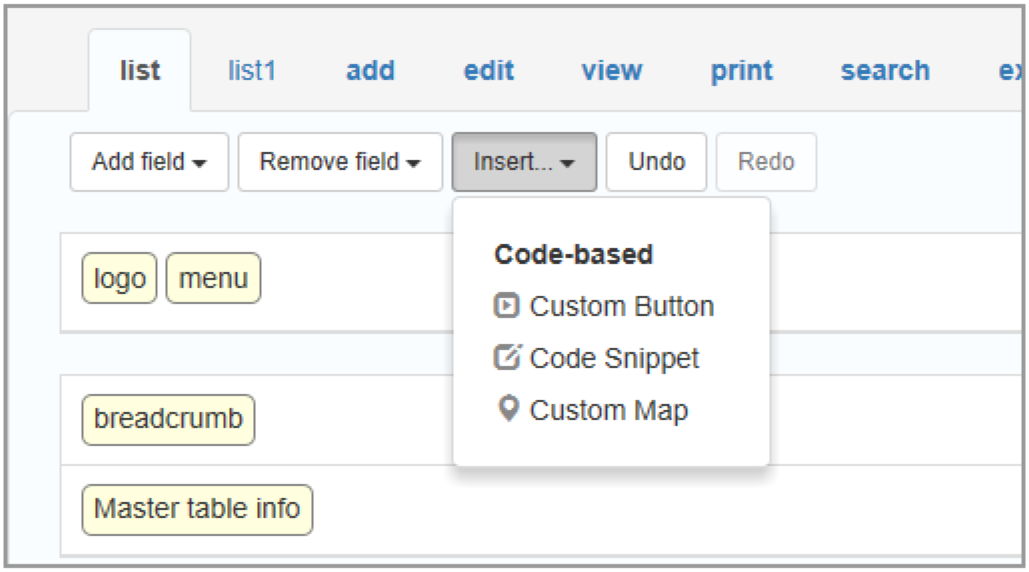


Below you can see buttons and controls available for that page, "Add menu link", "Add menu group", "Insert", "Undo" and "Redo".

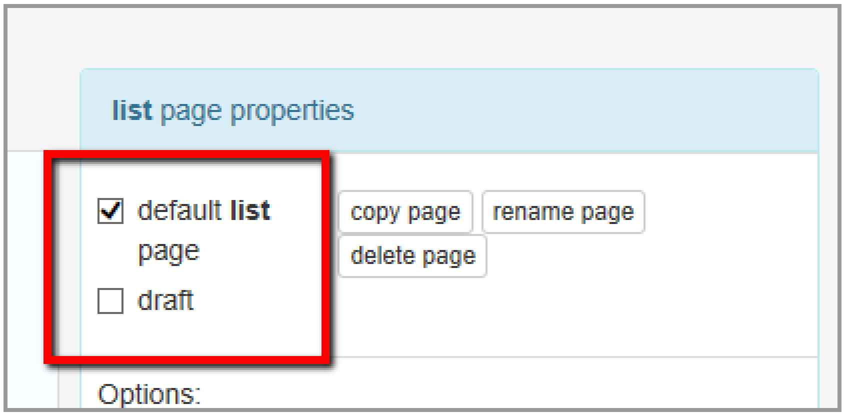
If you click on one of you project tables in the tables list you'll see tabs for project pages like list, add, edit and so on.



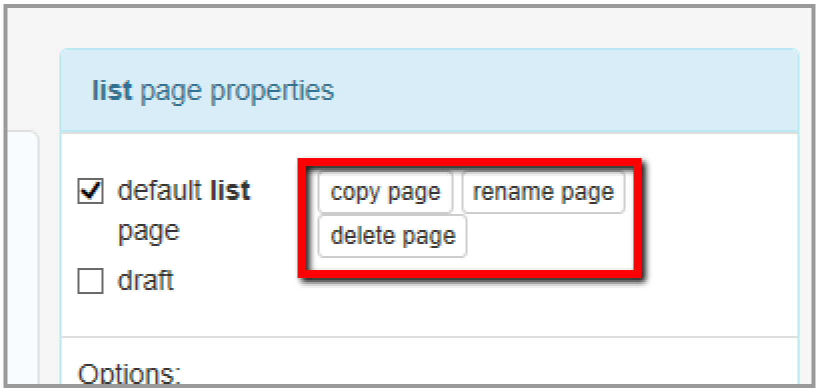
Now you can see buttons and controls available for that page, "Add field", "Remove field", "Insert", "Undo" and "Redo". You can find more info in the article Working with cells.
In the central part of the screen you'll see the preview area. You can click to select cells or elements or even drag and drop elements between cells.
To the right side of the screen you'll see available properties and options. Use "Insert" button to add code-based buttons, snippets and maps to the page. Use "Undo" and "Redo" buttons to undo/redo last change on the page.



Default/Draft page checkbox
This property is common for all pages. In the Page Designer you can create multiple pages of the same type i.e. two or three list pages based on the same table. Previously you had to create a custom view or two that were more difficult to manage and also created unnecessary pages. To add a version simply click on + and check the choose the type of the page to be created. Then press ok.

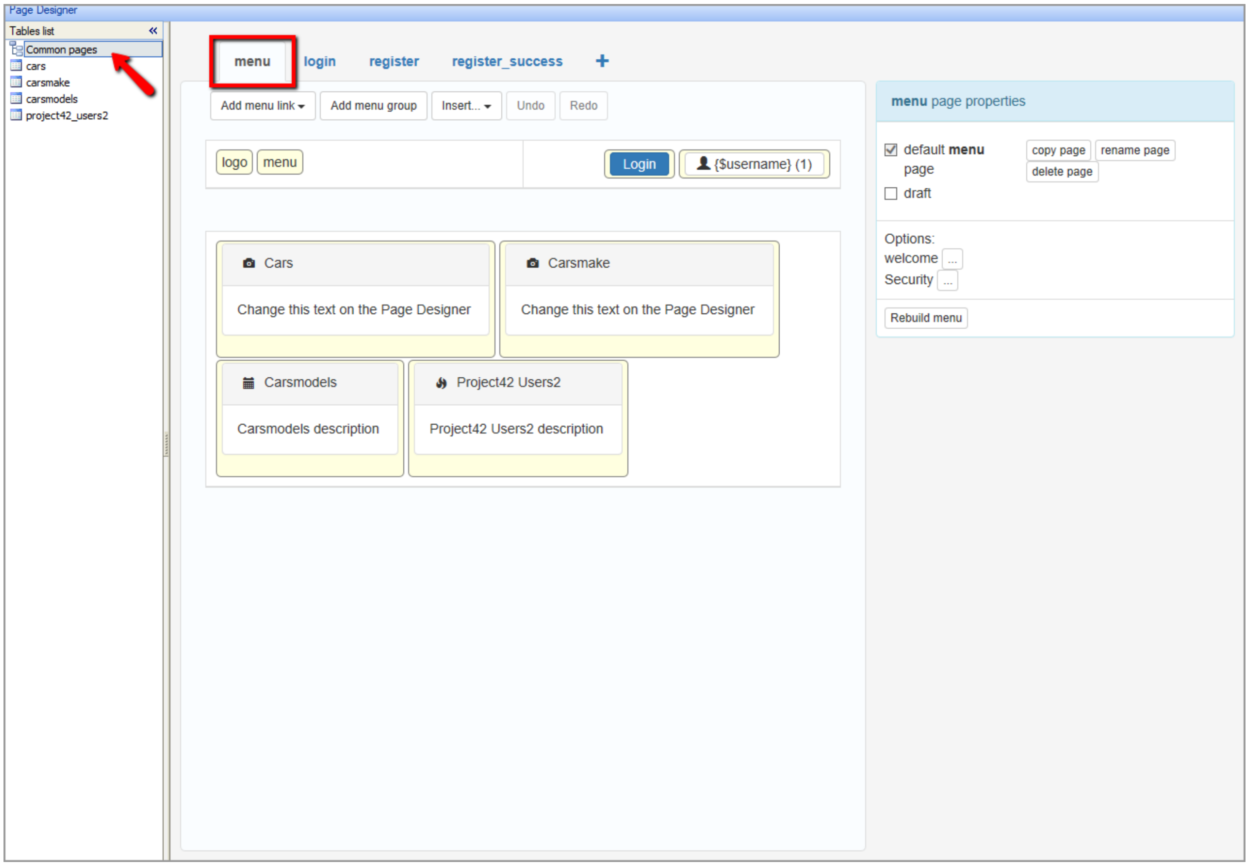


This will be useful in many scenarios. For instance you need to display a button on the list page but don't need to see it when the same list page is shown in dashboard. Or you need to design different list pages with different set of columns for different users. Now you can do this.
One of these pages is selected as the default. If page is marked as a draft it won't appear in generated application. Using drafts you can quickly switch between versions of the page.

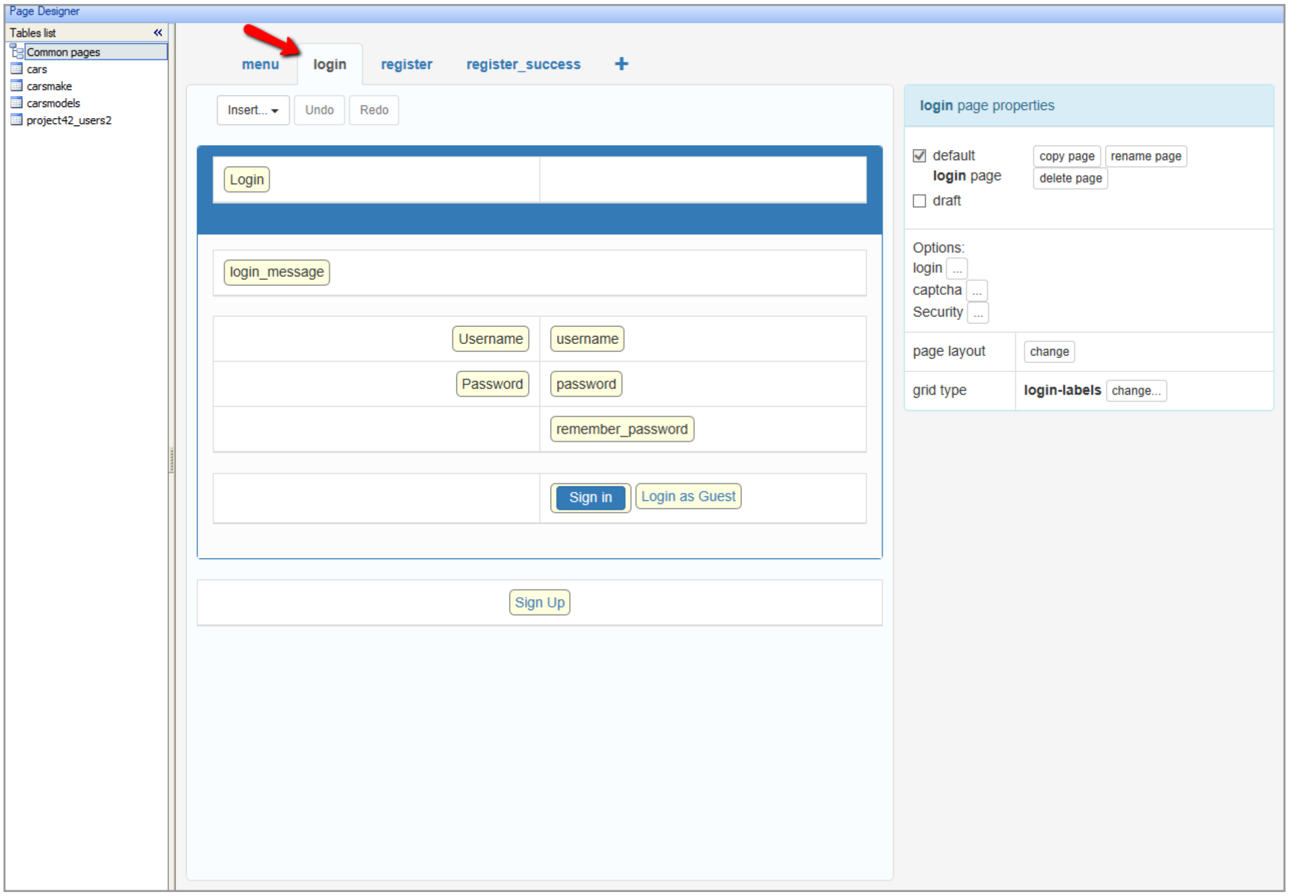


Copy page button - copies the current page. The auto incremented number will be added to the new new page.
Rename page button - press this button to rename the current page.
Delete page button - press this button to delete the current page.

For example let's see the Page Designer interface for the menu page. First click on the Common pages in the tables list. Menu tab will be opened in the top of the page, page preview will be shown in the central part of the application. On the right side you'll see "menu page Properties" and "Options" that can be changed for the menu page.

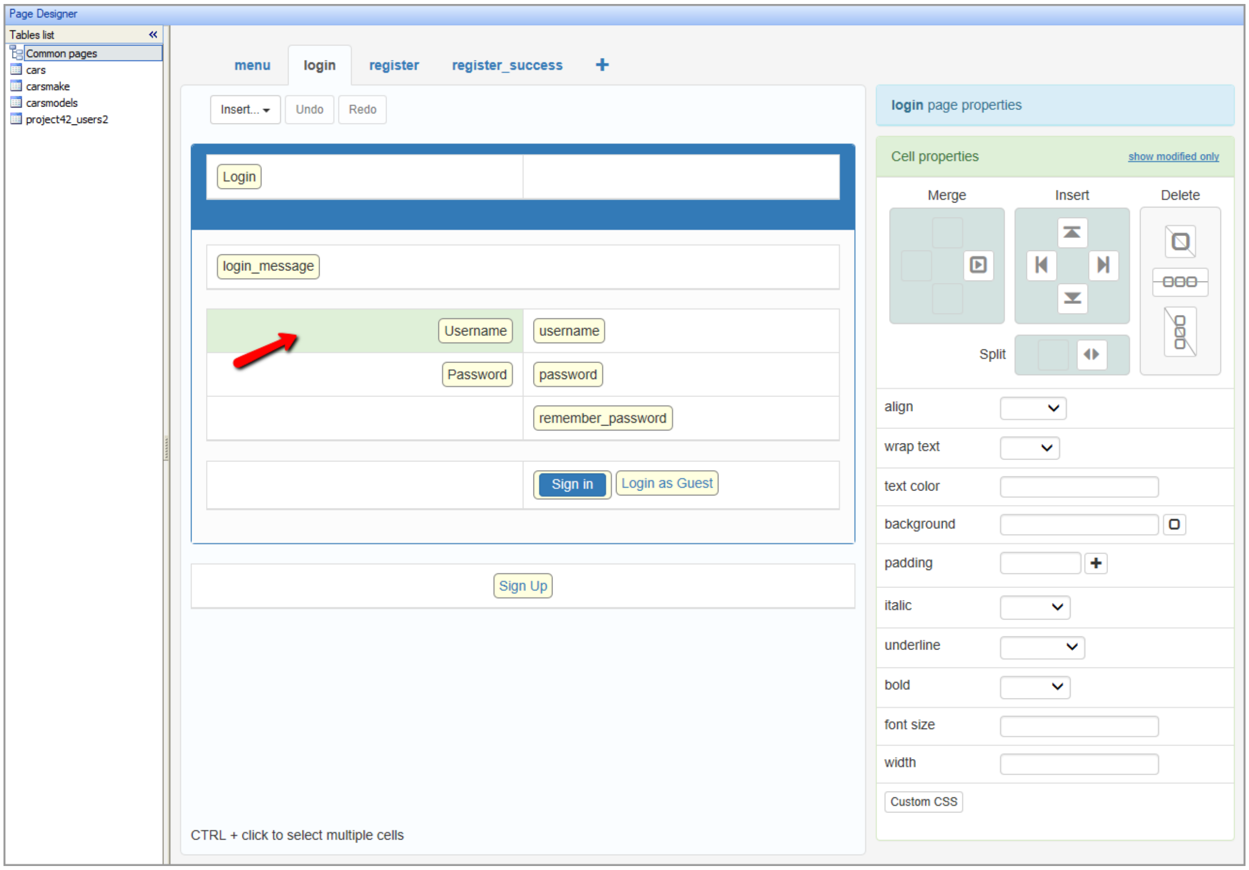


Click on the "login" tab to see the preview, properties and options available for the Login page:

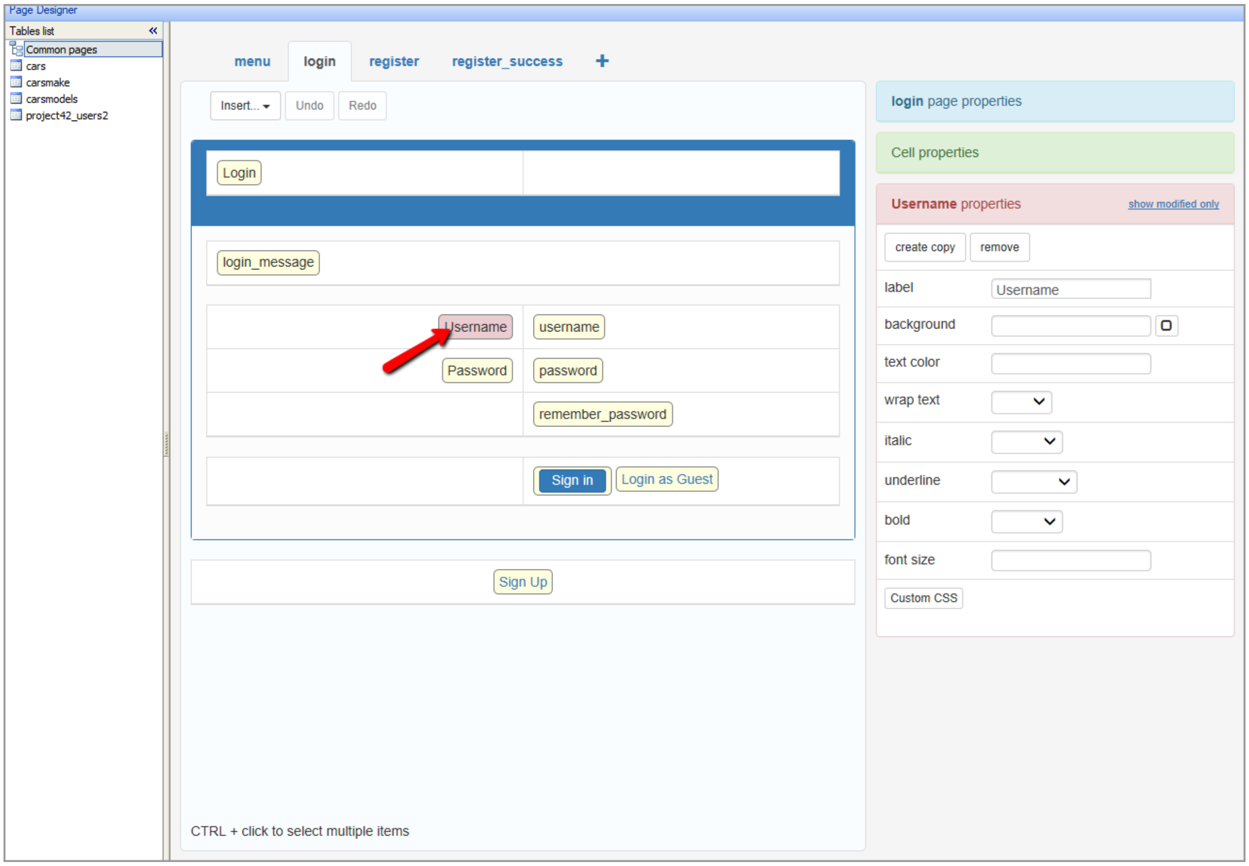


Click on one of the cells of the current pages, for example on Login page click on the cell containing "Username". The selected cell will he highlighted with light green fill and you'll see

Cell properties and options that are available for this cell to the right.



Now click on the "Username" element to see to the right properties and options available for it. The selected element will be highlighted with red fill.



label: Username

See also
Working with Common Pages
Working with Table Pages
Working with Cells
Working with Elements

2.16.2 Working with Common pages

Common pages

"Common pages" are pages like Menu Page, Login page, Register page, Change/Remind password and so on.

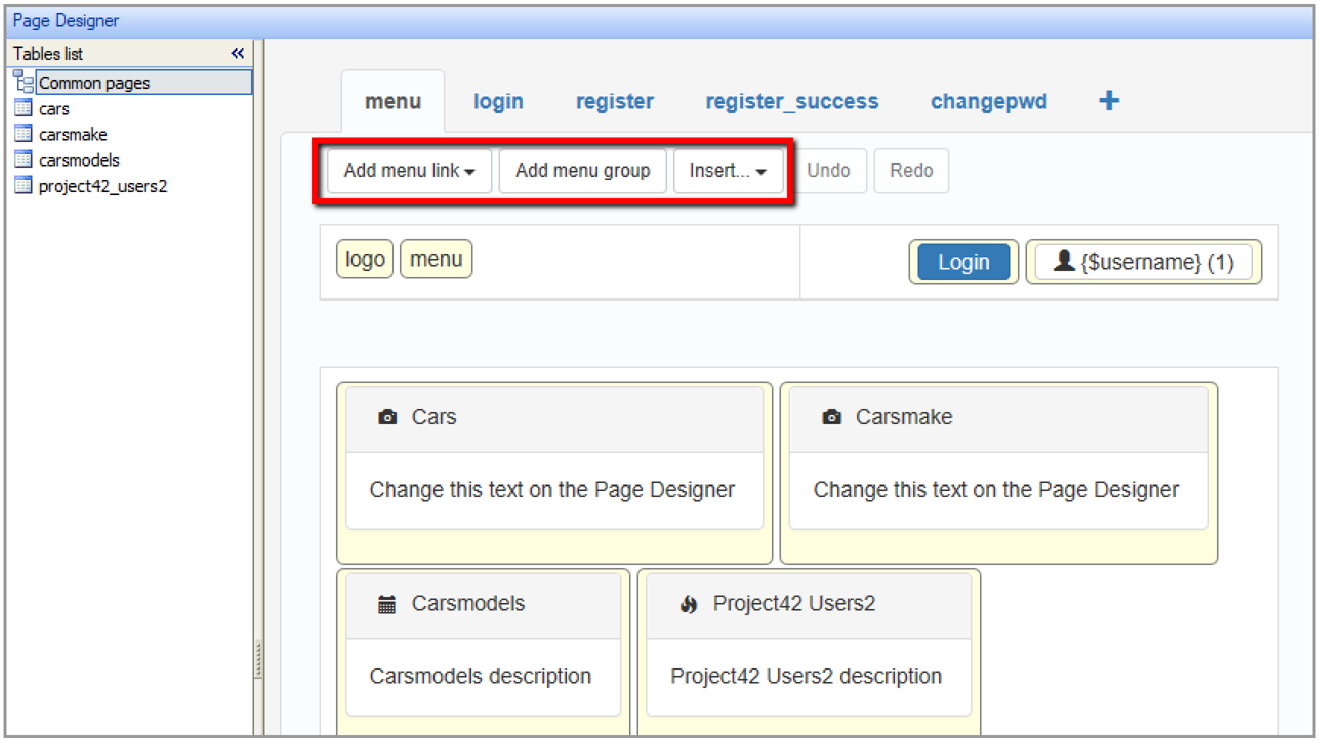
Add menu link button is used to add links to the menu page for the project tables. Links added will be shown on the generated welcome page.
Add menu group button is used for adding group to the menu page. You can add menu links into

different groups on the welcome page. To do this first add the new menu group and then drag and drop the table link to the new group.

All menu links will be added to the group that is currently highlighted in the Page designer window. If you need to move the link to another group just drag it to the other group or cell.

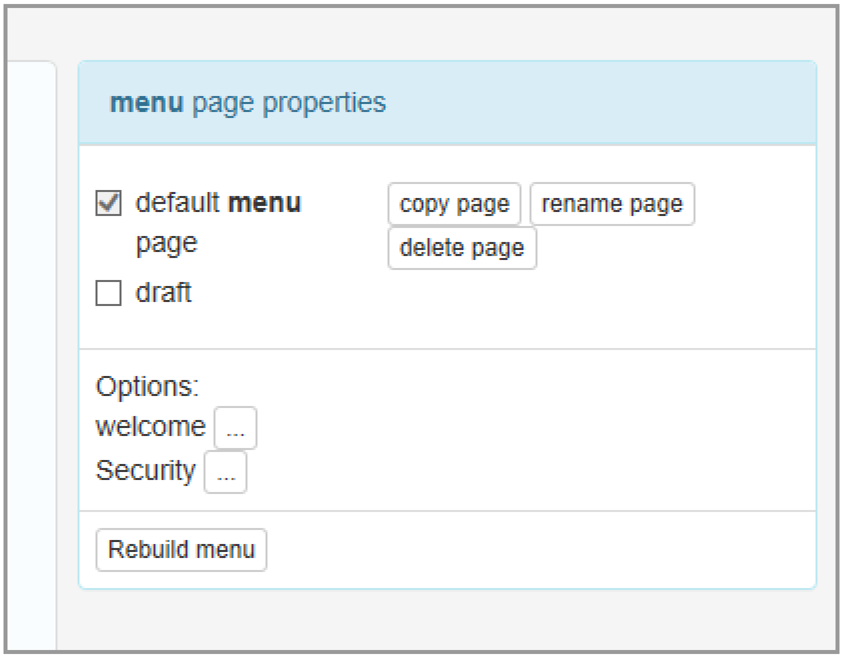
Insert button is used to add buttons, snippets and maps to the page.

Use **Undo** and **Redo** buttons to undo/redo last change on the page.



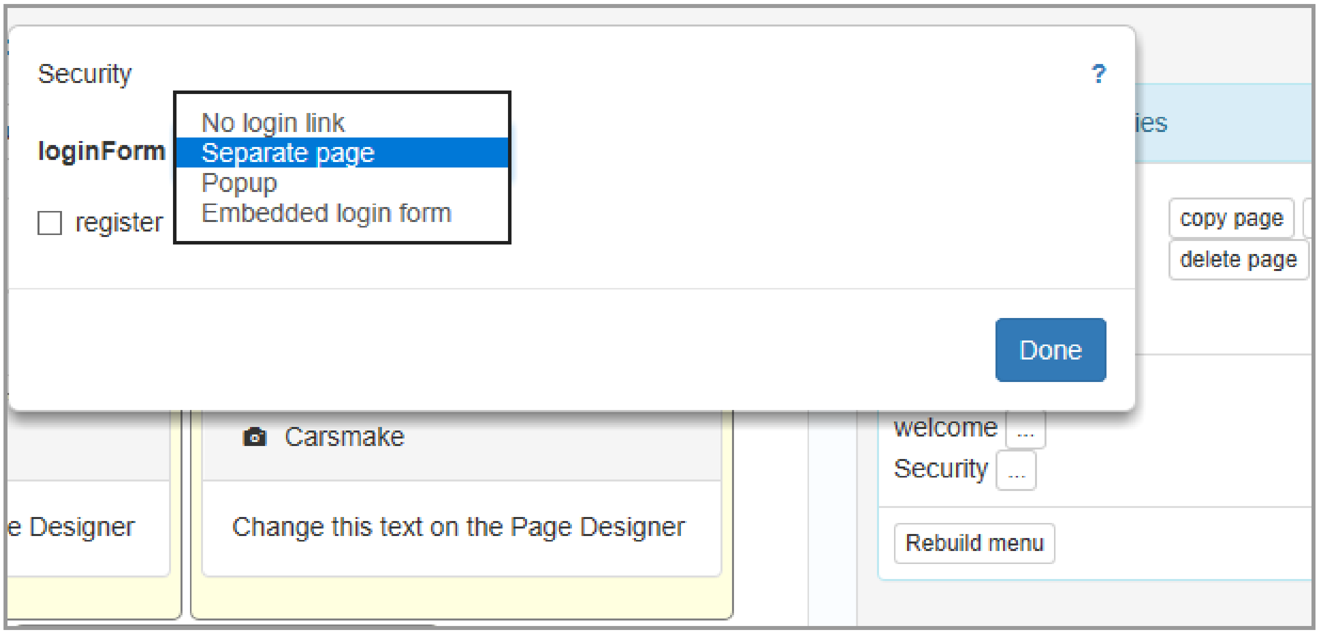
Menu Page Properties

On the right side you can see the menu page properties and Options:



Menu Page Options

Security



Press the "Security" button to choose the way the login form appearance form will be displayed

If you have enabled the change password and register page on the Security tab of the application you will also be able to turn on/off change password and register pages settings. These settings are also available on the user login settings form.

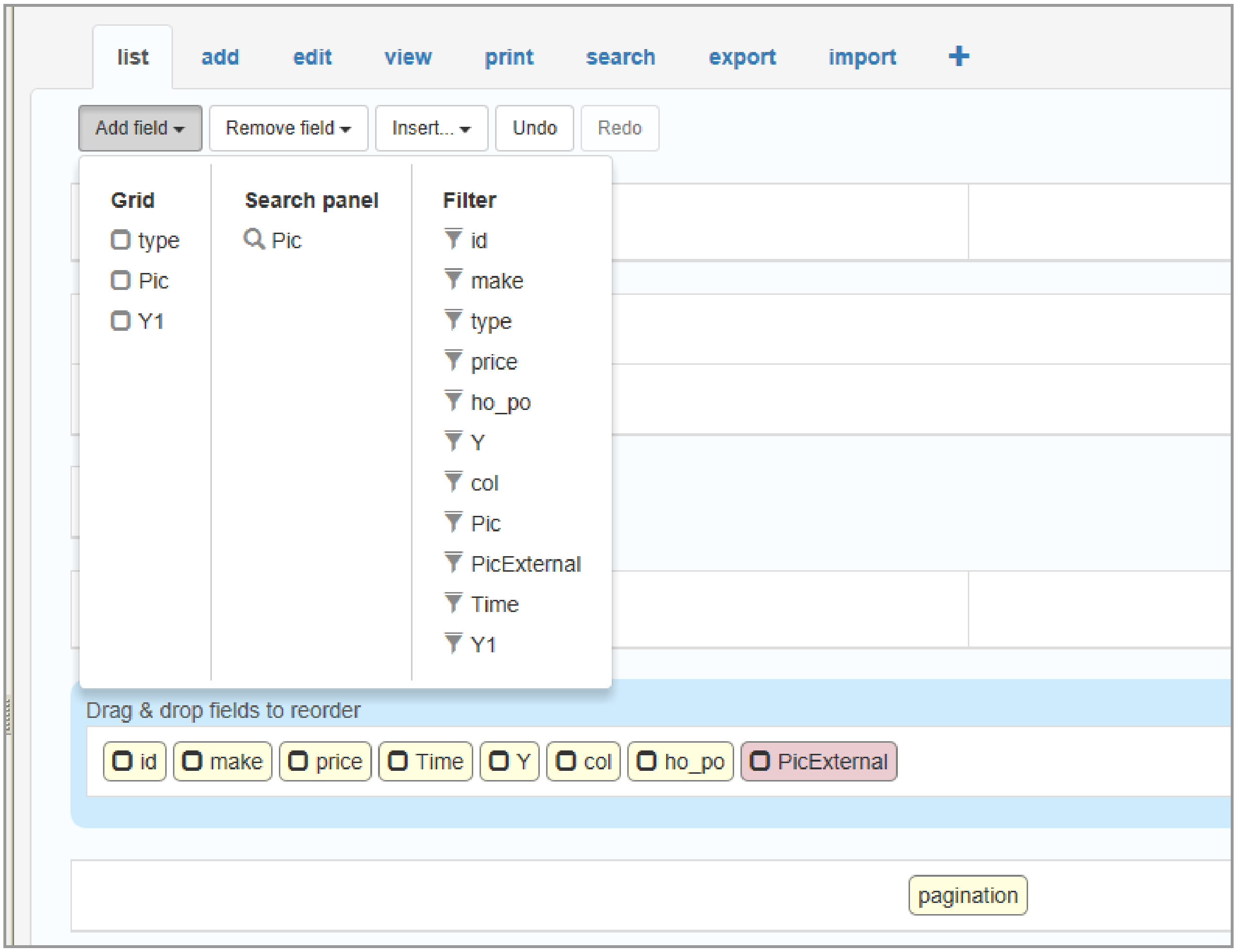
Rebuild menu button

If you added menu groups, added some cells to these groups and do some changes with the menu you press the Rebuild menu button.

2.16.3 Working with Table pages

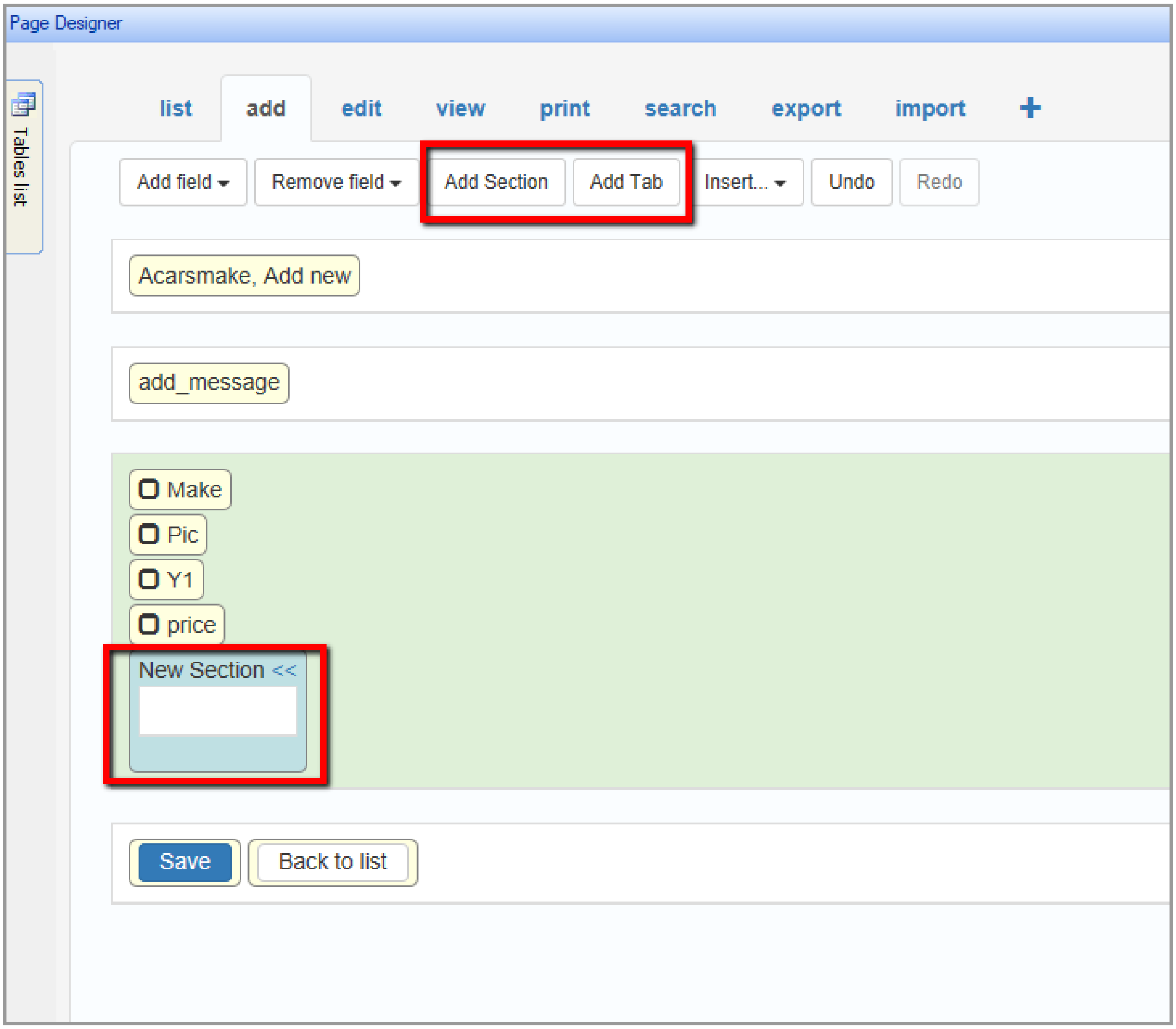
Table pages

Table pages are pages like List page, Add page, Edit page, View page, etc.
You can see **Add field**, **Remove field** buttons for this type of pages. Use these buttons to add table fields to the page grid, filter or search panels.
Click on the field name to add or remove it.

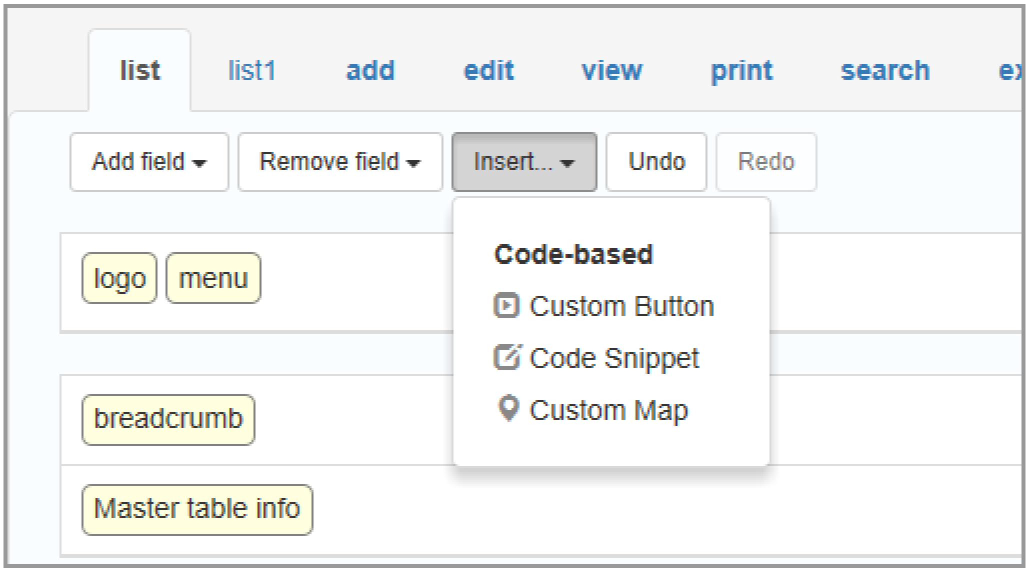


Add Section/Add Tab buttons are available for the Add, Edit, View page. After you press you can see the section or tab in the preview area. Now you need to drag and drop fields to the New section/Tab. To rename or format your section/tab click on it and modify its properties. See

Working with elements for details.



Insert button is used to add buttons, snippets and maps to the page.

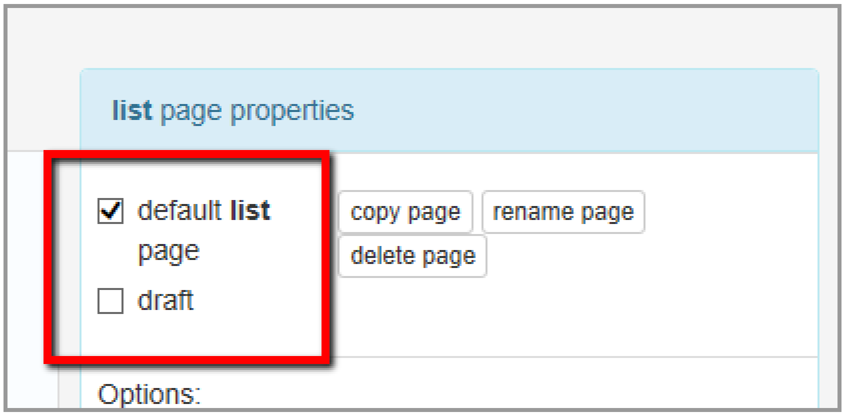


Use **Undo** and **Redo** buttons to undo/redo last change on the page.

Page properties

Default/Draft page checkbox

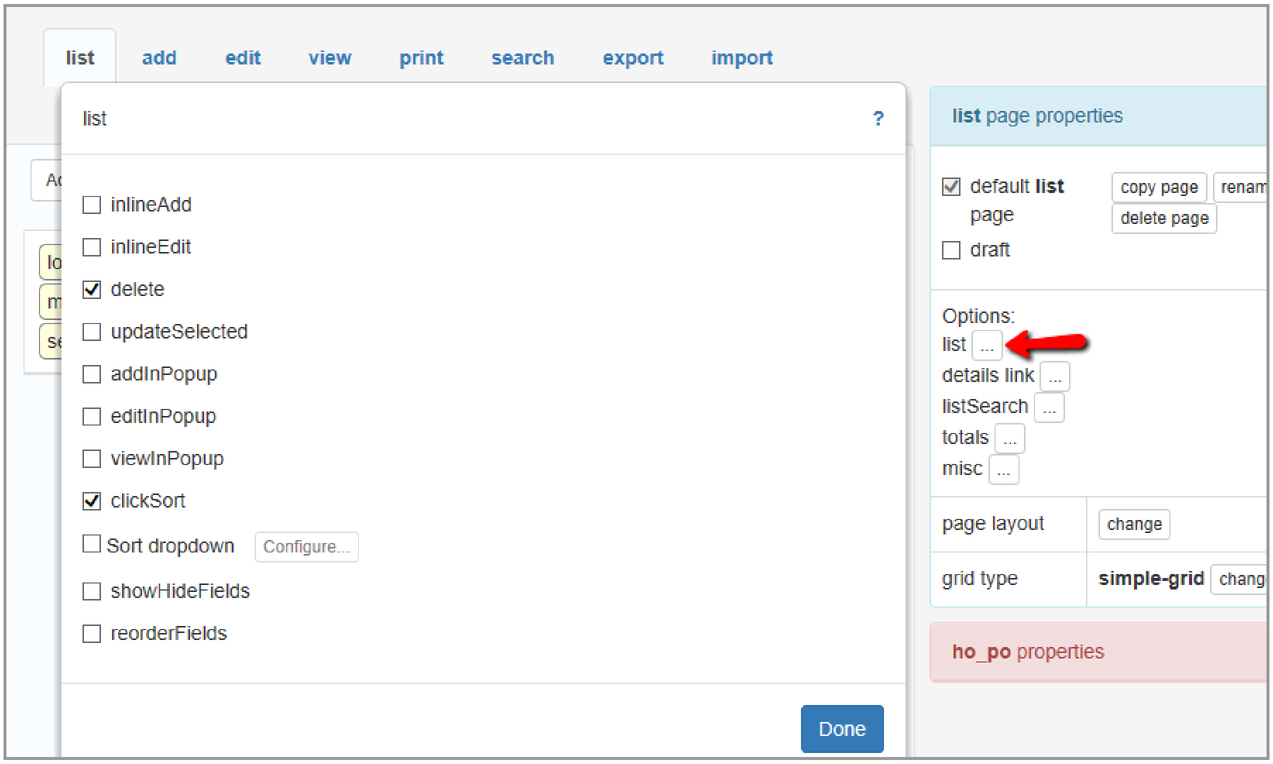
This property is common for all pages. In the Page Designer you can create multiple pages of the same type i.e. two or three list pages based on the same table. Previously you had to create a custom view or two that were more difficult to manage and also created unnecessary pages. To add a version simply click on + and check the choose the type of the page to be created. Then press ok.



Page Options

Here you can see options available for the page. For example List page has options to control Inline Add/Edit, Details link, List search or Totals. Add and Edit pages has the button to enable captcha on the generated page.

List options



list

□ inlineAdd

□ inlineEdit

☑ delete

□ updateSelected

□ addInPopup

□ editInPopup

□ viewInPopup

☑ clickSort

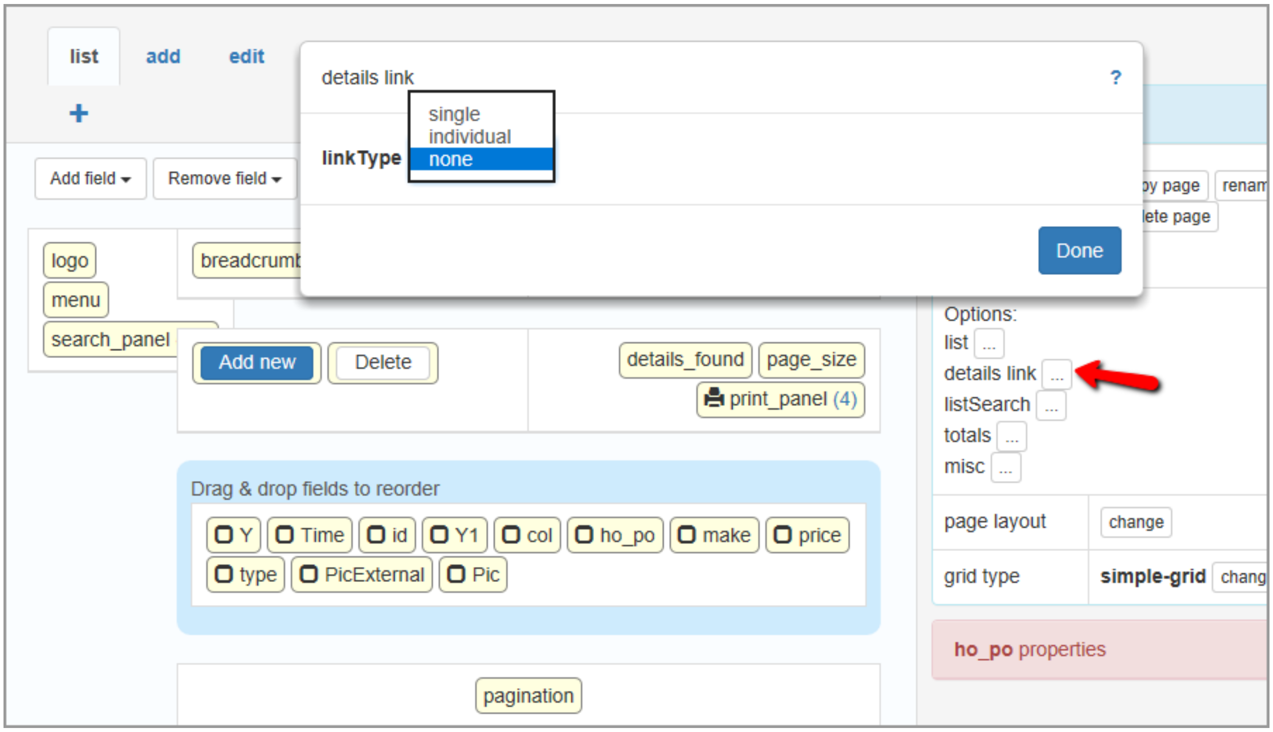
□ Sort dropdown Configure...

□ showHideFields

□ reorderFields

Done

Details link options

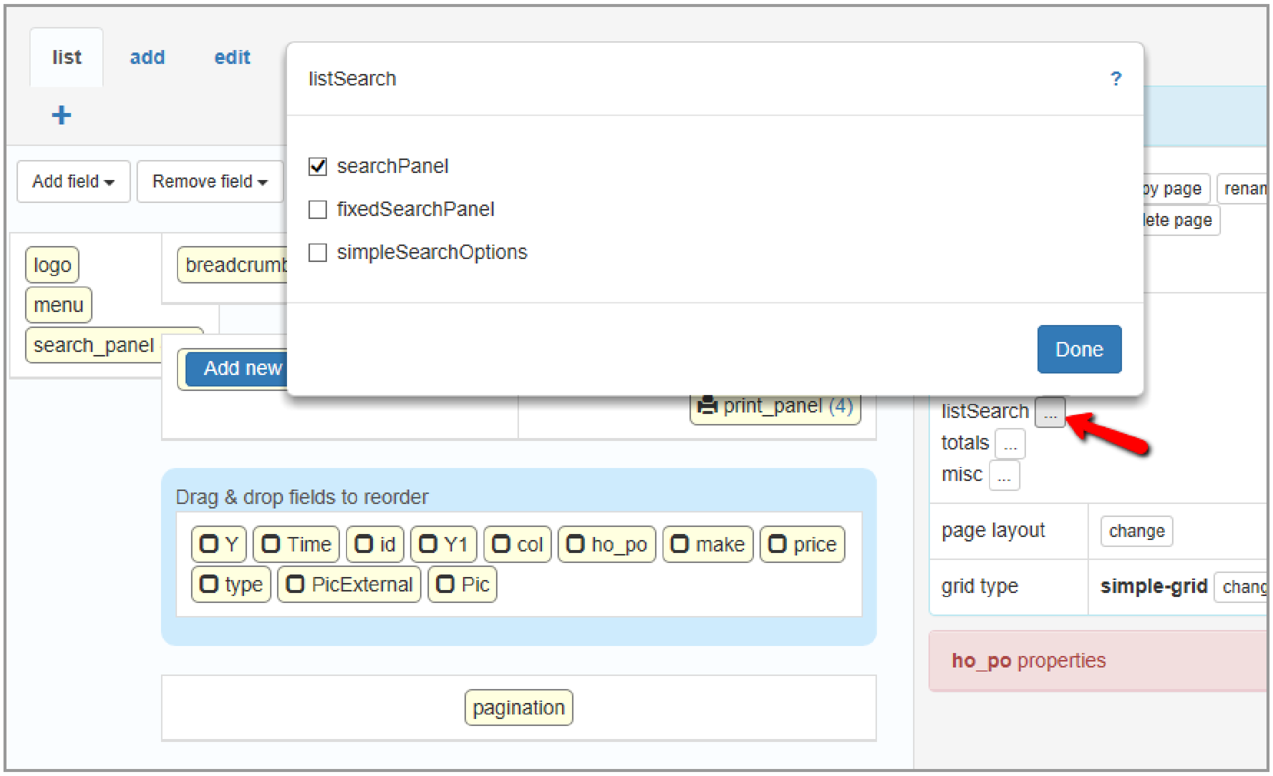


details link

linkType: single / individual / none (none selected)

Done

List search options



listSearch ?

☑ searchPanel

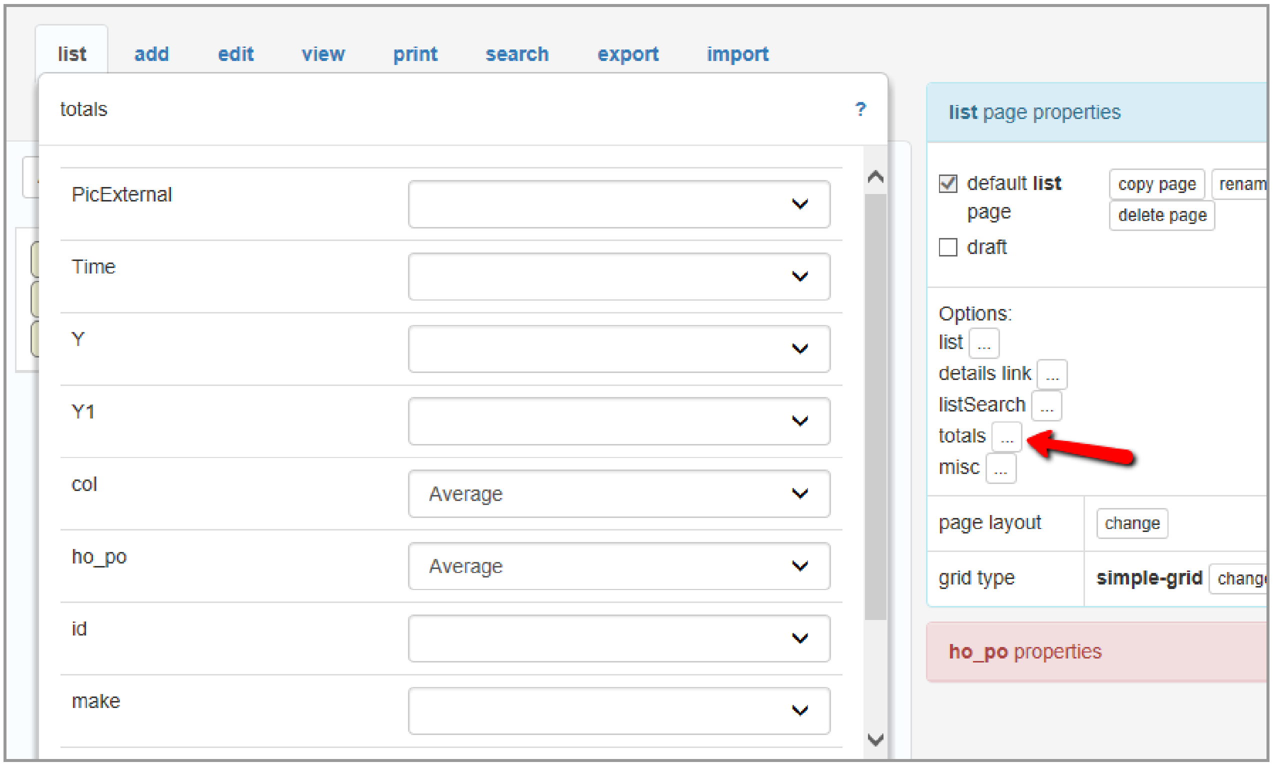
□ fixedSearchPanel

□ simpleSearchOptions

Done

Totals options

Click on this control when you want to add totals functions to the fields list page. You can also add the totals for each field in element's properties - Totals.



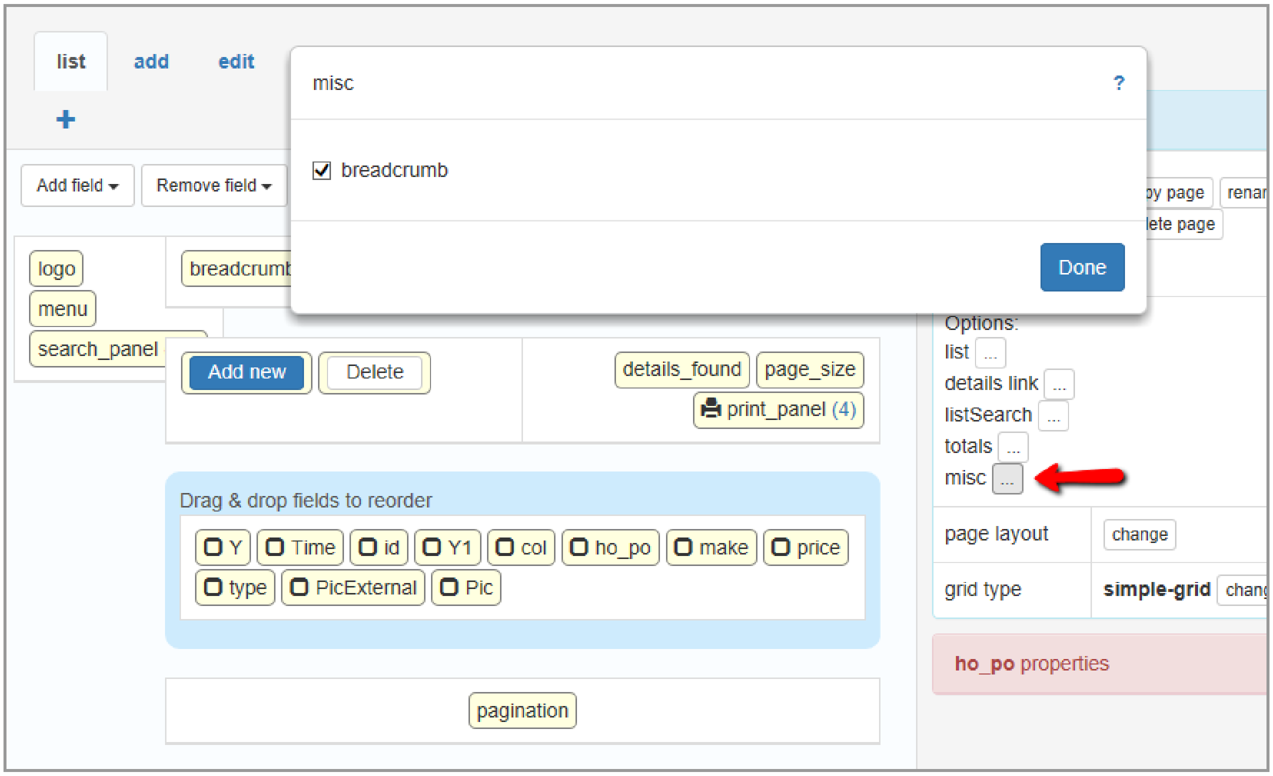
☑ default list page

□ draft

totals ?

PicExternal	
Time	
Y	
Y1	
col	Average
ho_po	Average
id	
make	

Misc options

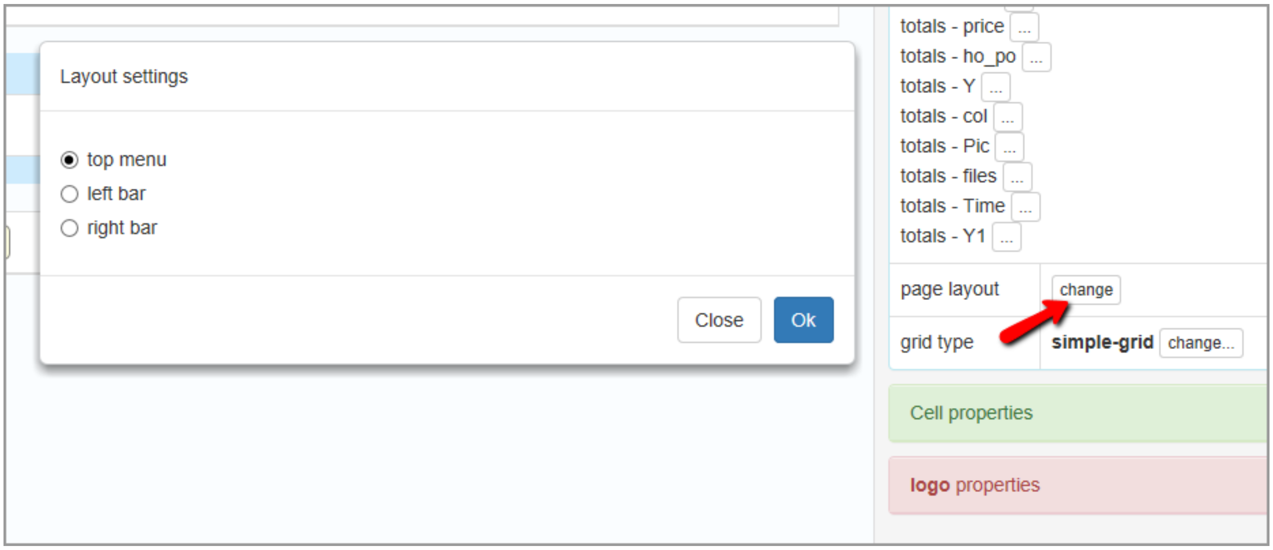


misc

☑ breadcrumb

Page layout

Press this button to open dialog with Layout settings available for the current page.

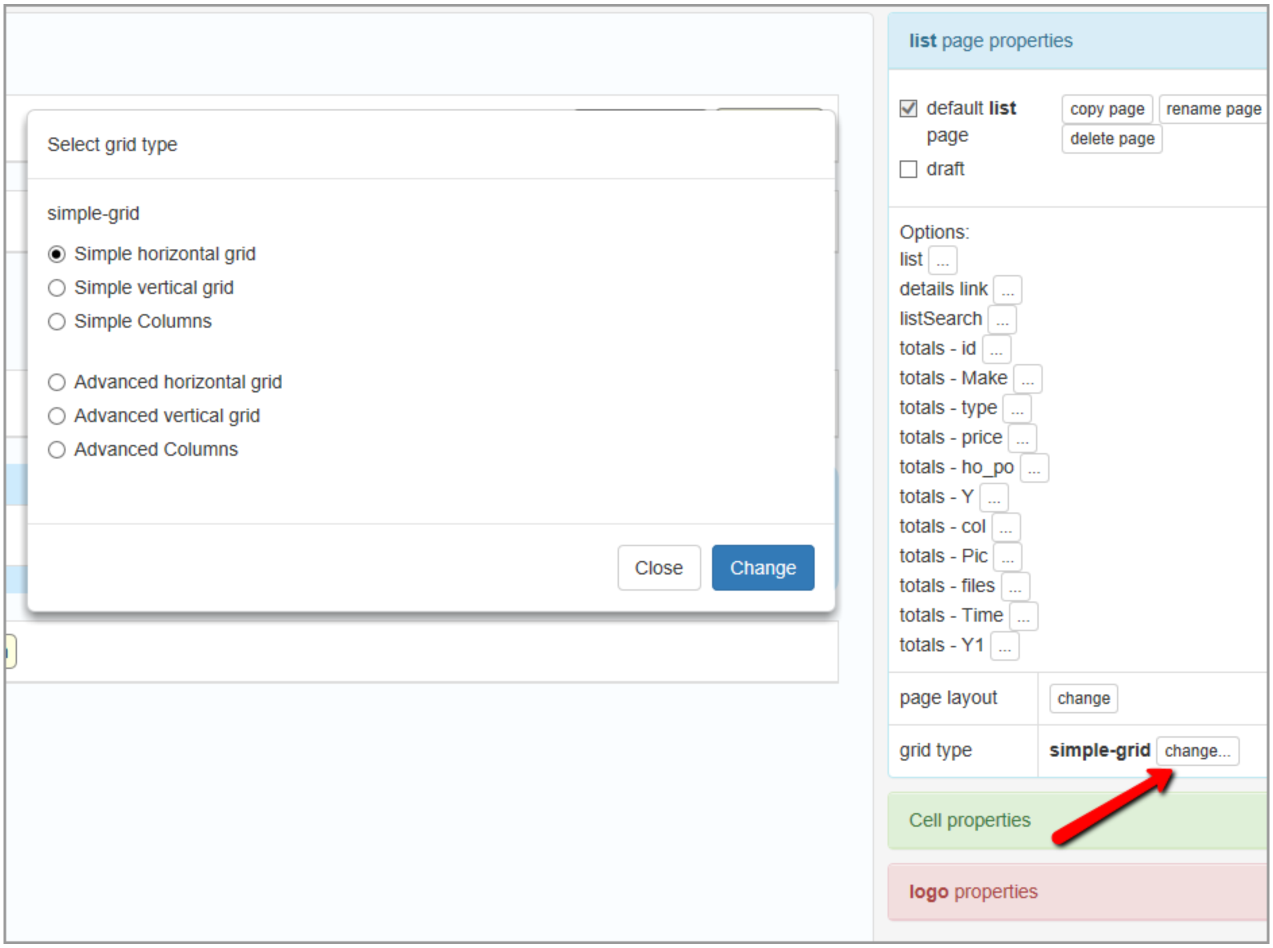


Layout settings

● top menu
○ left bar
○ right bar

Page grid type

Press this button to open dialog with Grid types available for the current page.



Select grid type

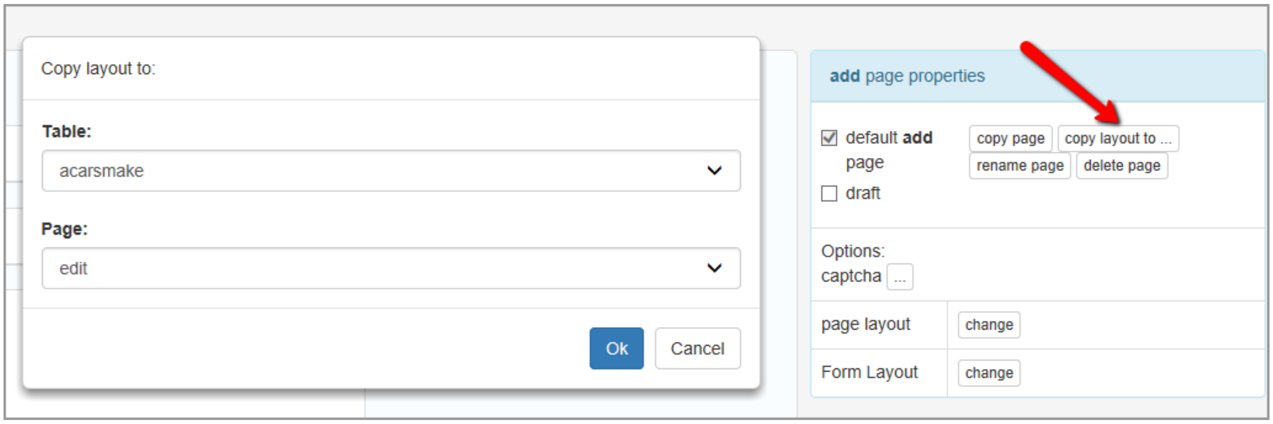
simple-grid

● Simple horizontal grid
○ Simple vertical grid
○ Simple Columns

○ Advanced horizontal grid
○ Advanced vertical grid
○ Advanced Columns

Copy layout to ...

This button is available for Add/Edit/View pages only. Opens the dialog where you can choose to which table/page the current layout should be copied to.



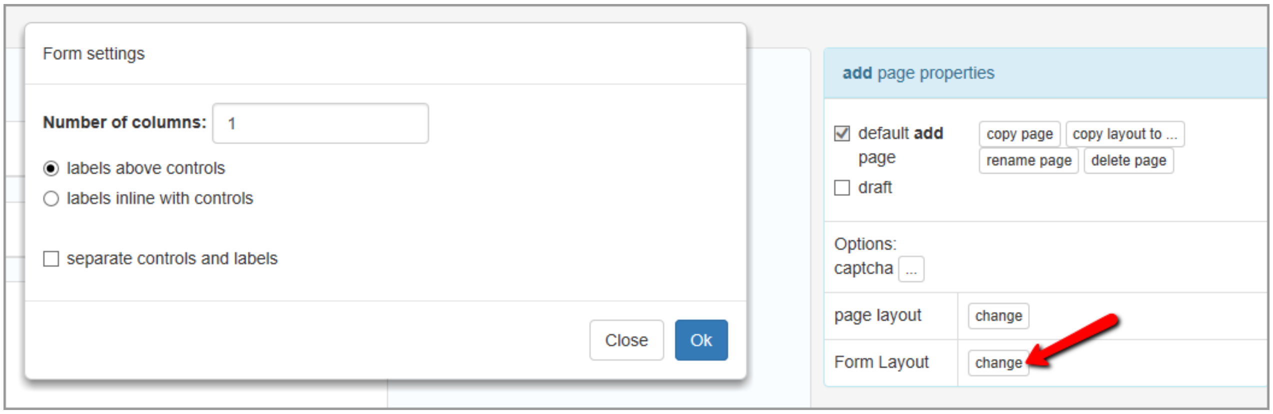
Copy layout to:

Table: acarsmake

Page: edit

Form layout

You can leave simple edit form or press the button "generate form" to set the numbers of columns on the Add/Edit page, labels position, and choose either separate controls and labels or no



Form settings

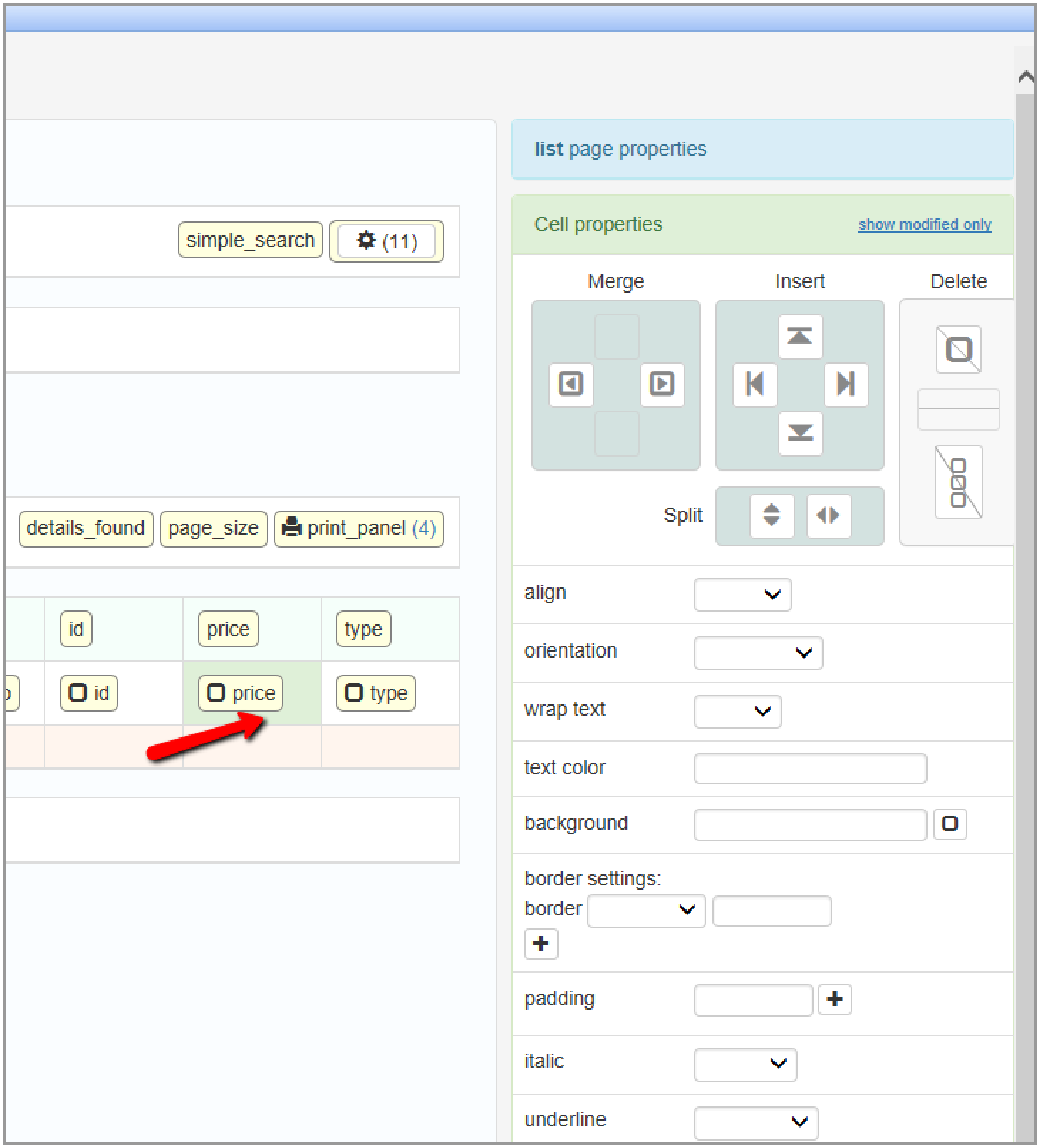
Number of columns: 1

● labels above controls
○ labels inline with controls

□ separate controls and labels

2.16.4 Working with cells

When you click on one of the cells in the Page Designer you will see the **Cell properties** to the right. Here you can manage settings that are available for the current cell.
You can click on two or more cells while holding CTRL button, in that case you'll see only properties that are available for all selected cells. Click again on one cell to disable multiple selection.



Merging cells

If you need to merge cells click on one of the cells - it should become highlighted and the Cells menu should appear to the right. Now press one of the Merge direction buttons.

Items that were inside the merging elements will be put into one cell.

Inserting cells

If you need to insert cells on the page click on one of the cells - it should become highlighted and the Cells menu should appear to the right. Now press one of the Insert direction buttons.

Deleting cells

If you need to delete cells on the page click on one of the cells - it should become highlighted and the Cells menu should appear to the right. Now press one of Delete buttons. You can delete a single cell, the whole row or the whole column.

Splitting cells

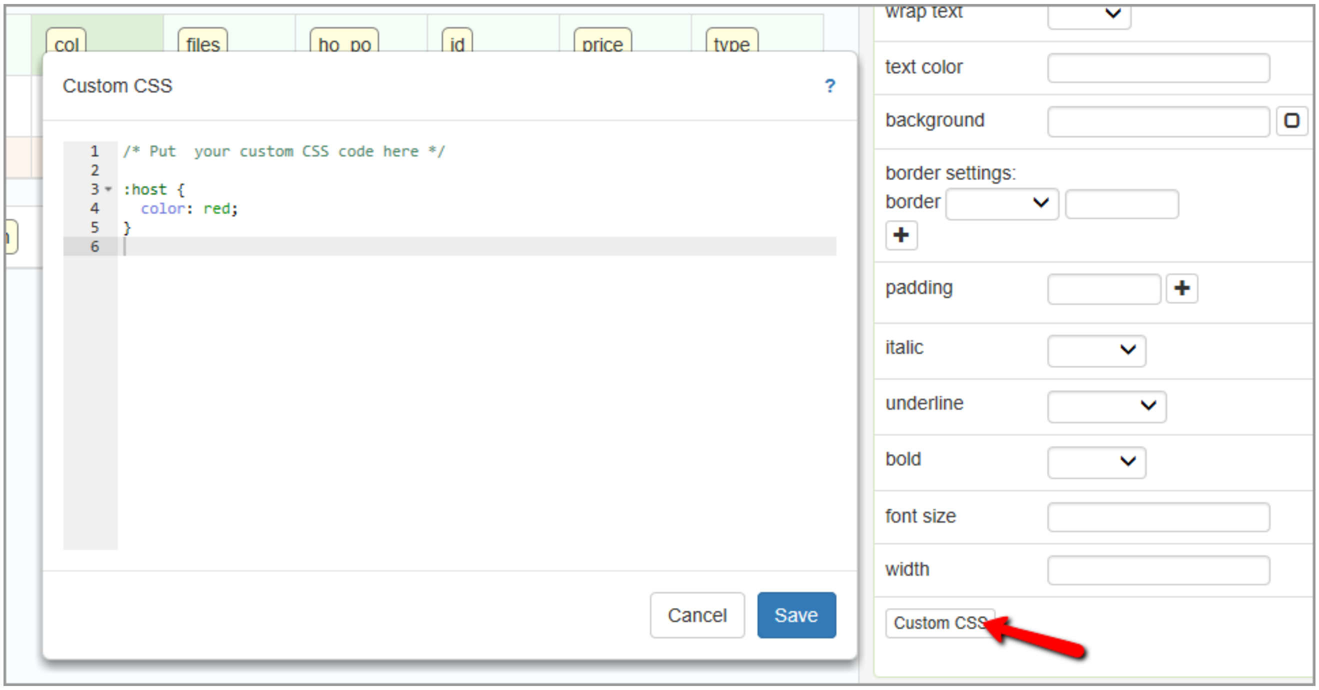
If you need to split cell on the page click on one of the cells - it should become highlighted and the Cells menu should appear to the right. Now press one of Split button. You can split a cell vertically or horizontally.

Formatting cells

To change the default formatting options click on one of the cells - it should become highlighted and the Cells properties should appear to the right. Now formatting controls to the right to change default settings.

Custom CSS

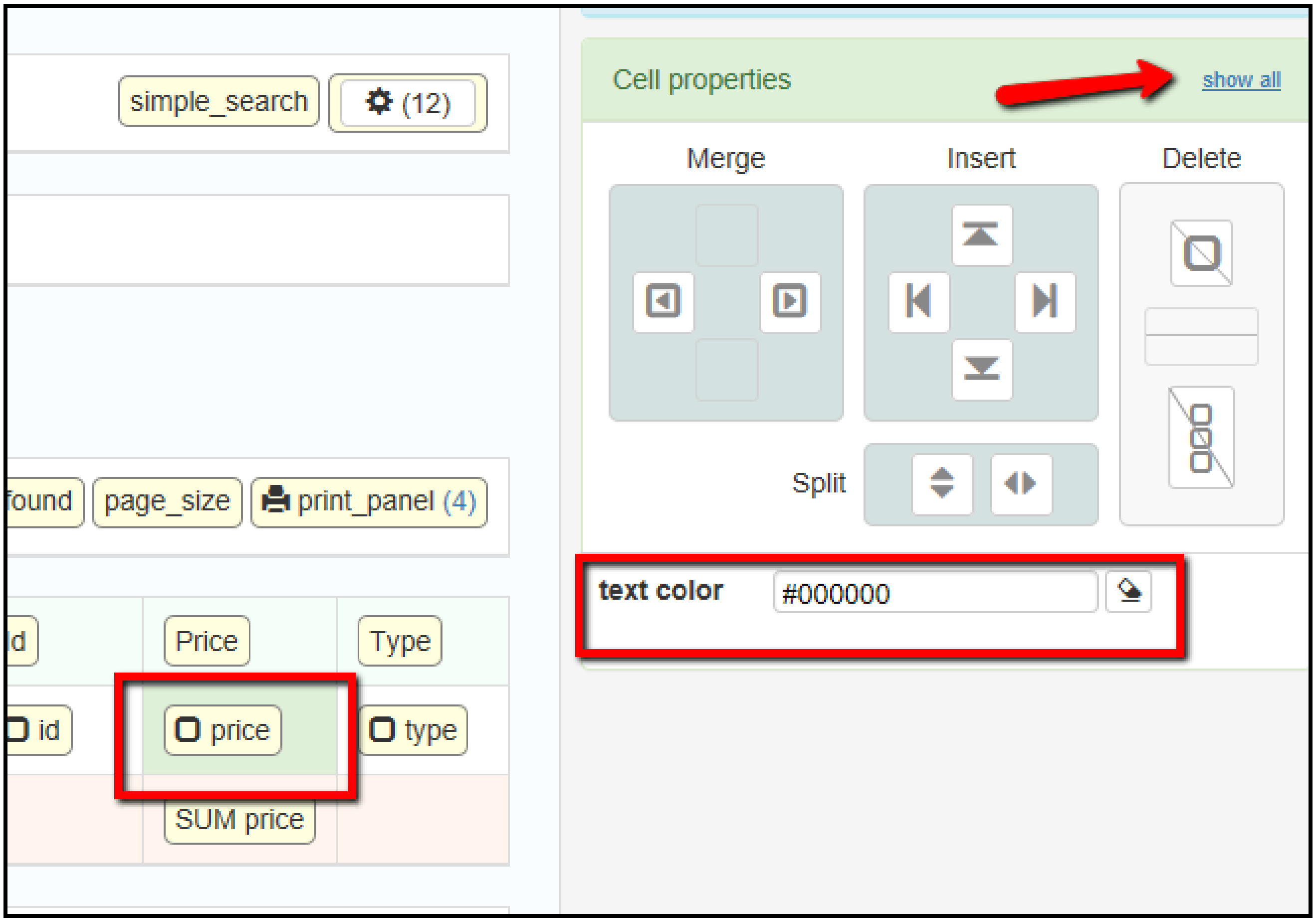
Press the **Custom CSS** button to open the dialog. here you can add the CSS code that should be applied to the cell. See more about Customizing CSS.



```
/* Put  your custom CSS code here */

:host {
  color: red;
}

```

You can change format for multiple cells at once. To do this press CTRL and click on multiple cells. Then change their format.

Press **"Show modified only"** link to see only modified cell properties.



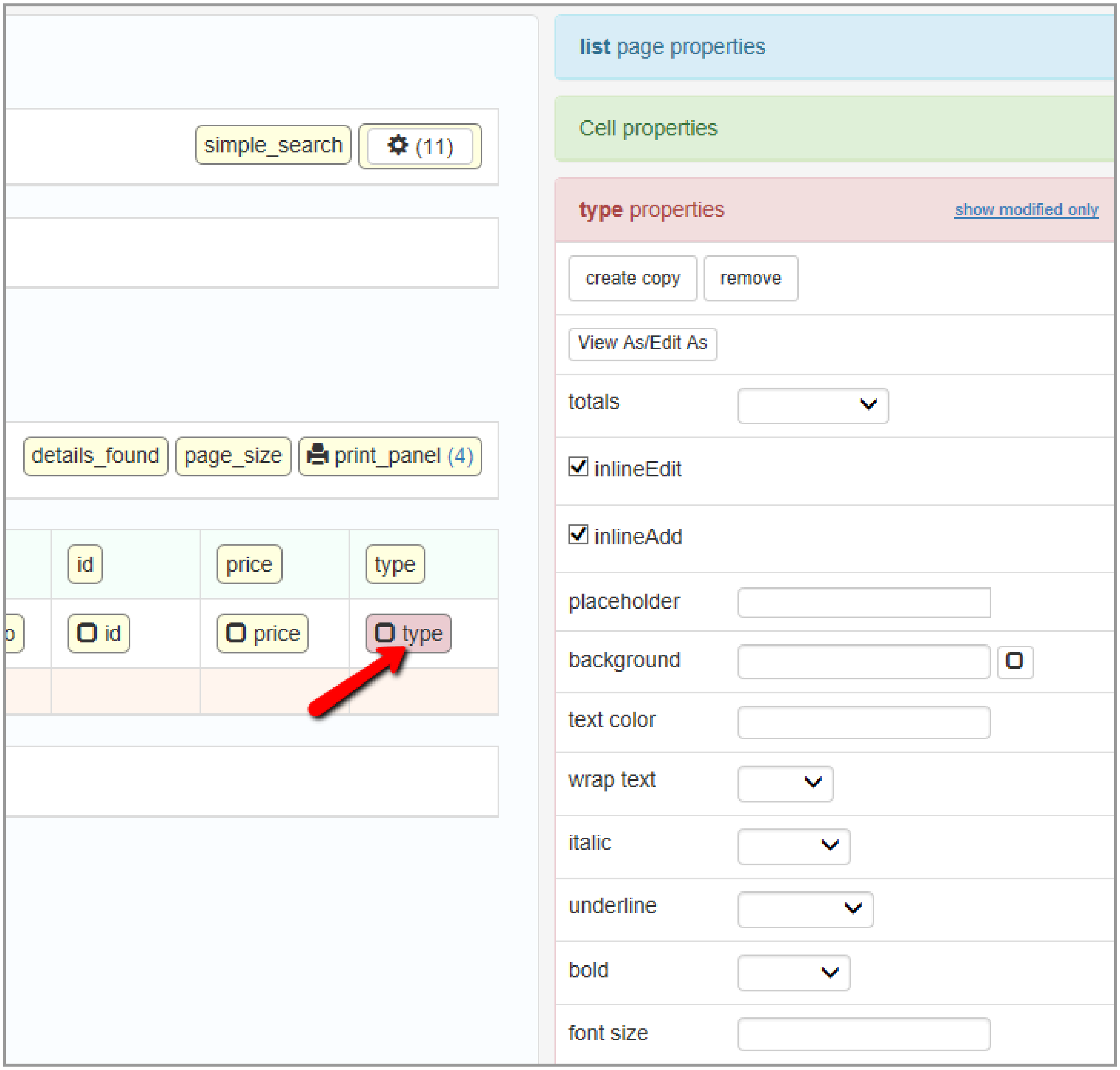
text color #000000

Press "**Show all**" to switch back to the default view.

2.16.5 Working with page elements

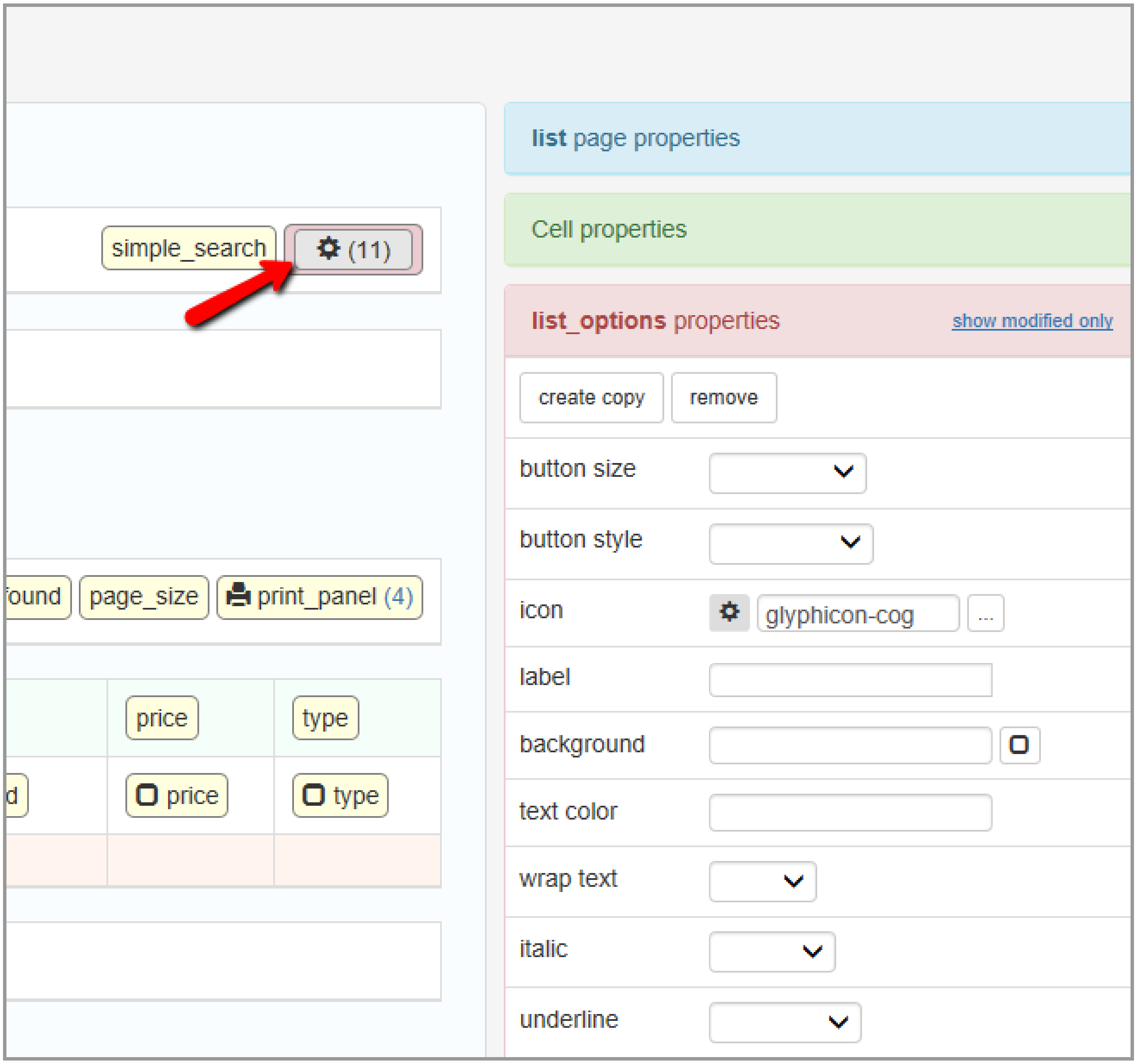
After clicking on element you can see its properties with settings that can be changed on the right part of the screen. You can also create a copy of an element or remove it.
You can also change the View As/Edit As settings for the element. To do this press the View As/Edit As button.

See more:
"View as" settings
"Edit as" settings



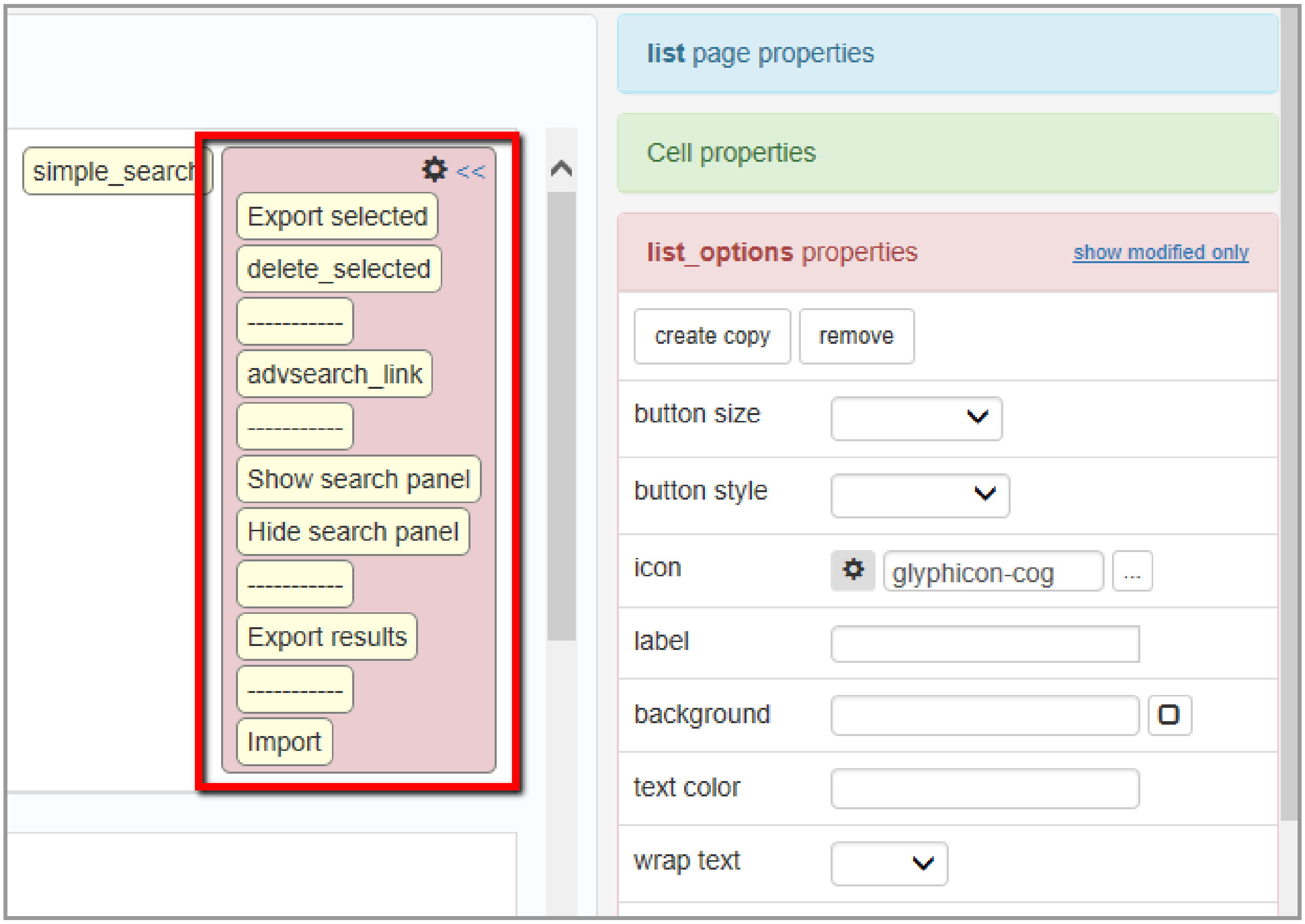
Different types of elements on the page have different properties.
Simple page elements like **logo, menu, simple search** and so on have only formatting options and Custom CSS.

Buttons have some additional properties like label, icon, button style and size.

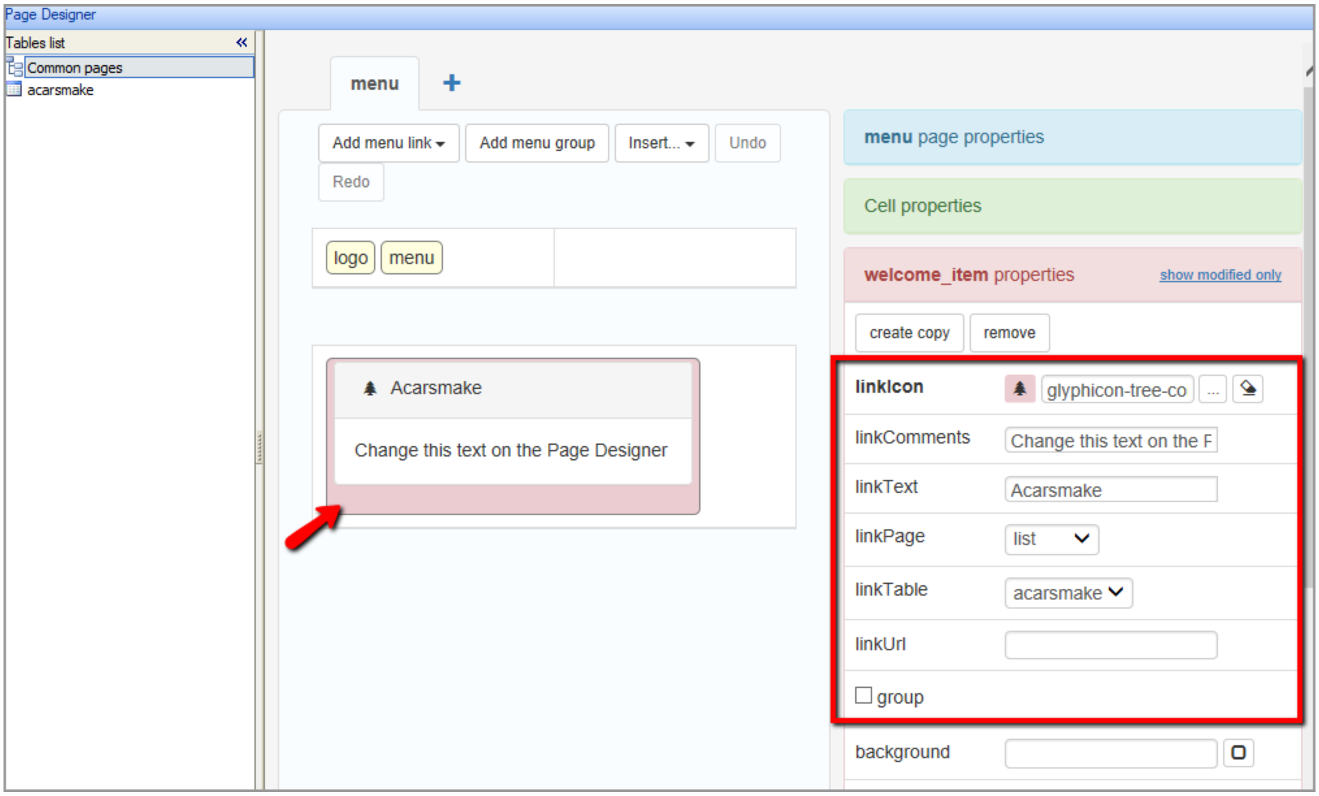


icon: glyphicon-cog

Some of the page elements like on the picture above are in fact groups of predefined simple elements. They have values in brackets, like search_panel(2) or print_panel(4). These numbers shows how much sub elements are inside this group. You can change settings for the whole group or for one of the element inside that group.



Elements on the menu page have some specific properties.



linkIcon glyphicon-tree-co

linkComments Change this text on the F

linkText Acarsmake

linkPage list

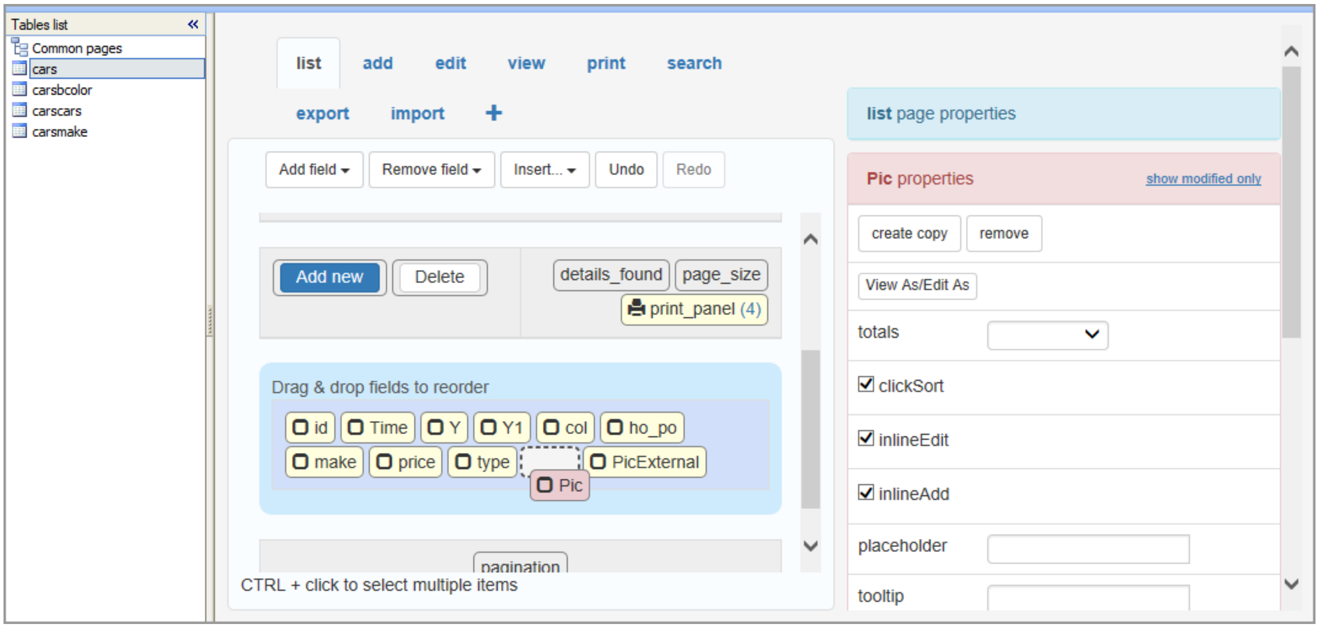
linkTable acarsmake

linkUrl

□ group

You can group/ungroup menu items if you need, change the link text, comments, and so on.

Table fields are also elements. To change fields order on specific page simply drag the field element to the new place.



CTRL + click to select multiple items

totals

☑ clickSort

☑ inlineEdit

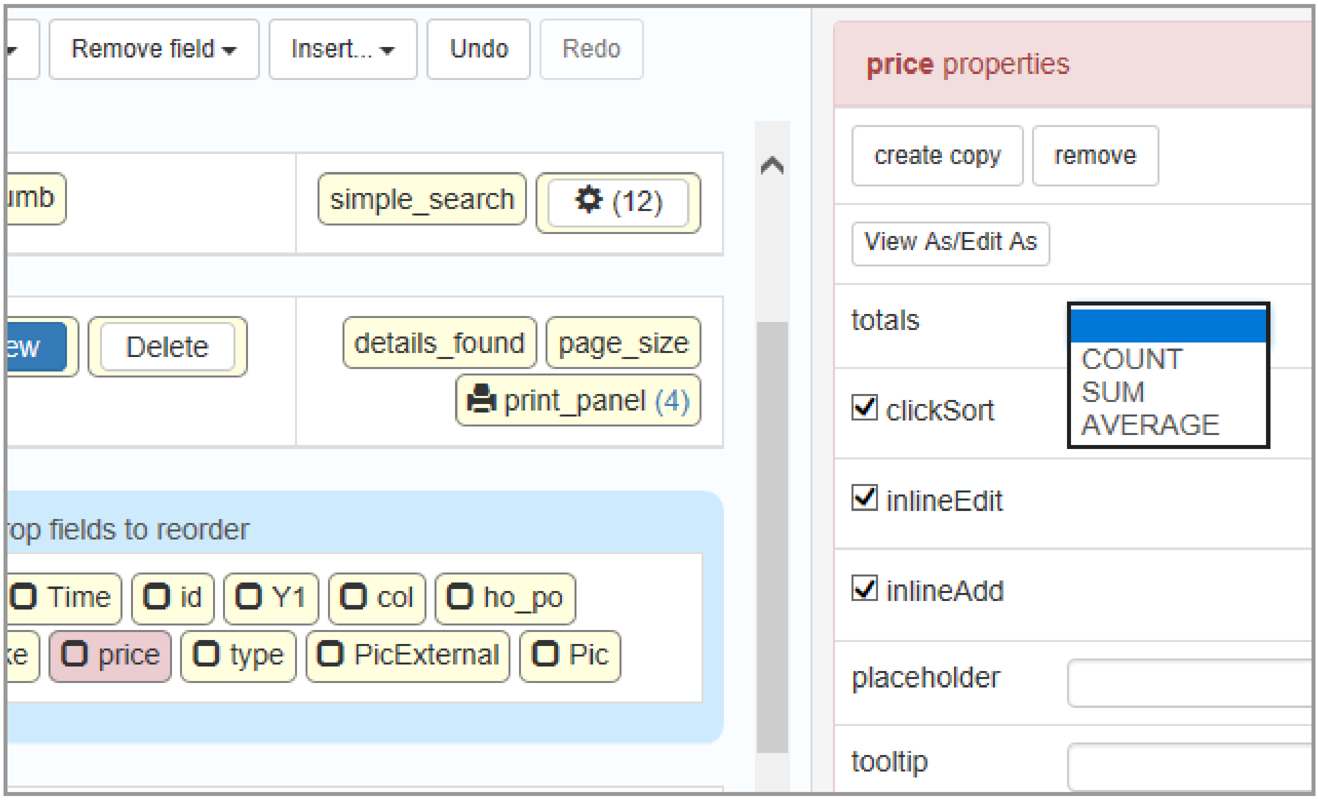
☑ inlineAdd

placeholder

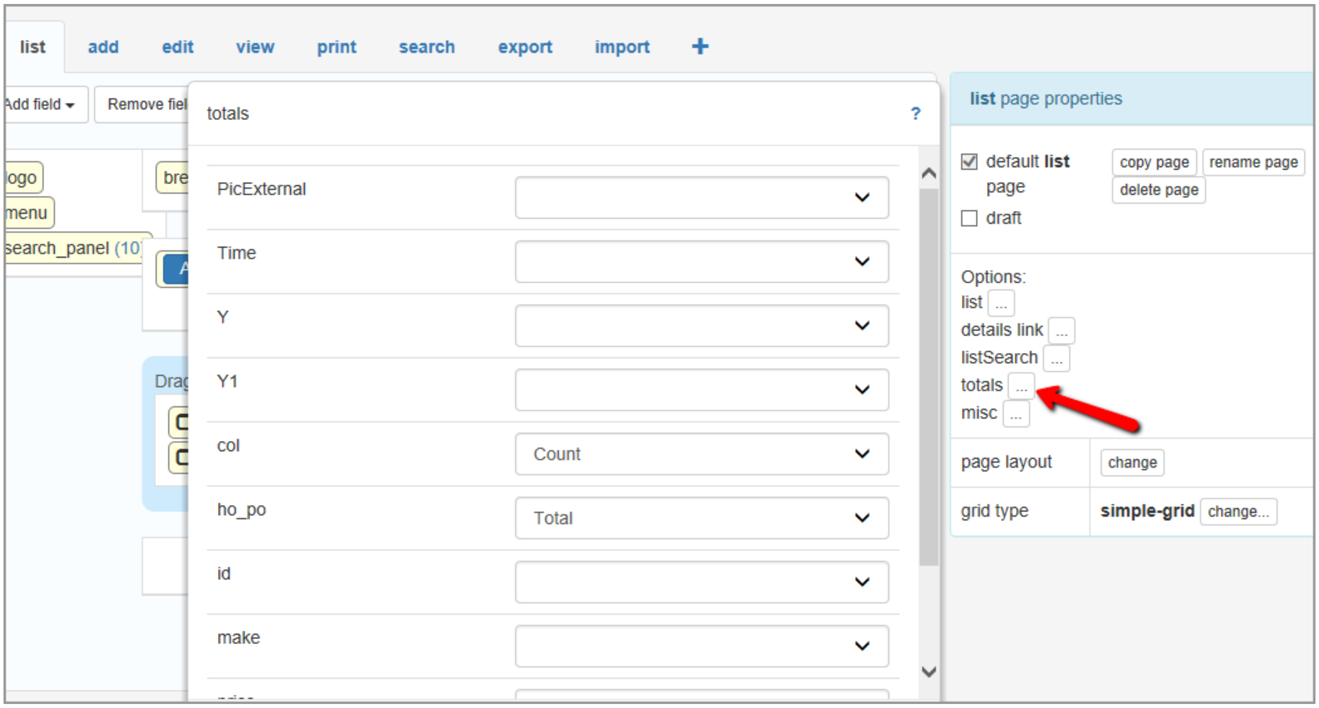
tooltip

Totals type

You can add totals calculation to some fields on the list/printer-friendly pages. Totals calculation is available with Simple horizontal and Advanced horizontal Grid type. You can choose between TOTAL, AVERAGE and COUNT aggregate functions. To do it select the field you want to add totals calculation to and in the properties on the right choose function type from **totals** drop-down box.



Another way of adding totals for fields is to click on the totals button in the list or print page properties. There you can choose totals functions for several fields at once.

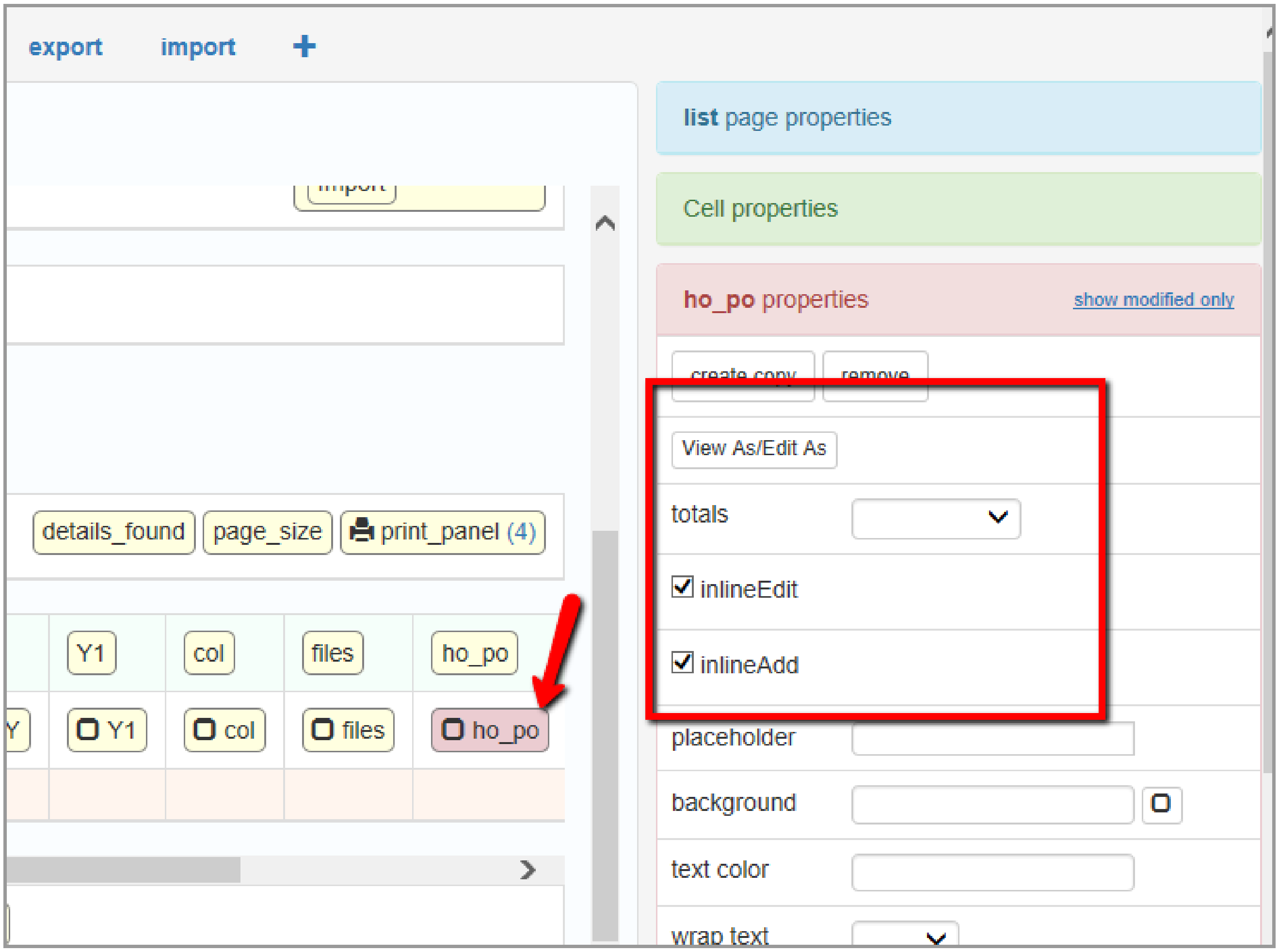


totals

PicExternal	
Time	
Y	
Y1	
col	Count
ho_po	Total
id	
make	

Grid elements have some special settings like tooltip, placeholder, Inline Add/Edit checkboxes. Changes in Inline Add/Edit made in the Page Designer would automatically apply on the Choose fields screen of the application.

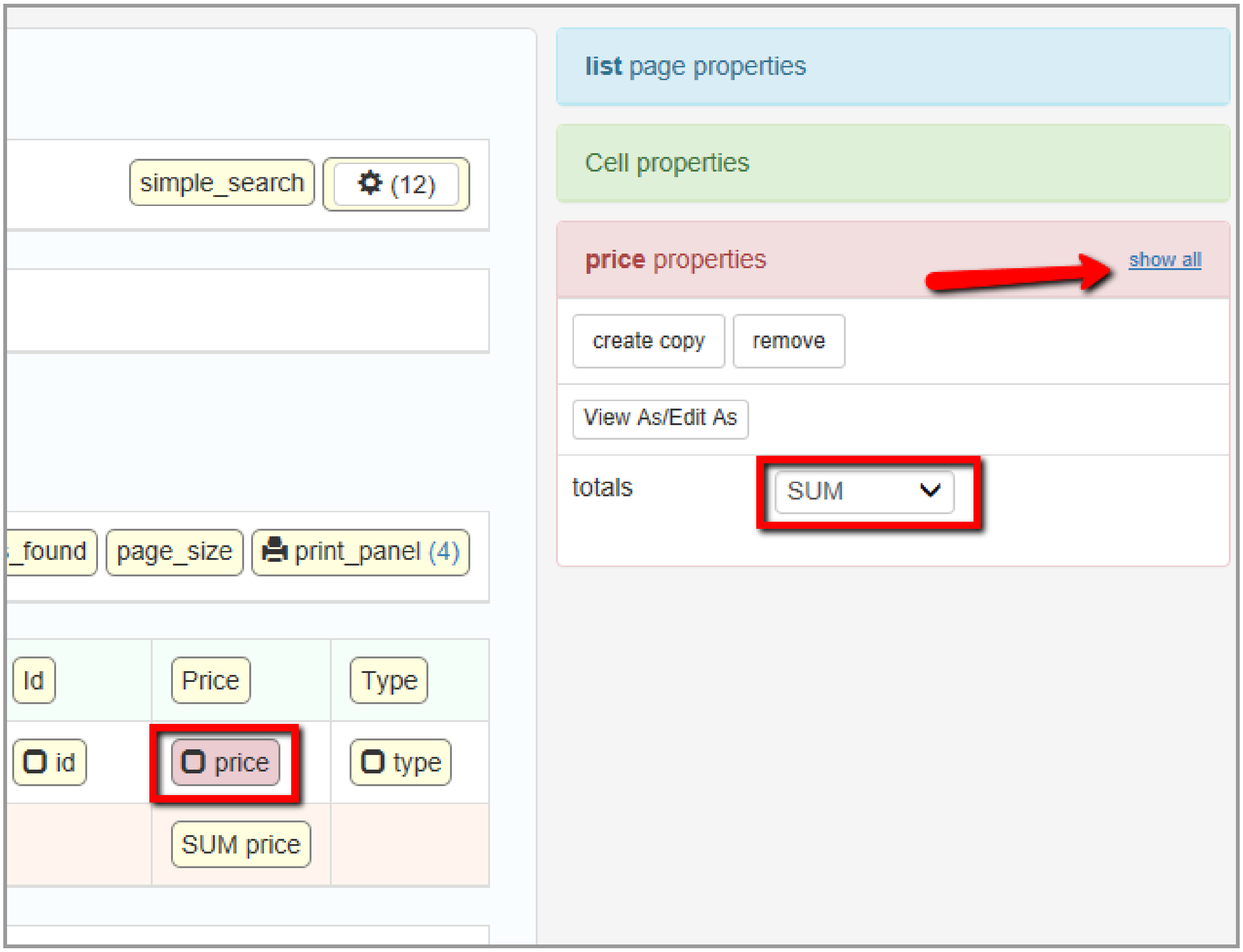
To change them click on the element in the grid, and then use controls that appear to the right. You can also access View As/Edit As settings and choose one of the calculated Totals from the right panel of a grid element.



☑ inlineEdit

☑ inlineAdd

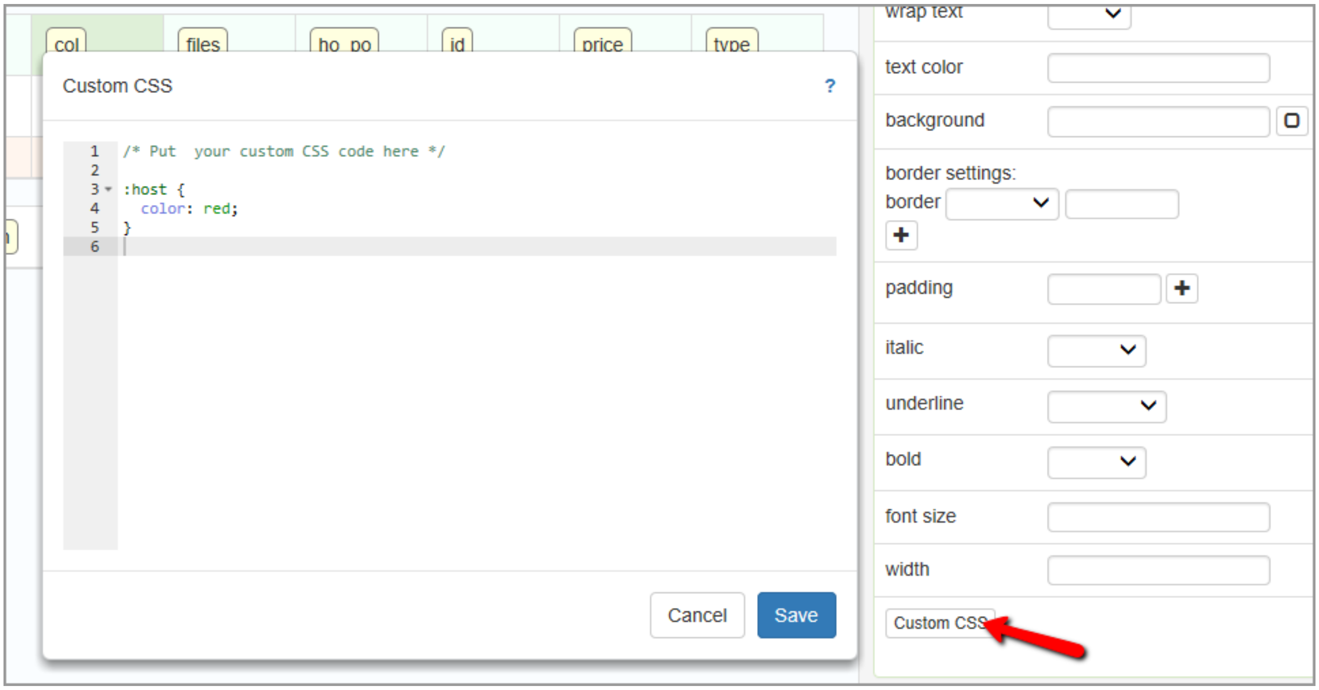
For Details icons in grid you can see "show records count", "badge-style record count" and "hide link when no child record exists" options.
Press **"Show modified only"** link to see only modified element properties.



Press "**Show all**" to switch back to the default view.
You can drag and drop element from one cell to another when you want to change its default position. Or in the grid to re-order the elements.

Custom CSS

Press the **Custom CSS** button to open the dialog. here you can add the CSS code that should be applied to the cell. See more about Customizing CSS.



```
/* Put  your custom CSS code here */

:host {
  color: red;
}
```

2.16.6 Page layout and grid type

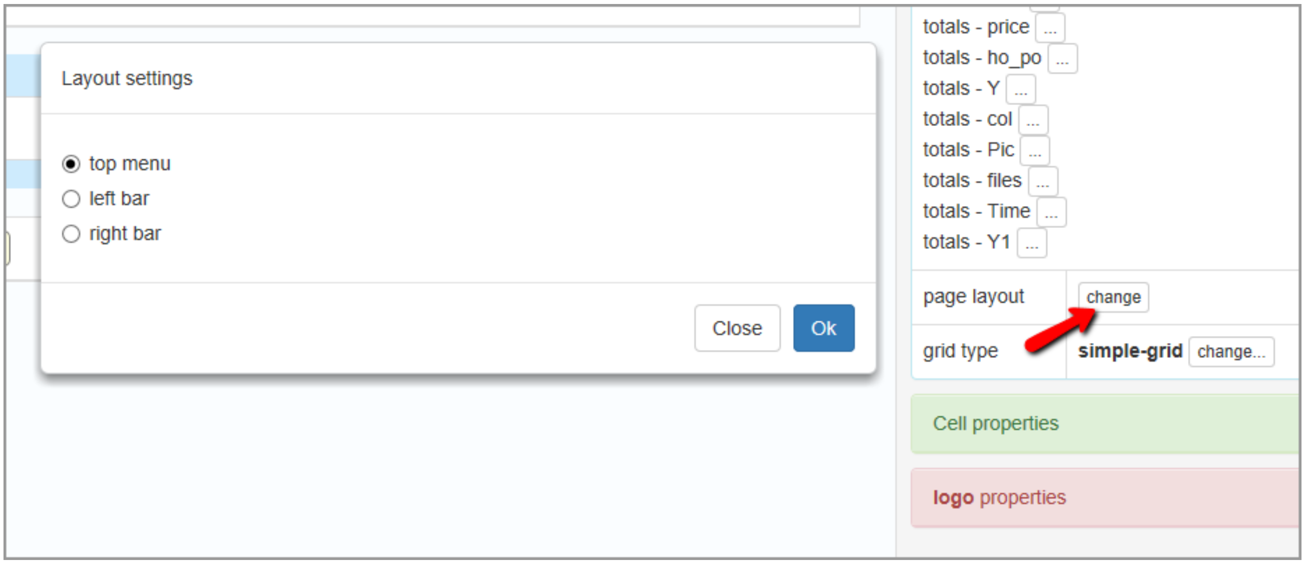
Quick jump

Page layout

Grid type

Page layout

Click **change** button on the List page properties **page layout** to select page layout:



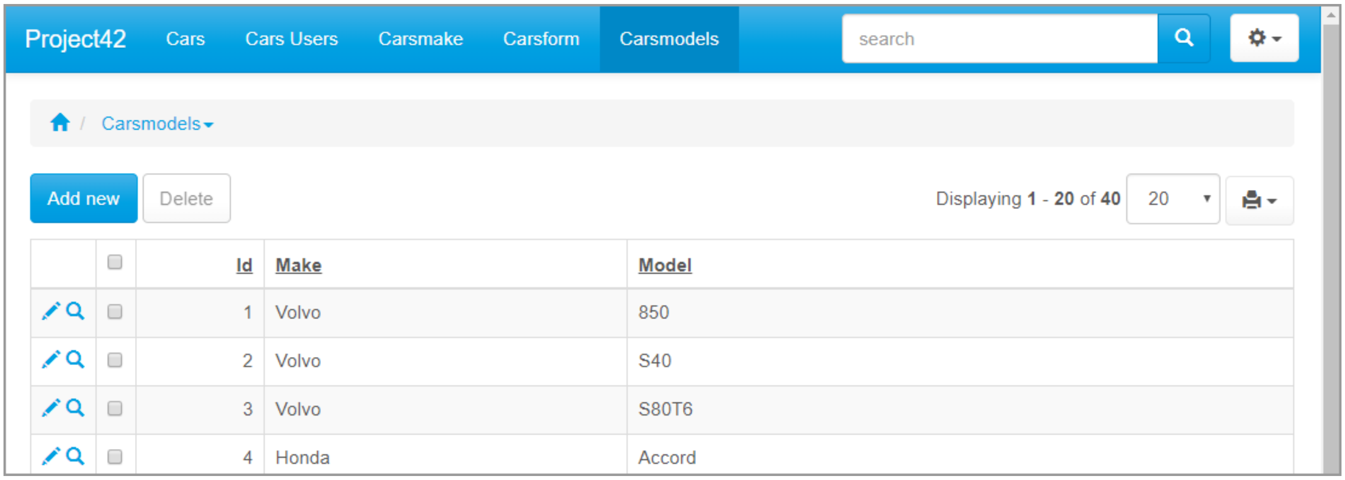
Layout settings

● top menu
○ left bar
○ right bar

totals - price ...
totals - ho_po ...
totals - Y ...
totals - col ...
totals - Pic ...
totals - files ...
totals - Time ...
totals - Y1 ...

page layout	change
grid type	**simple-grid** change...

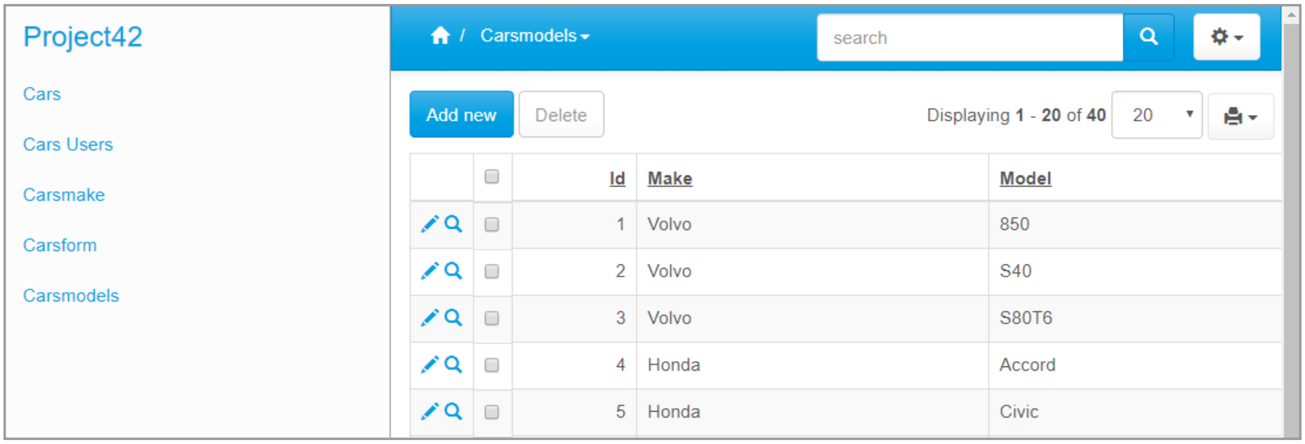
Example of top menu Page layout as seen in the browser:



Displaying **1** - **20** of **40** 20

	□	Id	Make	Model
	□	1	Volvo	850
	□	2	Volvo	S40
	□	3	Volvo	S80T6
	□	4	Honda	Accord

Example of left bar:



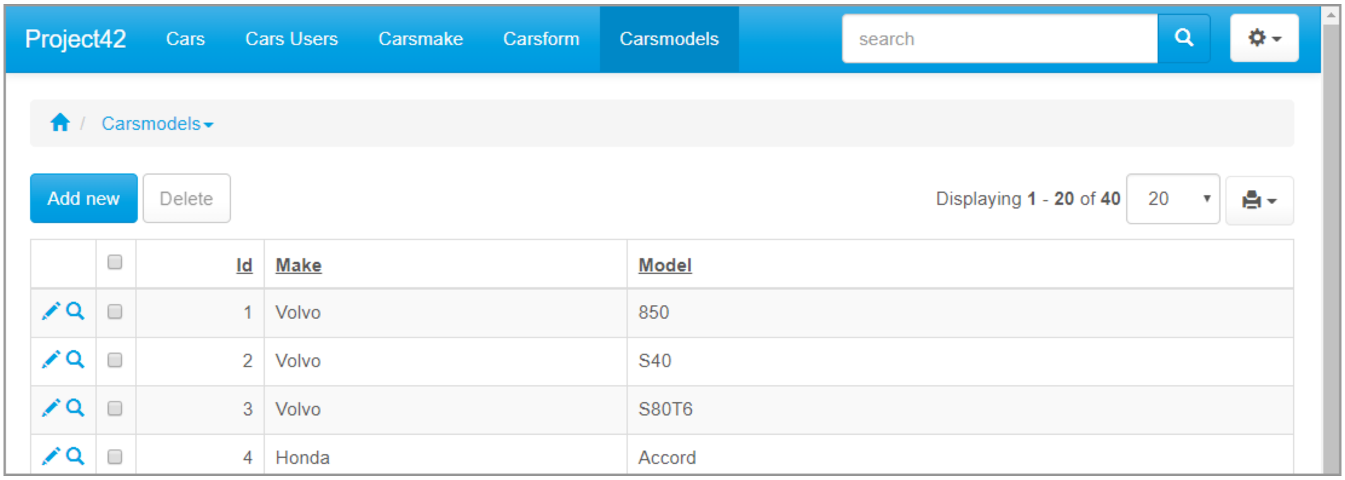
Id	Make	Model
1	Volvo	850
2	Volvo	S40
3	Volvo	S80T6
4	Honda	Accord
5	Honda	Civic

Example of right bar:

Grid type

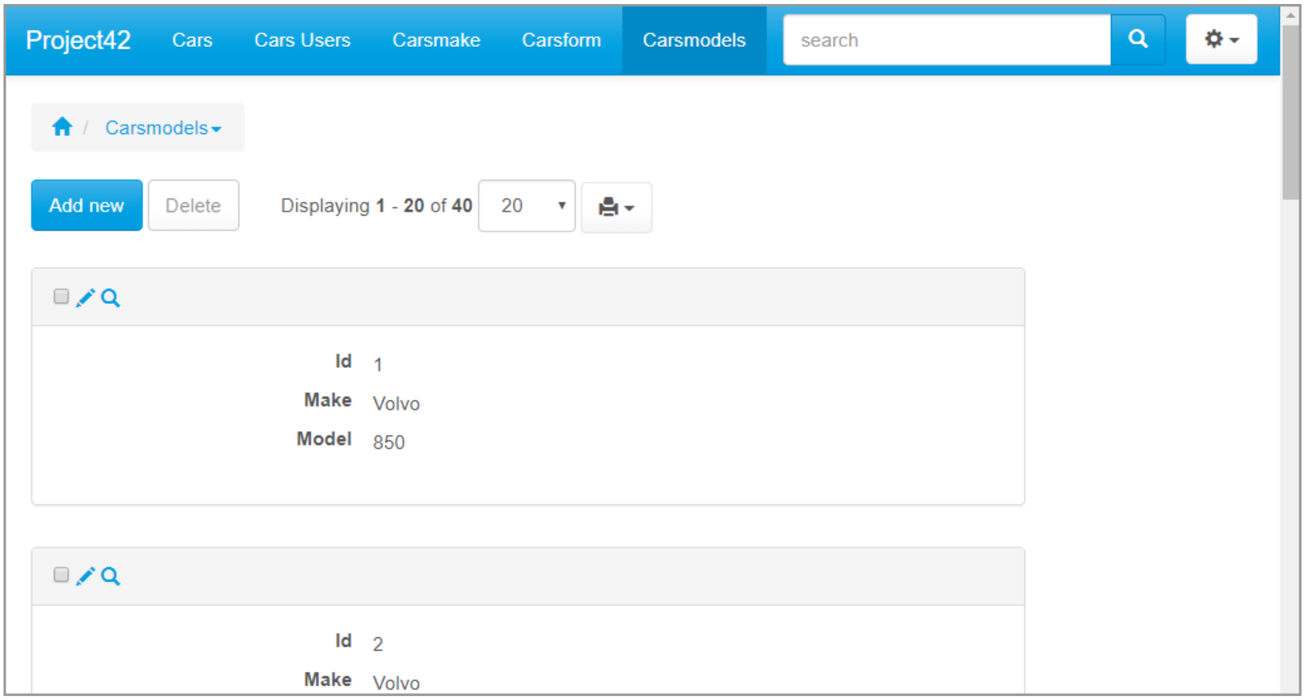
Page Designer allows to flexible change the appearance of the grid type. You can choose one of three simple types: horizontal, vertical or columns grid and one of three advanced modes. In the advanced mode you can easily change properties for labels or cells of the grid when you need to set some specific formatting. If you don't change anything in advanced mode the generated pages will look the same as in the simple mode.

Example of simple horizontal grid as seen in the browser:



Id	Make	Model
1	Volvo	850
2	Volvo	S40
3	Volvo	S80T6
4	Honda	Accord

Example of simple vertical grid:

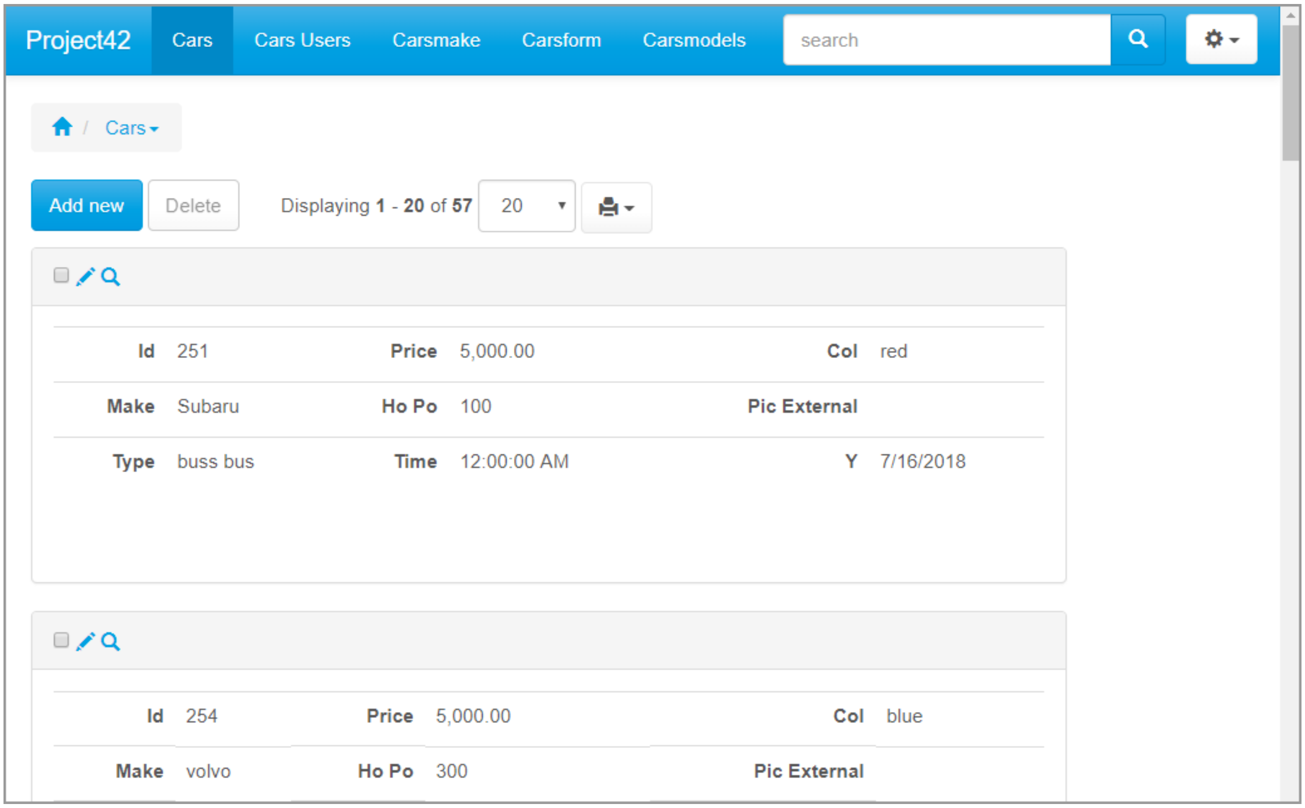


Displaying **1** - **20** of **40**

Id 1
Make Volvo
Model 850

Id 2
Make Volvo

Example of simple columns grid with 3 number of columns:



Displaying **1** - **20** of **57**

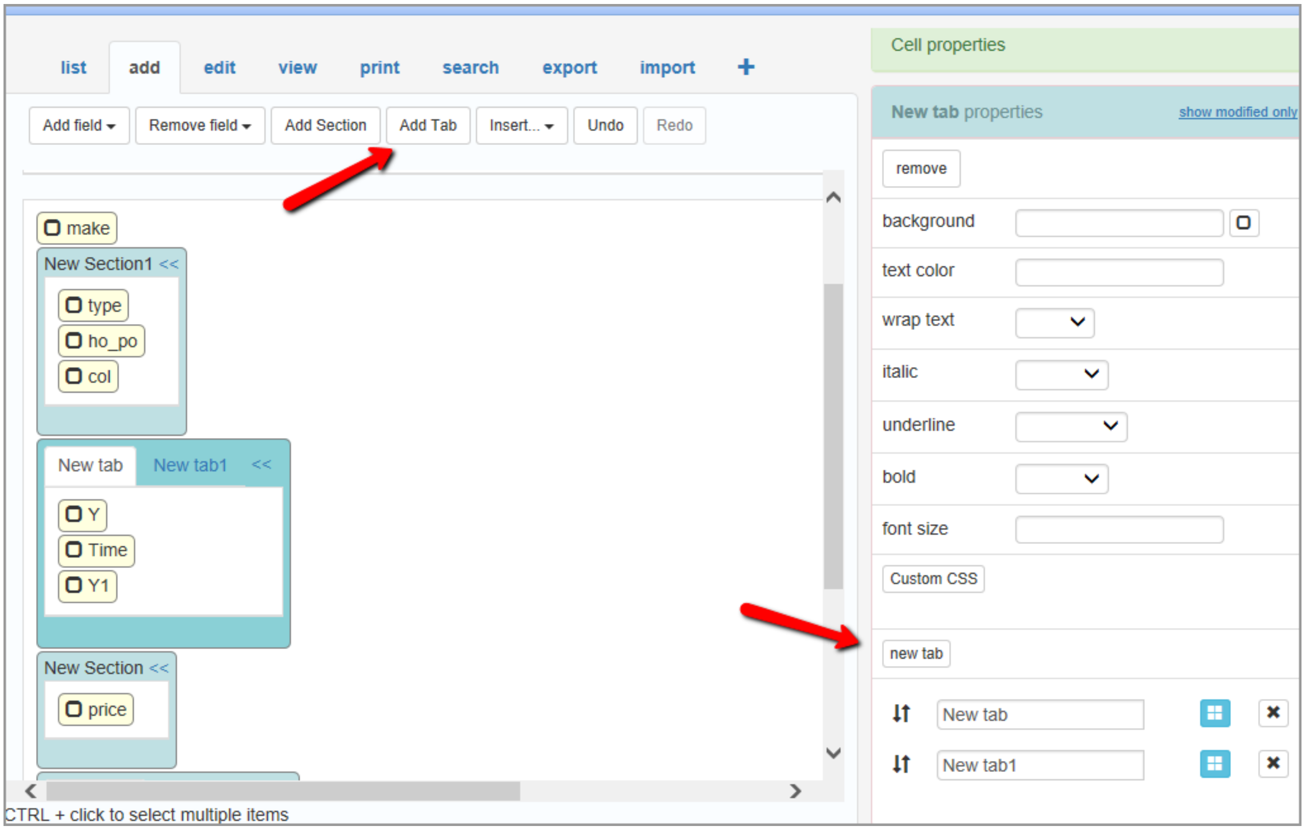
Id	251	**Price**	5,000.00	**Col**	red
Make	Subaru	**Ho Po**	100	**Pic External**	
Type	buss bus	**Time**	12:00:00 AM	**Y**	7/16/2018

Id	254	**Price**	5,000.00	**Col**	blue
Make	volvo	**Ho Po**	300	**Pic External**	

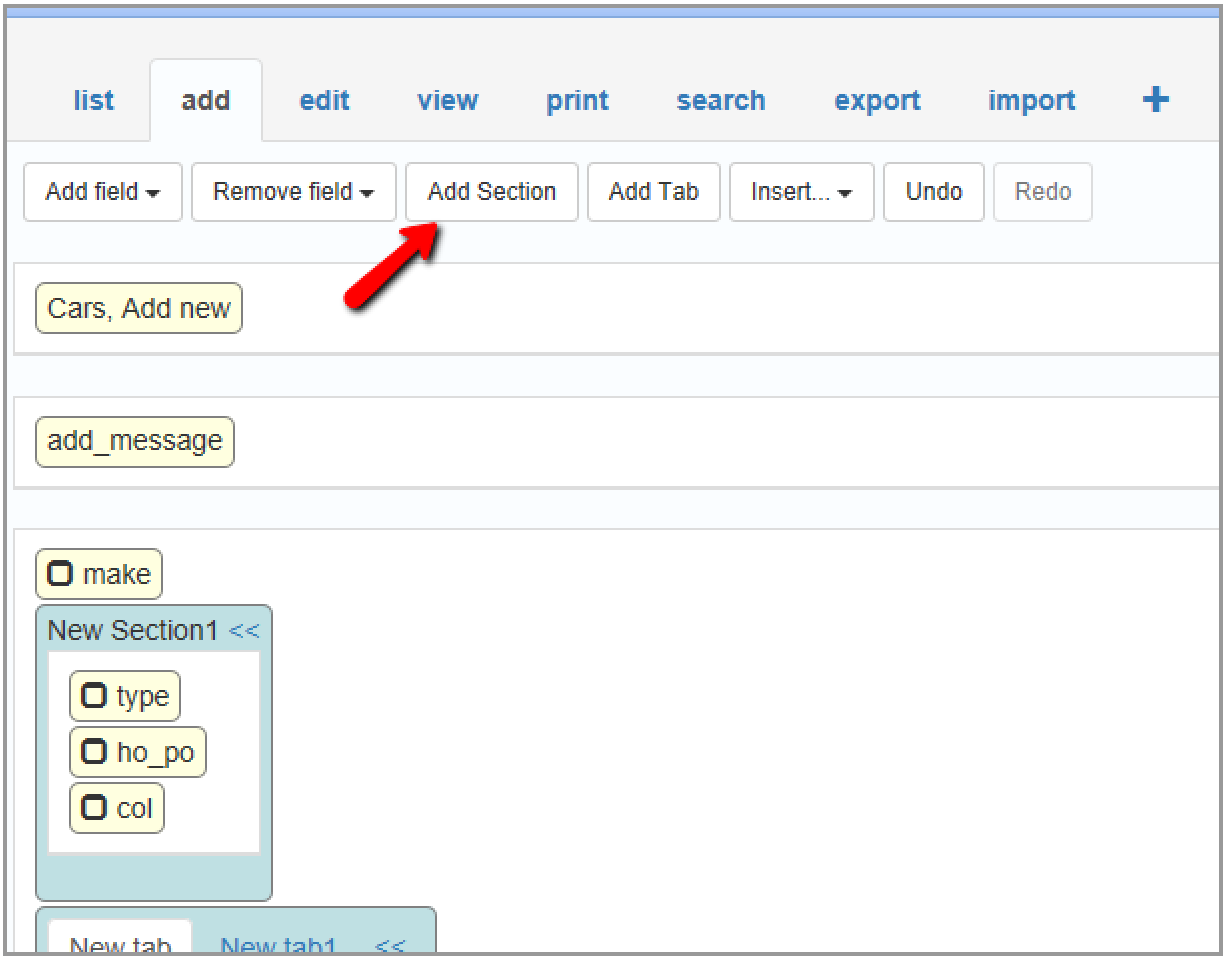
2.16.7 Tabs and sections

Tabs/Folding sections

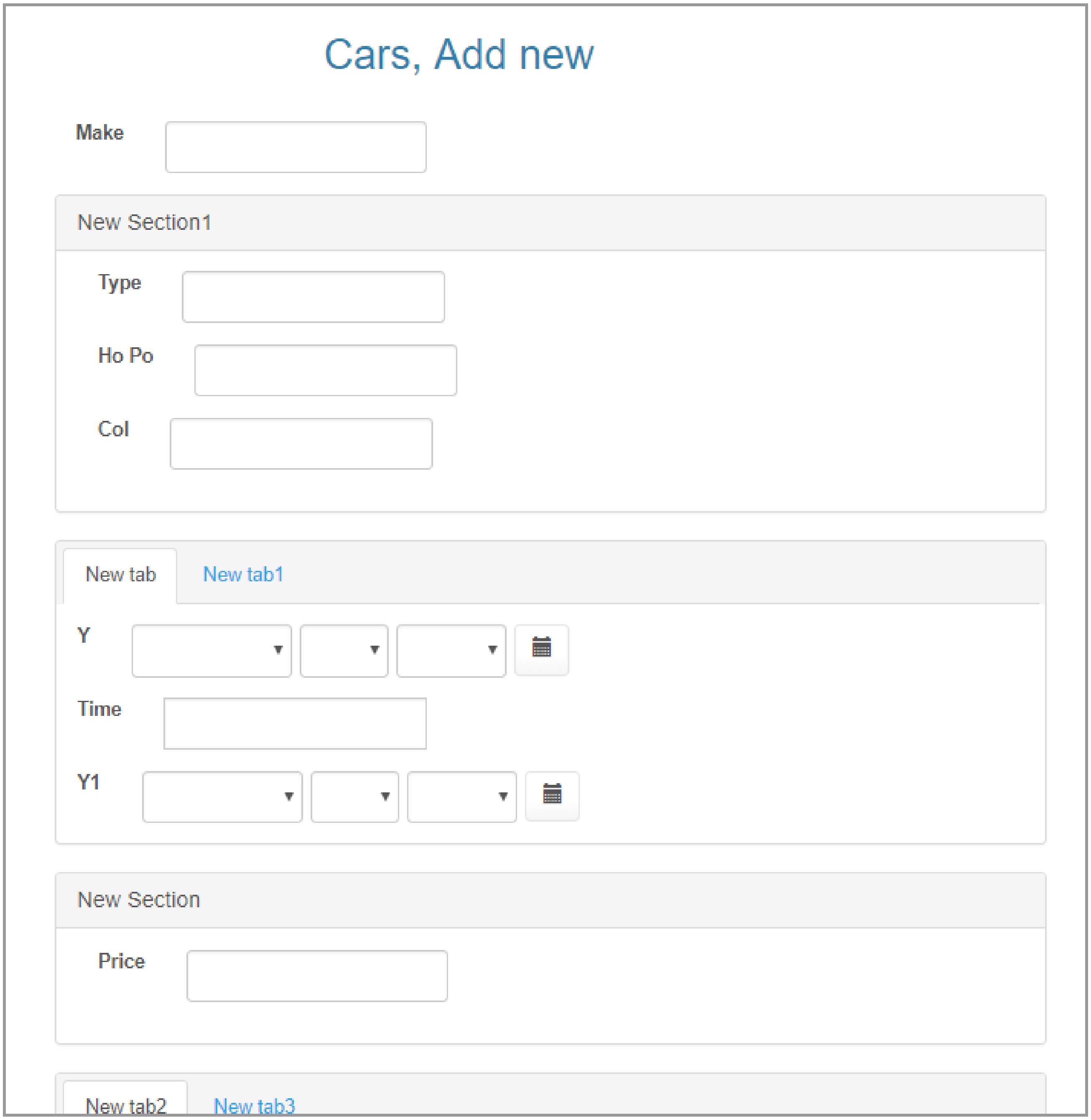
You can use tabs and folding sections on the Add/Edit/View pages. To create the new tabs group press the **Add tab** button on the top, then drag fields into them or when you need to re-order them. Press the **new tab** in the tab properties to add the new tab to the group.



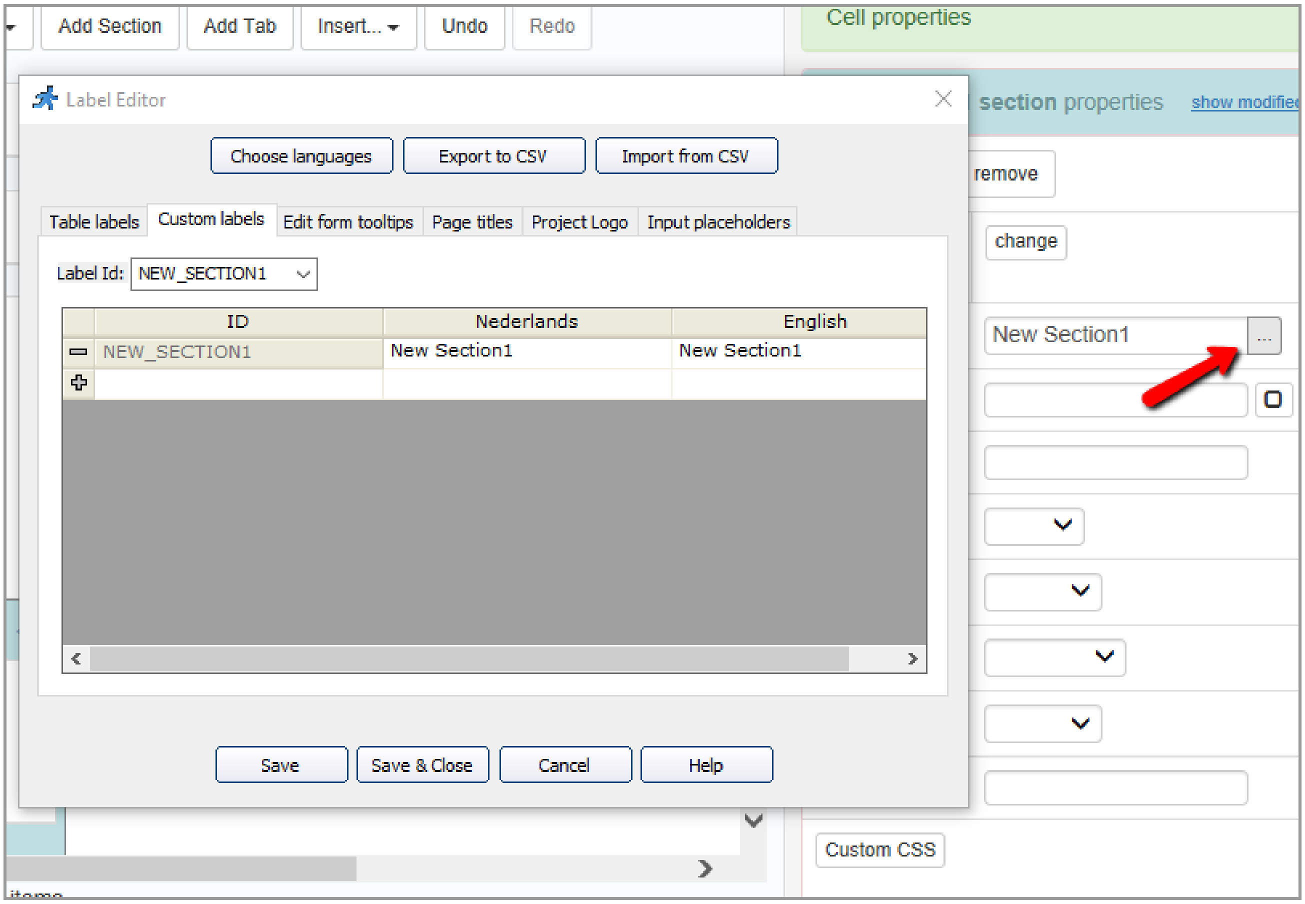
To create the new section use the **Add Section** button.



Here is how it looks on the generated Add page:



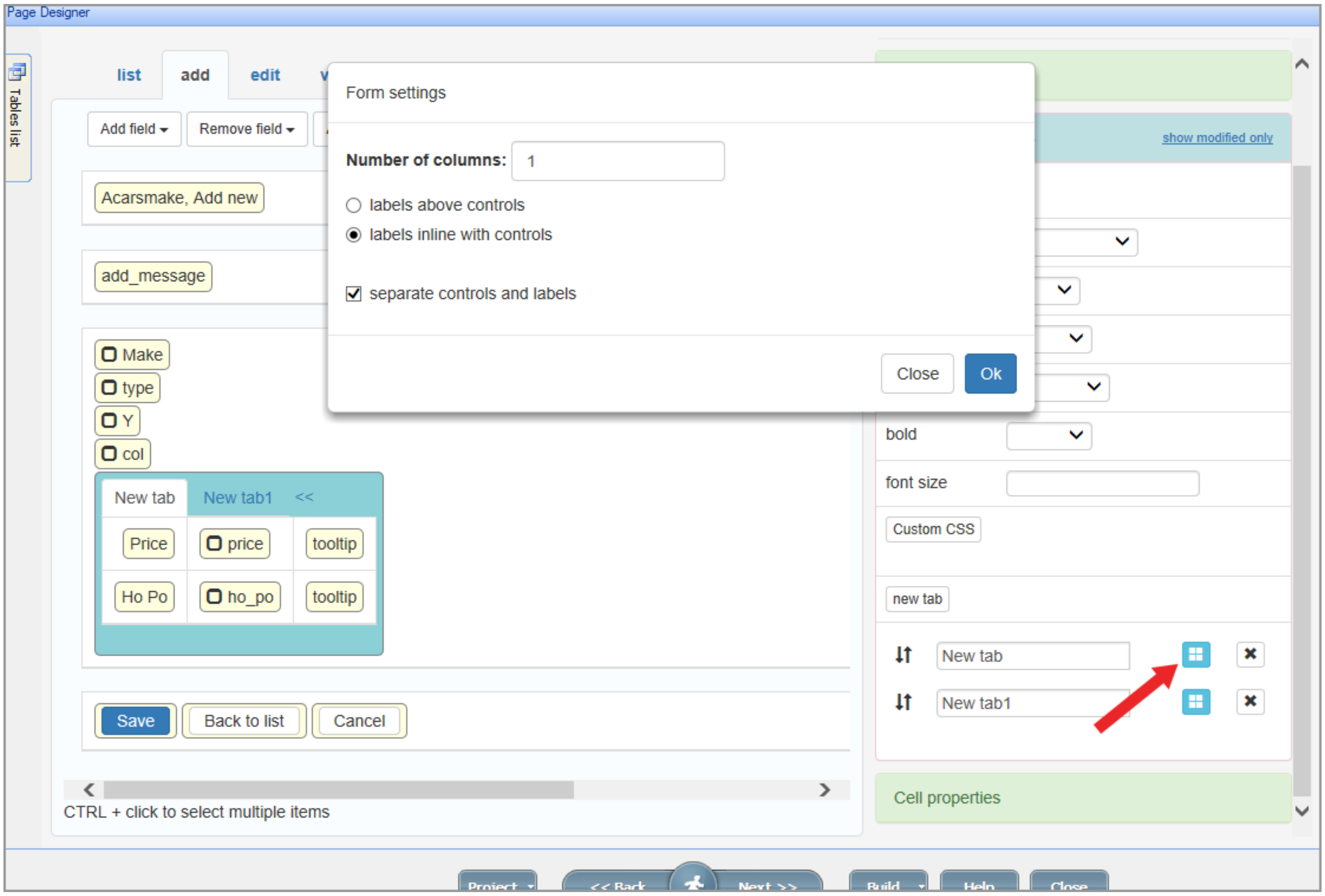
If you select several project languages in the Miscellaneous settings, you can translate the names of tabs and sections. Click the **...** button to open Label Editor and translate the names.



Label Id: NEW_SECTION1

ID	Nederlands	English
NEW_SECTION1	New Section1	New Section1

When working with tabs and sections on Add/Edit/View pages you can also change the way the labels and field values are shown - in columns, above the controls or inline with them.
You can also separate controls and labels to apply some custom formatting. This can be done for each Tab or section separately.
To do this press the Form Layout button then choose the desired settings for the Tab or Section.



Form settings

Number of columns: 1

○ labels above controls

● labels inline with controls

☑ separate controls and labels

Close Ok

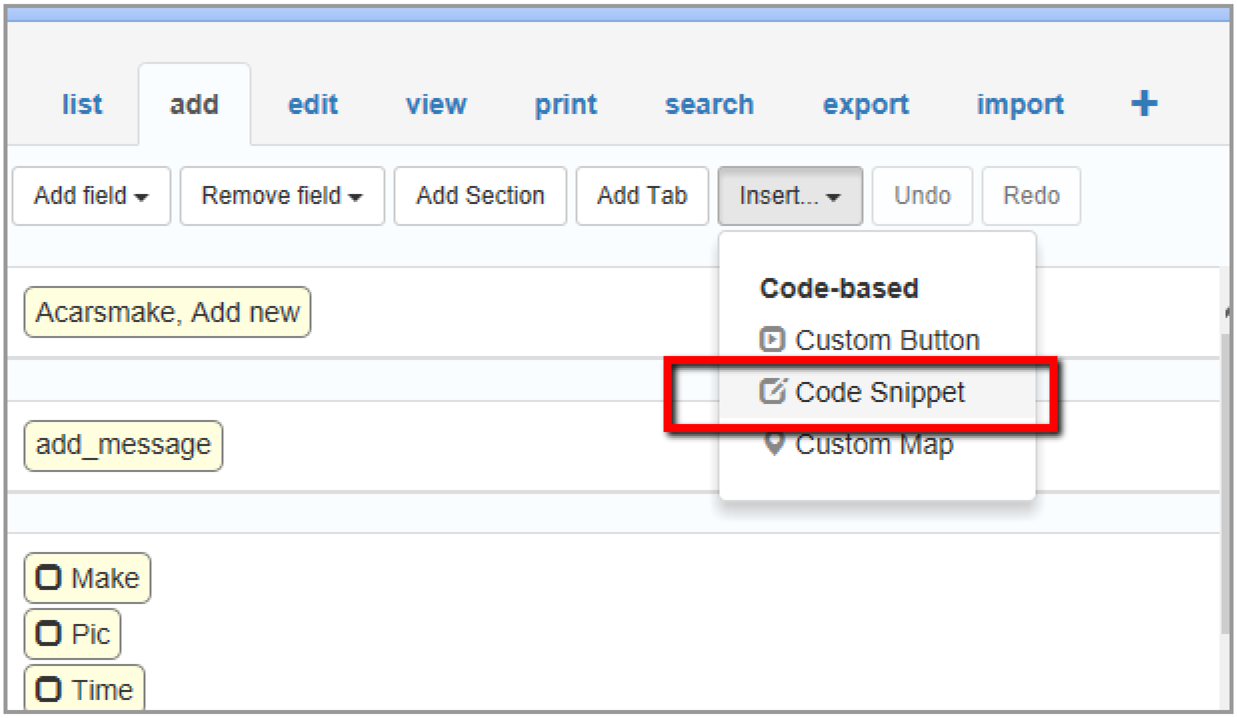
2.16.8 Adding PHP code snippet

Adding PHP code snippets you can modify the appearance and functionality of the prebuilt web pages. For example, you can add additional control elements to a web page or display some information.

To add a PHP code snippet, follow the instructions below.

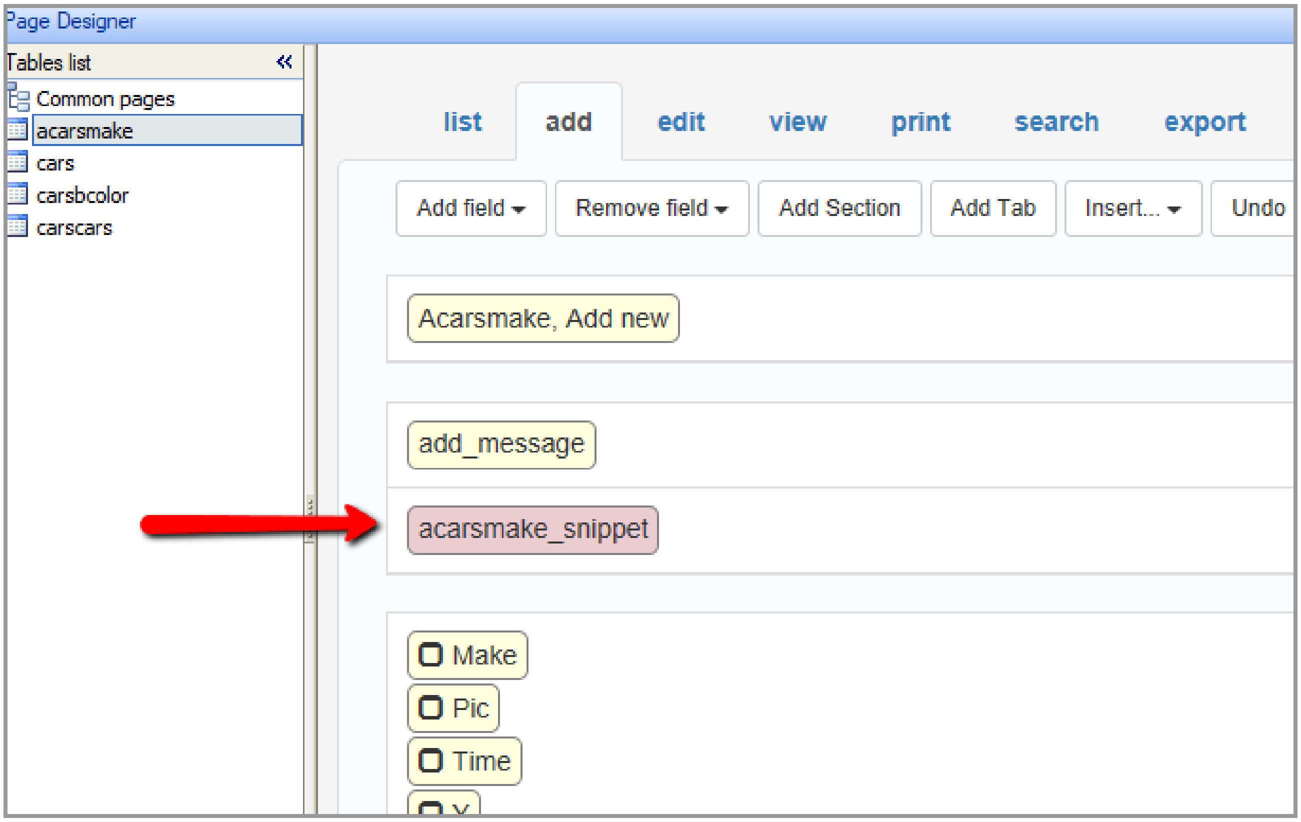
1. Proceed to the **Page Designer** and select a page you wish to modify.

2. Click **Insert - Code snippet** on the Main toolbar.



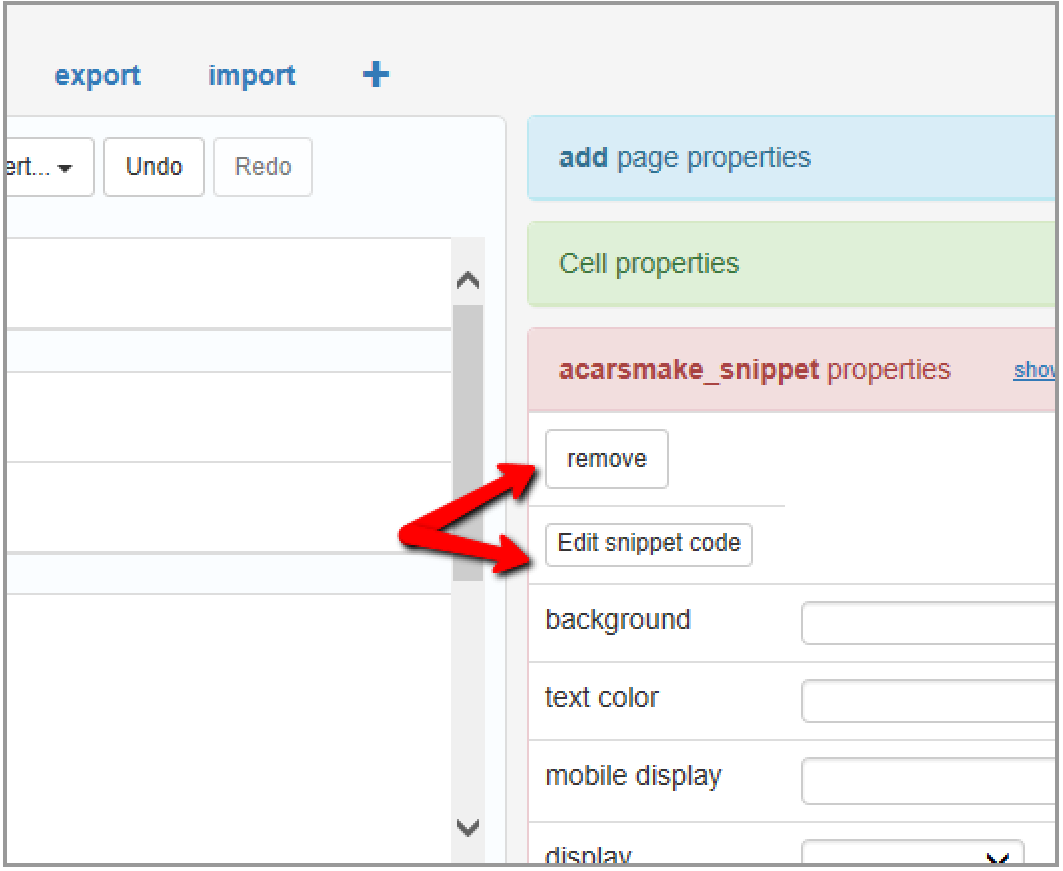
3. Add you PHP code and click **OK**.

4. New page element labeled **<Table name>_snippet** will appear on the page. If your code snippet displays some data on a web page, move the "snippet" element to where you wish to see that data.



To edit a code snippet, **Edit snippet code** button. Also you can edit a code snippet on the **Events** page.

To delete a code snippet, click **Remove** button.



Examples

1. Display current time.

PHP code:

```
echo now();
```

2. From the code snippet added to the Edit/View page you can get access to fields of the current record. For example, you can print the value of field *Make* using the following code:

```
global $pageObject;
$record = $pageObject->getCurrentRecord();
echo $record["Make"];
```

3. Add dropdown list box with values for search.

4. Show customer orders.

5. Show data from master table on details view/edit/add page.

2.16.9 Insert button

Quick jump

Insert button into datagrid

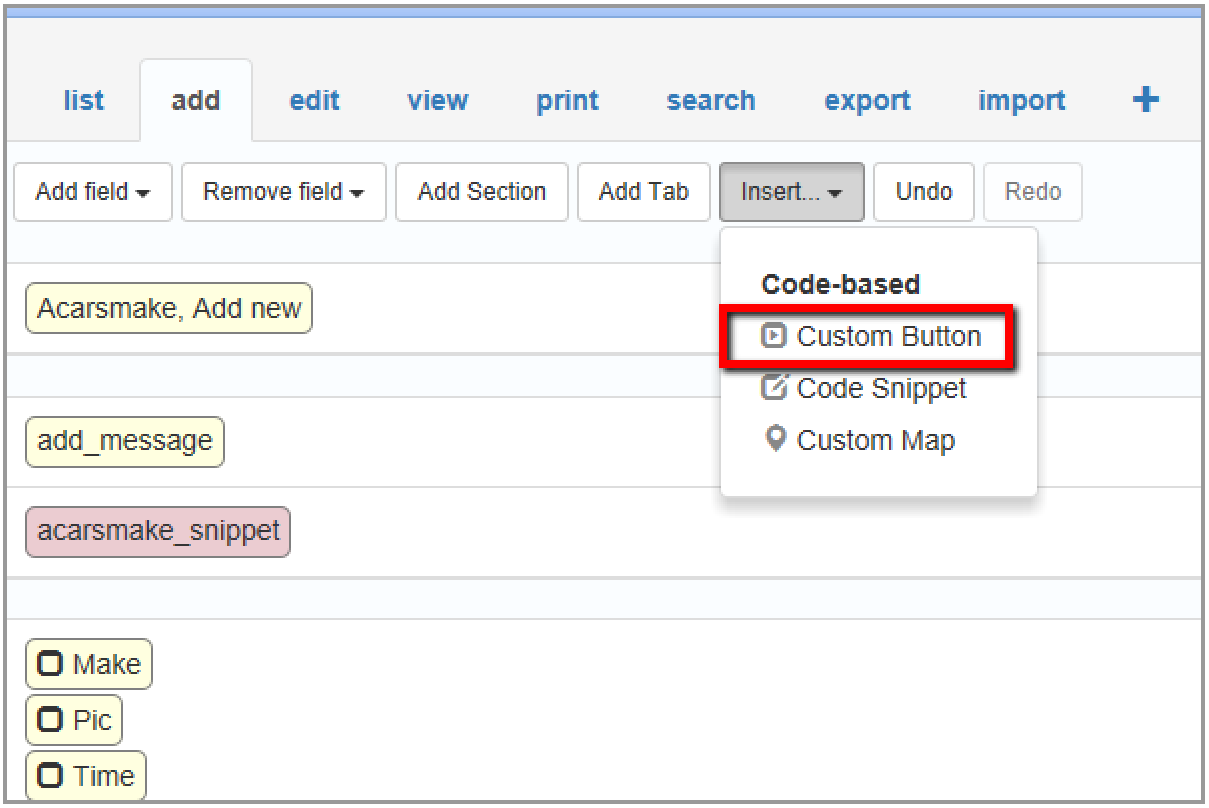
Passing data between events

Examples

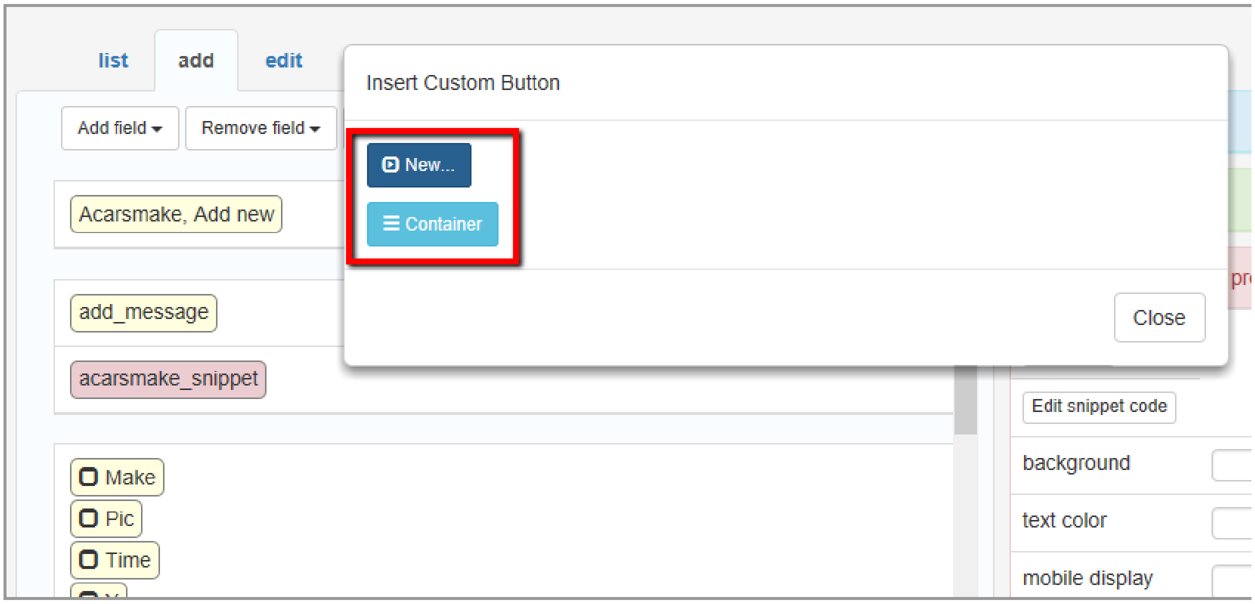
To add a button to a web page:

1. Proceed to the **Page Designer** and select a page you wish to modify.

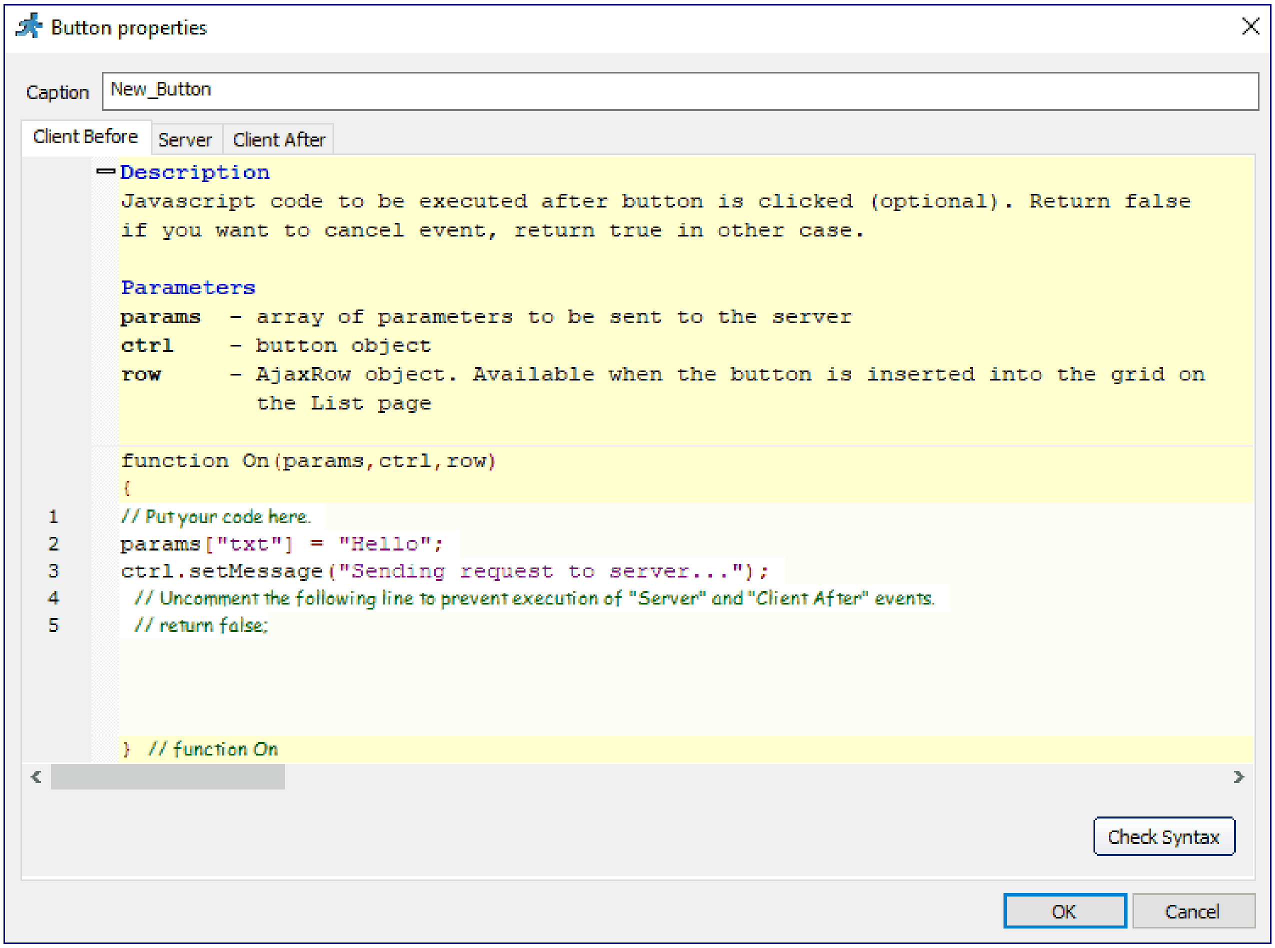
2. Click **Insert - Custom button** on the Main toolbar.



3. Choose the button or container

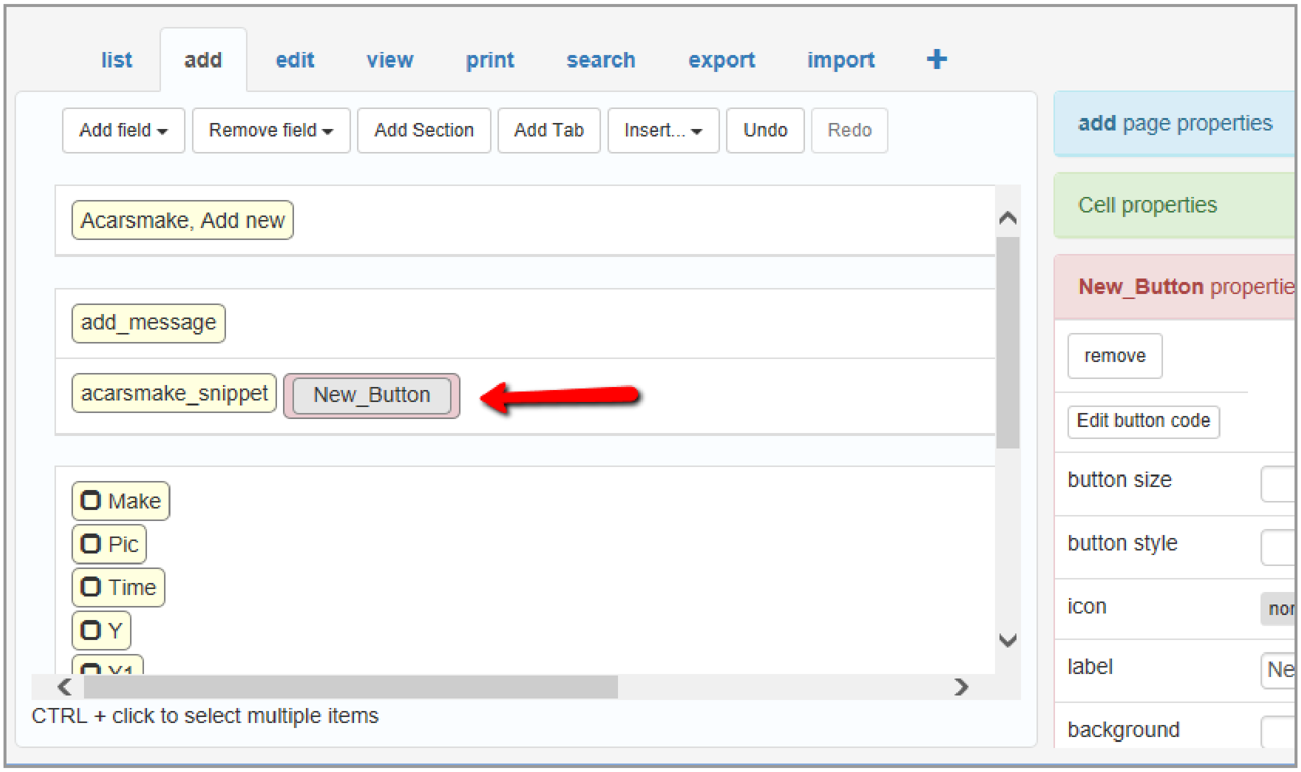


4. Type in the button caption and code to be executed after button is pressed. You can add both client-side (Javascript) and server-side (PHP) events. Click **OK**.



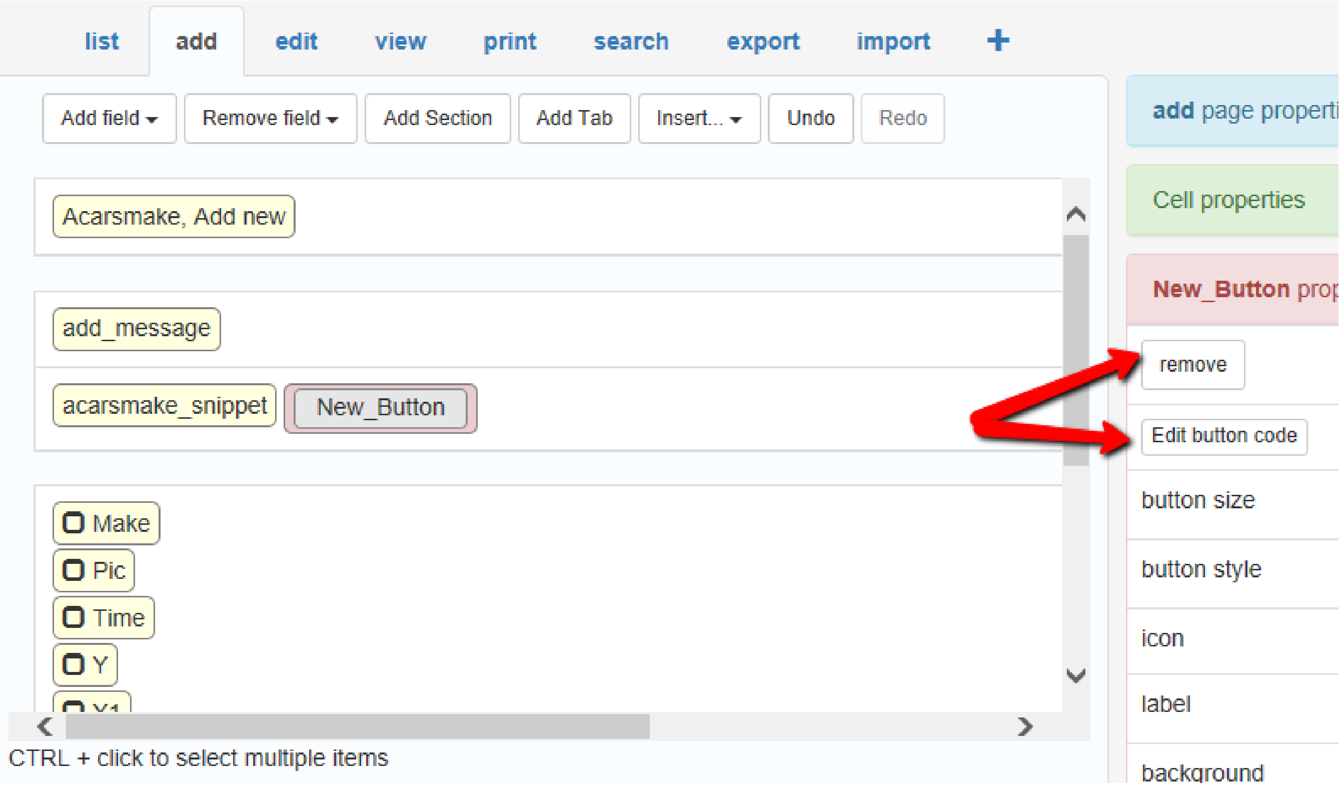
```
Description
Javascript code to be executed after button is clicked (optional). Return false
if you want to cancel event, return true in other case.

Parameters
params  - array of parameters to be sent to the server
ctrl    - button object
row     - AjaxRow object. Available when the button is inserted into the grid on
          the List page

function On(params,ctrl,row)
{
// Put your code here.
params["txt"] = "Hello";
ctrl.setMessage("Sending request to server...");
 // Uncomment the following line to prevent execution of "Server" and "Client After" events.
 // return false;



} // function On
```

3. Now the button is added to your web page. You can drag the button to change its location.



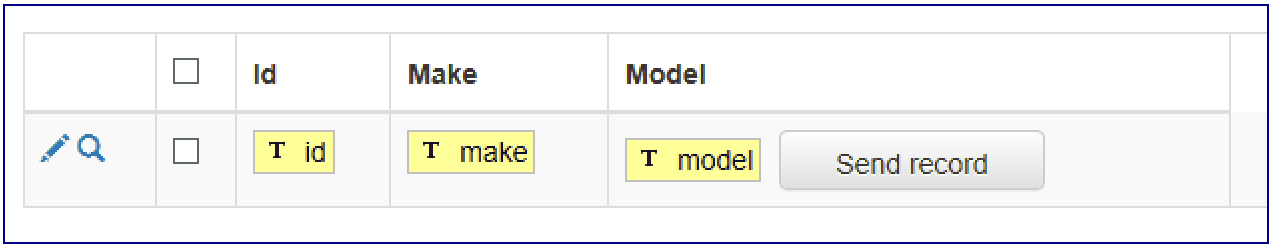
To edit the code, click **Edit button code** in **Properties**. You may also edit the code on the Events page.

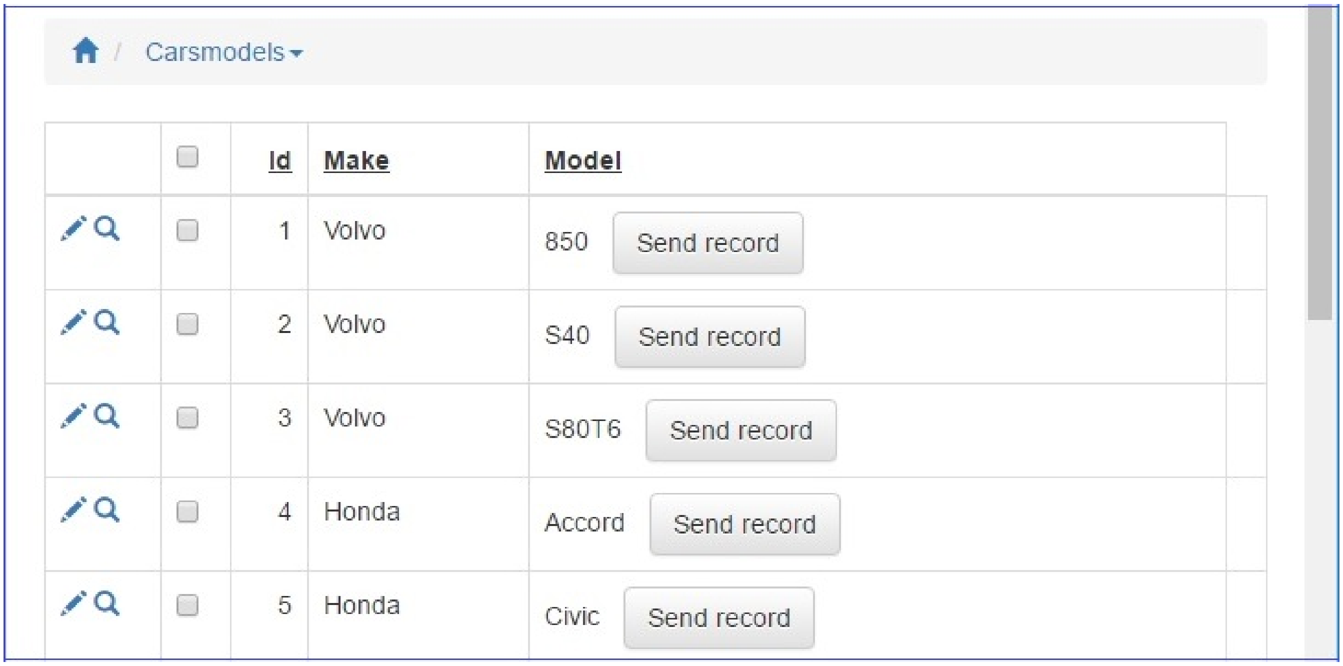
To delete button, click **Remove** in **Properties.**



Insert button into datagrid

You can insert a button into a data table to make it apply to each data record on a web page. This is possible only if you choose one of the Advanced Grid types on the Page Designer screen.





	☐	Id	Make	Model
	☐	1	Volvo	850 Send record
	☐	2	Volvo	S40 Send record
	☐	3	Volvo	S80T6 Send record
	☐	4	Honda	Accord Send record
	☐	5	Honda	Civic Send record

Note: a button inserted into a data grid will only work properly if you select a key field for the table on the Choose page screen.

You can use the *rowData* and *$button* objects to program buttons inserted into a grid. See also how to Hide buttons in select rows of a datagrid.

Passing data between events

1. Passing parameters from a ClientBefore event to an OnServer event.

Add the following code to the ClientBefore event:

```
params["test"]="Some value";
params["name"]="My name";
```

In the OnServer event use *$params["test"]* to access the value of a parameter named "test".

```
DB::Exec("insert into log (text) values
('".DB::prepareSQL($params["test"])."')");
```

2. Passing parameters from the OnServer event to the ClientAfter event.

OnServer event:

```
$result["response"]="Server response";
```

ClientAfter event:

```
alert(result["response"]);
```

Examples

Example 1. Inserting an "Add this product to shopping cart" button

For the Edit/View pages the *$keys* parameter in the *OnServer* event contains the information about the current record. You may use *$keys["KeyFieldName"] t*o access a specific key column.

For example, there is a *Products* table. To insert an **Add this product to shopping cart** button on the View page, add the following code to the *OnServer* event (**Server** tab):

```
global $dal;
$record = $button->getCurrentRecord();
if ($record["ProductID"])
{
//add new records to the ShoppingCart table
//save current username in the UserID field
$ShoppingCart = $dal->Table("ShoppingCart");
$ShoppingCart->Product = $record["ProductID"];
$ShoppingCart->Quantity = 1;
$ShoppingCart->UserID = $_SESSION["UserID"];
$ShoppingCart->Add();
}
$result["txt"] = "Product was added";
```

and this code to the *OnAfter* event (**Client After** tab):

```
var message = result["txt"] + ".";
```

For more information about using the Data Access Layer (DAL), see Data Access Layer.

Example 2. Send records selected on the List page via email

To send records selected on the List page via email, add the following code to the *OnServer* event (**Server** tab):

```
$email_msg = "";
$email_msg.= "List of records";
$i=1;
while($data = $button->getNextSelectedRecord())
{
  $email_msg.= "Record: ".($i++)."\r\n";
  $email_msg.= "Make: ".$data["make"]."\r\n";
  $email_msg.= "Qwerty: ".$data["qwerty"]."\r\n";
  $email_msg.= "\r\n";
}
//send email
$email = "test@test.com";
$subject = "Sample subject";
runner_mail(array('to' => $email, 'subject' => $subject, 'body' => $email_msg));
```

For more information see getNextSelectedRecord().

Example 3. Modifying the value of a field for all selected records on the List page

To modify the value of a field for all selected records on the List page, add the following code to the *OnServer* event (**Server** tab):

```
while($record = $button->getNextSelectedRecord())
{
$sql = "update Invoices set Status='Paid' where InvoiceID="
.$record["InvoiceID"];
DB::Exec($sql);
}
```

Example 4. Inserting a button redirecting to another page

To make a button redirect to another page, add the following code to the *OnBefore* event (**Client Before** tab) or *OnAfter* event (**Client After** tab):

```
location.href="http://cnn.com";
```

Example 5. Showing a message to the customer for a limited period of time

Let's say you have a button that retrieves a piece of sensitive info from the server that needs to be shown to the customer for a short period of time. After that this message needs to be hidden. To do this, add the following code to the *OnAfter* event (**Client After** tab):

```
// this message includes the SSN retrieved from the server
var message = result["txt"];
ctrl.setMessage(message);

// clear the message after 5 seconds
setTimeout(function(ctrl) {
  ctrl.setMessage("");
}
, 5000, ctrl);
```

Example 6.

Let's say you want to add a button to the View page that redirects to one of the application pages, depending on the value of the *Category* field. To do so you may add this code to the *OnServer* event (**Server** tab):

```
// get the value of the current record and pass it to the ClientAfter event
$result["record"] = $button->getCurrentRecord();
```

and this code to the *OnAfter* event (**Client After** tab):

```
if (result.record["Category"]=='Schools')
location.href="Schools_list.php";
else if (result.record["Category"]=='Organisations')
```

```
location.href="Organisations_list.php";
else if (result.record["Category"]=='Careers')
location.href="Careers_list.php";
```

For more information see getCurrentRecord().

Example 7.

To refresh the page add the following code to the OnAfter event (Client After tab):

```
location.reload();
```

2.16.10 Insert Map

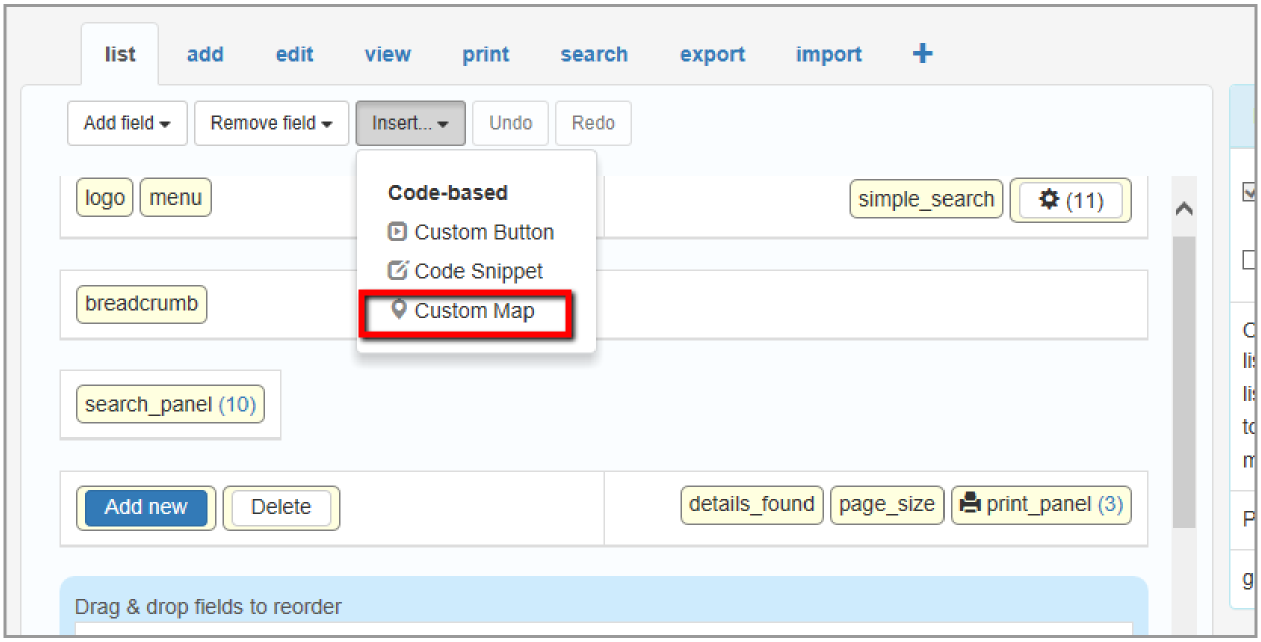
Quick jump
How to place description on each map marker
How to display GPS locations (map markers) by date
Maps API and geocoding

You can display the location data as separate map for each table record or insert large map to the **List** page. PHPRunner supports several map providers so you can select the suitable one on the **Miscellaneous** page -> Map settings.

To insert large map to the **List** page, follow the instructions below:

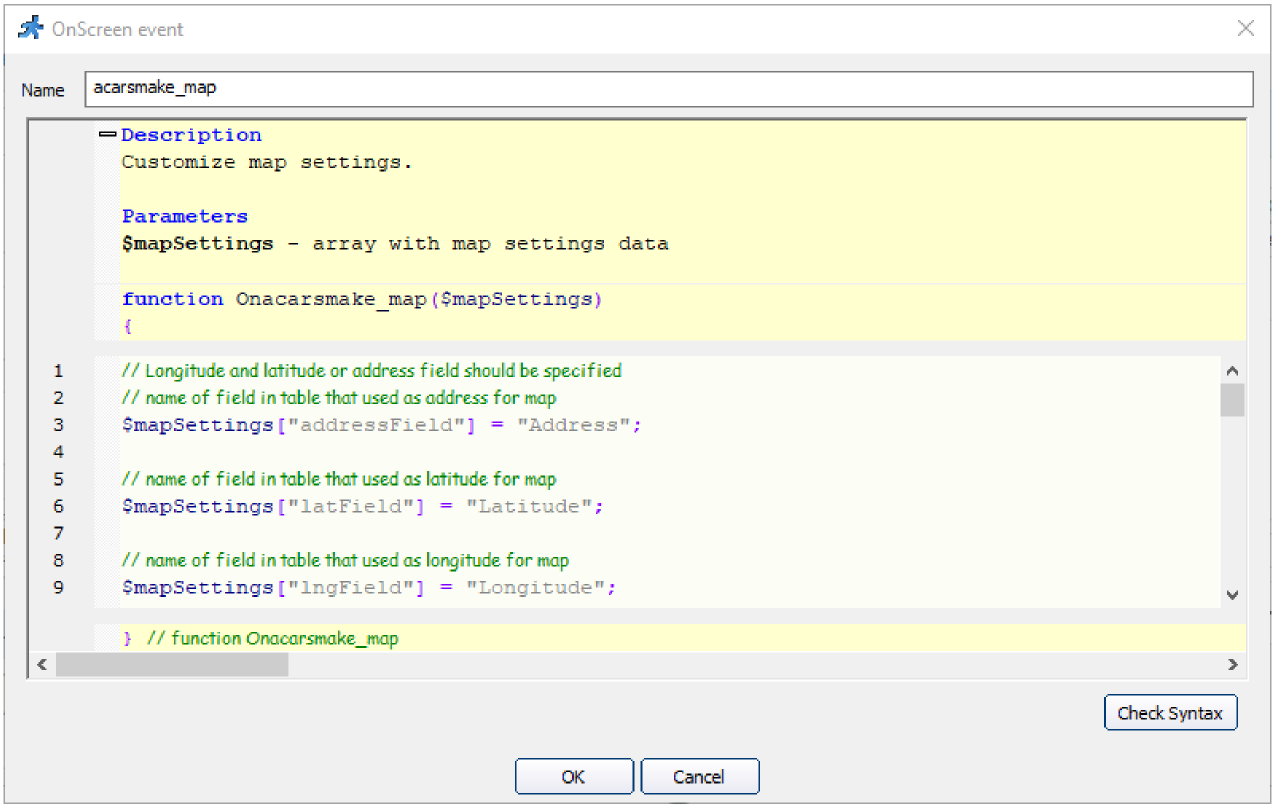
1. Proceed to the **Page Designer** and select a page you wish to modify.

2. Click **Insert - Custom Map** on the Main toolbar.



Address field must be a text field. If you do not have a field with address and only want to use Latitude/Longitude fields, comment out the following line:

```
//$mapSettings["addressField"] = "Address";
```



Name: acarsmake_map

```
Description
Customize map settings.

Parameters
$mapSettings - array with map settings data

function Onacarsmake_map($mapSettings)
{

// Longitude and latitude or address field should be specified
// name of field in table that used as address for map
$mapSettings["addressField"] = "Address";

// name of field in table that used as latitude for map
$mapSettings["latField"] = "Latitude";

// name of field in table that used as longitude for map
$mapSettings["lngField"] = "Longitude";

} // function Onacarsmake_map
```

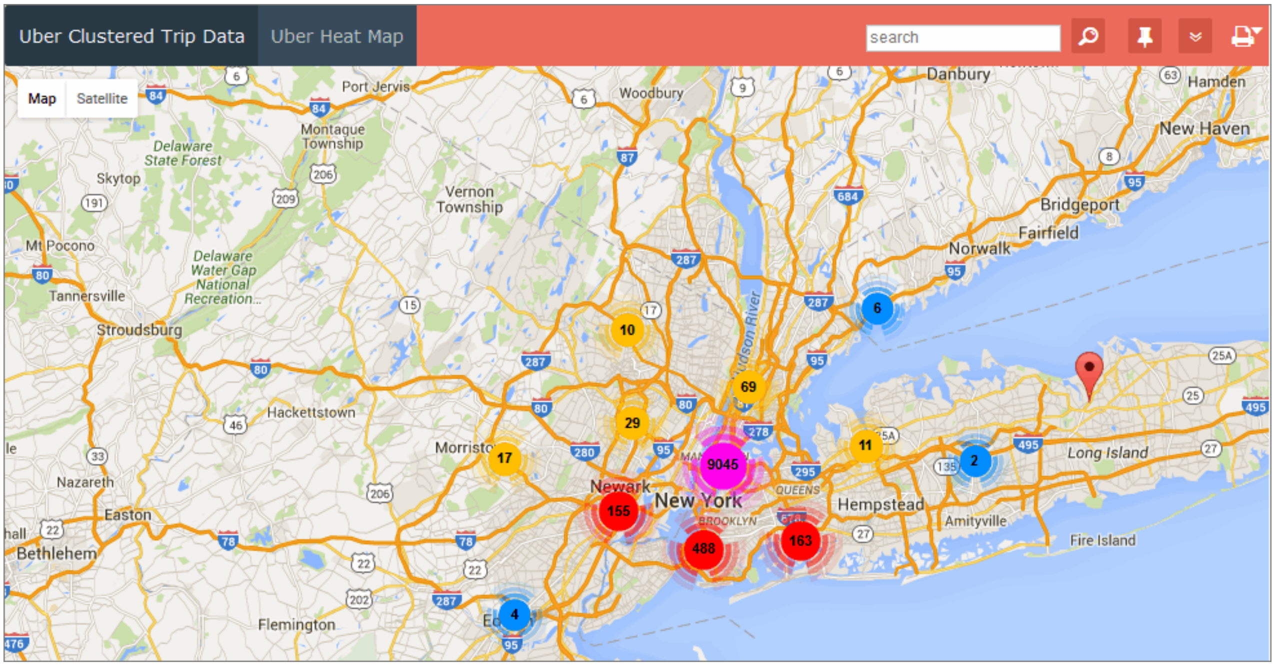
If you use Google Maps and store latitude/longitude information, you can enable clustering or heat map option.

To enable clustering, add this code:

```
$mapSettings['clustering'] = true;
```

If you like you can use custom clustering icons
There is a 'source' subfolder in your project folder. Proceed there and create 'images' folder there. Place there images named m1.png, m2.png, m3.png, m4.png and m5.png, These are icons that will be used in Google maps when clustering is turned on.

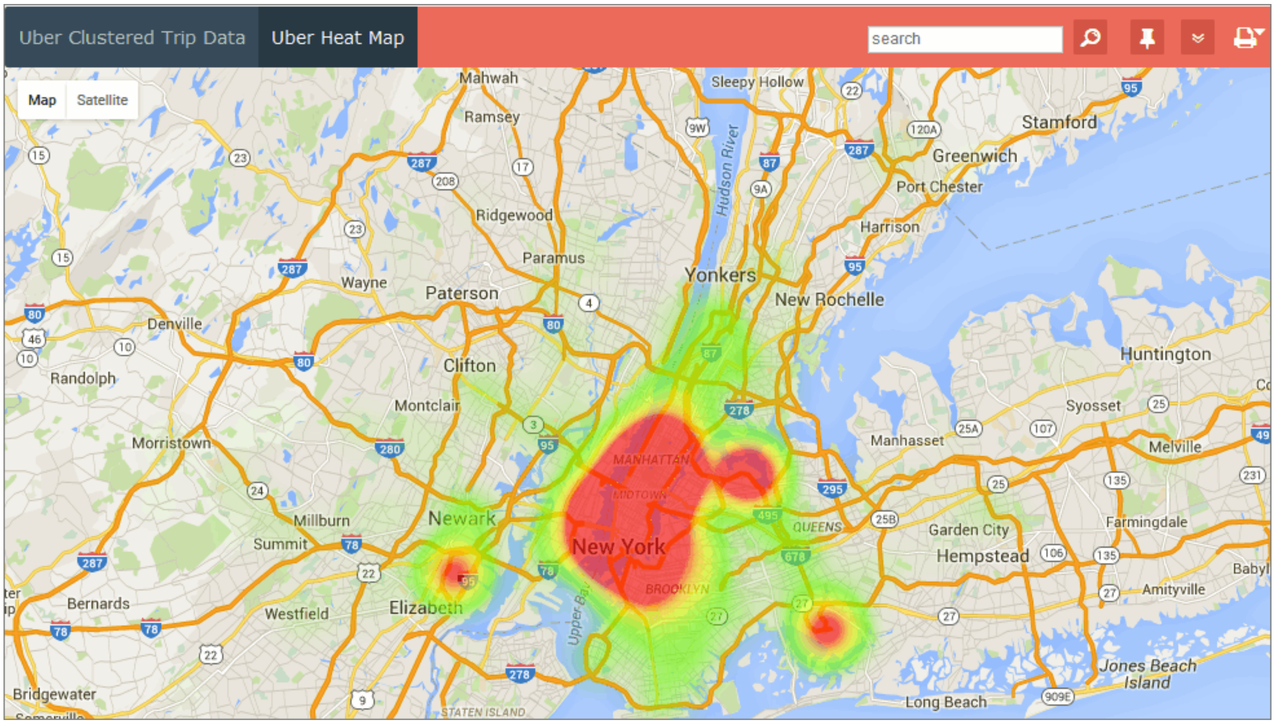
Example:



To enable heat map, add this code:

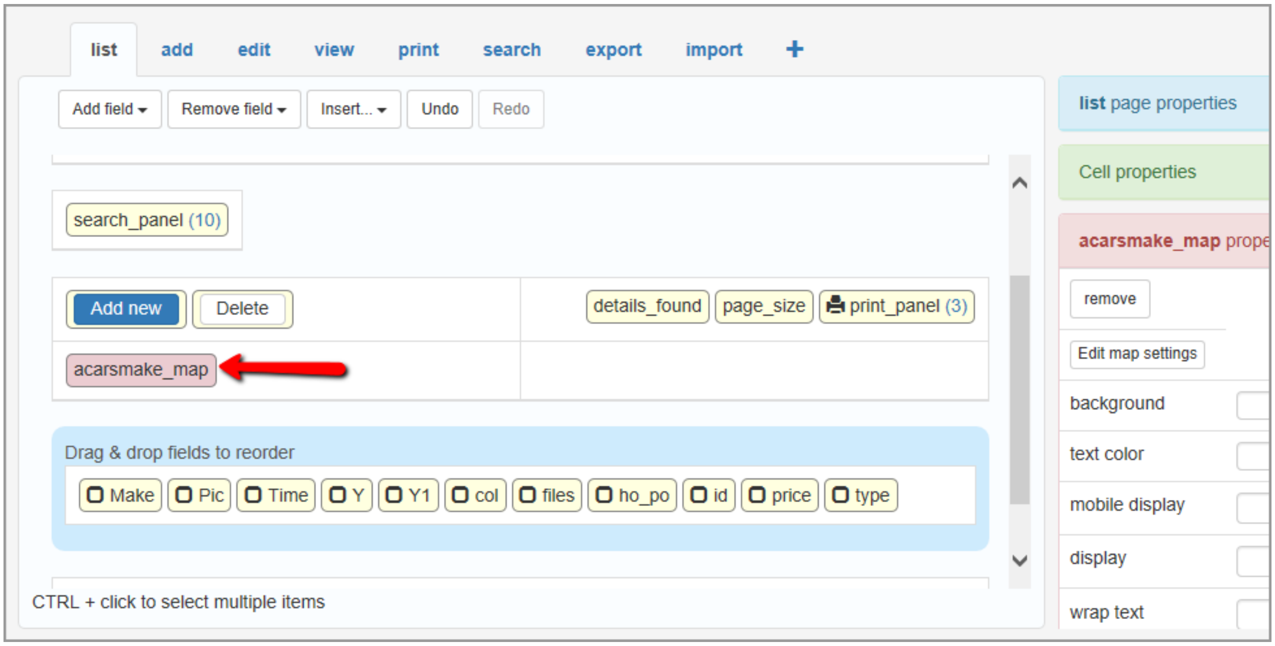
```
$mapSettings['heatMap'] = true;
```

Example:

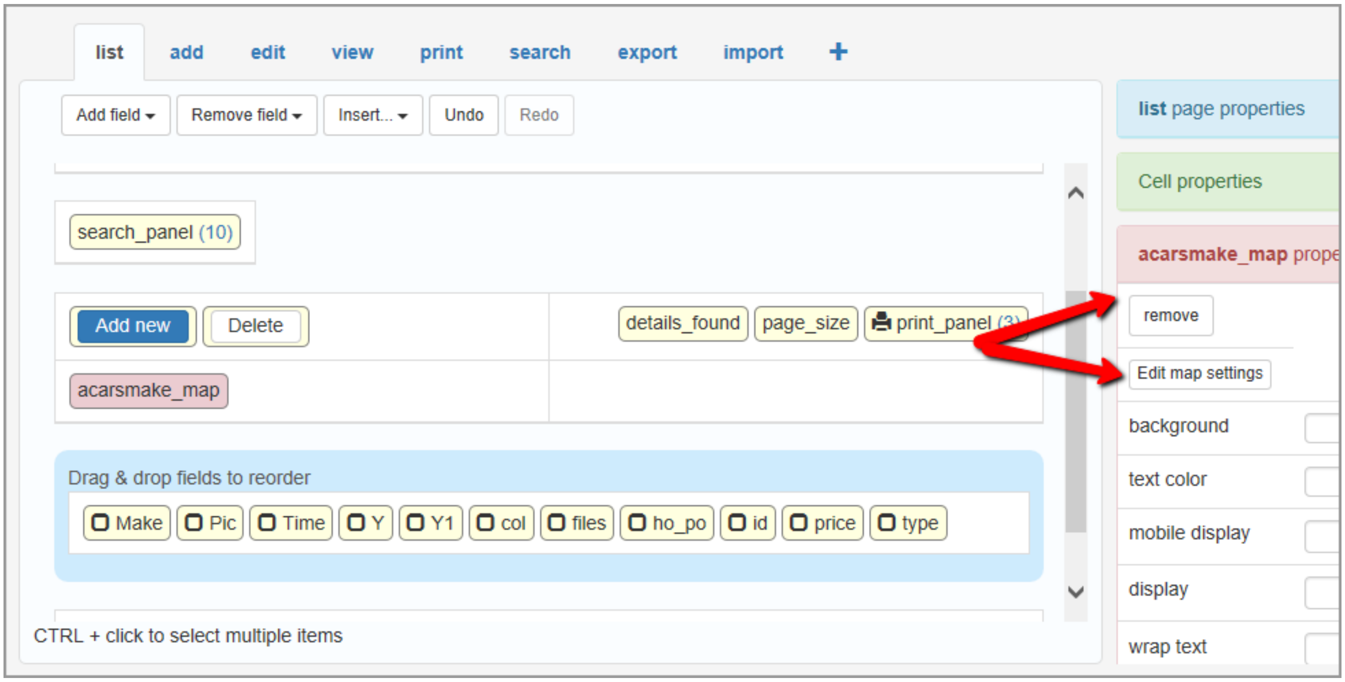


3. Click **OK**.

4. Now map is added to your web page. You can drag the map element to change its location.



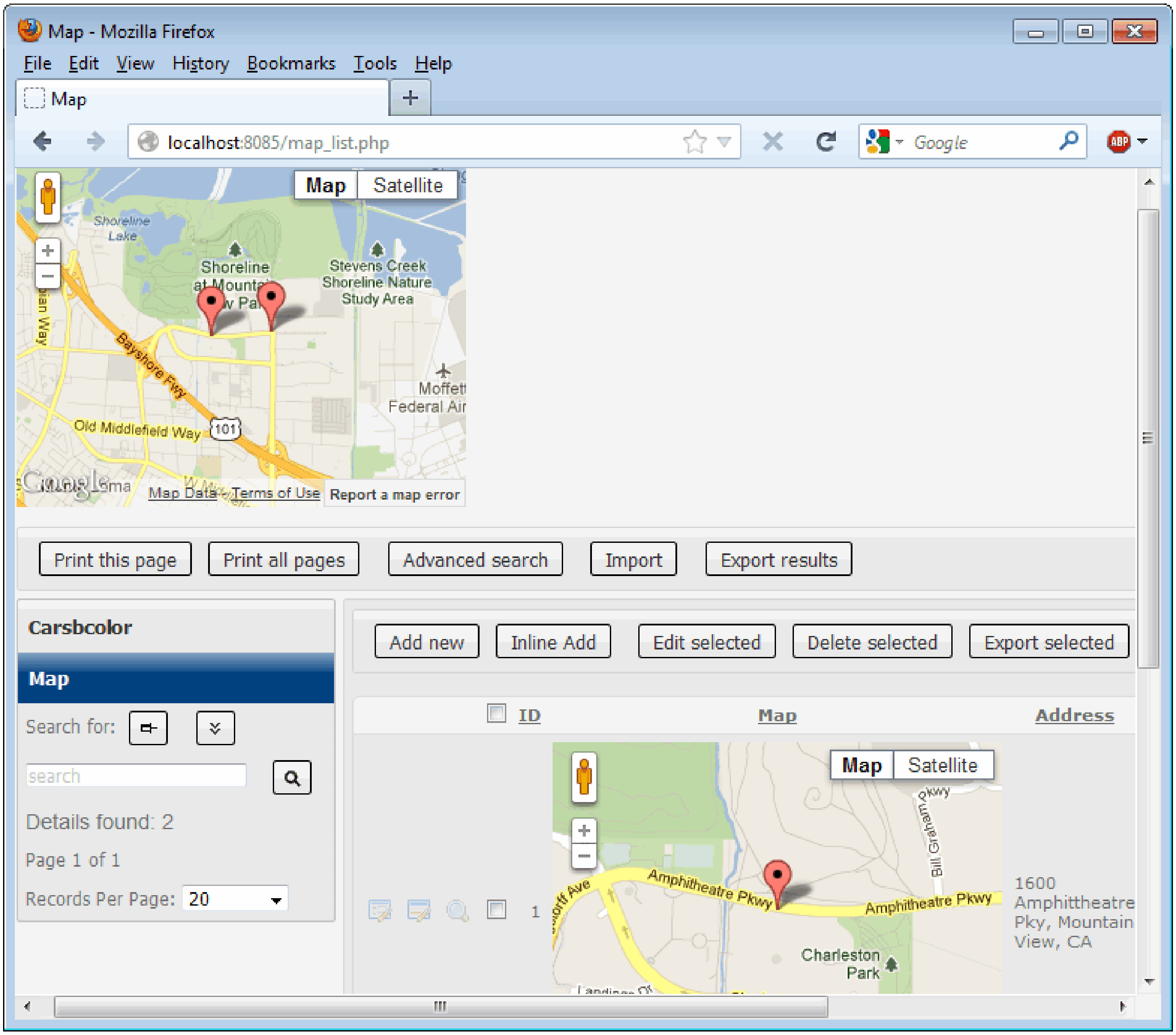
To edit the code, click **Edit map settings**. To delete it click **Remove**.



Also you can edit the code on the Events page.

Example

Note that clickable markers on the large map will point to the **View** record page.



Details found: 2

Page 1 of 1

Records Per Page: 20

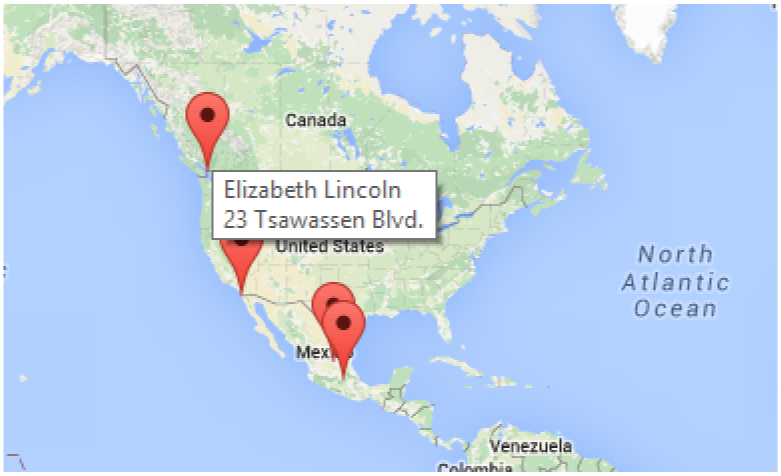
ID	Map	Address
1		1600 Amphittheatre Pky, Mountain View, CA

How to place description on each map marker

1. Modify SQL Query to create a calculated field with custom name and address.

```
SELECT
        CustomerID,
        CompanyName,
        ContactName,
        ContactTitle,
        Address,
        Lat,
        Lng,
concat(ContactName, '\n', Address) as DisplayOnMap
FROM customers
```

Notice '\n', it allows to create multiline descriptions. concat() function is MySQL specific. Similar functions exist in all databases.



2. Insert a map into List page. Here are settings I have used for this specific table.

```
// Longitude and latitude or address field should be specified
// name of field in table that used as address for map
$mapSettings["addressField"] = "DisplayOnMap";

// name of field in table that used as latitude for map
$mapSettings["latField"] = "Lat";

// name of field in table that used as longitude for map
$mapSettings["lngField"] = "Lng";
```

Latitude and Longitude fields are required as we use address field for marker description purposes.

How to display GPS locations (map markers) by date

If you need to display GPS locations (map markers) by date, for example, new geo coordinates that were inserted into database on a specific day add a WHERE clause to the SQL query of the table where you insert your map.

For instance to select today's data you can use something like this:

```
SELECT * FROM mytable WHERE DATE(posted) = CURDATE()
```

Maps API and geocoding

Maps API providers put limits on number of geocoding requests you can send per day and also limited requests rate. For example, if you use Google Maps API and have a page with 20 records with a map each only about 10 maps will be displayed properly. To overcome those limits you need to open a fairly expensive Premier account with Google that costs $10,000 per year.

The solution is to use latitude/longitude pairs instead of addresses for mapping purposes. Enable the Geocoding option on the **Choose pages** screen to update latitude/longitude information each time when a record is added or updated. The main question is how to convert existing addresses to latitude/longitude pairs? We have added a small utility to PHPRunner that will do all the job for you. Here are instructions:

1. Proceed to the table that stores addresses and add two new fields to that table. Fields need to be able to store floating point numbers. In MySQL use Decimal(10,6) or Double. In MS Access use Number with 'Field size' Double.

2. Proceed to the field that is setup as *Map*. Make sure that Address and Latitude/Longitude fields are selected.

3. Proceed to the List page, choose to display all records, select all records, click 'Edit', then 'Save all'. It will take some time to update all records but you only need to do this once. New/updated records will update longitude/latitude automatically.

To speed up this process you may want temporarily remove 'View as' Map field from the List page and add it back once geocoding process is finished.

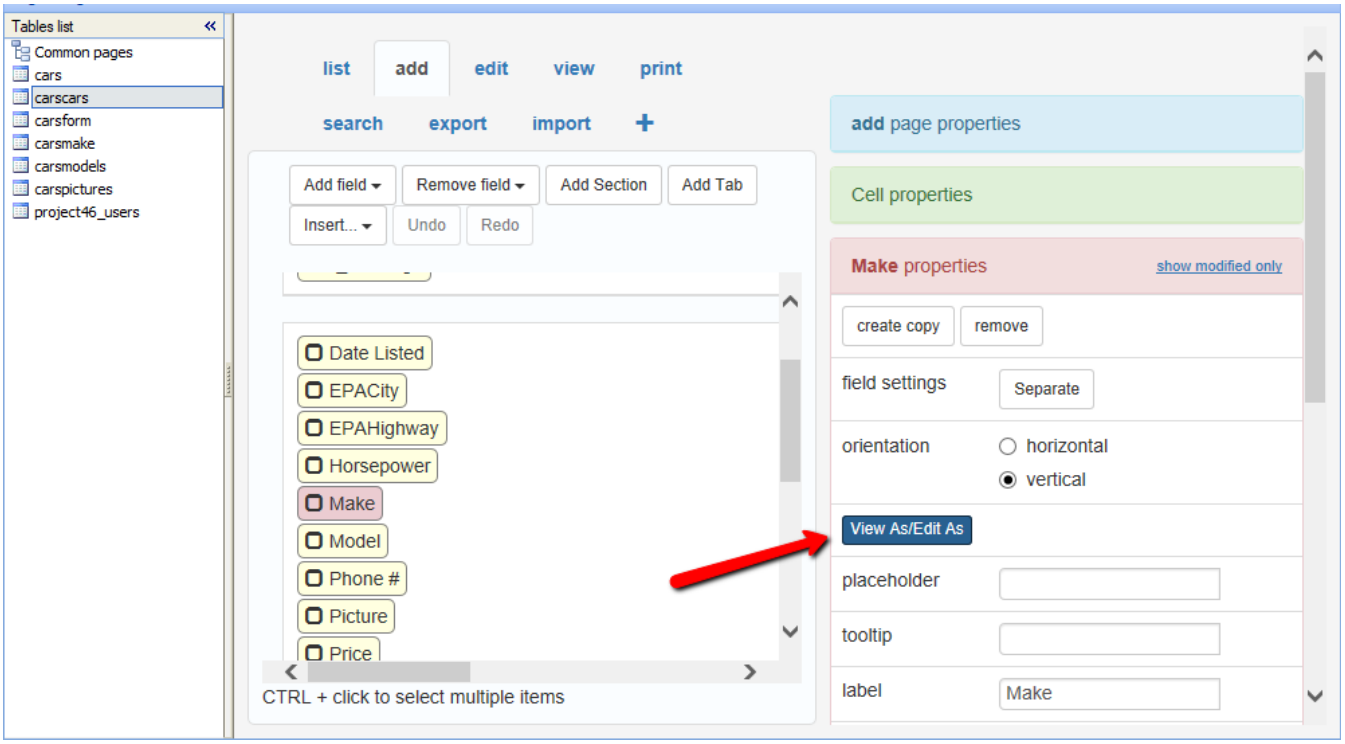
2.16.11 "View as" settings

2.16.11.1 "View as" settings

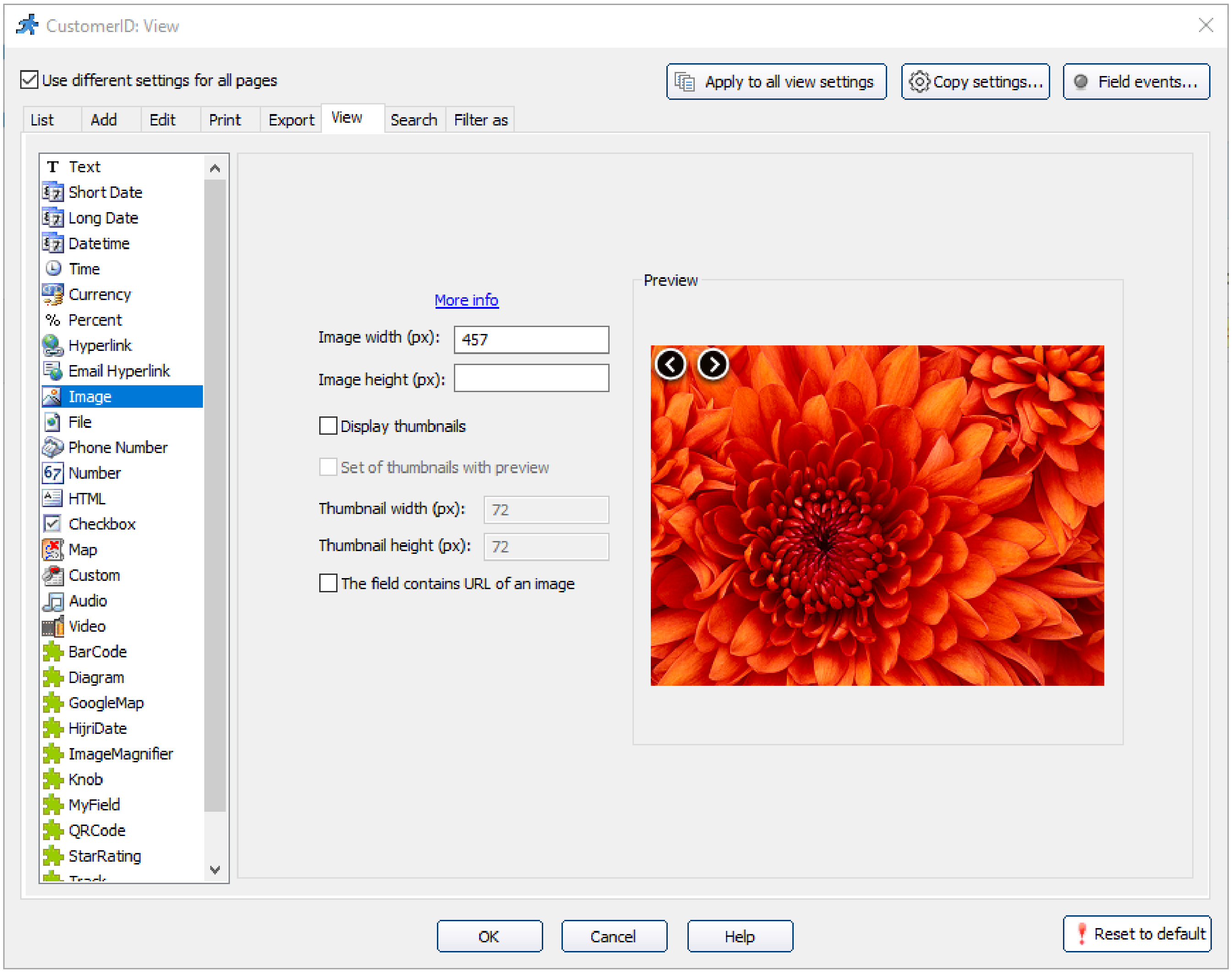
Quick jump		
Short Date	Long Date	Datetime
Time	Currency	Percent
Hyperlink	Email Hyperlink	File
Image	Phone number	Number
HTML	Checkbox	Map
Custom	Audio	Video
QRCode		

You can customize your data appearance on the List/View/Print/Export pages using formatting options on the **"View as" settings** dialog. You can define your field settings for all pages at once or separately for each page.

To control the filed appearance on the page click on the field and then in **Properties click** on the **View As/Edit as** button.



To set different field formats for different pages enable the **Use different settings for all pages** check box. Depending on selected format you will see different box-dialogs.



☑ Use different settings for all pages

Image width (px): 457

Image height (px):

□ Display thumbnails

□ Set of thumbnails with preview

Thumbnail width (px): 72

Thumbnail height (px): 72

□ The field contains URL of an image

Copy settings option allows you to copy field settings from another field.

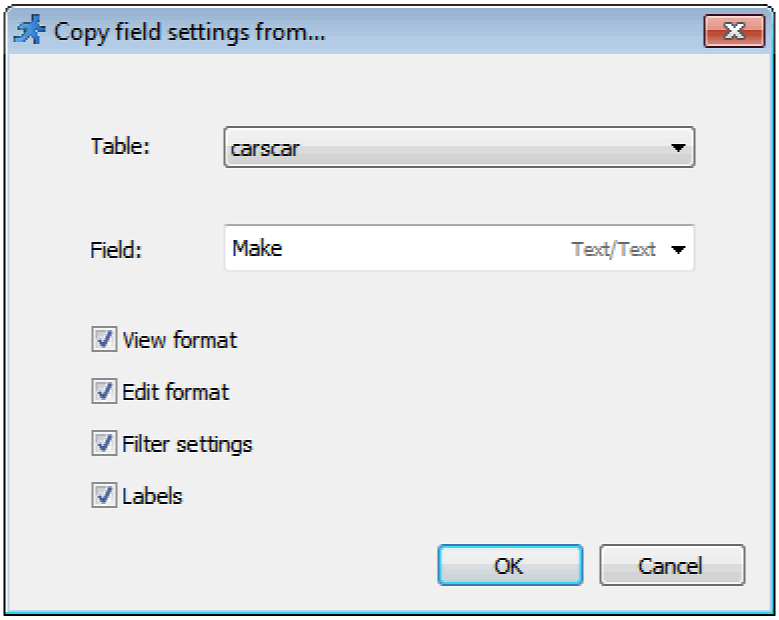


Table: carscar

Field: Make Text/Text

☑ View format

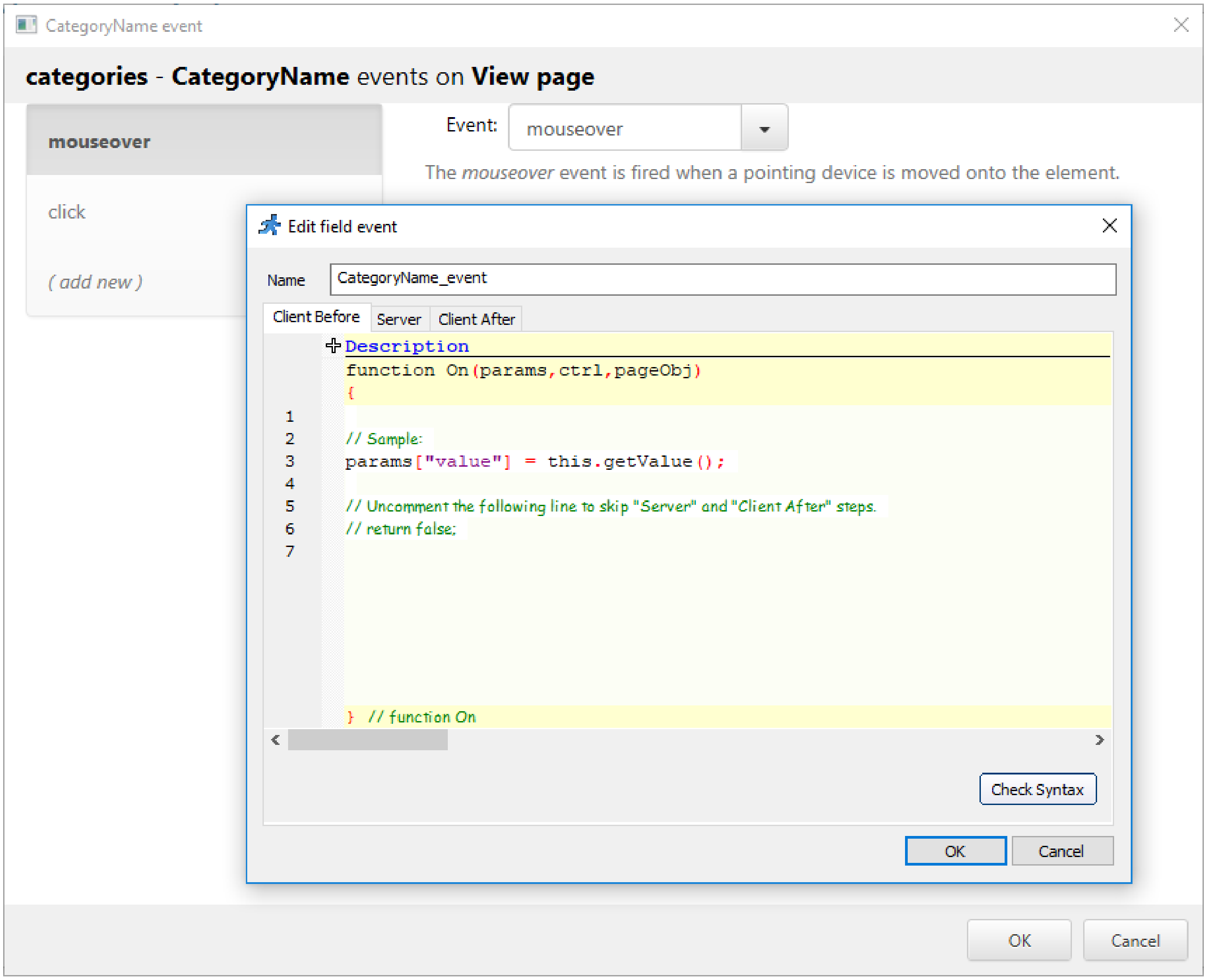
☑ Edit format

☑ Filter settings

☑ Labels

Field events option allows to perform an action when cursor enters edit field or leaves it or mouse is over a field. Perform any sort of validation, make other fields hidden or required etc. Designed to work on Add, Edit, View and Register pages.
For example, the *mouseover* event is fired when a pointing device is moved onto the element.

See also How to access fields in the field events



categories - **CategoryName** events on **View page**

Event: mouseover

The *mouseover* event is fired when a pointing device is moved onto the element.

Name: CategoryName_event

```
Description
function On(params,ctrl,pageObj)
{

// Sample:
params["value"] = this.getValue();

// Uncomment the following line to skip "Server" and "Client After" steps.
// return false;

} // function On
```

Short Date

Dates will be displayed in short format (02/17/2003).

Long Date

Dates will be displayed in long format (17 February 2003).

Datetime

Datetime values will be displayed as date and time (02/17/2003 14:22:03).

Time

Datetime values will be displayed as time (14:22:03).

Currency

Numeric values like 14000 will be displayed as $14 000.00 (actual format depends on your system regional settings like currency symbol, decimal symbol etc.).

Percent

Example: 0.38 will be displayed as **38%**.

Hyperlink

Choose this format if you store hyperlinks in this database field. Those hyperlinks will be made clickable automatically.

Email Hyperlink

Choose this format if you store email addresses in this database field. It will be converted into *mailto* HTML code automatically.

File

Choose this format if you store files in this field. For more information, see "View as" settings: File.

Image

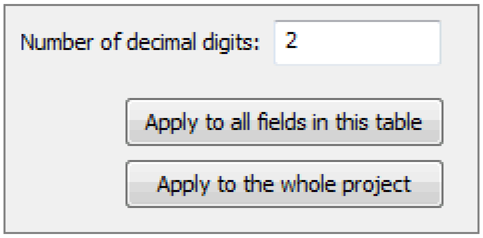
Choose this format if you store images in this field. For more information, see "View as" settings: Image.

Phone number

Formats number as a US phone number. Supports 7-digit or 10-digit numbers (123) 456-7890 or 123-4567.

Number

Choose this format if you like to format this field as a number. You can set the default number of digits after the comma for all numeric fields.



Number of decimal digits: 2

HTML

Use this view type when you store formatted HTML code in database field and wish to display this HTML code on the list page.

Checkbox

Use this view format to present field value as a check box. Works best with the following data types:

- MS Access: Yes/No field
- SQL Server: TINYINT or BIT field
- MySQL: TINYINT
- Oracle: NUMBER(1)

Map

Allows adding maps to your web pages. For more information, see "View as" settings: Map.

Custom

Allows formatting field values by adding php code. For more information, see "View as" setting: Custom.

Audio

Choose this format if you store audio files in this field. For more information, see "View as" settings: Audio.

Video

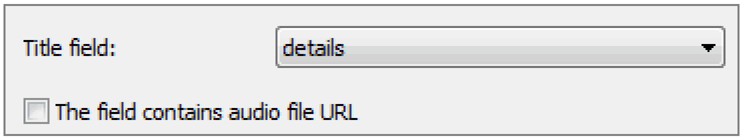
Choose this format if you store video files in this field. For more information, see "View as" settings: Video.

QRCode

The QRCode control allows you to add QR Codes to your pages. For more information, see "Edit as" settings: QRCode.

2.16.11.2 Audio

Select the field that contains the audio titles. Audio titles are displayed instead of the audio file URL/path.



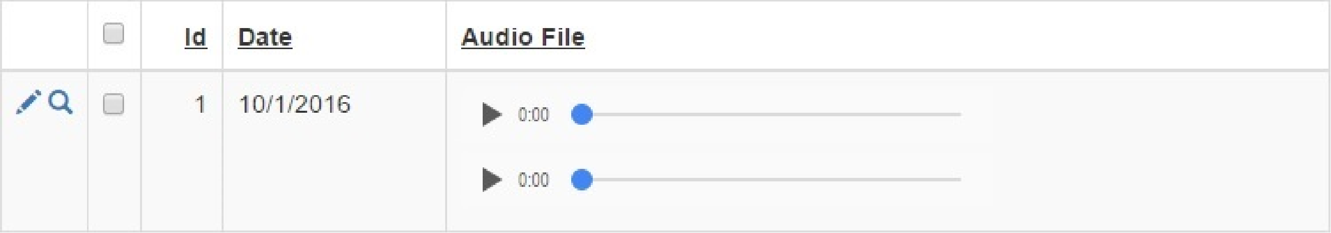
Title field: details ▾

□ The field contains audio file URL

If the selected field contains audio file URL, select the corresponding check box.

Edit as type can be either *Text field* (enter audio file name there) or *File/Image* (upload files).

An example of how audio files appear in the browser:



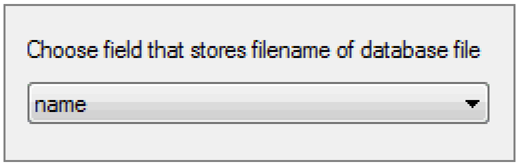
	□	Id	Date	Audio File
	□	1	10/1/2016	▶ 0:00 ▶ 0:00

2.16.11.3 File

File format is available for Binary and Text fields. Depending on field type you will be able to choose either folder where files are located (for text field) or field name that stores filenames (for binary field).

Binary field

In this case file is stored in the database and you need to choose a field that stores name of database file. This filename is required to set correct file type when you retrieve uploaded file from the database. If you don't choose filename field or leave it empty, you will be presented with **Open with** dialog every time you download this file from the database.

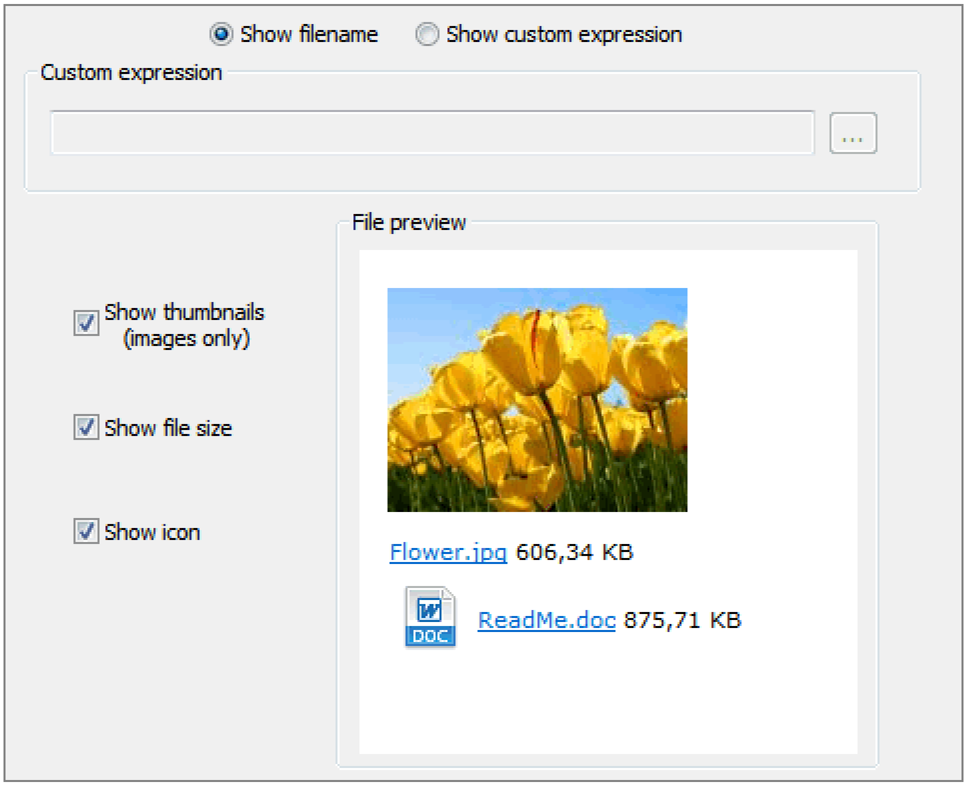


Choose field that stores filename of database file

name ▾

Text field

You can select to display filename or custom expression instead of filename. If you want to display the thumbnail image, file size or icon, select the corresponding check boxes.



● Show filename ○ Show custom expression

Custom expression

File preview

☑ Show thumbnails (images only)

☑ Show file size

☑ Show icon

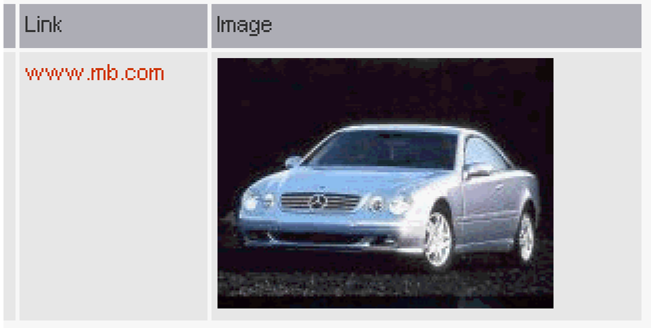
Flower.jpg 606,34 KB

ReadMe.doc 875,71 KB

2.16.11.4 Image

Image format is available for Binary and Text fields.

Example of how image is displayed in the browser:



Link	Image
www.mb.com	

Binary field

In this case images are stored in the database in the binary fields. PHPRunner creates code that extracts images from the database on the fly. Supported image formats are JPEG, GIF and BMP.

You can define fixed image size using the **Image width** and **Image height** text boxes.

If you want to display the thumbnail image, you should select the **Display thumbnails** check box and choose the field name that stores the thumbnail.

Note: you need an additional binary field (field of *MEDIUMBLOB* type) to store thumbnails. This field is the auxiliary one and we do not recommend to display it on your pages. You can display/hide fields on the [Choose fields]() page.

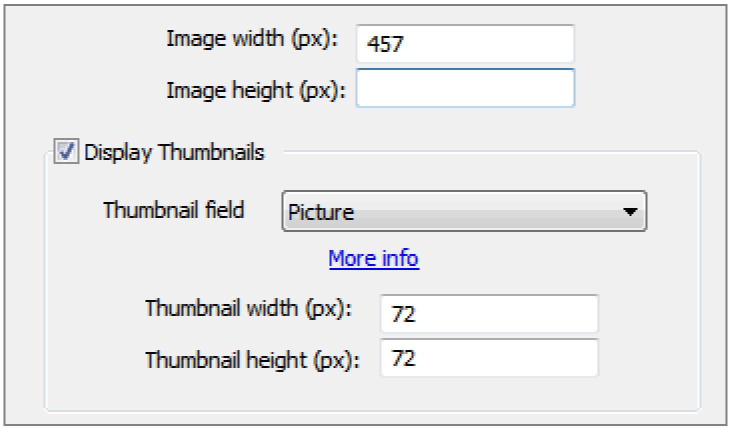


Image width (px): 457

Image height (px):

☑ Display Thumbnails

Thumbnail field: Picture

Thumbnail width (px): 72

Thumbnail height (px): 72

Text field

You can define fixed image size using the **Image width** and **Image height** text boxes.

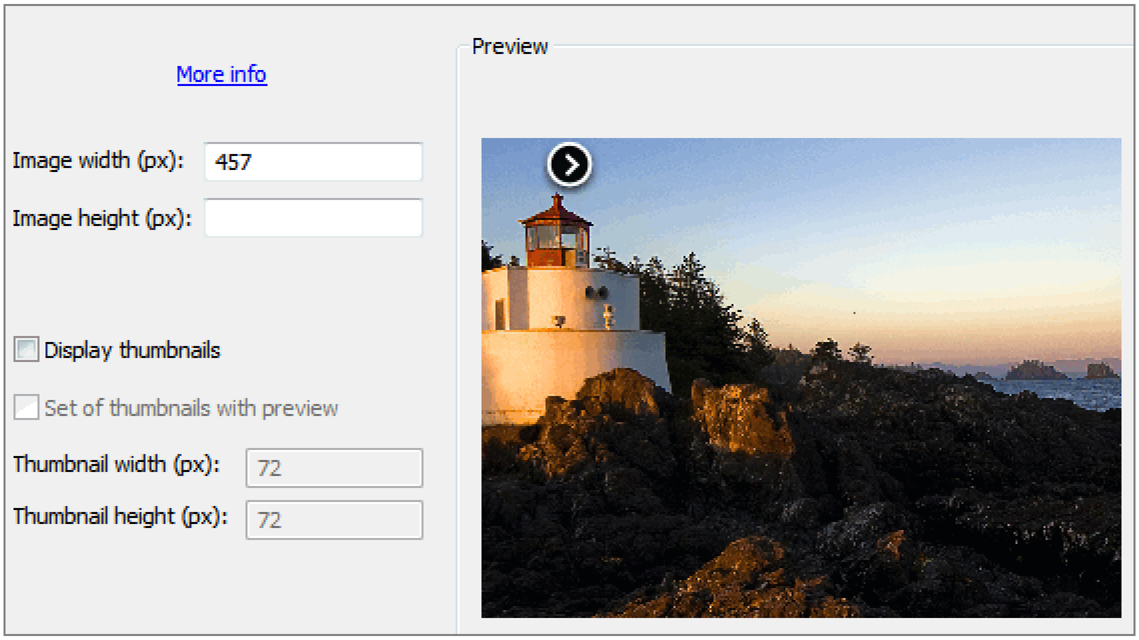


Image width (px): 457

Image height (px):

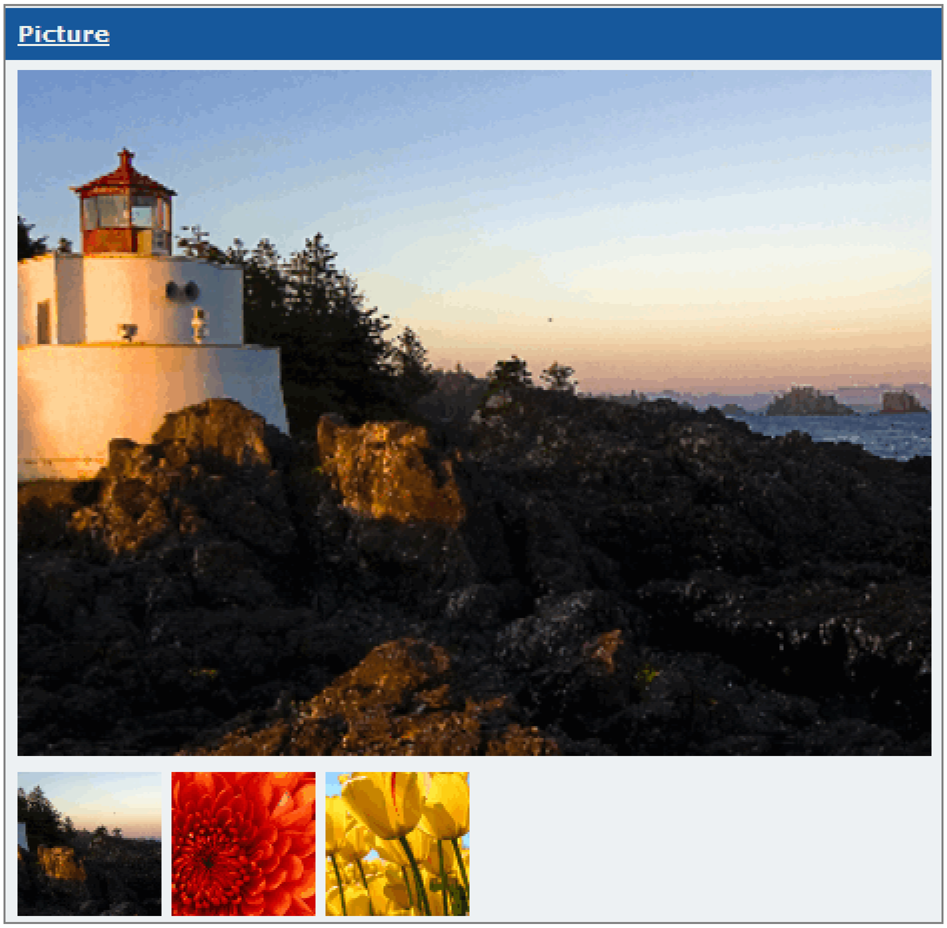
□ Display thumbnails

□ Set of thumbnails with preview

Thumbnail width (px): 72

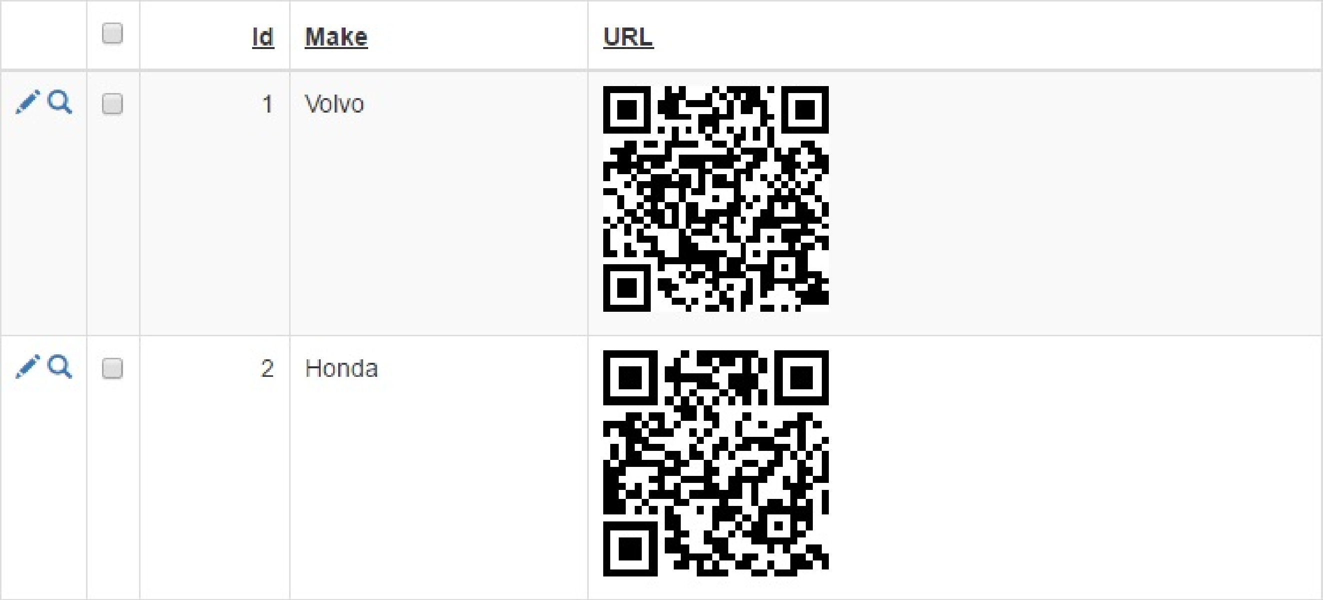
Thumbnail height (px): 72

If you want to display the thumbnail image, select the **Display thumbnails** check box. Then you can define fixed thumbnail image size. To show list of thumbnails with one large image preview, select the **Set of thumbnails with preview** check box.



2.16.11.5 QRCode

QRCode control allows you to generate QR Codes and display them on your web pages. QR Code (it stands for "Quick Response") is a cell phone readable bar code that can store website URL's, plain text, phone numbers, email addresses and any other alphanumeric data.

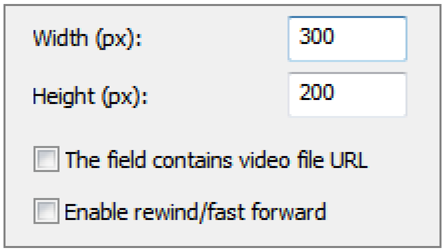


	□	Id	Make	URL
	□	1	Volvo	
	□	2	Honda	

Click **Add initialization script** button to customize the QRCode control.

2.16.11.6 Video

Select **The field contains Video file URL** check box if the field stores URL of the video file, for example *http://yourwebsite.com/folder/somefile.mp4*. Do not use this option if you use *File/Image* as *Edit as* type for this field.



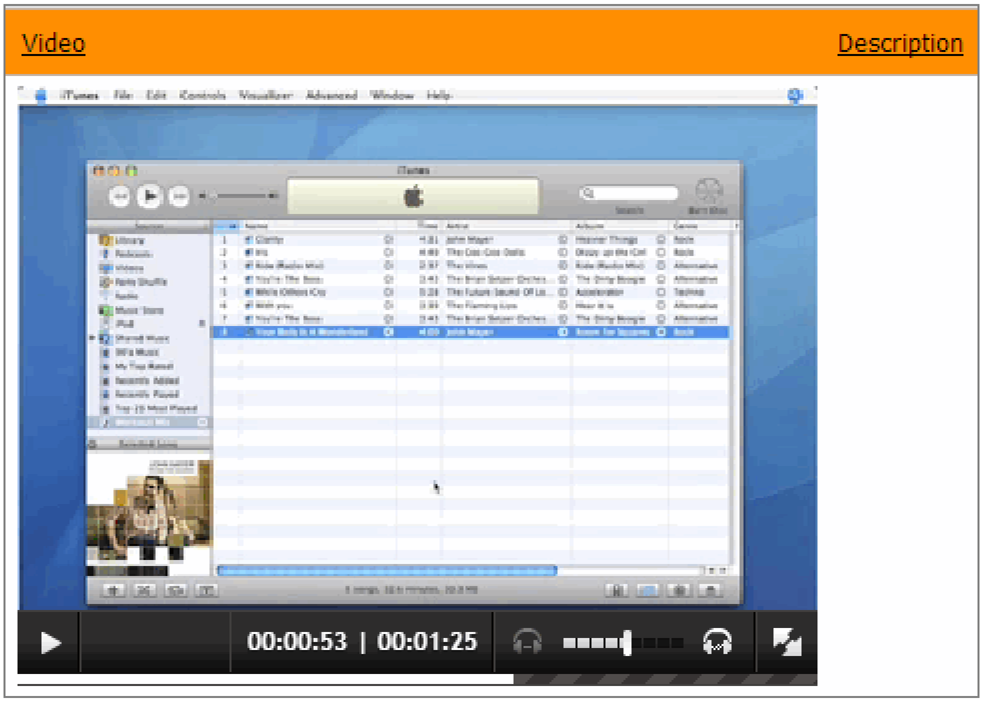
Width (px):	300
Height (px):	200

□ The field contains video file URL

□ Enable rewind/fast forward

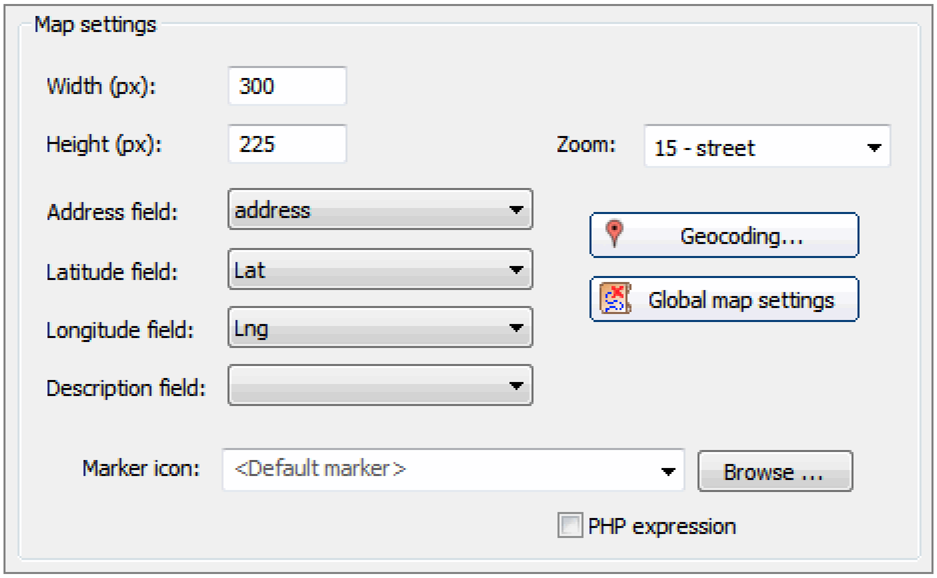
Edit as type can be either *Text field* (enter video file name there) or *File/Image* (upload files).

An example of how video files appear in the browser:



2.16.11.7 Map

Allows you to display location data as a separate map for each table record. To define the location you may use the address or latitude/longitude coordinates. It is highly recommended to use Latitude and Longitude fields to make maps function properly. You may use any widely-used address format (e.g. "1600 Amphitheatre Parkway, Mountain View, CA"). The number given in the **Zoom** field will define the level of zoom on your map.


Map settings

Width (px): 300

Height (px): 225

Zoom: 15 - street

Address field: address

Latitude field: Lat

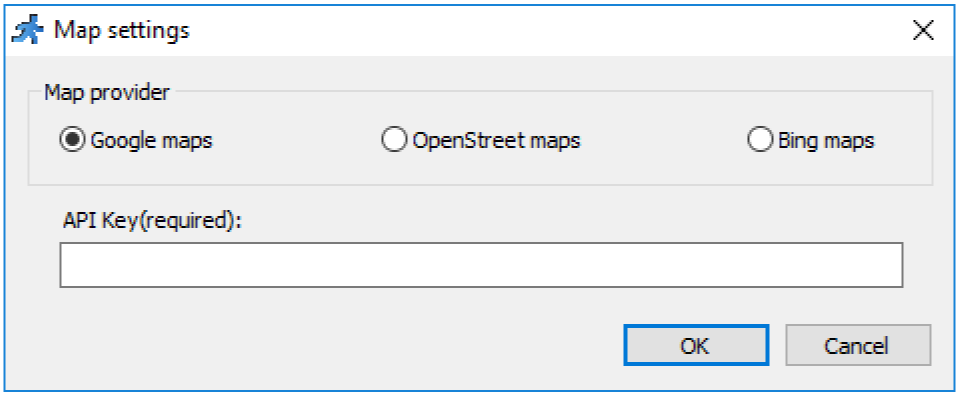
Longitude field: Lng

Description field:

Marker icon: <Default marker>

□ PHP expression

You may access **Global map settings** to select a map provider. Enter API Key (if required). You can get API keys for development purposes for free at https://developers.google.com (for Google maps) or at bingmapsportal.com (for Bing maps).

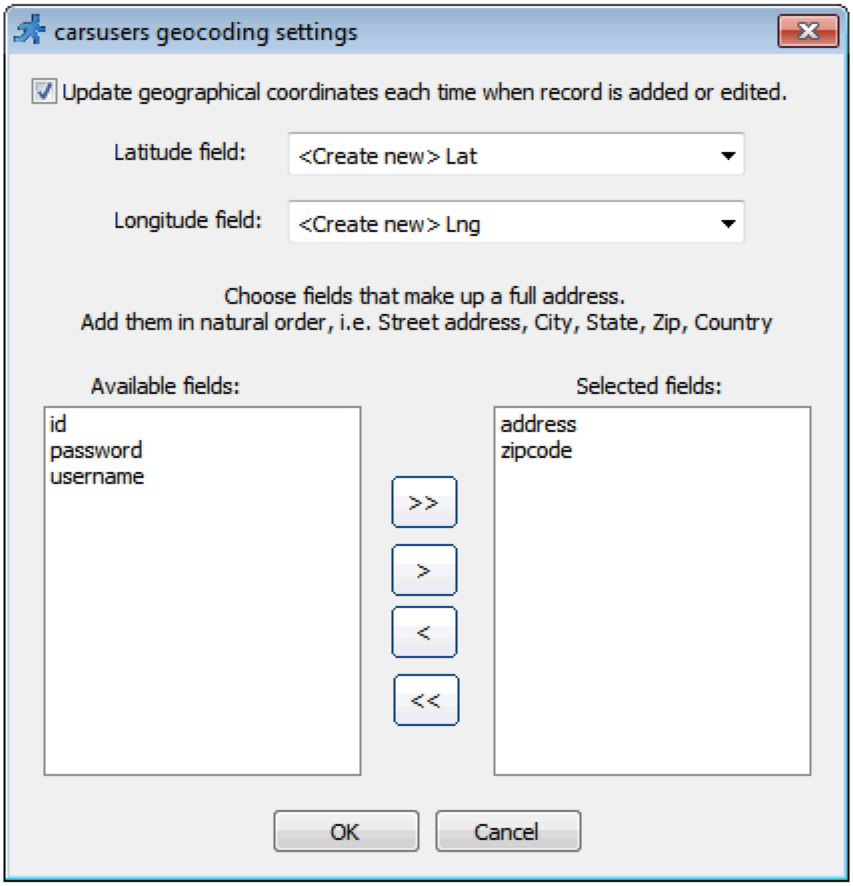

Map settings

Map provider

● Google maps ○ OpenStreet maps ○ Bing maps

API Key(required):

Use the **Geocoding** option to update latitude/longitude information each time a record is added or updated. Select existing fields for storing latitude and longitude data or create new ones. Then select address field(s) to be used during geocoding.


carsusers geocoding settings

☑ Update geographical coordinates each time when record is added or edited.

Latitude field: <Create new> Lat

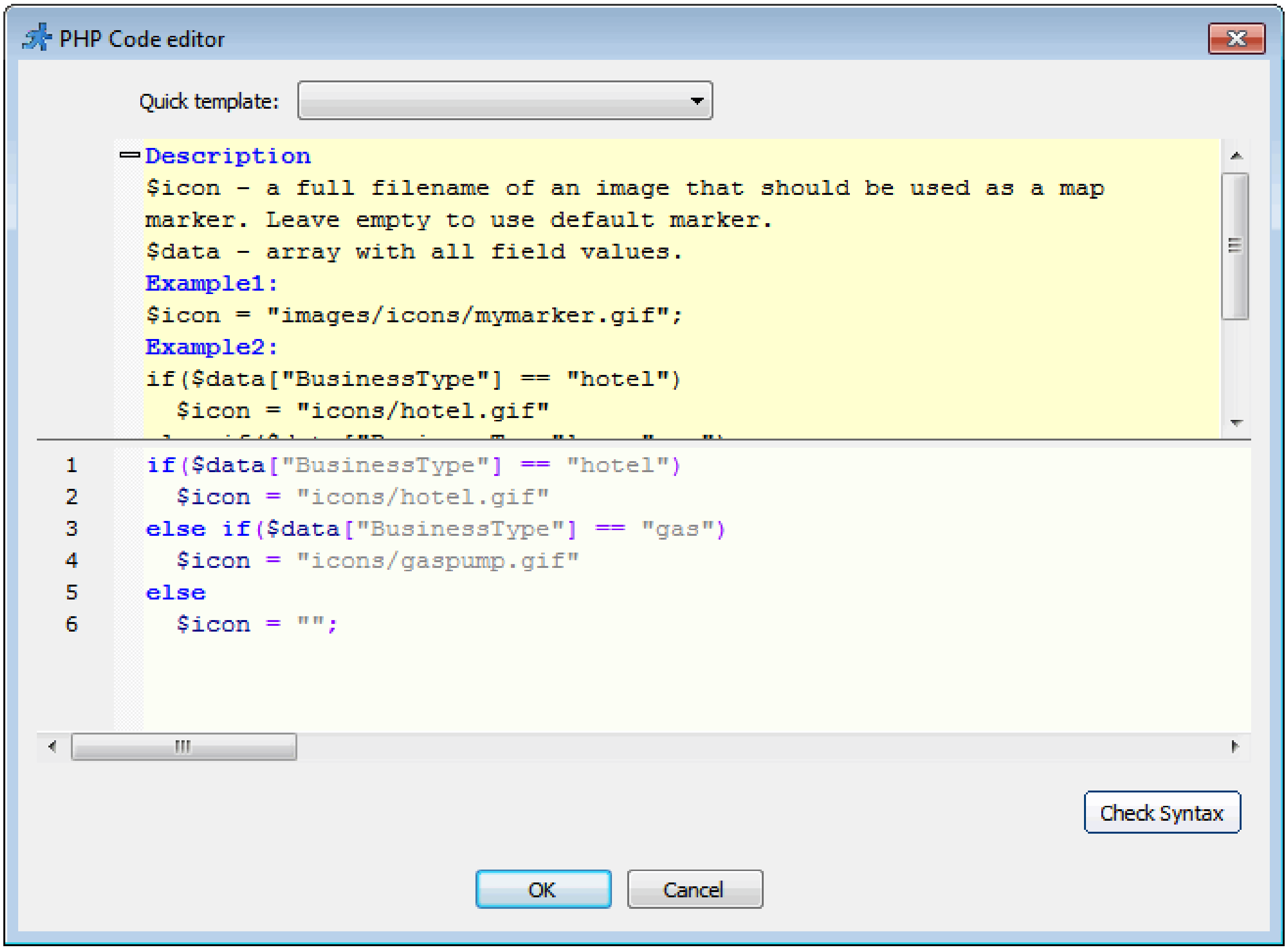
Longitude field: <Create new> Lng

Choose fields that make up a full address.
Add them in natural order, i.e. Street address, City, State, Zip, Country

Available fields: id, password, username

Selected fields: address, zipcode

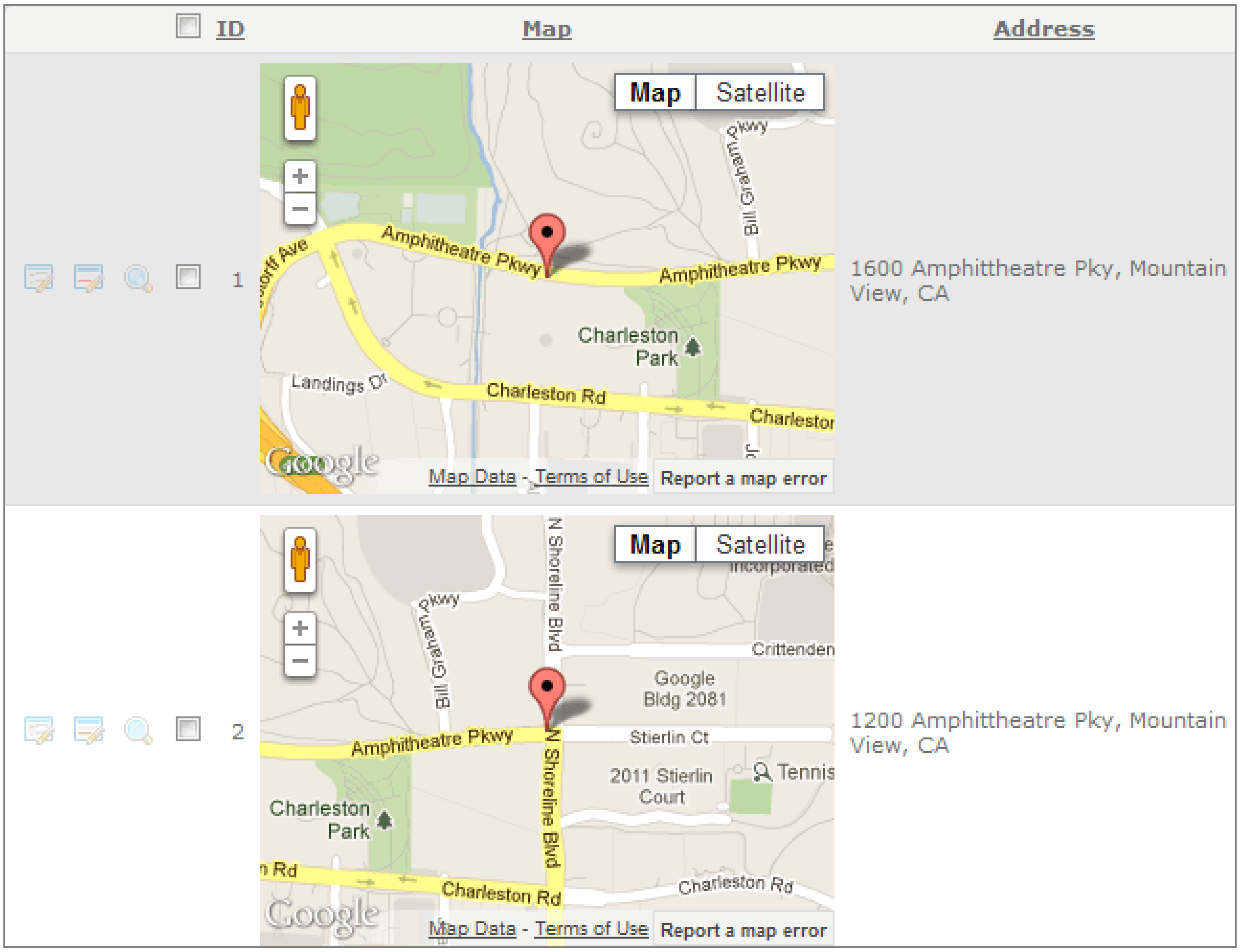
You may set a custom icon for map marker pins. Click **Browse** next to the **Marker icon** and select the icon image file. You may also set different icons for different map objects by using the **PHP expression** option.



```
Description
$icon - a full filename of an image that should be used as a map
marker. Leave empty to use default marker.
$data - array with all field values.
Example1:
$icon = "images/icons/mymarker.gif";
Example2:
if($data["BusinessType"] == "hotel")
  $icon = "icons/hotel.gif"
```

```
if($data["BusinessType"] == "hotel")
  $icon = "icons/hotel.gif"
else if($data["BusinessType"] == "gas")
  $icon = "icons/gaspump.gif"
else
  $icon = "";
```

An example of how maps are displayed in a browser. Note that double left clicking will zoom in the image and double right clicking will zoom out.



□ ID	Map	Address
□ 1		1600 Amphitheatre Pky, Mountain View, CA
□ 2		1200 Amphitheatre Pky, Mountain View, CA

For more information about inserting large maps to the **List** page, see Insert map.

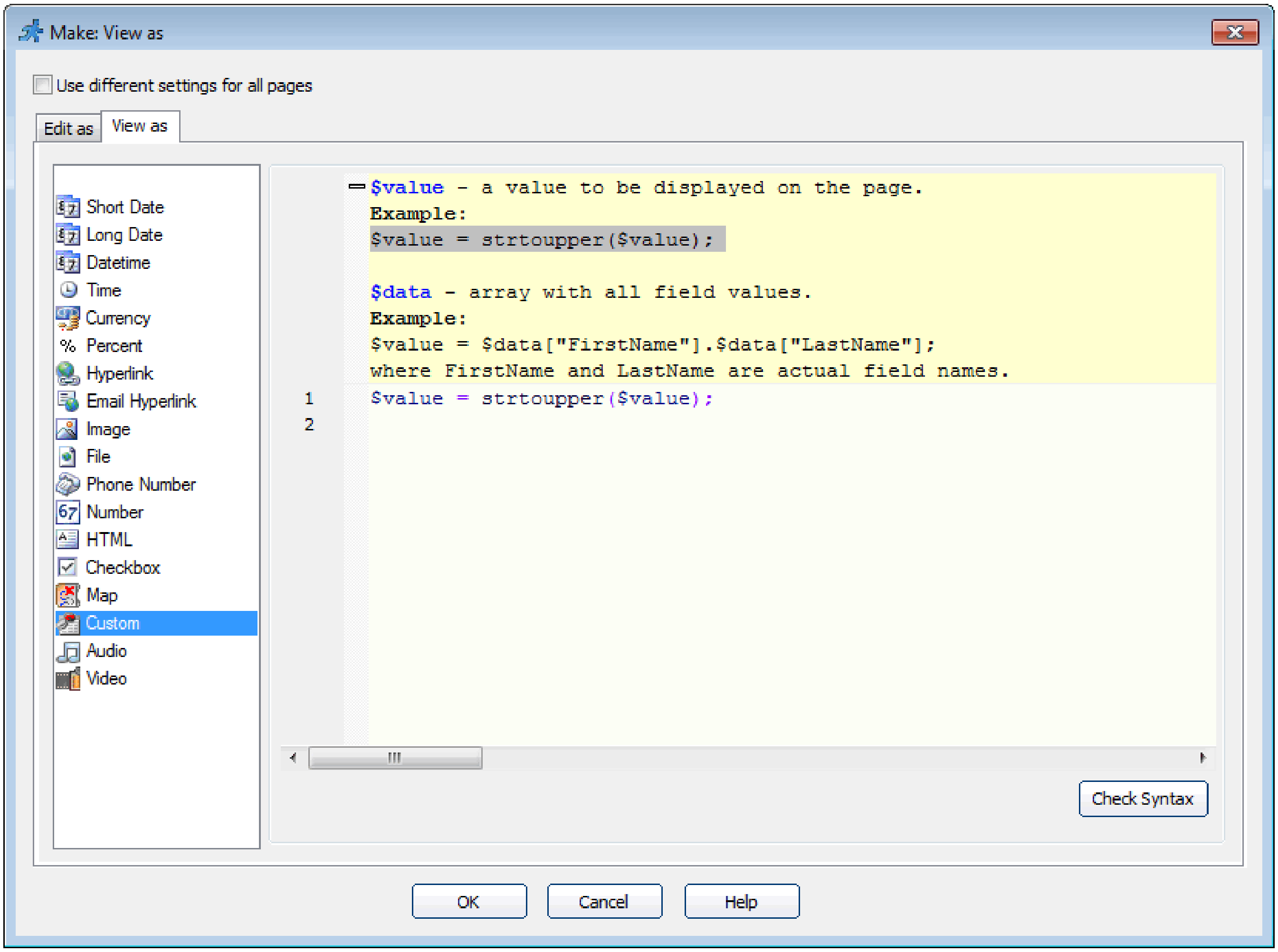
Maps API and geocoding

Maps API providers limit the number of geocoding requests you are allowed to send per day and also limit request rates. For example, if you use Google Maps API and have a page with 20 records with a map for each, only about 10 maps will be displayed properly. To bypass these limitations you need to purchase a fairly expensive Premier account with Google that costs $10,000 per year.

The solution is to use latitude/longitude pairs instead of addresses for mapping purposes. The main question now is how to convert existing addresses to latitude/longitude pairs? For more information, read Insert Map - Geocoding.

2.16.11.8 Custom

You can format field values by adding PHP code.



```
$value - a value to be displayed on the page.
Example:
$value = strtoupper($value);

$data - array with all field values.
Example:
$value = $data["FirstName"].$data["LastName"];
where FirstName and LastName are actual field names.
$value = strtoupper($value);
```

Input value is stored in the variable *$value*. Output value is assigned to the same variable *$value*. You can access other fields of the same data record as *$data["FieldName"]*.

If you chose *Lookup wizard* in the "Edit as" settings, use *$value* to access the Display field value and *$data["LinkFieldName"]* to access the Link field value.

Examples:

1. Convert a string into the upper case:

```
$value = strtoupper($value);
```

2. Format 10-digit phone number into the following format (xxx) xxx-xxx:

```
if (strlen($value)==10)
{
  $value="(" . substr($value,0,3) . ") " . substr($value,3,3) . "-" . substr($valu
}
```

3. Display value of field *FirstName* as *<FirstName> <LastName>* (if *LastName* field defined):

```
if ($data["LastName"])
  $value = $value." ".$data["LastName"];
```

4. Display a number in black color if number is positive and red font if number is negative:

```
if ($value>0)
  $color="black";
else
  $color="red";
$value="<font color='$color'>$value</font>";
```

5. Display a field containing email address as email hyperlink (**mailto** function used). The value of subject in **mailto** function is set to a value of another field:

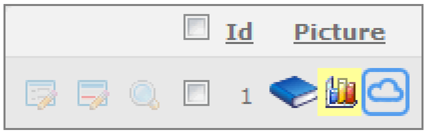
```
$value = "<a href='mailto:".$value."?subject=".$data["SubjectField"]."'>Send emai
```

6. Display a field value in one line (without line breaks):

```
$value = str_replace(" ", "&nbsp;",$value);
```

7. Display all images uploaded via multiupload:

```
$filesArray = my_json_decode($value);
foreach ($filesArray as $imageFile) {
   $imageValue .= "<img alt=\"".htmlspecialchars($imageFile["usrName"])."\"
src=\"".htmlspecialchars($imageFile["name"])."\">";
}
$value = $imageValue;
```



8. Display phone number as click-to-call link for mobile browsers:

```
$value = '<a href="tel:'.$value.'">Call us!</a>';
```

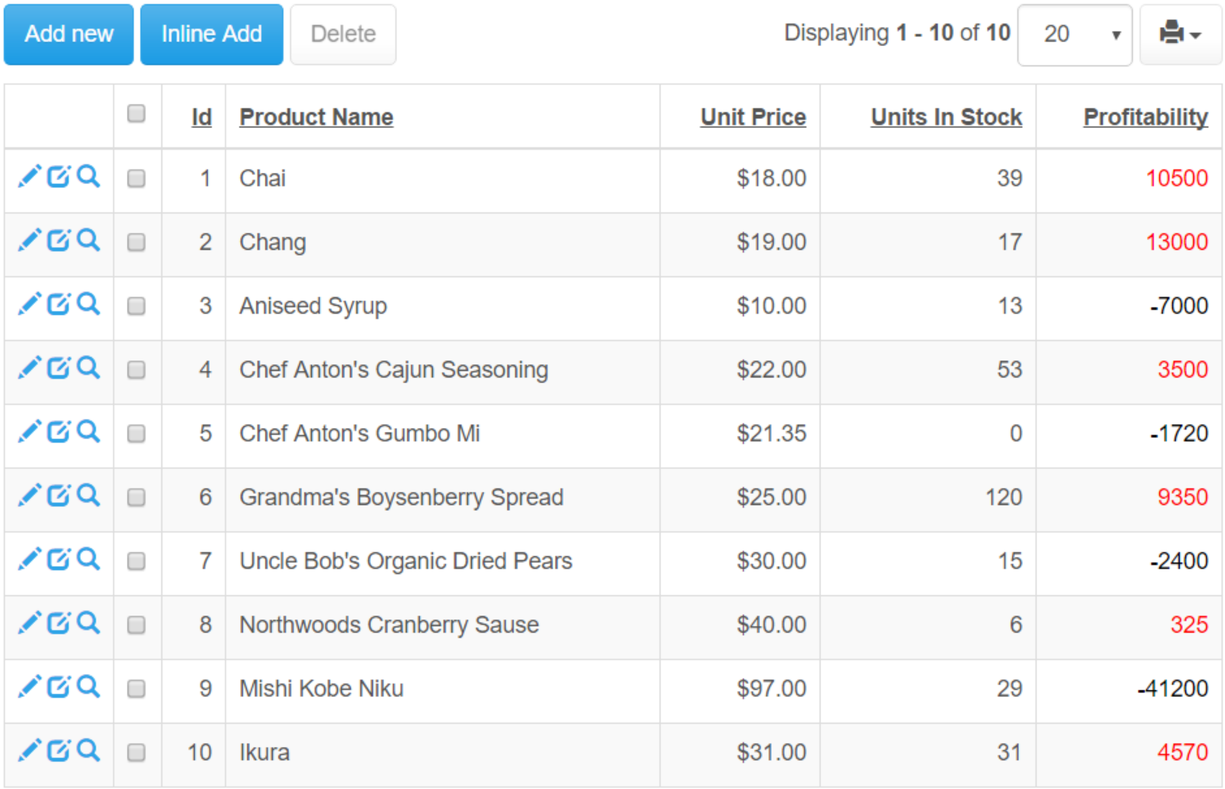
See also

- Conditional formatting

2.16.11.9 Conditional formatting

How to use conditional formatting in PHPRunner projects.

The first example illustrates the profitability of the product line where we have a list of products with their respective monthly profit figures. All positive numbers are displayed in black and all negative in red. Changing the font color of this field based on its value helps the viewer immediately spot the losing products and take necessary actions.



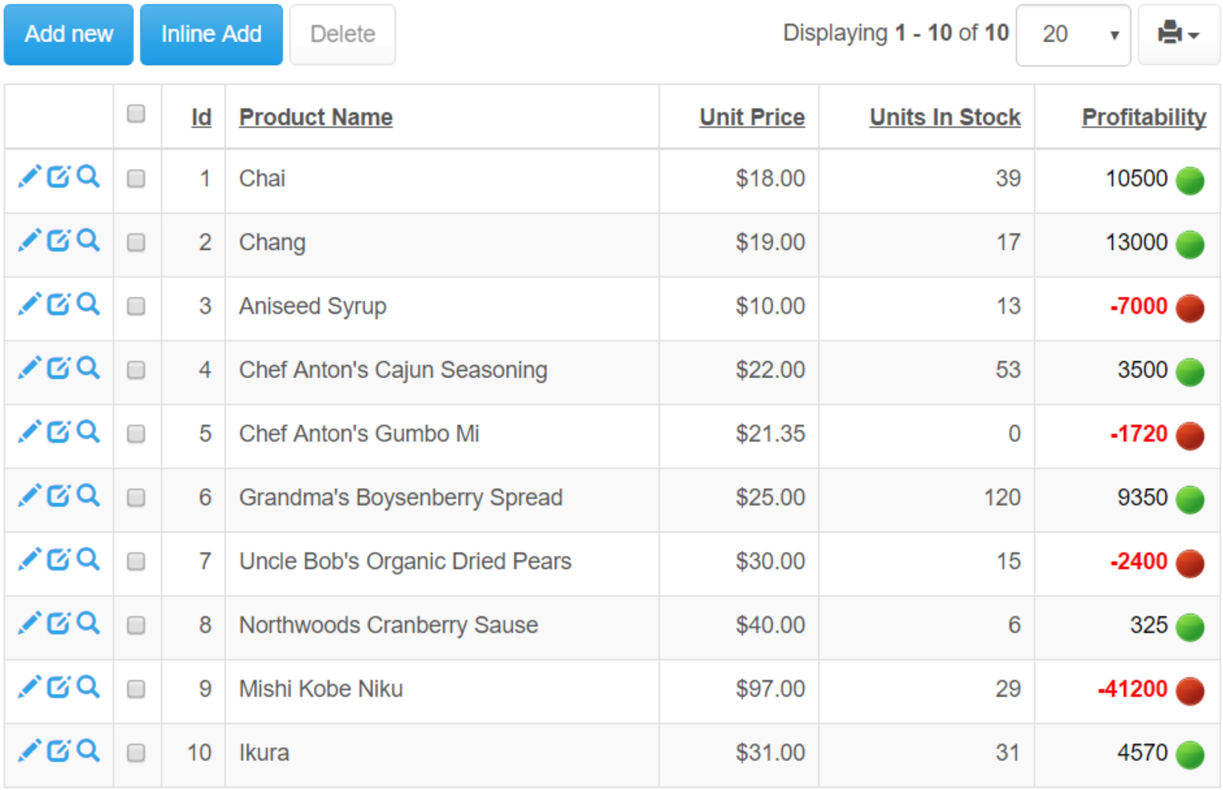
	□	Id	Product Name	Unit Price	Units In Stock	Profitability
	□	1	Chai	$18.00	39	10500
	□	2	Chang	$19.00	17	13000
	□	3	Aniseed Syrup	$10.00	13	-7000
	□	4	Chef Anton's Cajun Seasoning	$22.00	53	3500
	□	5	Chef Anton's Gumbo Mi	$21.35	0	-1720
	□	6	Grandma's Boysenberry Spread	$25.00	120	9350
	□	7	Uncle Bob's Organic Dried Pears	$30.00	15	-2400
	□	8	Northwoods Cranberry Sause	$40.00	6	325
	□	9	Mishi Kobe Niku	$97.00	29	-41200
	□	10	Ikura	$31.00	31	4570

To set the formatting proceed to Visual Editor and double-click on the field you want to format. In our case it is the field "Profitability". Select 'Custom' View As option. Add your code in the custom code editor. If the value of the current field is greater than zero, the font color will be black, otherwise red. Always remember to check the syntax of your code.

```
if ($value > 0) {
     $color="black";
} else {
   $color="red";
}
$value="<span style='color: " . $color . "'>" .$value . "</span>";
```

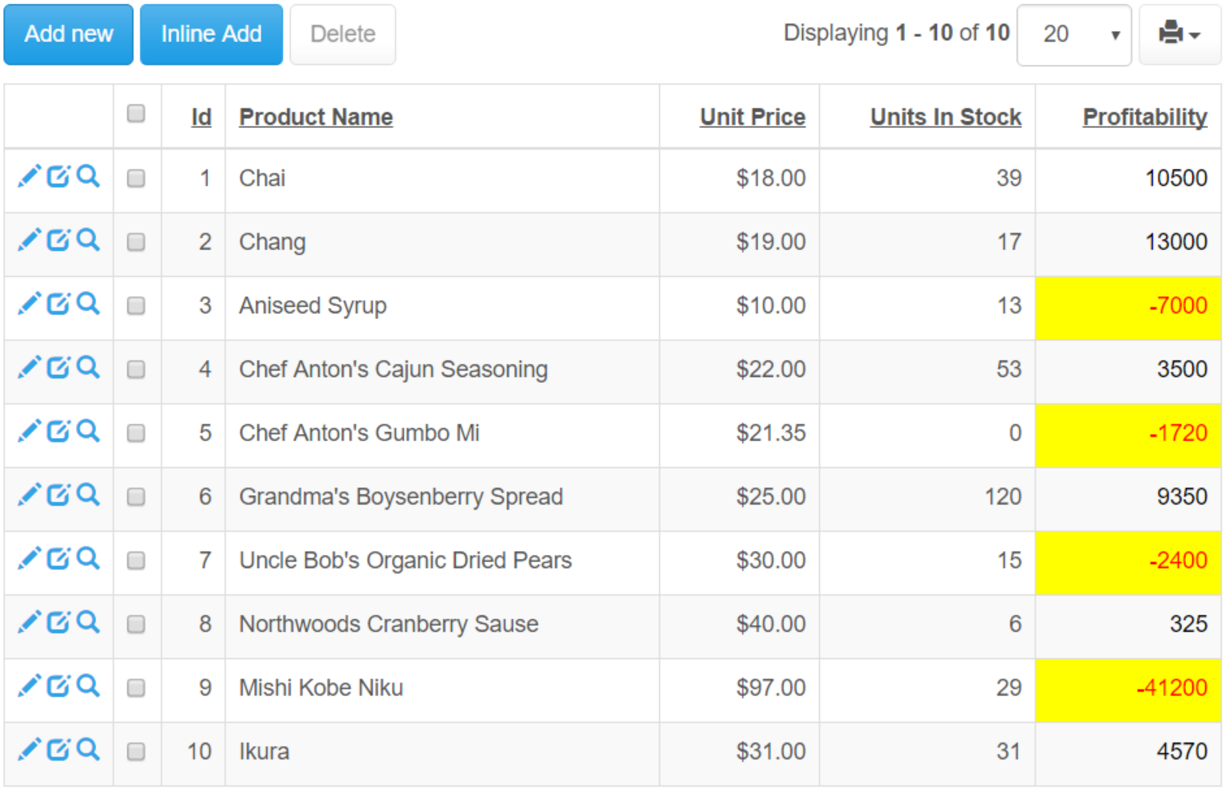
You can also represent the positive and negative results with images. For example there are the red and green circles were added to highlight the performance of each product. To make this work you will simply need to add two more lines to your custom code. Don't forget to place the actual images in the project folder.

```
if ($value > 0) {
  $value =$value. ' <img src="green.png" alt="" />';
   $color="black";
} else {
   $value ='<strong>'.$value. '</strong> <img src="red.png" alt='' />';
   $color="red";
}
$value="<span style='color: " . $color . "'>" . $value . "</span>";
```



	□	Id	Product Name	Unit Price	Units In Stock	Profitability
✎ ✎ 🔍	□	1	Chai	$18.00	39	10500 🟢
✎ ✎ 🔍	□	2	Chang	$19.00	17	13000 🟢
✎ ✎ 🔍	□	3	Aniseed Syrup	$10.00	13	**-7000** 🔴
✎ ✎ 🔍	□	4	Chef Anton's Cajun Seasoning	$22.00	53	3500 🟢
✎ ✎ 🔍	□	5	Chef Anton's Gumbo Mi	$21.35	0	**-1720** 🔴
✎ ✎ 🔍	□	6	Grandma's Boysenberry Spread	$25.00	120	9350 🟢
✎ ✎ 🔍	□	7	Uncle Bob's Organic Dried Pears	$30.00	15	**-2400** 🔴
✎ ✎ 🔍	□	8	Northwoods Cranberry Sause	$40.00	6	325 🟢
✎ ✎ 🔍	□	9	Mishi Kobe Niku	$97.00	29	**-41200** 🔴
✎ ✎ 🔍	□	10	Ikura	$31.00	31	4570 🟢

Now, lets say we want to highlight not only the font color itself but also change the background of the cell to yellow.



Displaying **1 - 10 of 10**

	□	Id	Product Name	Unit Price	Units In Stock	Profitability
	□	1	Chai	$18.00	39	10500
	□	2	Chang	$19.00	17	13000
	□	3	Aniseed Syrup	$10.00	13	-7000
	□	4	Chef Anton's Cajun Seasoning	$22.00	53	3500
	□	5	Chef Anton's Gumbo Mi	$21.35	0	-1720
	□	6	Grandma's Boysenberry Spread	$25.00	120	9350
	□	7	Uncle Bob's Organic Dried Pears	$30.00	15	-2400
	□	8	Northwoods Cranberry Sause	$40.00	6	325
	□	9	Mishi Kobe Niku	$97.00	29	-41200
	□	10	Ikura	$31.00	31	4570

Because we are no longer working with just the field values, but with the table structure, we will have to use the events to set the background color.

Proceed to the Events screen in the software and select 'After record processed' event for the list page of the Products table. The code will be similar to the one we used earlier. If the value of the 'Profitability' field is less than zero and will set the background of the cell to yellow. You can refer to the software manual for more examples and code syntax.

```
if ($data["Profitability"]<0)
    $record["Profitability_css"].='background:yellow';
```

Let’s say you want to take it a step further and highlight the entire row of the losing products.

In the same event you can use the same code for the condition check and only change the code for setting the background of the row. You code should look like this. Once again you can find the exact syntax for this example in the software manual.

```
if ($data["Profitability"] < 0)
    $record["css"]="background:yellow;";
```

And if you want to change the background color of all rows regardless of the condition, you can simply comment out the condition in this event:

```
    $record["css"]="background:yellow;";
```

See also

- The video tutorial
- How to change row background color
- How to change cell background color

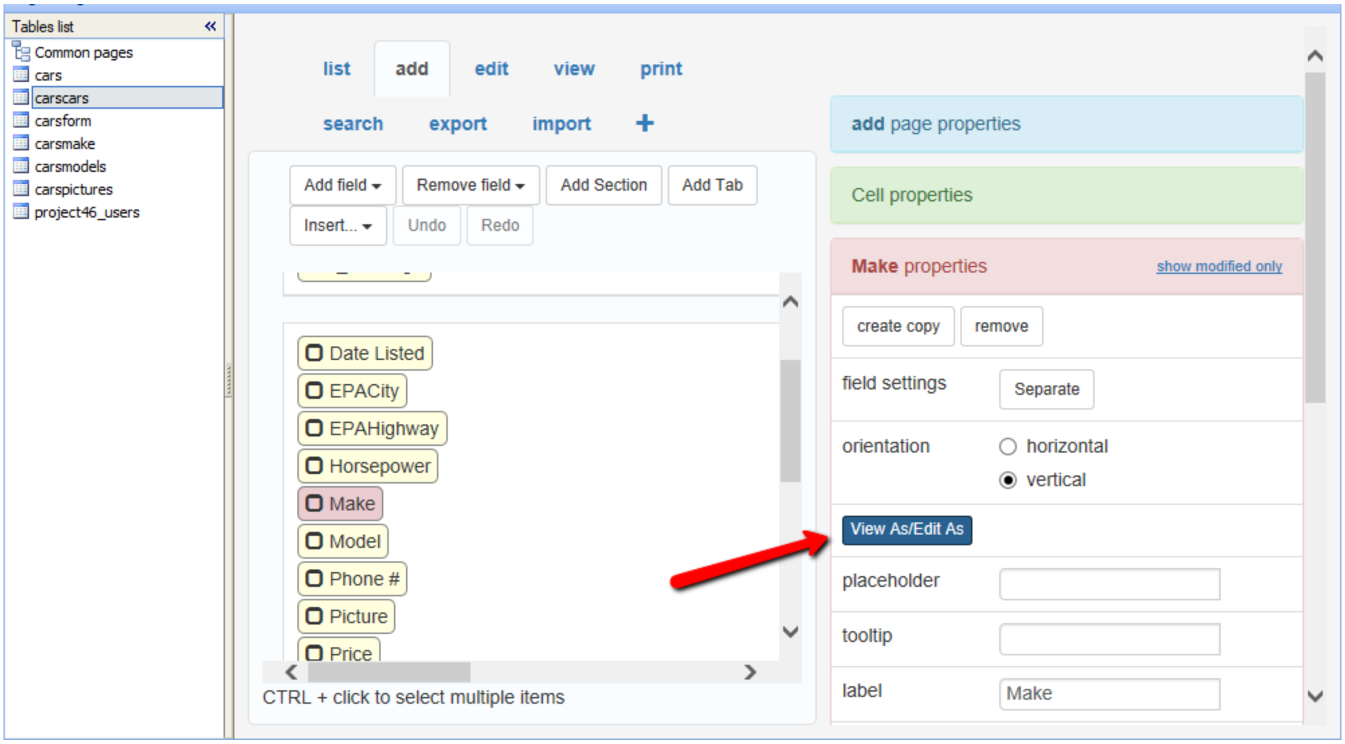
2.16.12 "Edit as" settings

2.16.12.1 "Edit as" settings

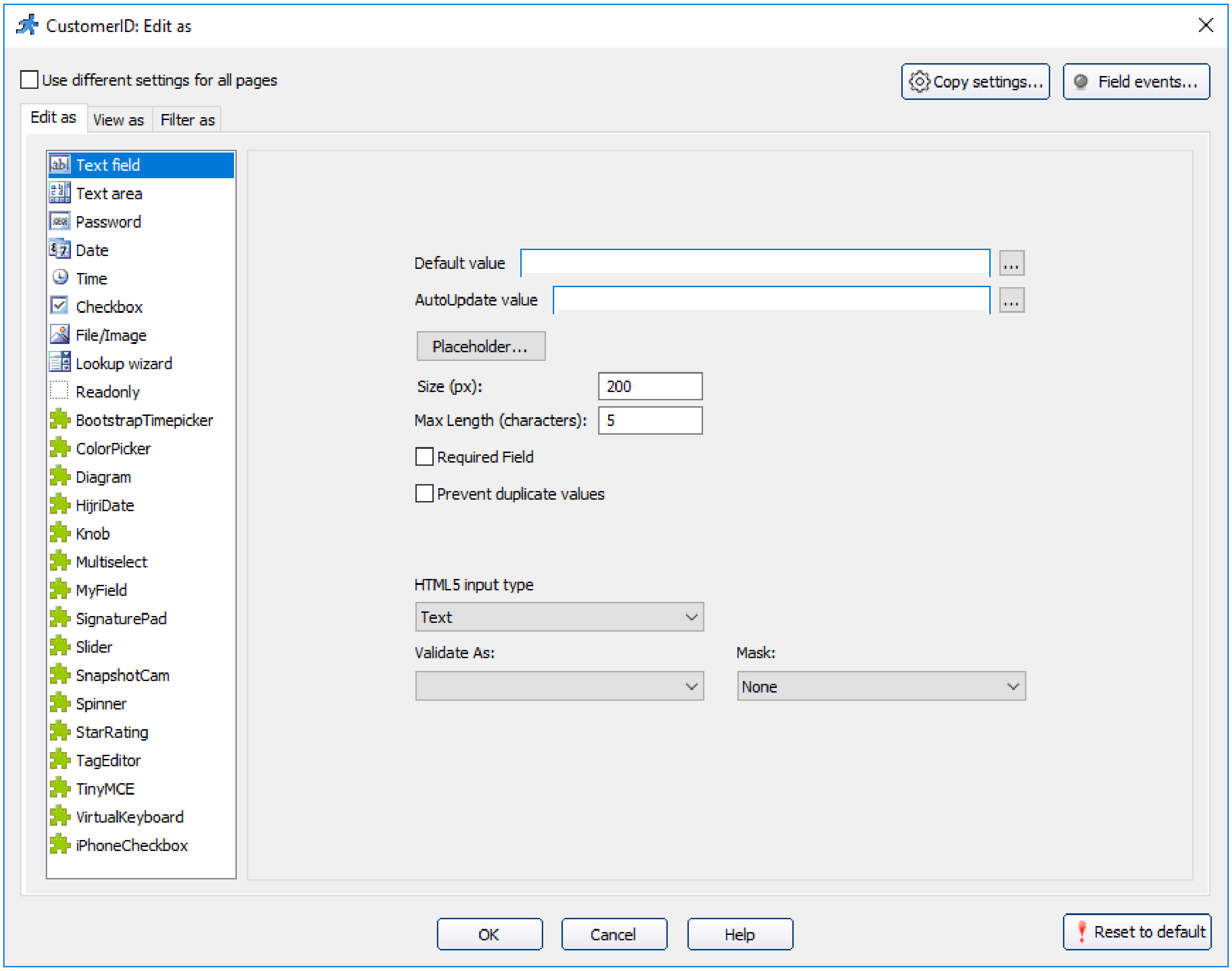
Quick jump		
Text field	Text area	Password
Date	Time	Checkbox
File/Image	Lookup wizard	Readonly
ColorPicker	SignaturePad	

You can customize your data appearance on the Add/Edit/Search using formatting options on the **"Edit as" settings** dialog. You can define your field settings for all pages at once or separately for each page.

To control the filed appearance on the page right-click the field and choose **Properties**. Note that fields format you've chosen is preserved on the Advanced search page, for example, drop-down boxes, date-pickers, check boxes formats.



To set different field formats for different pages enable **Use different settings for all pages** check box. Depending on selected format you will see different box-dialogs.



Size (px): 200

Max Length (characters): 5

□ Required Field

□ Prevent duplicate values

HTML5 input type: Text

Mask: None

Quick jump

Text field	Text area	Password
Date	Time	Checkbox
File/Image	Lookup wizard	Readonly
ColorPicker	SignaturePad	

Common options for "Edit as" formats

Validate as

For more information about validation, see Validation types.

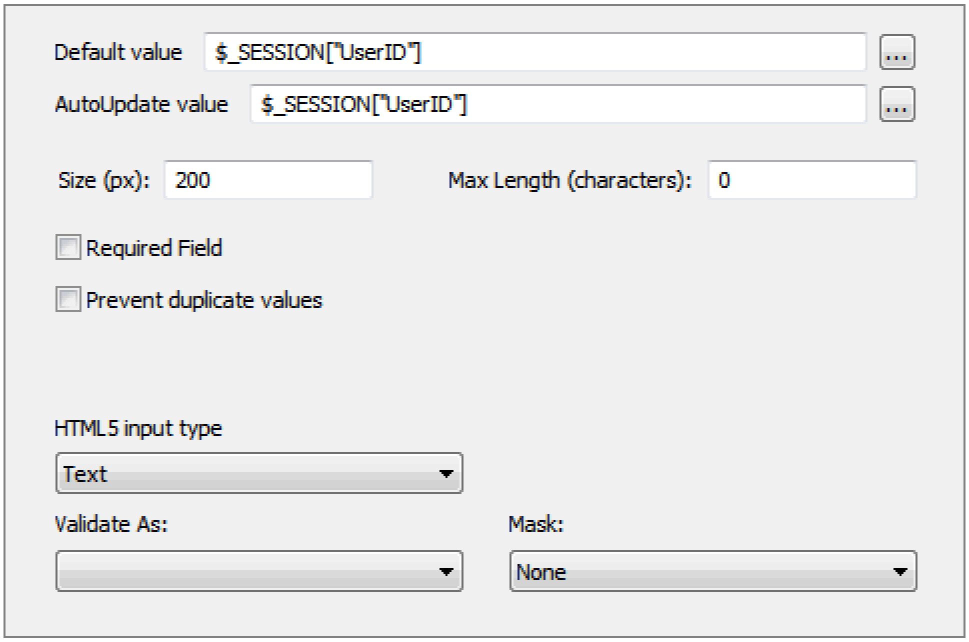
Default value

Default value will be assigned to a field directly on the Add/Search pages. Default value should be a valid PHP expression. Text expressions must be quoted.

Example of default value	Description
35	Number.
"ABC"	Text.
now()	Current date/time.
$_SESSION["UserID"]	Username of the person who created or updated the record.
$_SERVER['REMOTE_ADDR']	IP address.

AutoUpdate value

AutoUpdate value will be assigned to a field every time record is updated on the Edit page. You can use this feature to keep track of who and when updated the record. AutoUpdate value should be a valid PHP expression. Text expressions must be quoted.



Default value $_SESSION["UserID"]

AutoUpdate value $_SESSION["UserID"]

Size (px): 200

Max Length (characters): 0

□ Required Field

□ Prevent duplicate values

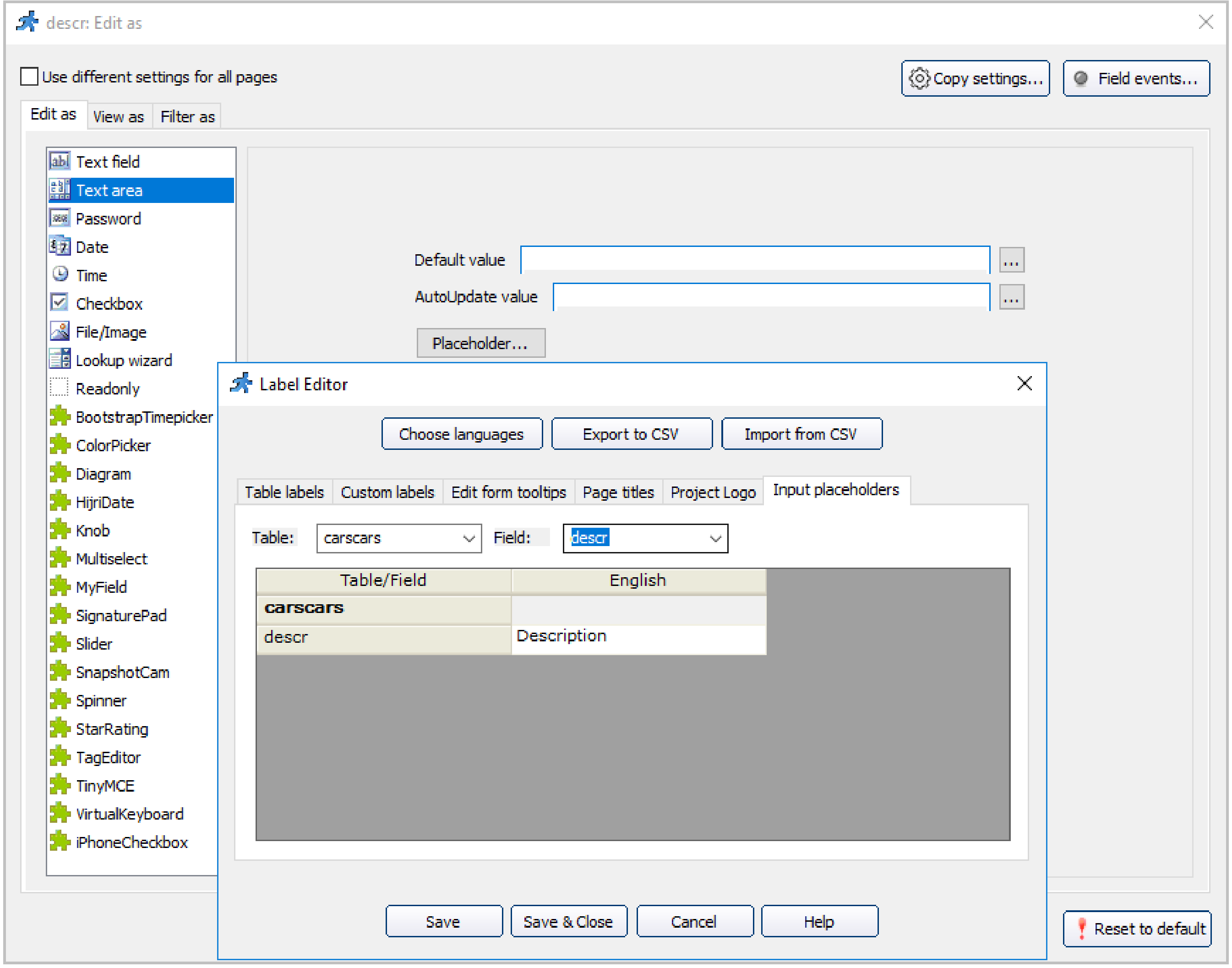
HTML5 input type: Text

Validate As:

Mask: None

Placeholder

Placeholders are in-place tooltips that disappear as soon user starts typing something in that field.



☐ Use different settings for all pages

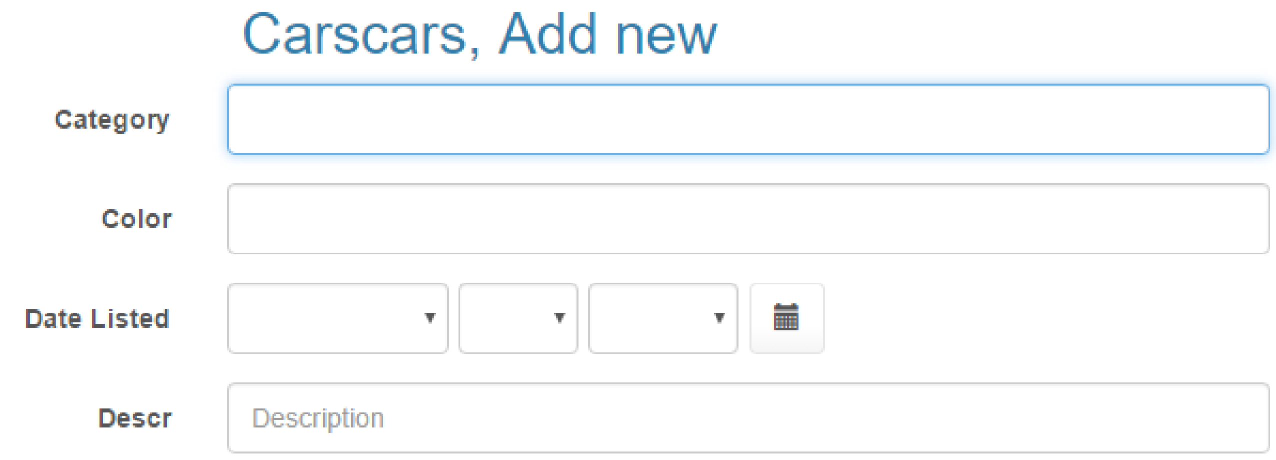
Default value

AutoUpdate value

Table: carscars Field: descr

Table/Field	English
carscars	
descr	Description

On the generated page placeholder looks like this:



Carscars, Add new

Category

Color

Date Listed

Descr: Description

You can change placeholders also in the Label Editor.
Feel free to check the following live demo that showcases placeholders.

Prevent duplicate values

Use this option to prevent users from entering duplicate values. Enter a message that will be displayed when users enter value that already exists in database to **Message** field.

Example of AutoUpdate value	Description
35	Number.
"ABCDEF"	Text.
now()	Current date/time.
$_SESSION["UserID"]	Username of the person who created or updated the record.
$_SERVER['REMOTE_ADDR']	IP address.

Copy settings option allows you to copy field settings from another field.

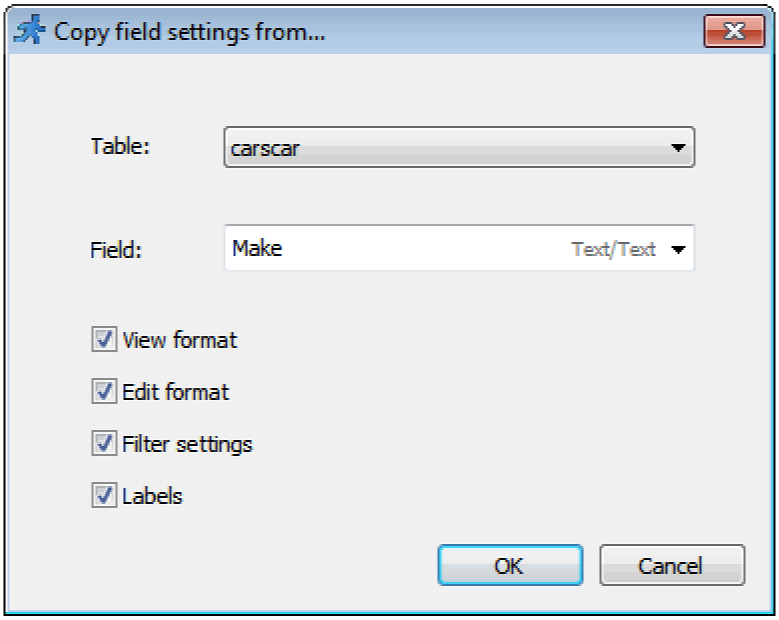


Table: carscar

Field: Make Text/Text

☑ View format

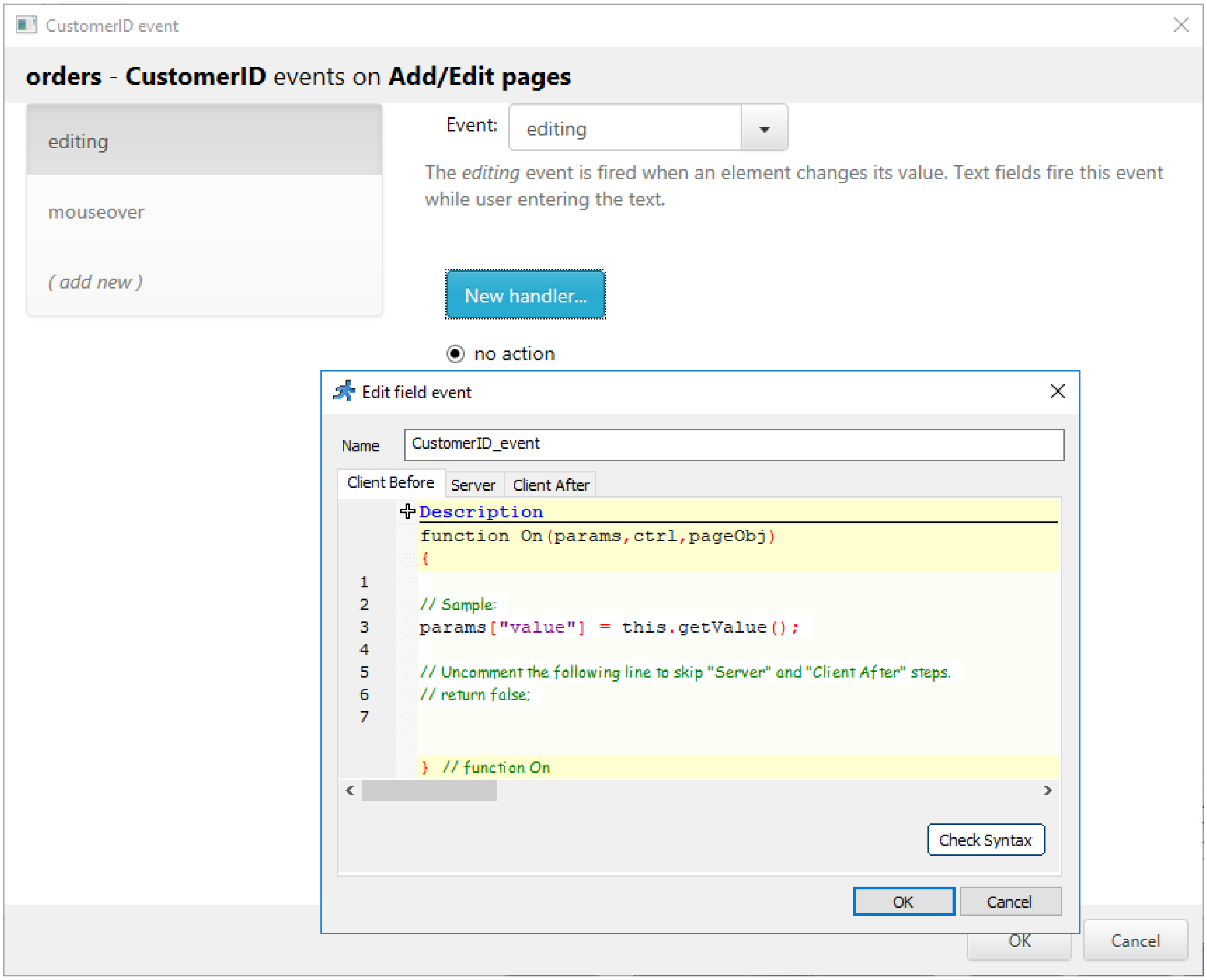
☑ Edit format

☑ Filter settings

☑ Labels

Field events option allows to perform an action when cursor enters edit field or leaves it or mouse is over a field. Perform any sort of validation, make other fields hidden or required etc. Designed to work on Add, Edit, View and Register pages.
For example, the *editing* event is fired when an element changes its value. Text fields fire this event while user entering the text

See also How to access fields in the field events



orders - **CustomerID** events on **Add/Edit pages**

Event: editing

The *editing* event is fired when an element changes its value. Text fields fire this event while user entering the text.

Name: CustomerID_event

Description

```
function On(params,ctrl,pageObj)
{

// Sample:
params["value"] = this.getValue();

// Uncomment the following line to skip "Server" and "Client After" steps.
// return false;


} // function On
```

Text field

Simple text box. For more information, see "Edit as" settings: Text field.

Text area

Multiline text area. For more information, see "Edit as" settings: Text area.

Password

Password field. All entered characters appear as "*".

Date

Date edit control. For more information, see "Edit as" settings: Date.

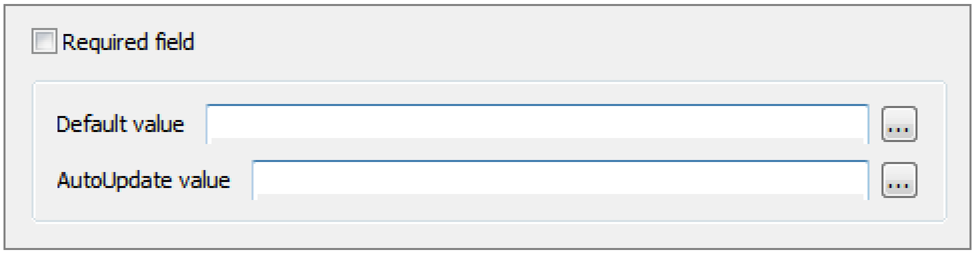
Time

Time edit control. For more information, see "Edit as" settings: Time.

Checkbox

Check box control. Works best with the following data types:

- MS Access: *Yes/No* field;
- SQL Server: *TINYINT* or *BIT* field;
- MySQL: *TINYINT* field;
- Oracle: *NUMBER(1)* field.



□ Required field

Default value

AutoUpdate value

Select the **Required field** option to make the check box required so that it should be selected before submitting a form. Use this option for the fields such as the Accept License Agreement and Terms.

File/Image

Choose this format if you store files/images in this field. For more information, see "Edit as" settings: File/Image.

Lookup wizard

Drop-down box with list of values. For more information, see "Edit as" settings: Lookup wizard.

Readonly

Use this format for a field that should not be edited.

ColorPicker

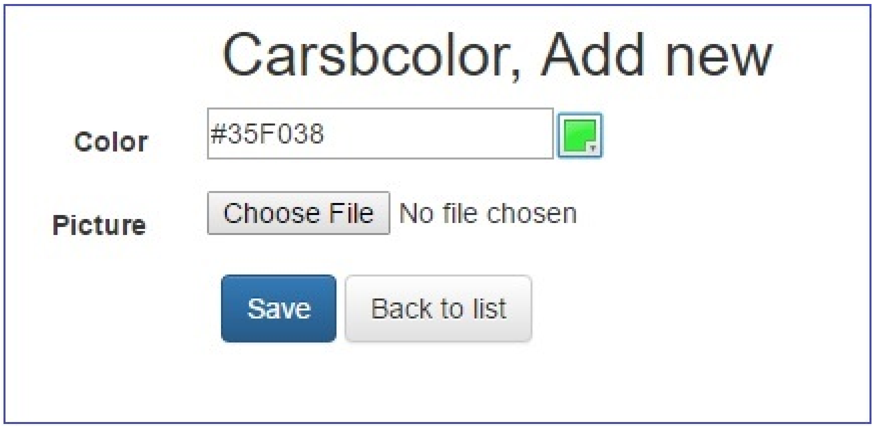
The Color picker control that allows users select color the same way they do in Adobe Photoshop. For more information, see "Edit as" settings: ColorPicker.

SignaturePad

The SignaturePad control allows you to add signature pad to your forms. For more information, see "Edit as" settings: SignaturePad.

2.16.12.2 ColorPicker

Color picker control allows users select color the same way they do in Adobe Photoshop. SignaturePad works with both mouse and touch devices. You may select **ColorPicker** as *Edit as* type for any text field and enjoy your color picker on Add/Edit pages.



Carsbcolor, Add new

Color: #35F038 🟩

Picture: Choose File No file chosen

Click **Add initialization script** button to customize the ColorPicker control.

Note that Color picker control is a custom plugin. From now on you also can create your own Edit control plugins. For more information, see How to create your own Edit control.

You may consider some additional enhancements:

1. It would be nice if instead of hex color value we can show some visual representation of selected color on List/View pages. We'll do so choosing 'View as' type 'Custom' and putting the following code there:

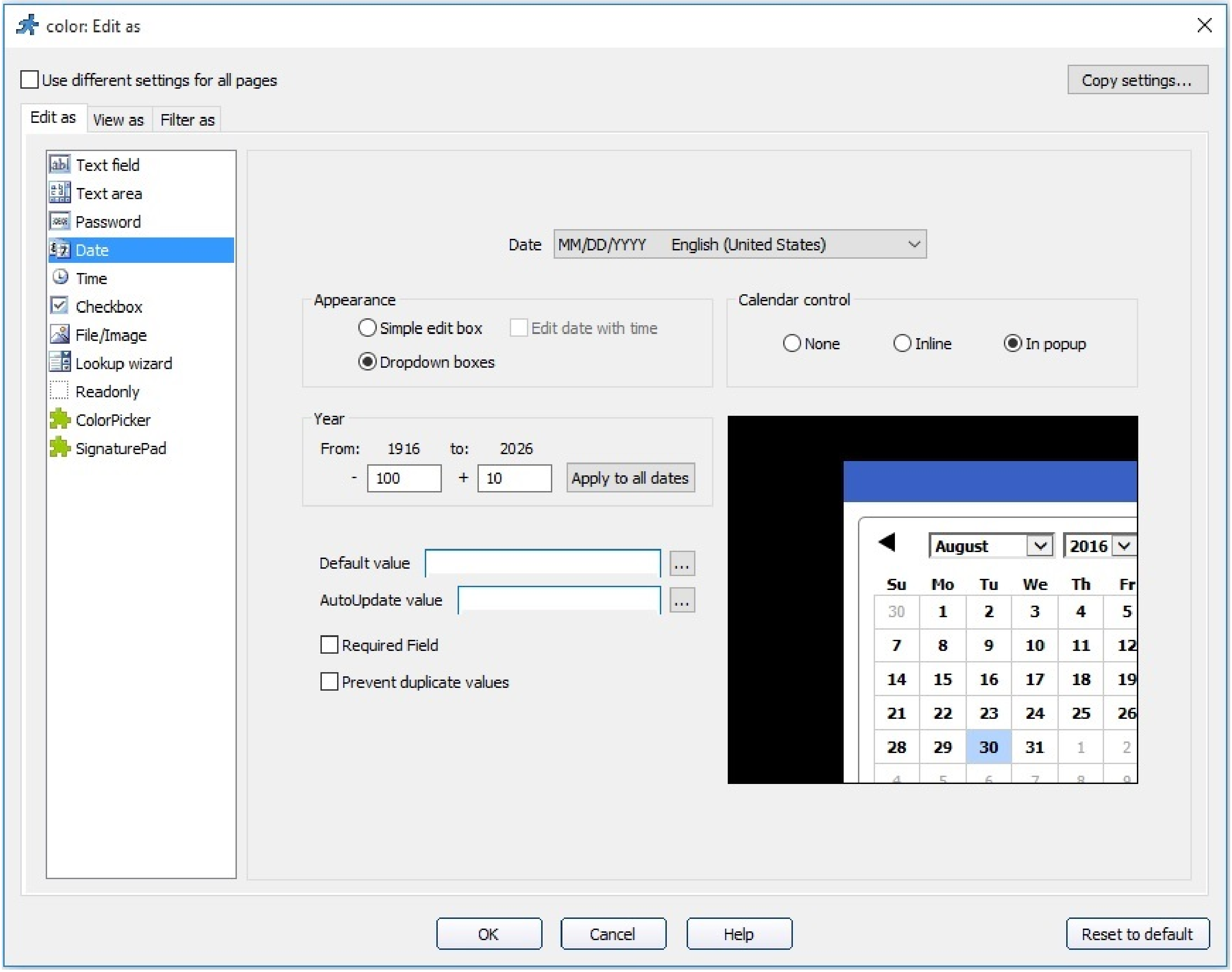
```
$value="<span style='padding: 5px; background:
".$value."'>".$value."</span>";
```

2. By default PHPRunner sets focus to the first edit control when Add or Edit page is loaded. This is not a desired behaviour for colorpciker control as we do not want to see colorpicker popup window to open every time page is loaded. To prevent this from happening implement [setFocus] function - simply return false every time.

2.16.12.3 Date

Date edit control. Choose between:

- simple edit box;
- simple edit box with datepicker (inline or popup mode);
- dropdown boxes with day, month and year selection;
- dropdown boxes with datepicker (inline or popup mode).



color: Edit as

□ Use different settings for all pages

Edit as | View as | Filter as

Text field
Text area
Password
Date
Time
Checkbox
File/Image
Lookup wizard
Readonly
ColorPicker
SignaturePad

Date: MM/DD/YYYY English (United States)

Appearance
○ Simple edit box □ Edit date with time
● Dropdown boxes

Calendar control
○ None ○ Inline ● In popup

Year
From: 1916 to: 2026
- 100 + 10

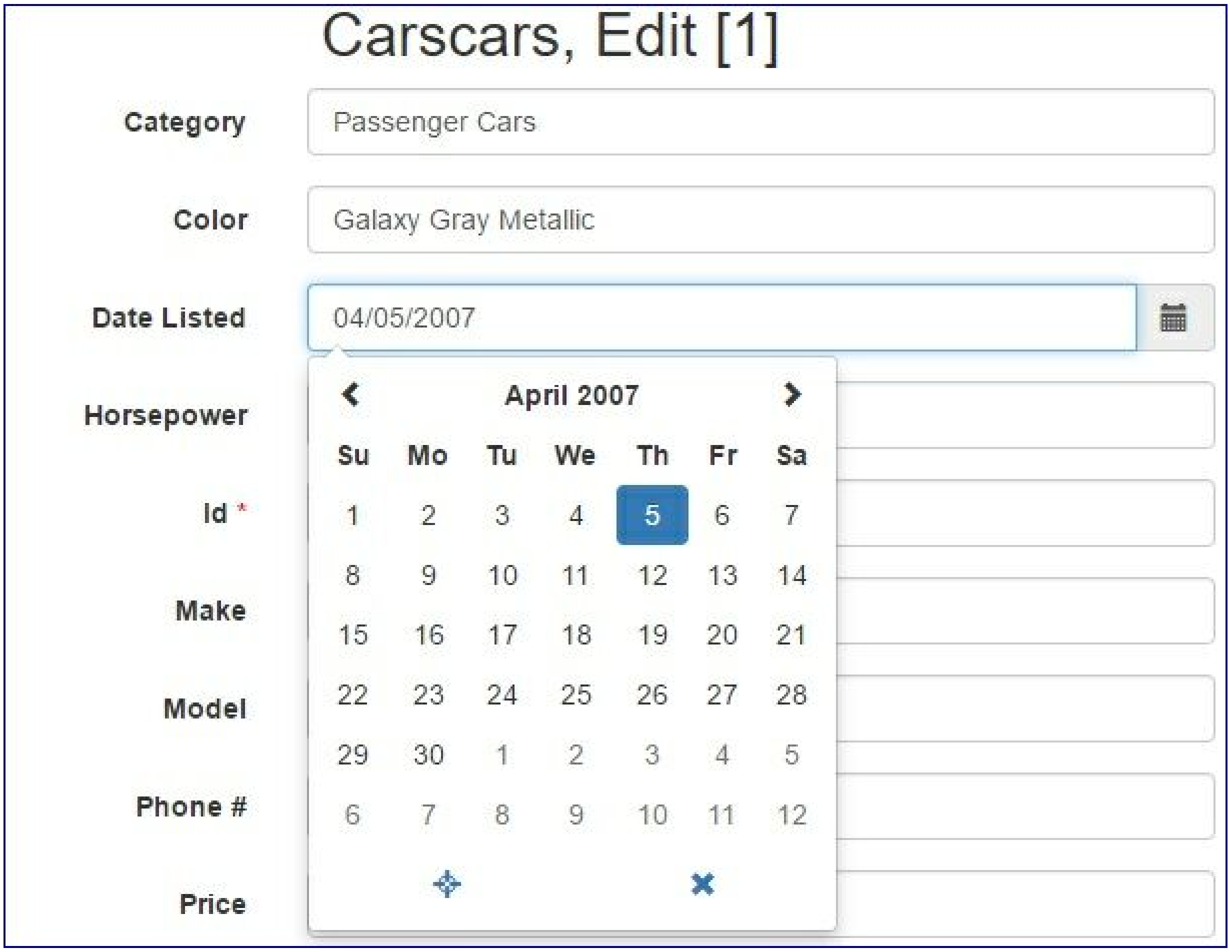
Default value:
AutoUpdate value:

□ Required Field
□ Prevent duplicate values

You can set start and end year range. Simple edit box also allows to enter time.

Default value will be assigned to a field directly on the Add/Search pages. AutoUpdate value will be assigned to a field every time record is updated on the Edit page. You can use this feature to keep track of who and when updated the record. Default and AutoUpdate values should be valid PHP expressions.

Simple edit box with datepicker (inline mode):



Carscars, Edit [1]

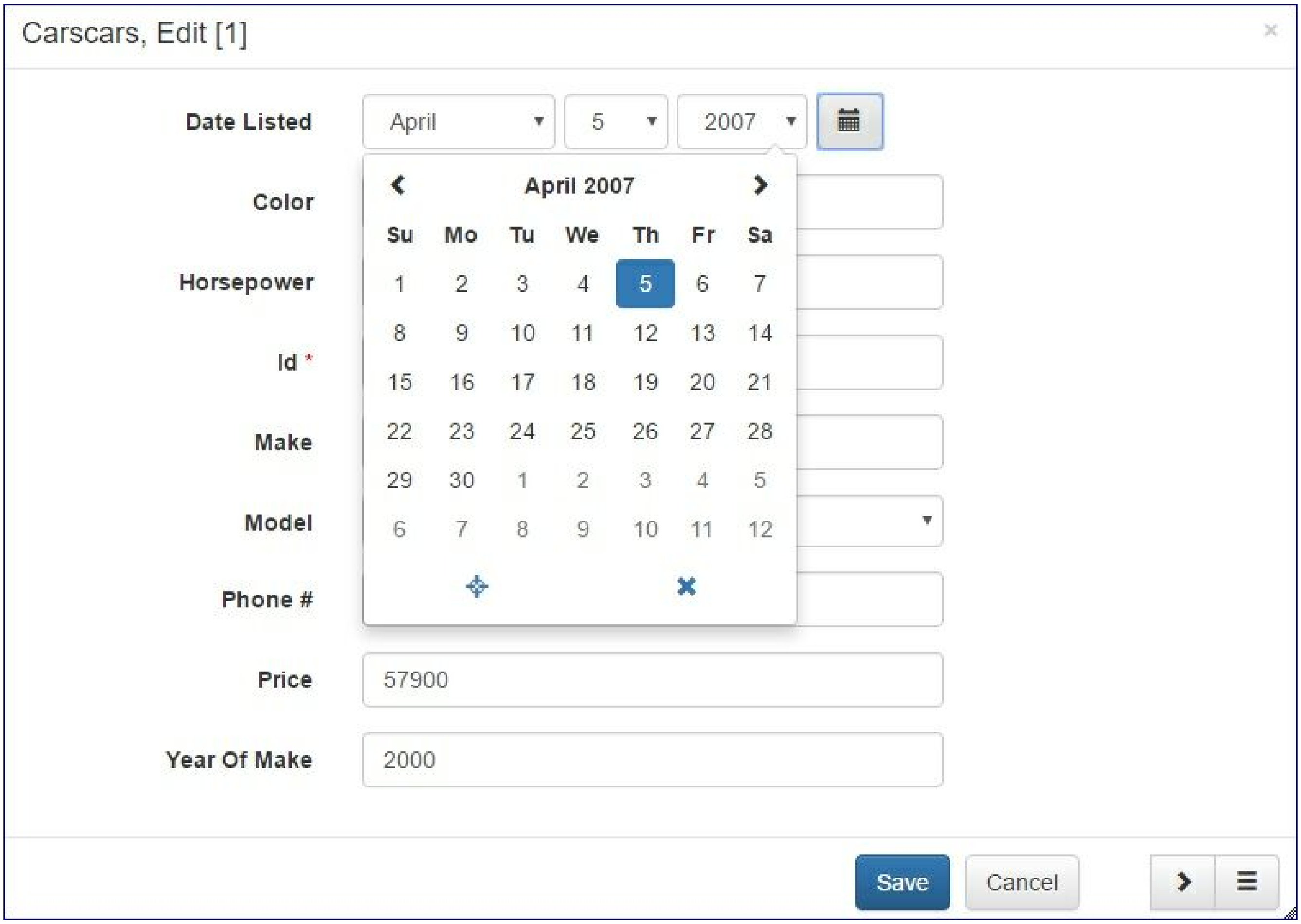
Category	Passenger Cars
Color	Galaxy Gray Metallic
Date Listed	04/05/2007
Horsepower	
Id *	
Make	
Model	
Phone #	
Price	

April 2007

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

1

Dropdown boxes with datepicker (popup mode):



Carscars, Edit [1]

Date Listed: April 5 2007

April 2007

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Color

Horsepower

Id *

Make

Model

Phone #

Price: 57900

Year Of Make: 2000

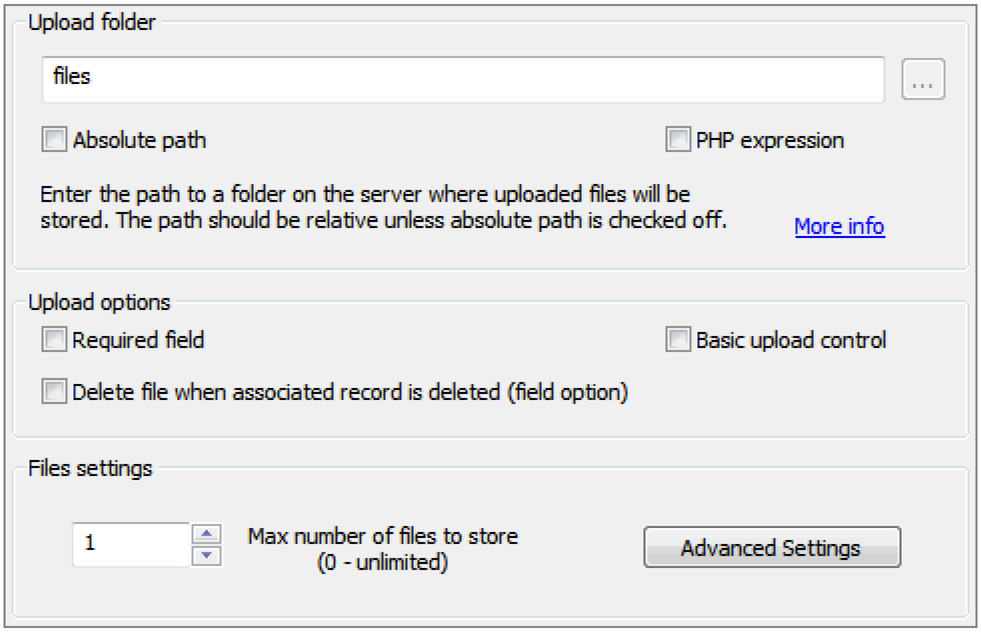
2.16.12.4 File/Image

Depending on the field type this control allows uploading images and files to the database (binary field) or to some directory on the Web server (text field).

Text field

File/Image upload control allows selecting and uploading multiple files at once. You only need one database field to store all file names. For multiple file upload long text field is recommended, for example, Memo in MS Access, Mediumtext in MySQL, TEXT or Varchar(max) in SQL Server.

Since images and files will be uploaded to some directory on the server, you need to enter the path to that directory. The path should be relative to the folder with generated pages. Since some hosting providers do not allow making reference to the directories, which are above the current one, you can use the **Absolute path** option and specify the full path to the directory on the server, in which files will be uploaded. E.g. *C:\Inetpub\WWWROOT\project1\files*.



Upload folder

files

□ Absolute path □ PHP expression

Enter the path to a folder on the server where uploaded files will be stored. The path should be relative unless absolute path is checked off.

Upload options

□ Required field □ Basic upload control

□ Delete file when associated record is deleted (field option)

Files settings

1 Max number of files to store (0 - unlimited)

Path to upload folder may contain PHP expression (select **PHP expression** option). So you do not need to write event code to save each user's files to separate folder. Sample upload path in this case is:

```
$folder = $_SESSION["UserID"];
```

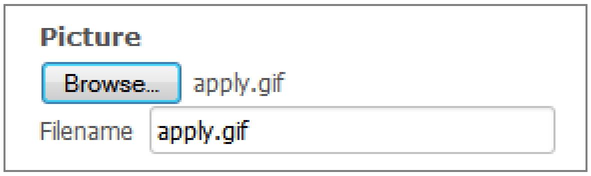
Another example is when a person needs to upload files to a folder (e.g. *http://mywebsite.com/files/images)* and it is not possible to use relative path *.. / files / images*. In this case select both **PHP expression** and **Absolute path** check boxes and use the following code:

```
$folder = $_SERVER["DOCUMENT_ROOT"]. "/files/images";
```

Use the **Delete file when associated record is deleted** option to make sure that a file will be deleted when associated record is deleted from the database. This option is global and if enabled is applied to all tables and fields.

You can restrict the maximum number of files to store. Enter "0" for unlimited number of files. Note that default value is "1" and you need to increase it to enable multiple upload.

Enable **Basic upload control** option to select the old style file upload control that lets users upload one file at a time.

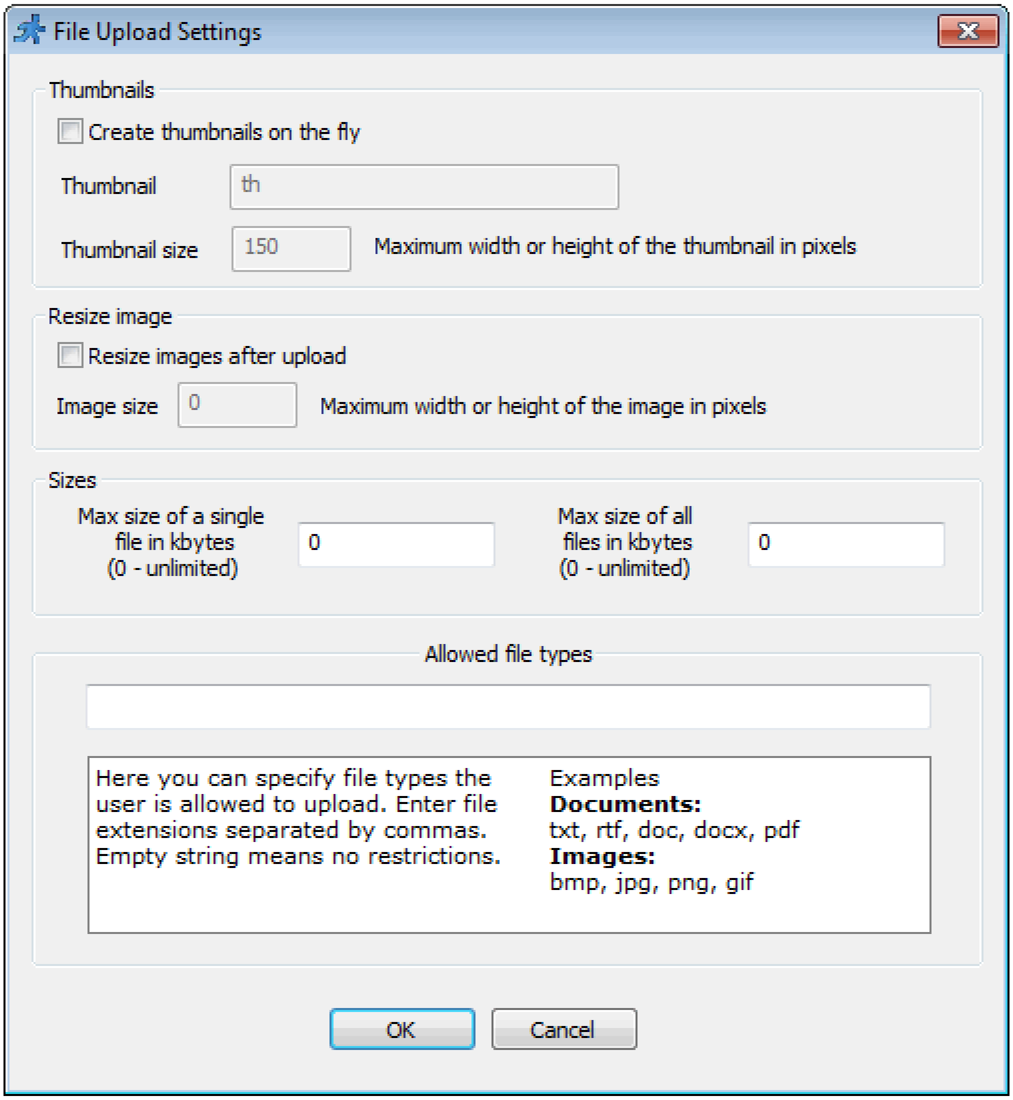


Picture

apply.gif

Filename apply.gif

Advanced settings



File Upload Settings

Thumbnails

□ Create thumbnails on the fly

Thumbnail: th

Thumbnail size: 150 Maximum width or height of the thumbnail in pixels

Resize image

□ Resize images after upload

Image size: 0 Maximum width or height of the image in pixels

Sizes

Max size of a single file in kbytes (0 - unlimited): 0

Max size of all files in kbytes (0 - unlimited): 0

Allowed file types

Here you can specify file types the user is allowed to upload. Enter file extensions separated by commas. Empty string means no restrictions.

Examples
Documents:
txt, rtf, doc, docx, pdf
Images:
bmp, jpg, png, gif

PHPRunner allows creating thumbnails on the fly. To do this, select the **Create thumbnails of the fly** check box and choose thumbnail size and thumbnail prefix. For example, if the original image filename is *example.gif* and thumbnail image filename is *th_example.gif* correspondingly, then thumbnail prefix is *th_*. Note that thumbnail images are stored in the same folder as the images.

Note: to use thumbnails functionality you need to have the GD library installed. On Windows you need to uncomment php_gd2.dll extension in your php.ini file.

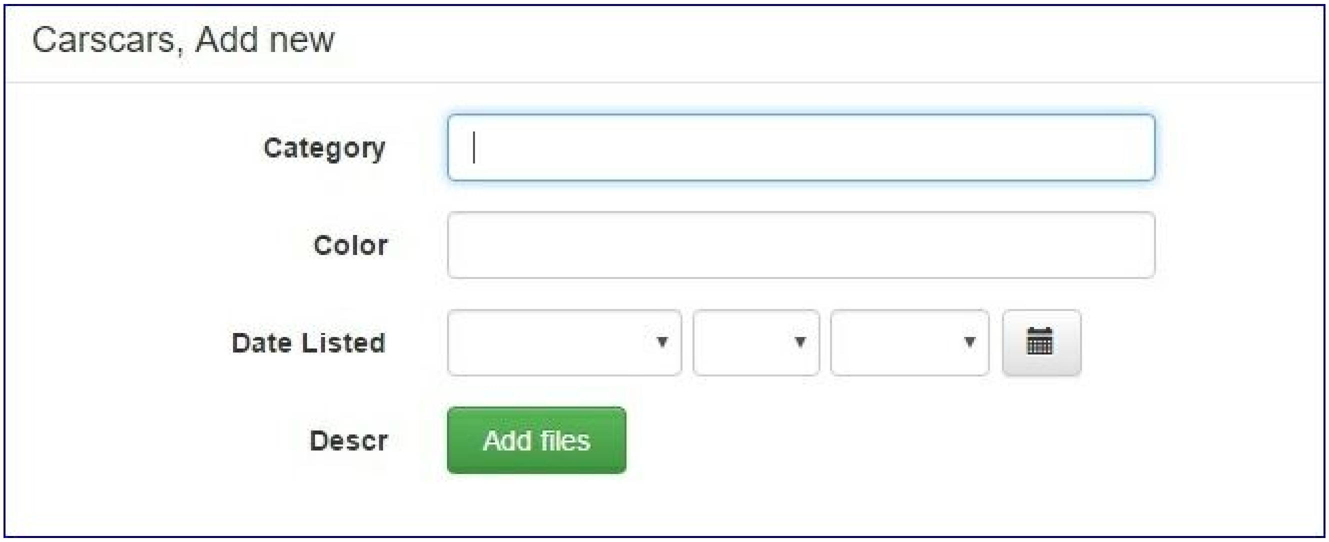
To resize images on upload, select the **Resize images after upload** option and select max width or height of resulting image.

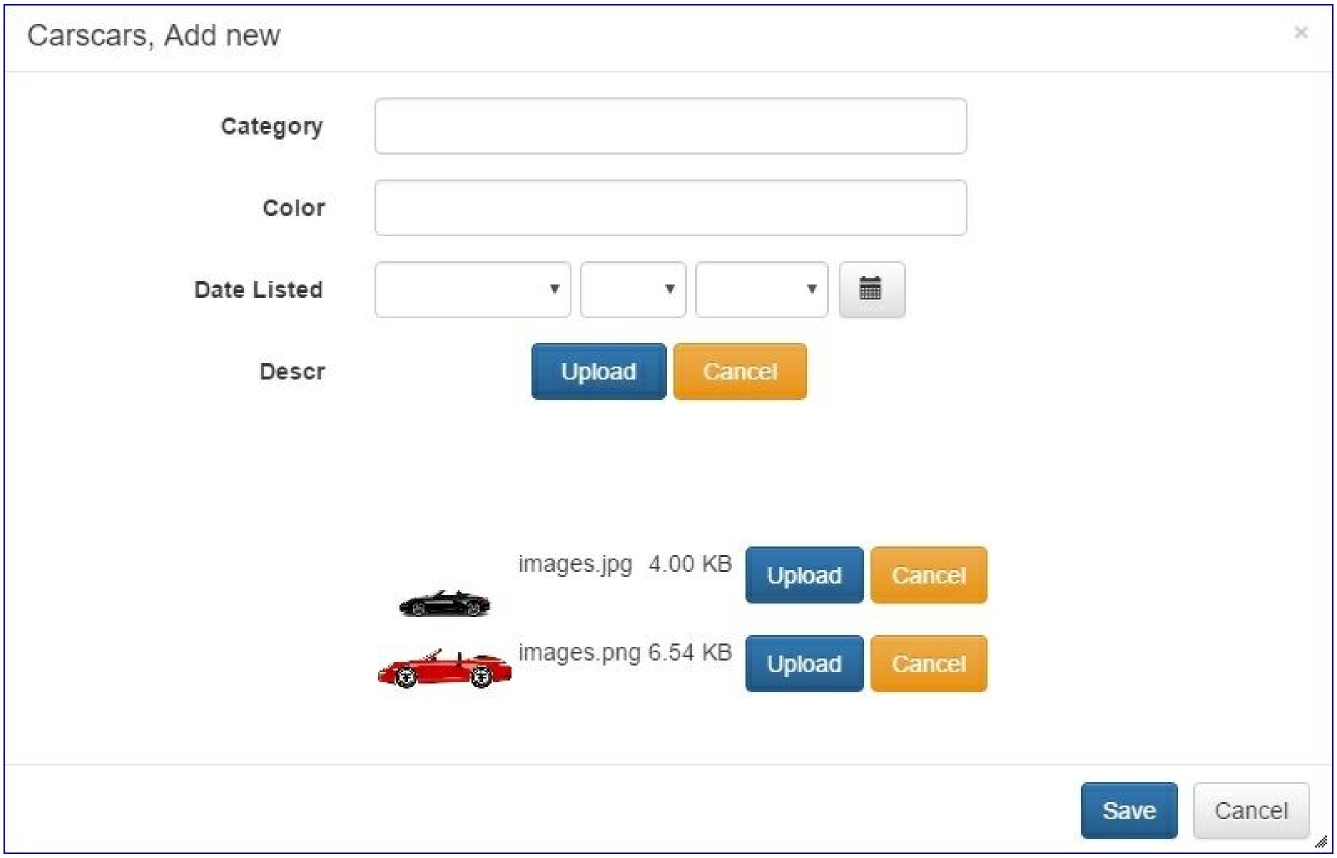
You can restrict the uploaded file size by setting maximum size of a single file and maximum size of all uploaded files.

Add list of file extensions allowed to upload under **Allowed file types**. Empty string means no restrictions.

Upload control in generated application

You can drag one or more files from your desktop right to the upload control on web page. Drag and drop feature is not supported in Internet Explorer.

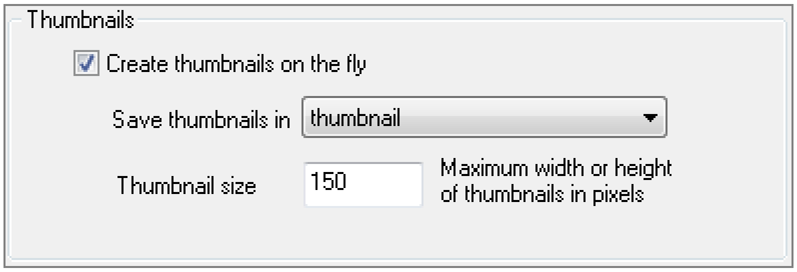




Binary field: Image

To create thumbnails on the fly, select the **Create thumbnails of the fly** check box and choose the field name to save thumbnails in.

Note: you need an additional binary field (field of *MEDIUMBLOB* type) to store thumbnails. This field is the auxiliary one and we do not recommend to display it on your pages. You can display/hide fields on the Choose fields page.



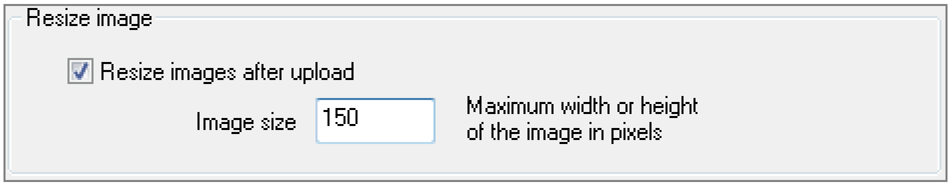
Thumbnails

☑ Create thumbnails on the fly

Save thumbnails in: thumbnail

Thumbnail size: 150 — Maximum width or height of thumbnails in pixels

To resize images on upload, select the **Resize images after upload** option and select max width or height of resulting image.



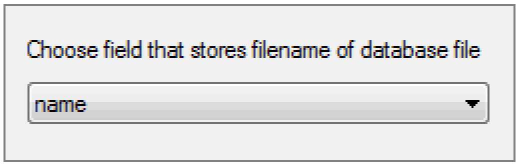
Resize image

☑ Resize images after upload

Image size: 150 — Maximum width or height of the image in pixels

Binary field: File

Choose field that stores name of database file. This filename is required to set correct file type when you retrieve uploaded file from the database. If you don't choose filename field or leave it empty, you will be presented with **Open with** dialog every time you download this file from the database.



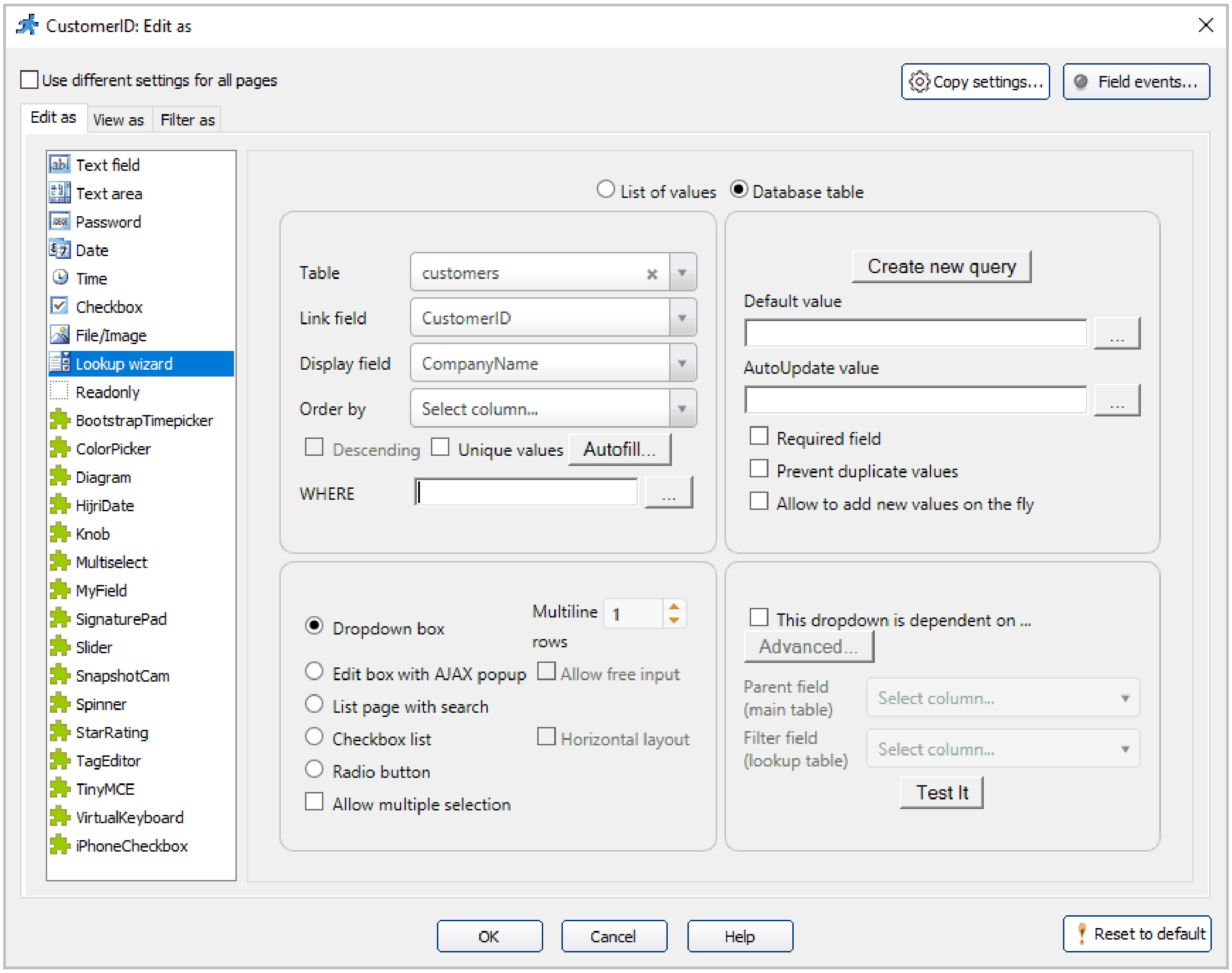
Choose field that stores filename of database file

name

2.16.12.5 Lookup wizard

Quick jump
List of values
Database table
Dependent dropdown lists
Appearance of lookup wizard
SQL variables in SQL and Lookup wizards

Lookup wizard is a list of values. Values can be entered manually (**List of values** option) or retrieved from the database table (**Database table** option).



CustomerID: Edit as

□ Use different settings for all pages

Edit as: Lookup wizard

○ List of values ● Database table

Table: customers

Link field: CustomerID

Display field: CompanyName

Order by: Select column...

□ Descending □ Unique values

WHERE:

Default value:

AutoUpdate value:

□ Required field

□ Prevent duplicate values

□ Allow to add new values on the fly

● Dropdown box

Multiline rows: 1

○ Edit box with AJAX popup □ Allow free input

○ List page with search

○ Checkbox list □ Horizontal layout

○ Radio button

□ Allow multiple selection

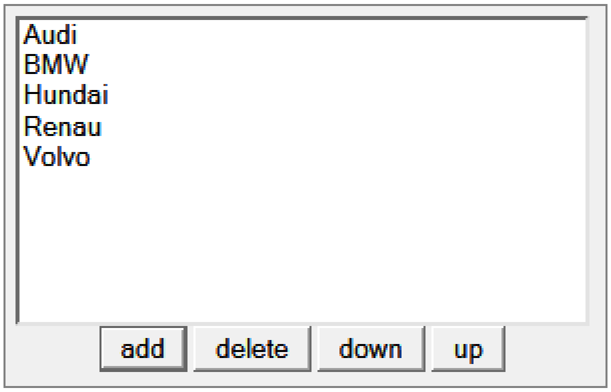
□ This dropdown is dependent on ...

Parent field (main table): Select column...

Filter field (lookup table): Select column...

List of values

Use **add/delete** buttons to add/delete values and **up/down** buttons to order them.


Audi
BMW
Hundai
Renau
Volvo

Database table

You can select the existing database table/custom view to retrieve values from (**Table** dropdown box) or create new custom view using **Create new query** button. For more information about creating SQL query, see About SQL query designer.

Note: if selected table is included in the project, the rules applied to the project tables (such as modified SQL query, Advanced Security, changes made to the SQLQuery object in the After table initialized event) will be also applied to the lookup table. For example, you can limit the list of items in dropdown with Advanced Security.

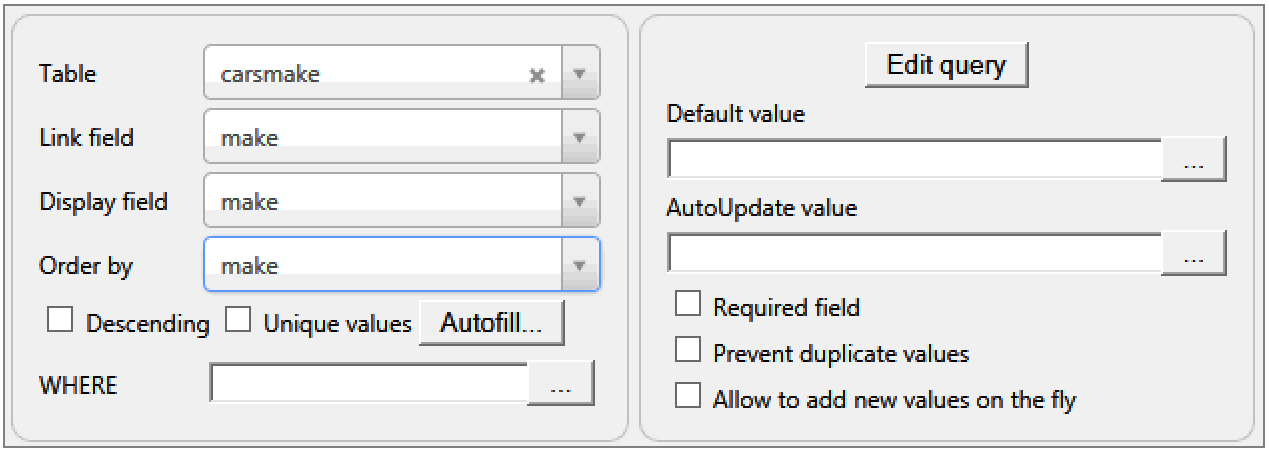
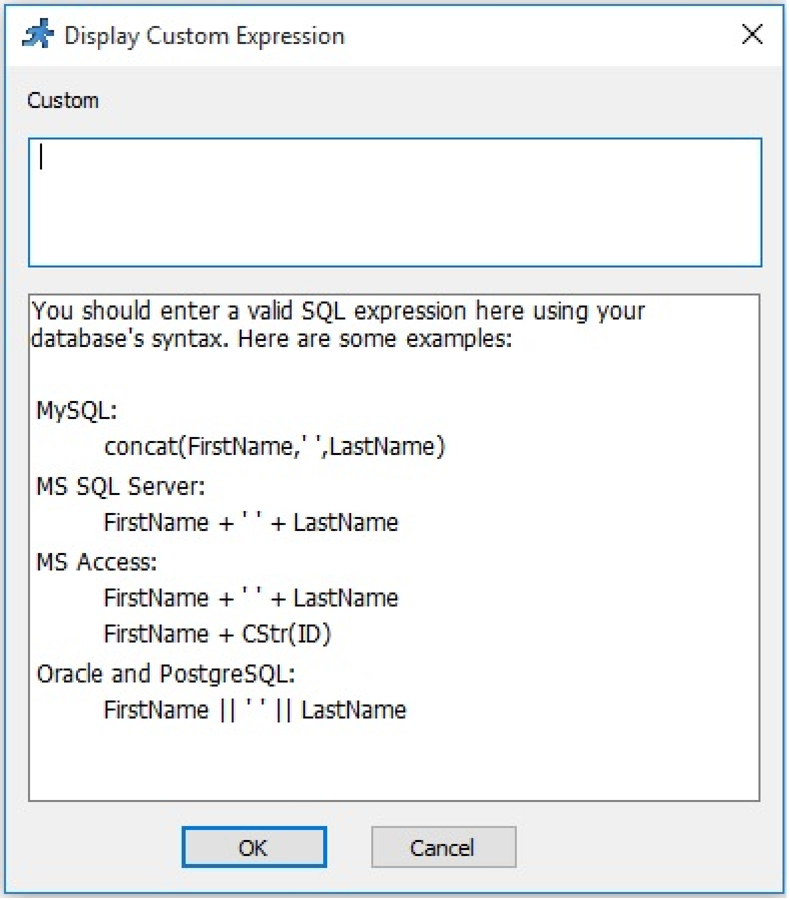


Table: carsmake
Link field: make
Display field: make
Order by: make
□ Descending □ Unique values
WHERE:
Default value:
AutoUpdate value:
□ Required field
□ Prevent duplicate values
□ Allow to add new values on the fly

Display field

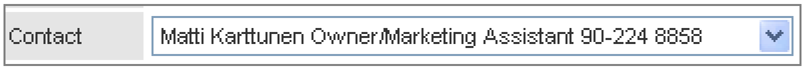
You can use Custom Expression for the **Display field** to display several values from different fields with custom design in dropdown box. Click the arrow near **Display field** and then **<Custom expression>**.



You should enter a valid SQL expression here using your database's syntax. Here are some examples:

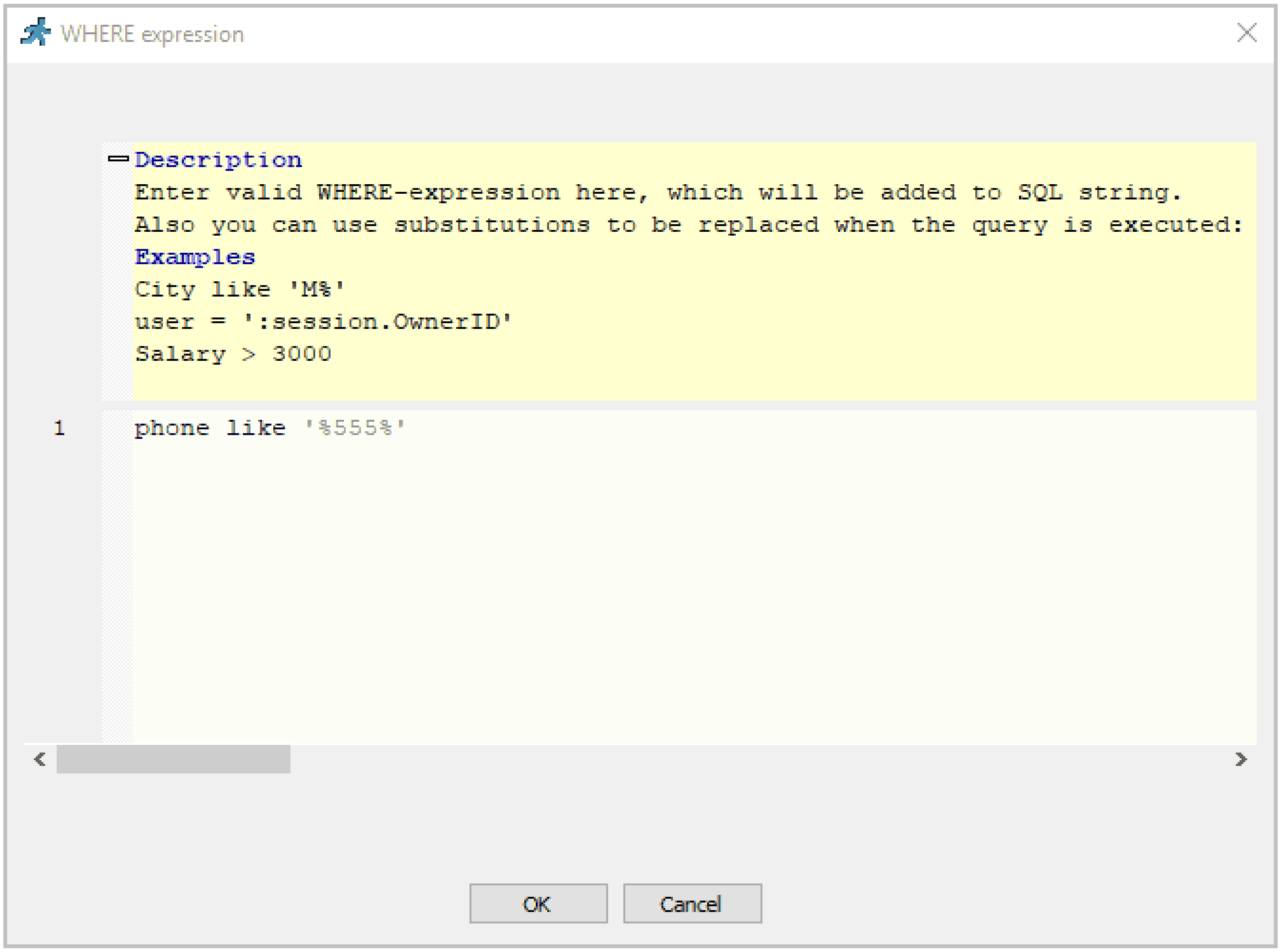
```
MySQL:
        concat(FirstName,' ',LastName)
MS SQL Server:
        FirstName + ' ' + LastName
MS Access:
        FirstName + ' ' + LastName
        FirstName + CStr(ID)
Oracle and PostgreSQL:
        FirstName || ' ' || LastName
```

How it looks in the browser:



WHERE expression

The **WHERE** expression allows to sort dropdown box content. In our example if you put **"phone like '%555%'"** into WHERE box only Contacts with 555 in phone number will be displayed.



```
Description
Enter valid WHERE-expression here, which will be added to SQL string.
Also you can use substitutions to be replaced when the query is executed:
Examples
City like 'M%'
user = ':session.OwnerID'
Salary > 3000
```

```
phone like '%555%'
```

To use table variables of the session in a WHERE clause, table names should be written as follows:

```
$_SESSION["Cars_masterkey1"]
$_SESSION["_Cars_OwnerID"]
```

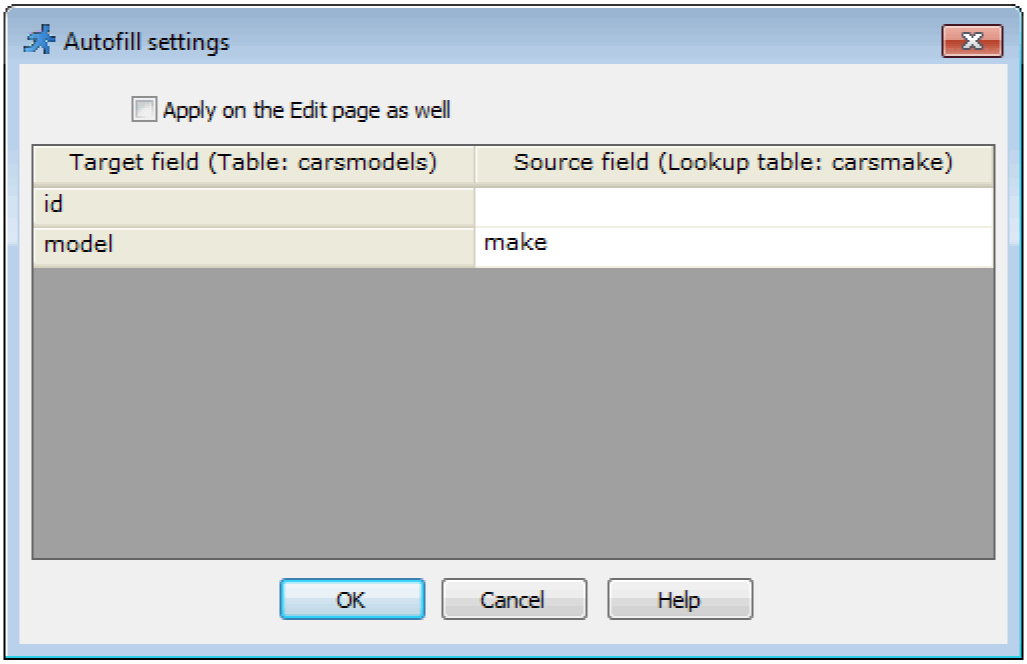
You can use SQL variables in a WHERE expression:

```
CustomerID= ':user.CustomerID'
CustomerID= ':session.UserID'
```

See more examples of SQL variables

Autofill

You can auto-fill several fields on the Add/Edit pages with values from the lookup table.

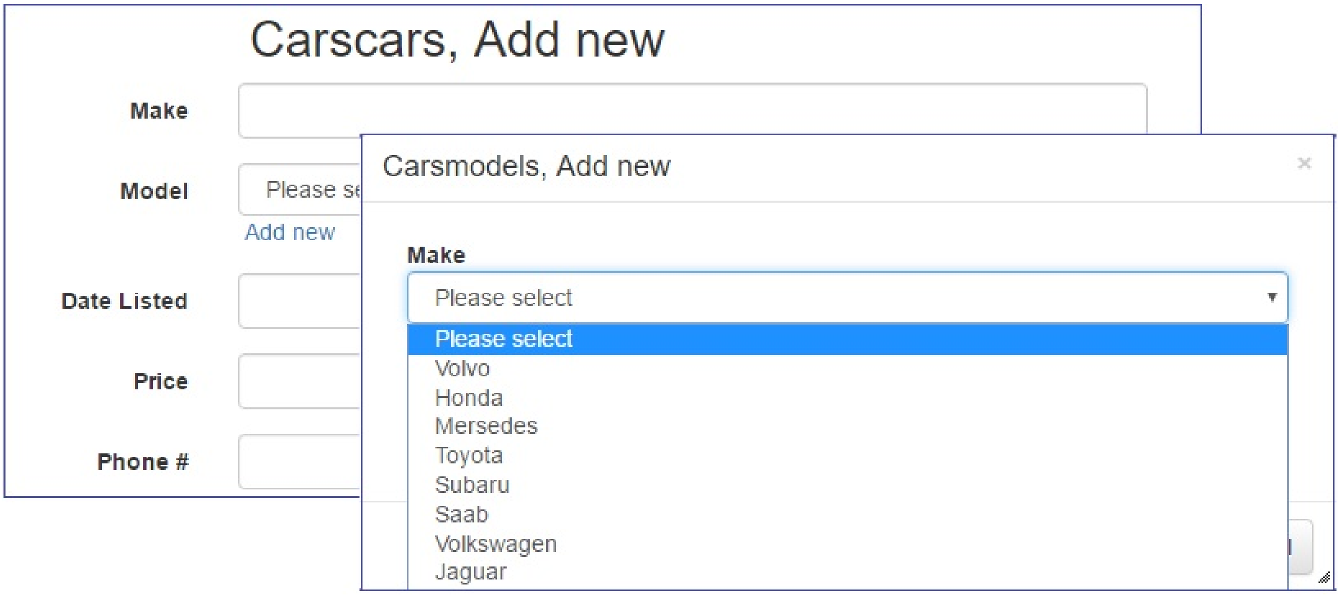


□ Apply on the Edit page as well

Target field (Table: carsmodels)	Source field (Lookup table: carsmake)
id	
model	make

Add new items on the fly

This option puts **Add new** link next to the dropdown box allowing to add new items right on Edit/Add page. Add new item popup is a fully-featured add page.



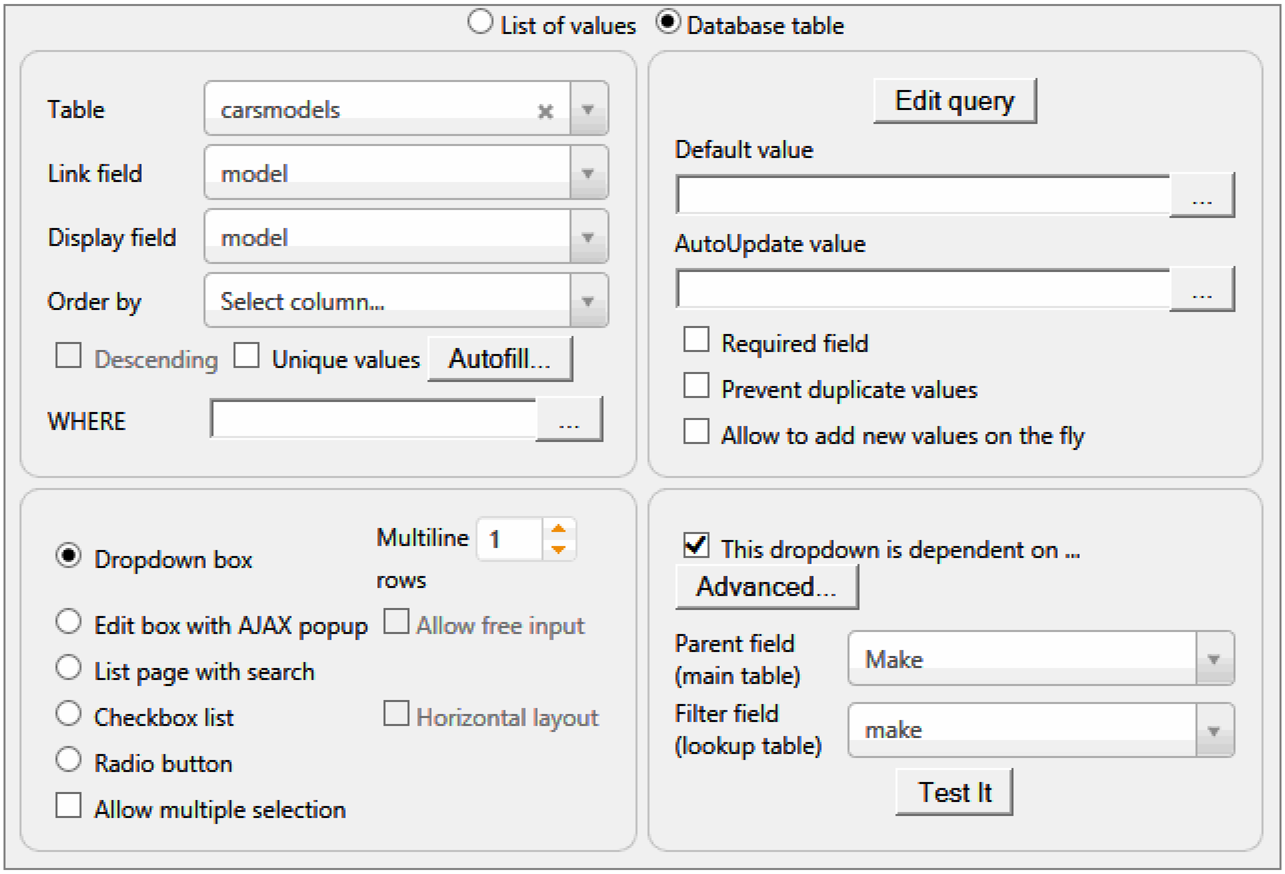
Dependent dropdown lists

You can use dependent dropdown lists, where values shown in the second dropdown list depend on the value you've chosen in the first one.

Let's make the **Model** field content depends on the **Make** field value:

1. Set **Lookup wizard** as "Edit as" type for the **Make** and **Model** fields.

2. For the **Model** field select **This dropdown is dependent on** check box. Select **Make** as parent field from the main table and filter field from lookup table.



○ List of values ● Database table

Table: carsmodels

Link field: model

Display field: model

Order by: Select column...

□ Descending □ Unique values Autofill...

WHERE: ...

Edit query

Default value: ...

AutoUpdate value: ...

□ Required field

□ Prevent duplicate values

□ Allow to add new values on the fly

● Dropdown box Multiline 1 rows

○ Edit box with AJAX popup □ Allow free input

○ List page with search

○ Checkbox list □ Horizontal layout

○ Radio button

□ Allow multiple selection

☑ This dropdown is dependent on ...

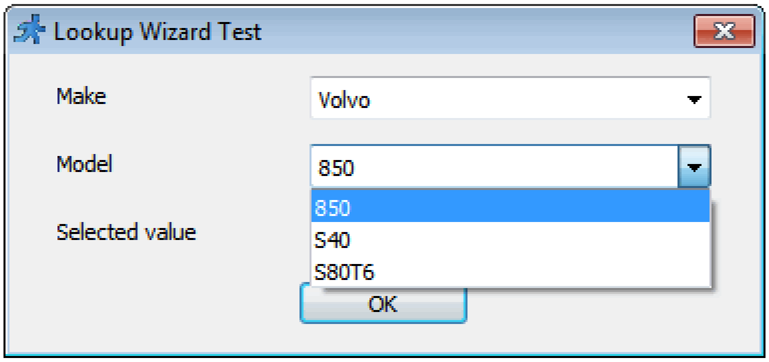
Advanced...

Parent field (main table): Make

Filter field (lookup table): make

Test It

3. Click **Test it** to check how it works.



Lookup Wizard Test

Make: Volvo

Model: 850

850

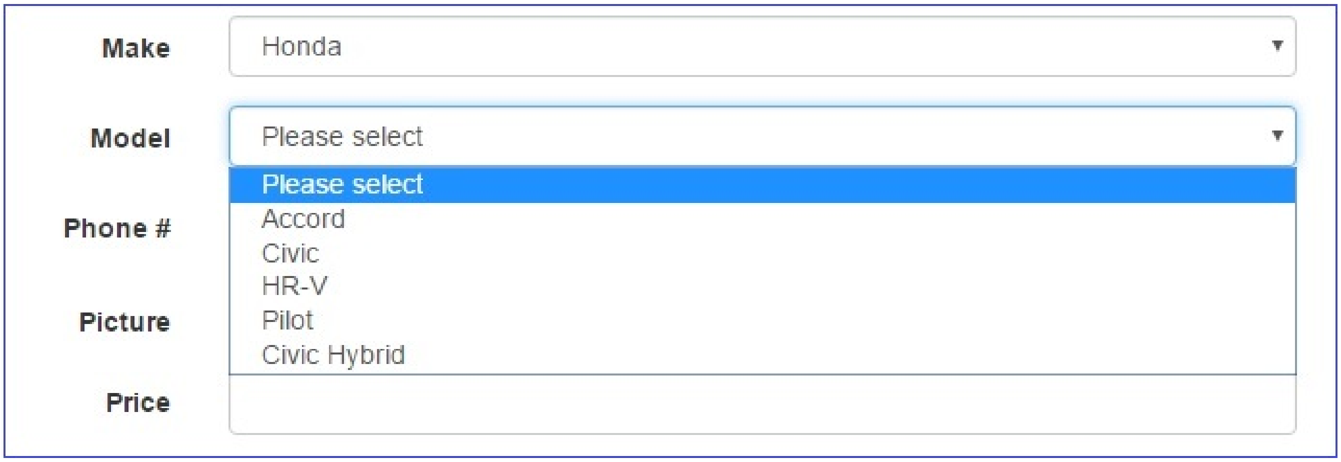
S40

S80T6

Selected value

OK

An example of Edit page with dependent dropdown lists:



Make: Honda ▾

Model: Please select ▾

Please select
Accord
Civic
HR-V
Pilot
Civic Hybrid

Phone #

Picture

Price

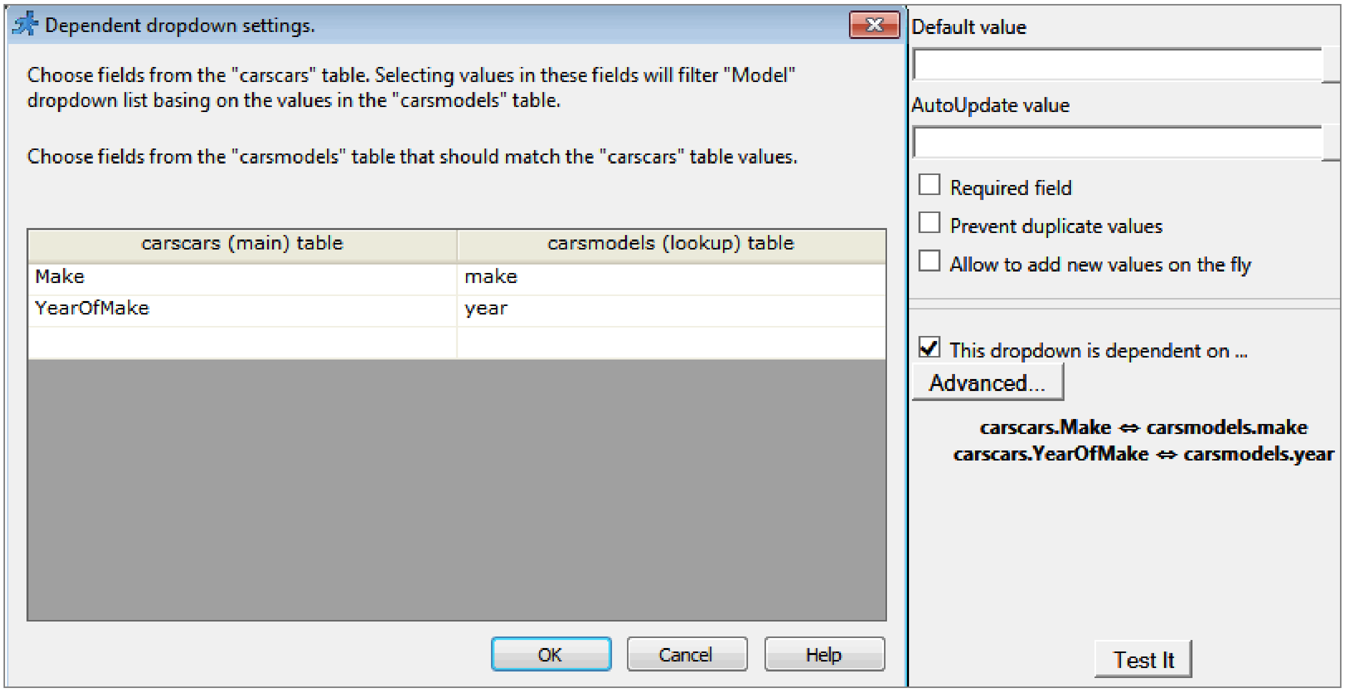
Cascading dropdown lists

You can also create chain of dependent dropdown lists where one dropdown list depends on two or more master dropdown controls. For example, Model field content depends on values of Make and YearOfMake fields. So you can select from only those models that were produced by selected company on selected year. To perform this:

1. Set **Lookup wizard** as "Edit as" type for the **Make**, **YearOfMake** and **Model** fields.

2. Make the **YearOfMake** dropdown list depends on **Make** dropdown list.

3. For the **Model** field select **This dropdown is dependent on** check box and click **Advanced**. Then select Make and YearOfMake fields and click OK.



Dependent dropdown settings.

Choose fields from the "carscars" table. Selecting values in these fields will filter "Model" dropdown list basing on the values in the "carsmodels" table.

Choose fields from the "carsmodels" table that should match the "carscars" table values.

carscars (main) table	carsmodels (lookup) table
Make	make
YearOfMake	year

Default value

AutoUpdate value

☐ Required field

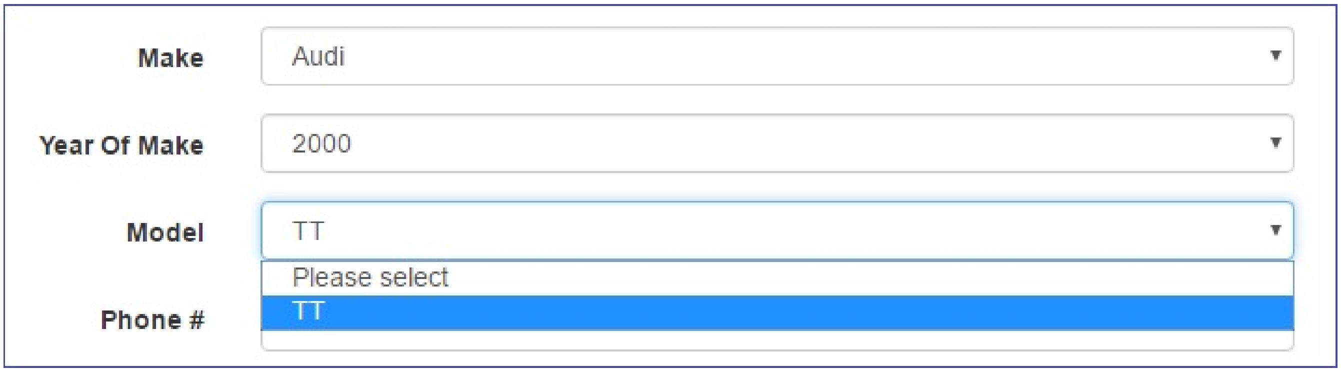
☐ Prevent duplicate values

☐ Allow to add new values on the fly

☑ This dropdown is dependent on ...

carscars.Make ⇔ carsmodels.make
carscars.YearOfMake ⇔ carsmodels.year

Click **Test it** to check how it works.



Make: Audi

Year Of Make: 2000

Model: TT

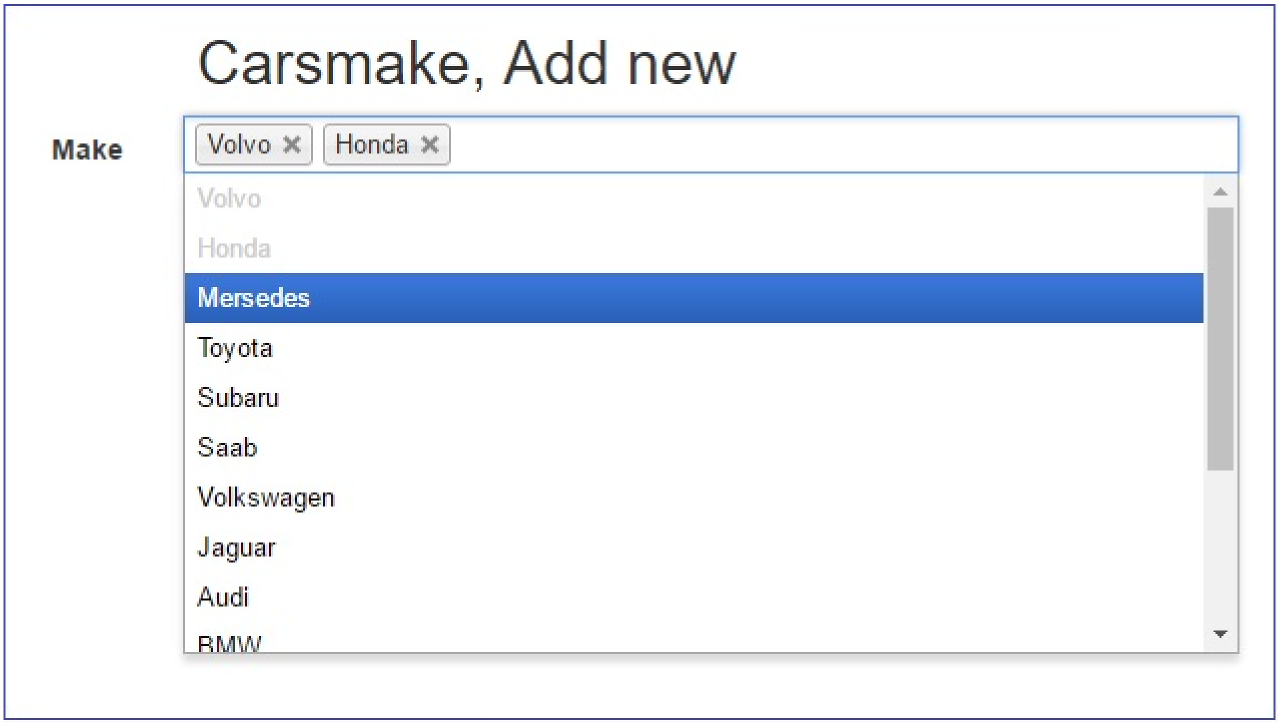
Phone #

Also more than one dependent dropdown box can be tied to the same master dropdown control.

Appearance of lookup wizard

Dropdown box

List of values is displayed as a dropdown box. If you set **Multiline** to any value greater then one this field will appear as a listbox on Add/Edit pages. Select **Allow multiple selection** check box to allow users select multiple values.

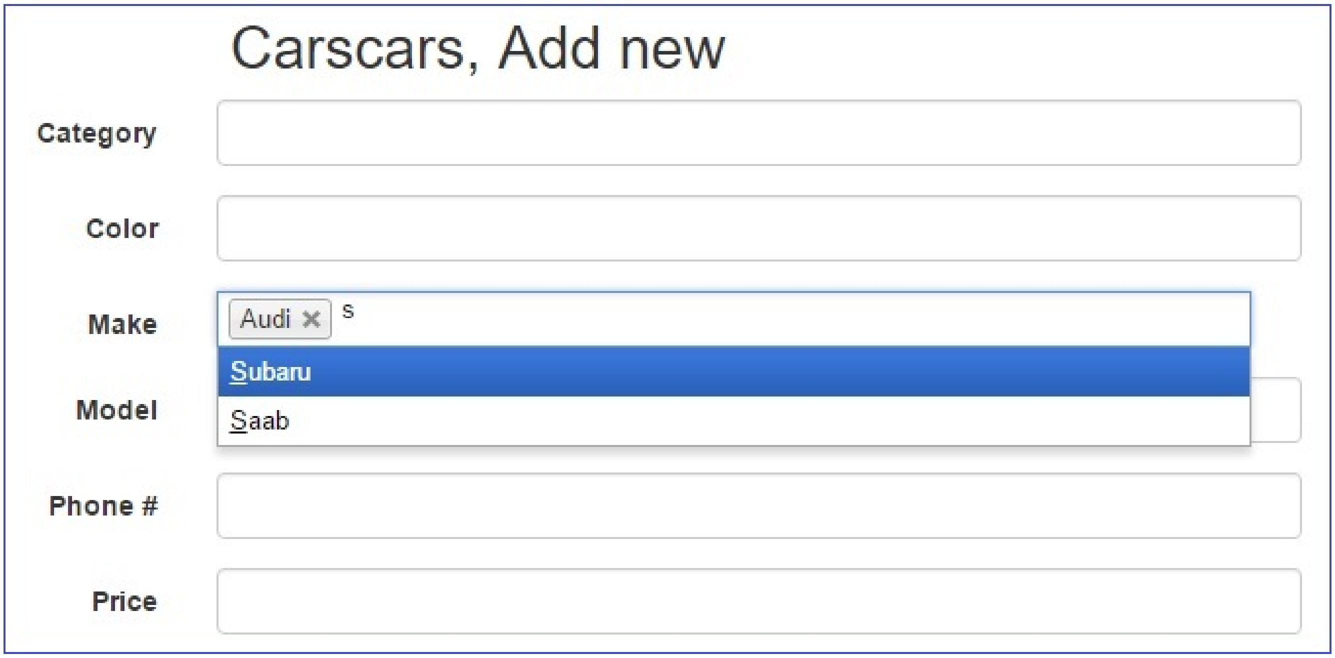


Carsmake, Add new

Make: Volvo ✕ Honda ✕

Edit box with AJAX popup

This feature is explained in AJAX-based features. Select **Allow multiple selection** check box to allow users select multiple values.

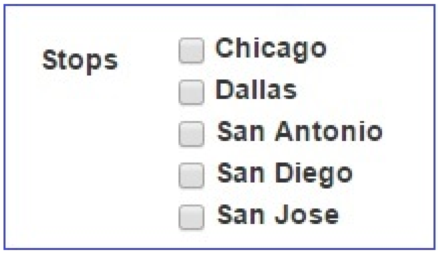


The application will look for the occurrence of the typed in string anywhere in the list. For example, when you enter 'co', it will show 'Corolla' and 'Accord'. If you want to change this behavior and make it look for the values starting with the entered value, i.e. 'Corolla' only, add the following code to the AfterAppInit:

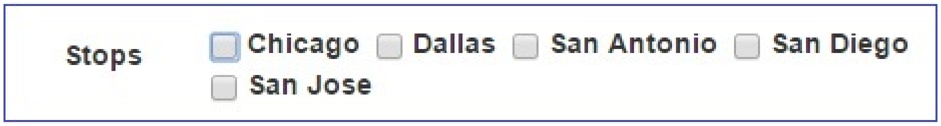
```
$ajaxSearchStartsWith = true;
```

Checkbox list

List of values is displayed as set of check boxes. A user can select one or several values.

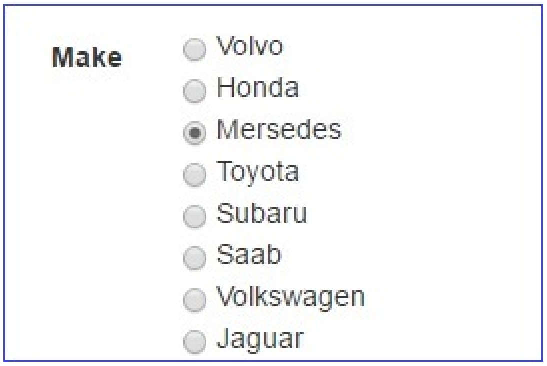


If you enable **Horizontal layout** check box, check boxes will be placed horizontally.



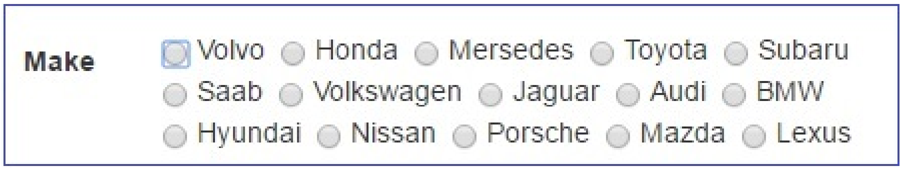
Radio button

List of values is displayed as set of radio buttons. A user can select only one value.



Make
○ Volvo
○ Honda
● Mersedes
○ Toyota
○ Subaru
○ Saab
○ Volkswagen
○ Jaguar

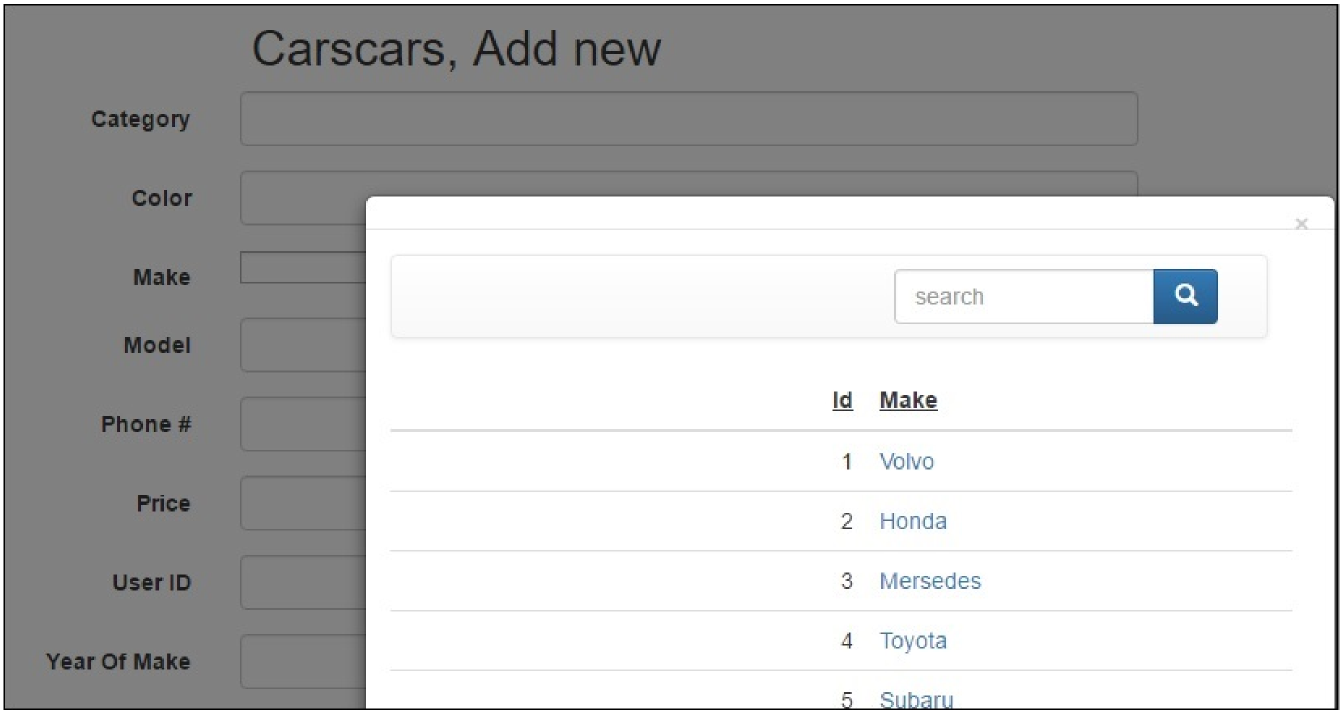
If you enable **Horizontal layout** check box, radio buttons will be placed horizontally.



Make ○ Volvo ○ Honda ○ Mersedes ○ Toyota ○ Subaru ○ Saab ○ Volkswagen ○ Jaguar ○ Audi ○ BMW ○ Hyundai ○ Nissan ○ Porsche ○ Mazda ○ Lexus

List page with search

Once **Select** button is clicked user is presented with the searchable lookup table. Lookup table appearance is fully customizable in Visual Editor. Select **Allow multiple selection** check box to allow users select multiple values.



Carscars, Add new

Category

Color

Make

Model

Phone #

Price

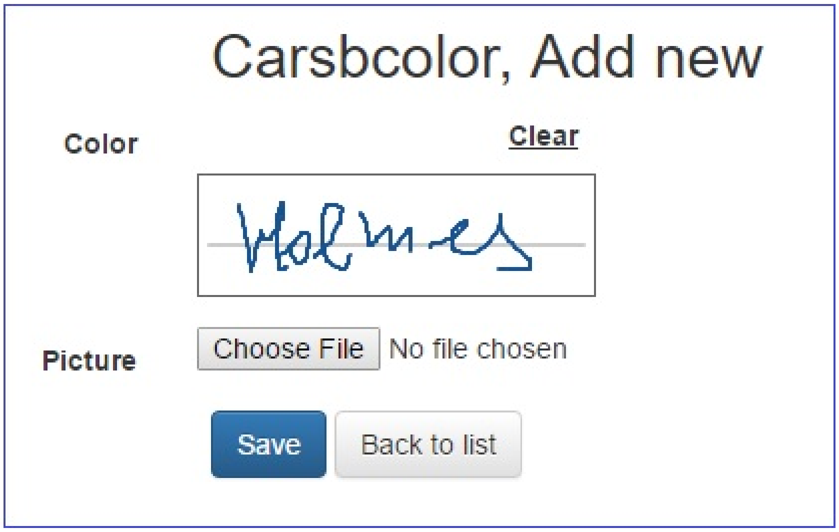
User ID

Year Of Make

Id	Make
1	Volvo
2	Honda
3	Mersedes
4	Toyota
5	Subaru

2.16.12.6 SignaturePad

The SignaturePad control allows you to add signature pad to your forms. SignaturePad works with both mouse and touch devices. Select *SignaturePad* as "Edit as" type for any text field with a length of 200 characters or more and enjoy your signature pad on Add/Edit pages. You should select *Image* as "View as" type for selected text field.



Carsbcolor, Add new

Color

Picture

No file chosen

Click **Add initialization script** button to customize the SignaturePad control.

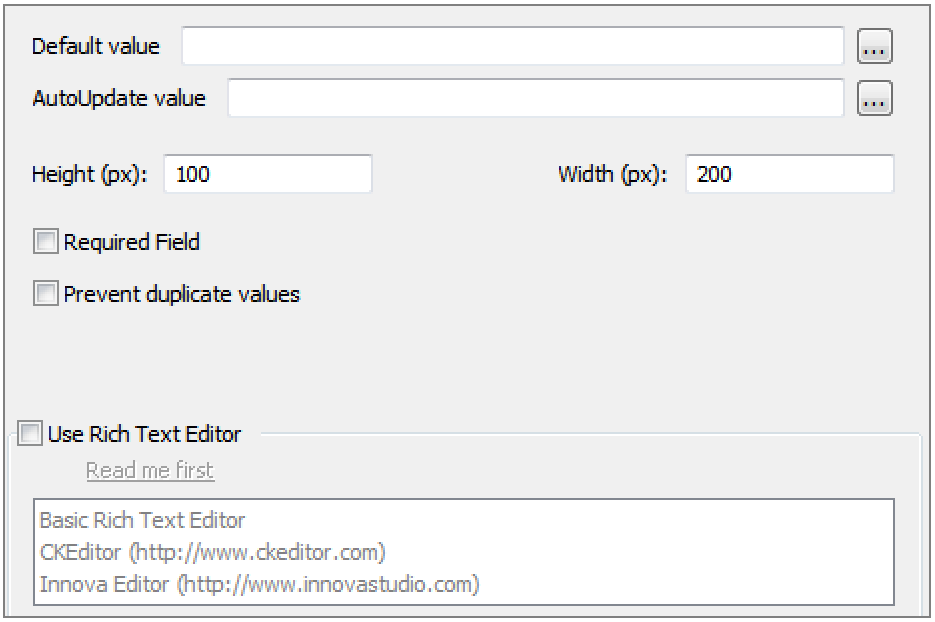
Note that Signature Pad control is a custom plugin. From now on you also can create your own Edit control plugins. For more information, see How to create your own Edit control.

Code with the example of settings for the SignaturePad control:

```
// signature field height
$this->settings["height"] = 100;
// signature field width
$this->settings["width"] = 300;
// signature background color
$this->settings["bgcolor"] = "#ffffff";
// set it to true to make signature field required
$this->settings["required"]=false;
// folder to store signature files
$this->settings["folder"]="files";
// signature background image
// $this->settings["bgimage"] = "http://website.com/images/background.png";
$this->settings["bgimage"] = "";
// signature pen color
$this->settings["color"] = "#145394";
// signature line width
$this->settings["linewidth"] = 2;
```

2.16.12.7 Text area

Multiline text area control. For advanced WYSIWYG edit capabilities select the **Use Rich Text Editor** check box. For more information about using Rich Text Editor, see Rich Text Editor plugins.



Default value ...

AutoUpdate value ...

Height (px): 100 Width (px): 200

□ Required Field

□ Prevent duplicate values

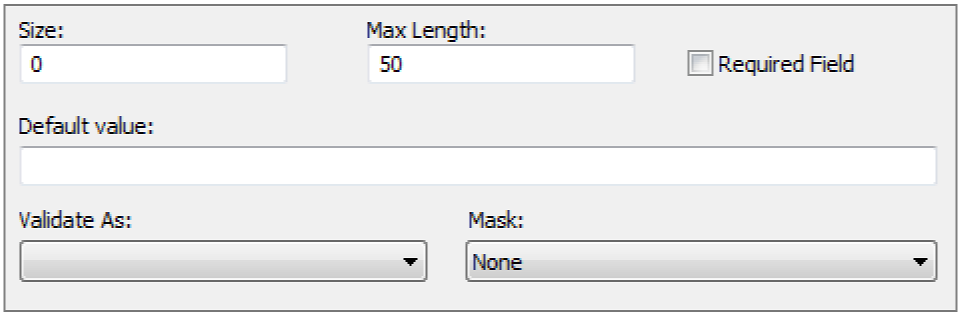
□ Use Rich Text Editor

Read me first

Basic Rich Text Editor
CKEditor (http://www.ckeditor.com)
Innova Editor (http://www.innovastudio.com)

2.16.12.8 Text field

Simple text box control.



Size: 0

Max Length: 50

□ Required Field

Default value:

Validate As:

Mask: None

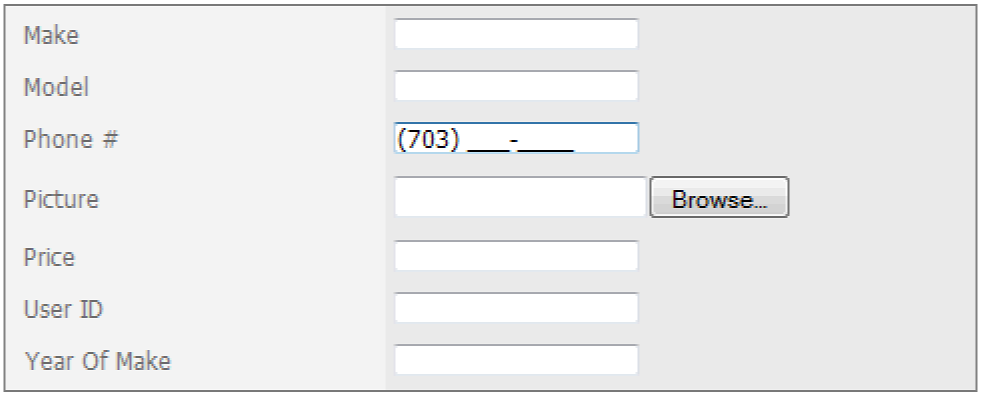
Validate As

For more information about validation, see Validation types.

Mask

Masked input allows you making a user to enter the data in certain format (dates,phone numbers, etc).

Example of phone numbers masked input:



Make	
Model	
Phone #	(703) ___-____
Picture	Browse...
Price	
User ID	
Year Of Make	

You can use one of the predefined mask format or create your own custom expression.

The following mask definitions are predefined:

- a - Represents an alpha character (A-Z,a-z)
- 9 - Represents a numeric character (0-9)
- * - Represents an alphanumeric character (A-Z,a-z,0-9)

A few examples of custom expression:

- US social security number: 999-99-9999
- Date: 99/99/9999
- Time (hh:mm:ss): 99:99:99
- Phone number: (999) 999-9999
- Phone number with optional extension: (999) 999-9999? x99999;

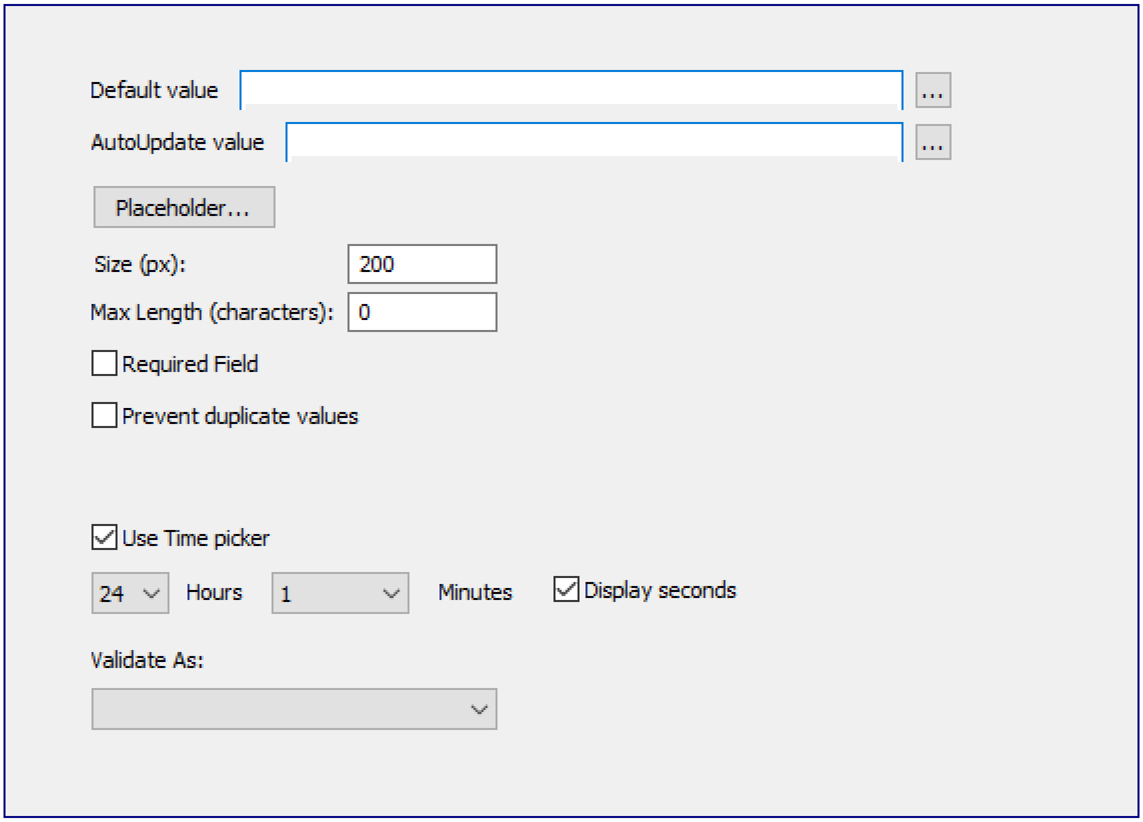
You can get more usage examples and documentation at http://digitalbush.com/projects/masked-input-plugin/.

2.16.12.9 Time

Time edit control. Use the **Use Time picker** check box to access time picker screen which allows you to select time instead of typing it. Select 12 or 24 hour format and one of available minute intervals. Enable the option "Display seconds" if you wish to display the time to the nearest minute.

Examples of default values are:
"9:30"
date("h:i:s")
date("h:i")

For more information about validation, see Validation types.



Default value

AutoUpdate value

Placeholder...

Size (px): 200

Max Length (characters): 0

□ Required Field

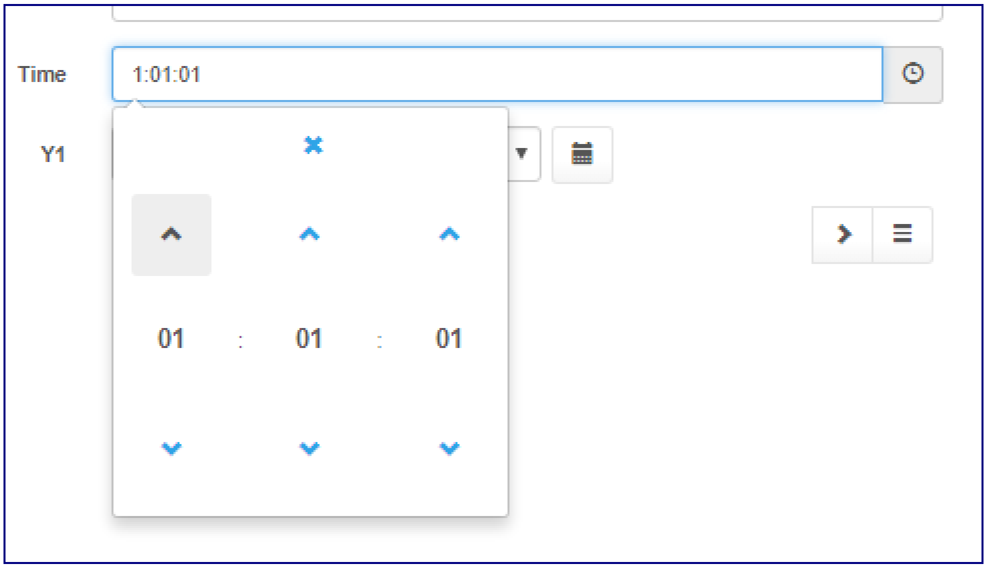
□ Prevent duplicate values

☑ Use Time picker

24 Hours 1 Minutes ☑ Display seconds

Validate As:

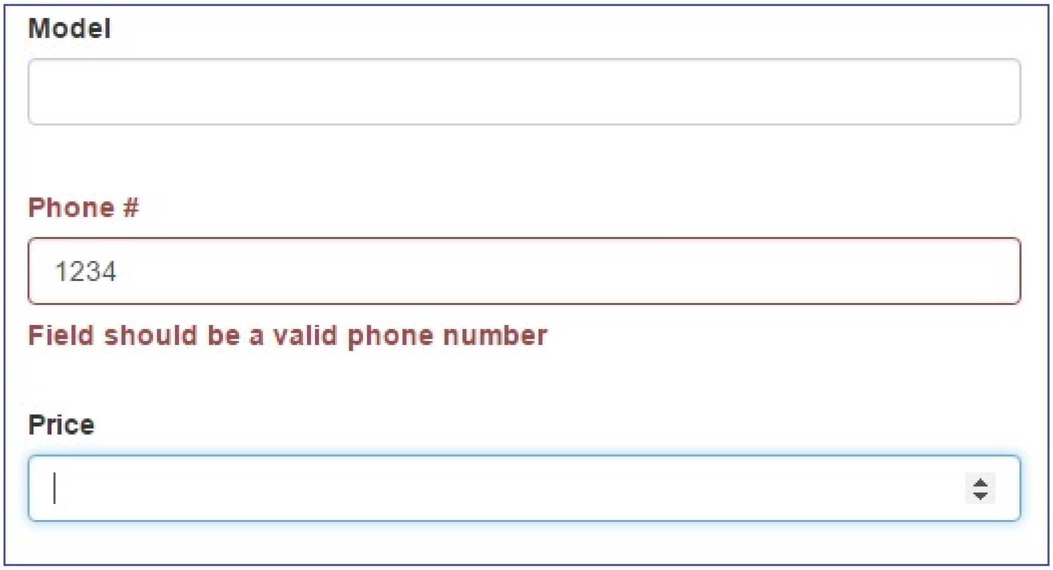
How it looks on the generated pages:



Time 1:01:01

2.16.12.1(Validation types

Validation types allow you to validate data that users enter on the add/edit page. You can use standard validation types or create your own validation plugins. If field value doesn't match defined format, web page users will see message saying what should be changed. Here is an example of such message:



Model

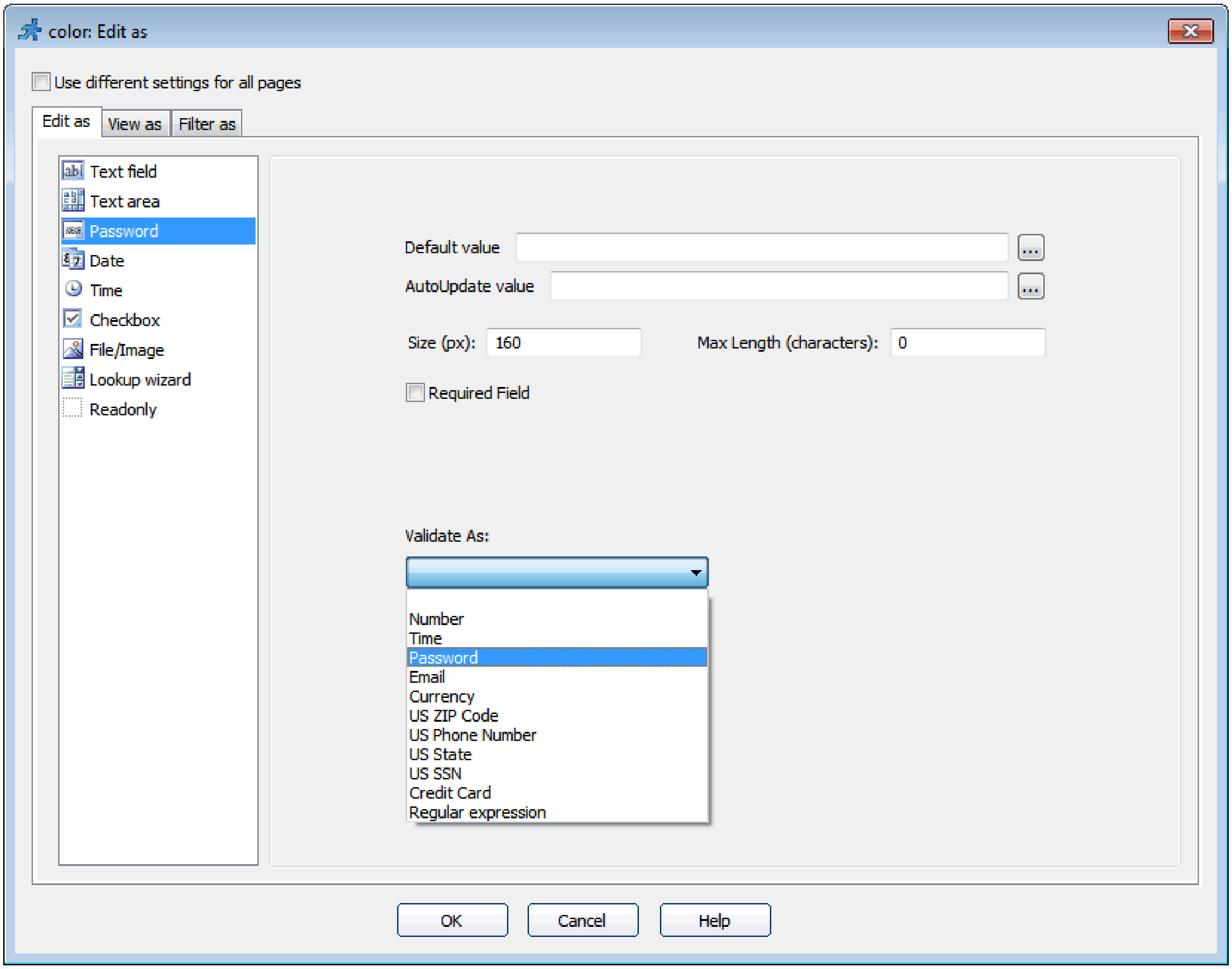
Phone #

1234

Field should be a valid phone number

Price

To add/edit validation rules, go to **Visual Editor** page, right-click a field and select **Properties**. **Validate As** check box is available for **Text field**, **Time**, **Password** formats at **Edit as** tab. Then you can choose one of validation types from the list.



□ Use different settings for all pages

Default value

AutoUpdate value

Size (px): 160

Max Length (characters): 0

□ Required Field

Validate As:

List of standard validation types

Number - field value should be a number.

Time - any valid time format that match regional settings.

Password - password field cannot be blank, cannot be 'Password' and should be at least 4 characters long.

Email - field value should be a valid email address.

Currency - numeric value. Decimal point is allowed. Examples: 13, 24.95

US Zip Code - five or ten digit number. Valid formats: 12345, 12345-6789 or 123456789

US Phone Number - numbers, spaces, hyphens, and parentheses are allowed. Examples: (123) 456-7890, 123 456 7890, 123 4567

US State - this field should be two letter US state abbreviation. Examples: AK, AL, CA, MN.

US SSN - nine digit US social security number. Valid formats: 123-45-6789 or 123 45 6789

Credit Card - valid credit card number.

ip_address - valid ip address (four numbers between 0 and 255 separated by periods). **ip_address** is the validation plugin. More info about adding your own validation types.

Regular expression - regular expression (regexp for short) is a special text string for describing a search pattern. You can get more information about basic syntax for setting regular expressions at http://www.regular-expressions.info/reference.html.

Examples of regular expressions:

- [abc] matches a, b or c

- [a-zA-Z0-9] matches any letter or digit;

- [^a-d] matches any character except a, b, c or d

- a{3} matches aaa

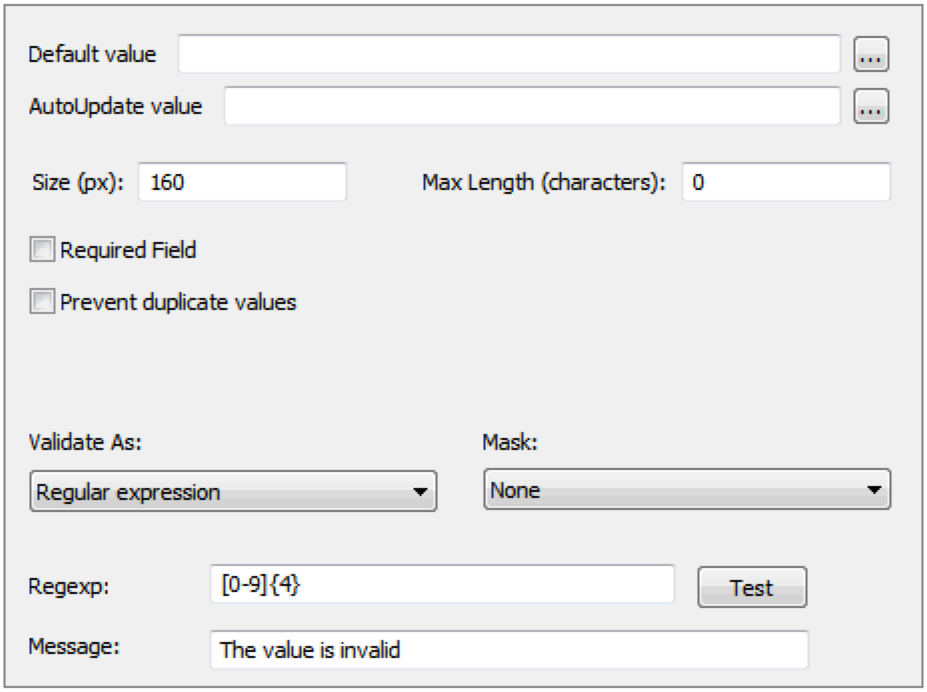
- regular expression for ip address:

  \b(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b matches 1.2.3.4

- regular expression for email address:

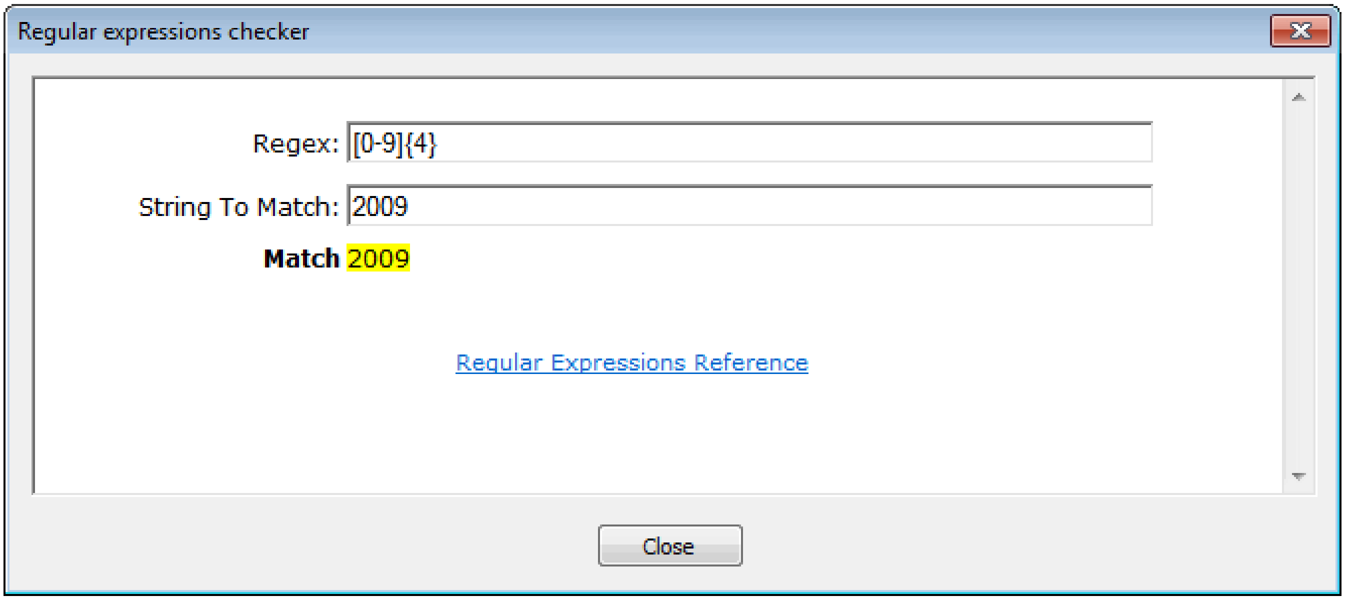
  ^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$ matches contact@abc.com

To define regular expression, add regular expression to **Regexp** field and message that will be displayed if value entered doesn't match defined regular expression to **Message** field. If multilanguage support is turned on in your project, you will see **Multilanguage** button that allows translating message into several languages. For more information about multiple language support, see Miscellaneous_settings.



Default value
AutoUpdate value
Size (px): 160
Max Length (characters): 0
□ Required Field
□ Prevent duplicate values
Validate As: Regular expression
Mask: None
Regexp: [0-9]{4}
Message: The value is invalid

You can test your regular expression by clicking **Test** button.



Regular expressions checker

Regex: [0-9]{4}
String To Match: 2009
Match 2009

How to add your own validation plugins

To add new validation type, you should create a file with javascript function that implements data validation and store it at **<PHPRunner install folder>\source\include\validate** folder, where <PHPRunner install folder> is a folder PHPRunner is installed to.

You can add any javascript code to your validation plugins. The javascript function should return nothing if the value was validated successfully and return error message if the value was not validated.

Note: The file name should match the function name.

Example:

As an example, let us create validation plugin that validates ip address.

Step 1. Create file **ip_address.js**.

Step 2. Add the following javascript function to this file:

```
function ip_address(sVal)
  {
  var regexp = /\b(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|
[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|
[01]?[0-9][0-9]?)/;
  if(!sVal.match(regexp))
    return 'This field should be a valid IP address';
  else
    return true;
  }
```

Step 3. Store file ip_address.js to the *<PHPRunner install folder>\source\include\validate* folder. In our case the folder is *C:\Program Files\PHPRunner10.0\source\include\validate*.

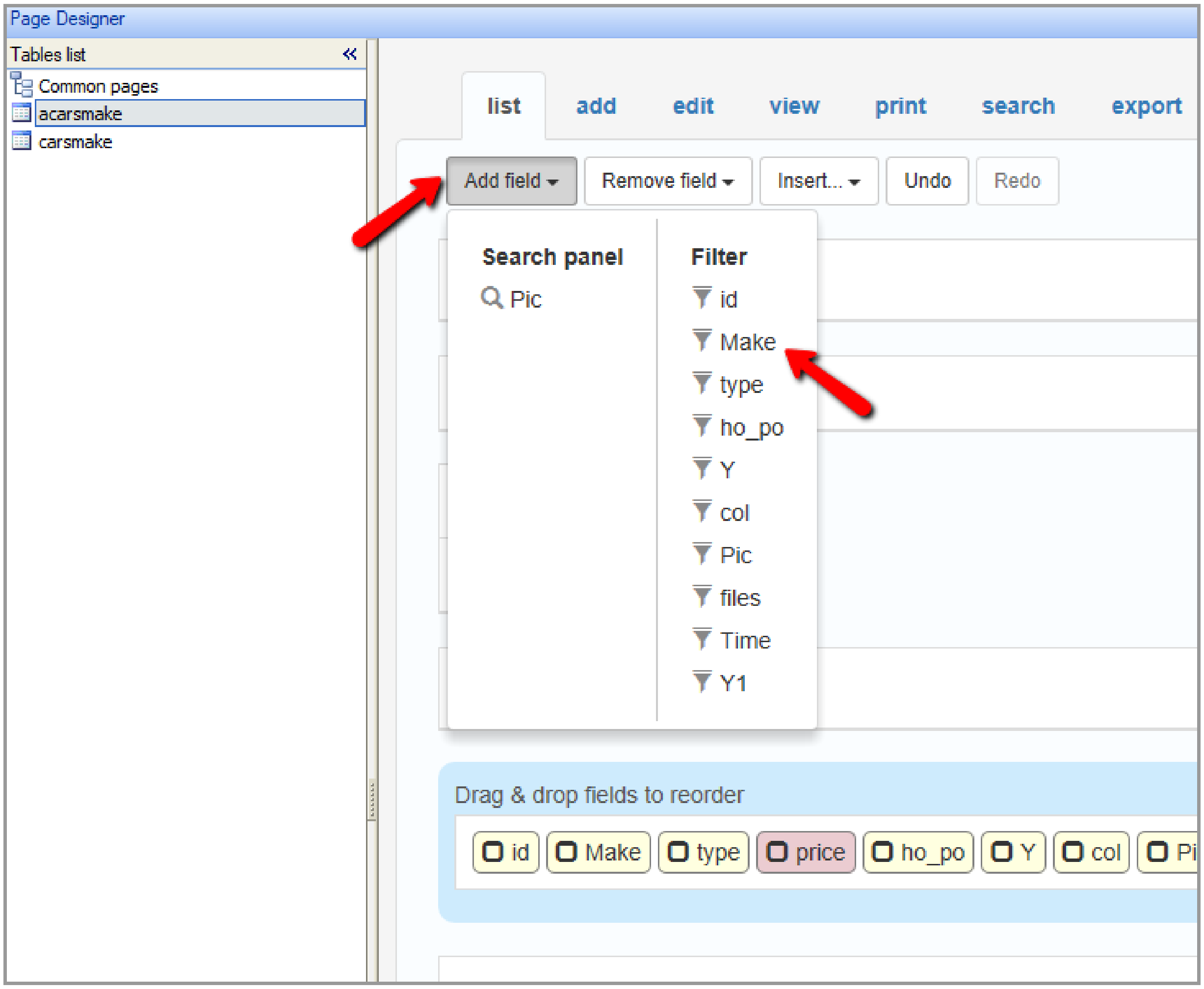
Step 4. Start PHPRunner. Now **ip_address** validation type is available in the list of validation types and you can use it in your projects.

If you want to add a multilanguage message (i.e. error message) to your validation plugin, you need to create a custom label in Label editor and use `Runner.getCustomLabel("LABEL1")` code in your javascript function, where **LABEL1** is custom label title. Note that **GetCustomLabel** function is applicable only for editing fields like Add/Edit/Register/List with inline add/edit.

2.16.13 "Filter as" settings

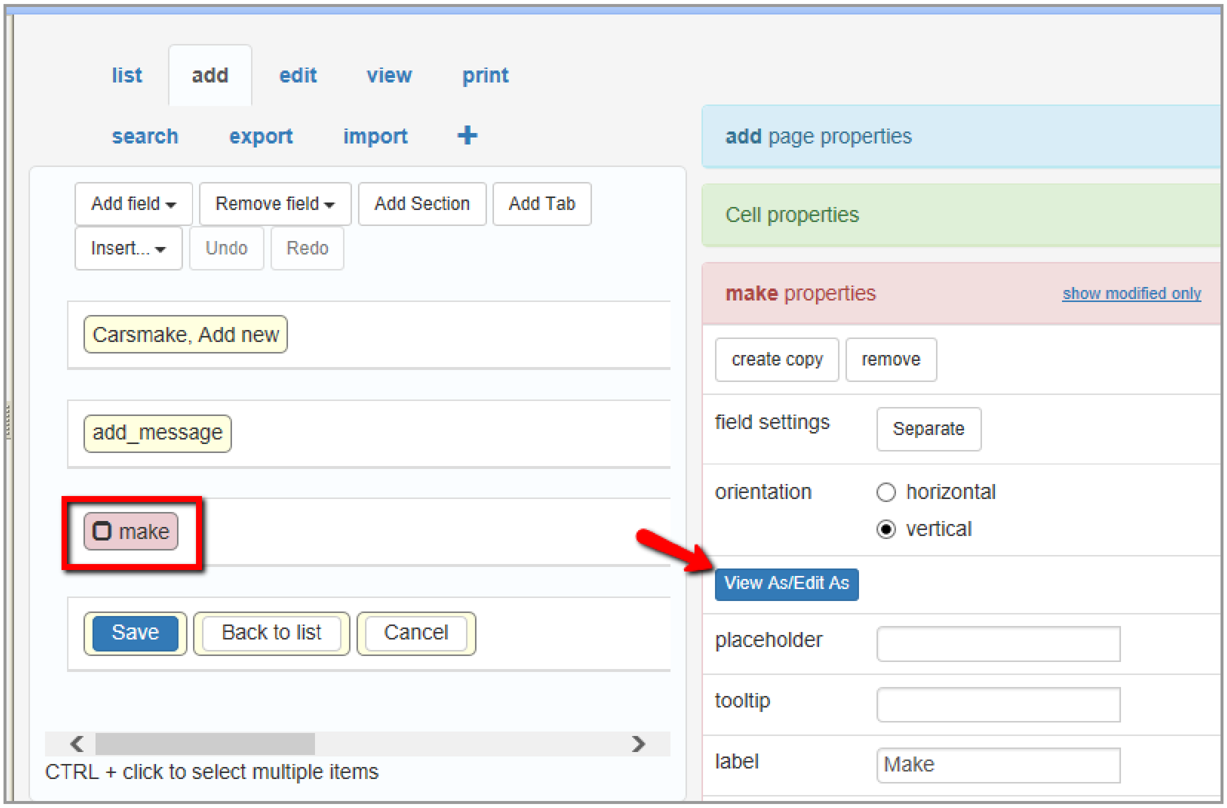
2.16.13.1 "Filter as" settings

You can choose what fields will be shown on the Filter panel by using the Add/Remove field buttons in the page designer.



You can customize your data appearance on the Filter Panel using formatting options on the **"Filter as" settings** dialog. To open this dialog select the field and click on the **View As/Edit**

As button in the field properties on the right panel.



field settings: Separate

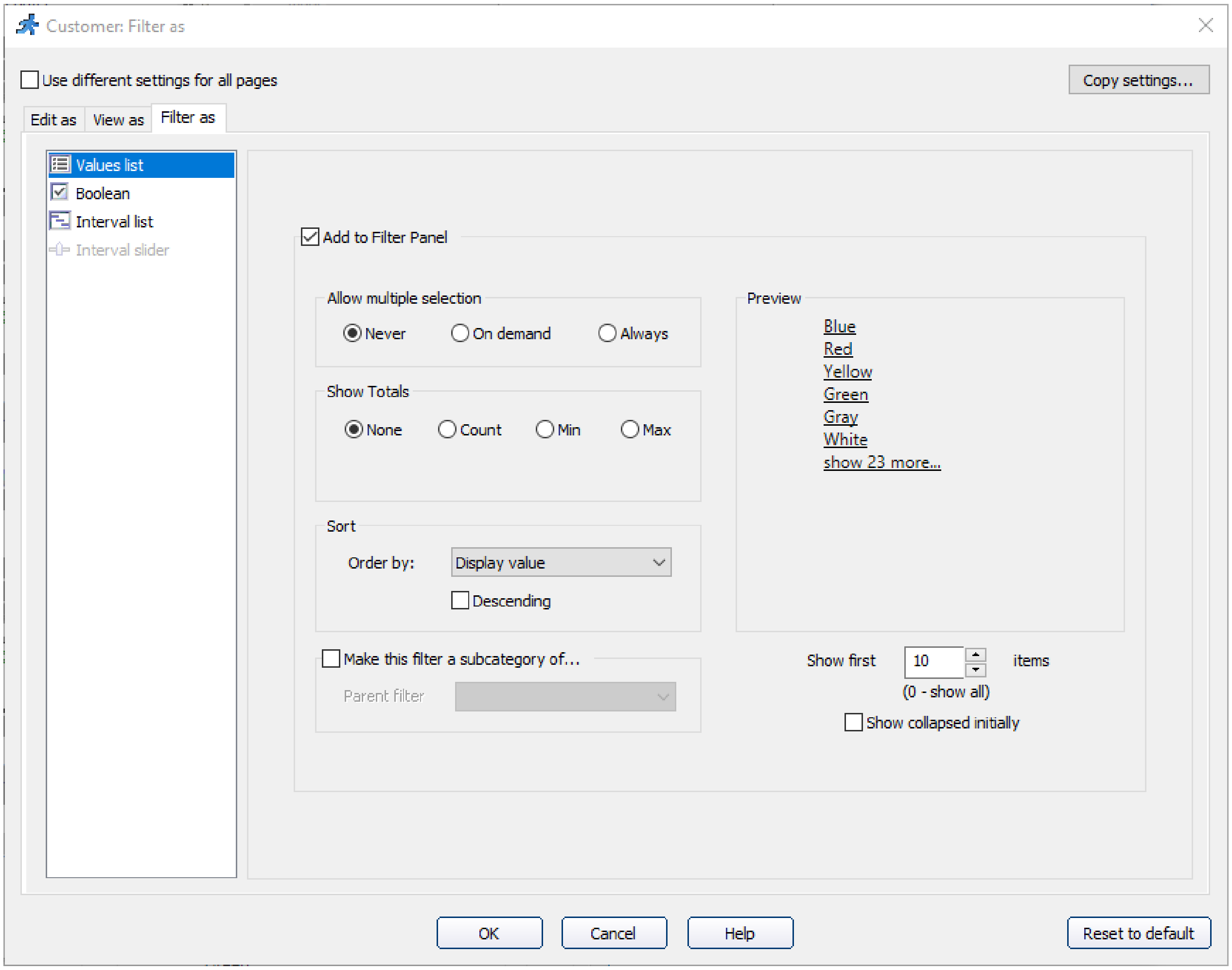
orientation: ○ horizontal ● vertical

placeholder:

tooltip:

label: Make

On the Filter As tab select Filter format on the left and set appropriate options.



Customer: Filter as

□ Use different settings for all pages

Copy settings...

Edit as | View as | Filter as

Values list
Boolean
Interval list
Interval slider

☑ Add to Filter Panel

Allow multiple selection

● Never ○ On demand ○ Always

Show Totals

● None ○ Count ○ Min ○ Max

Sort

Order by: Display value

□ Descending

□ Make this filter a subcategory of...

Parent filter

Preview

Blue
Red
Yellow
Green
Gray
White
show 23 more...

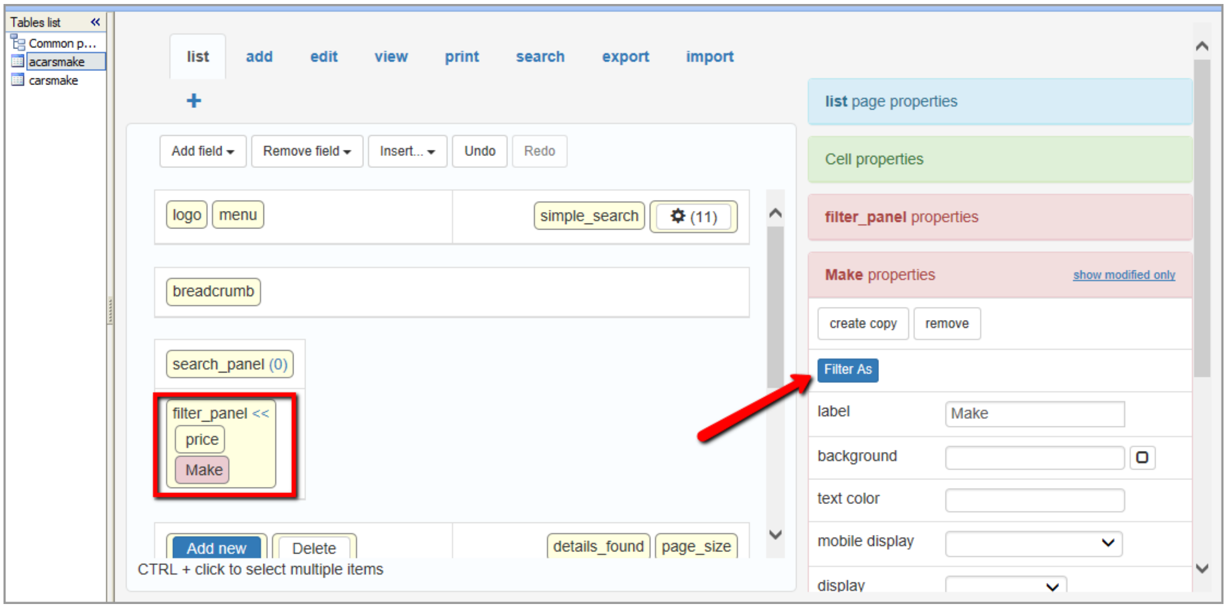
Show first 10 items
(0 - show all)

□ Show collapsed initially

OK | Cancel | Help | Reset to default

Now you can see that the field is displayed on the Page Designer inside the Filter panel element. Those fields have Filter As button in properties to the right.

You can click them to open the Filter as settings of the field.

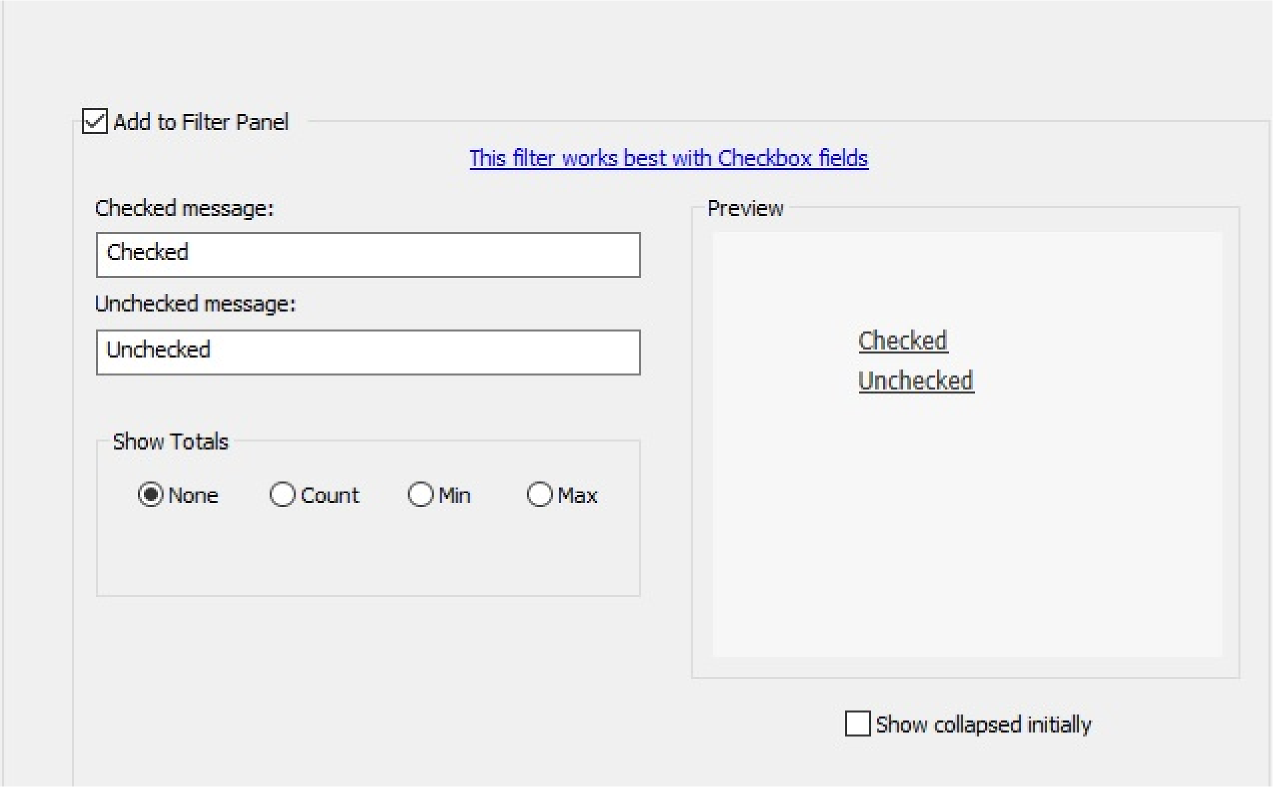


Another way to enable/disable filter on your pages is to use Search and Filter settings.

"Filter as" formats
Values list
Boolean
Interval list
Interval slider

2.16.13.2 Boolean

Displays filter as Checked/Unchecked values on default or your custom values (i.e. Yes/No, True/False). This filter works best with Checkbox fields. You can select whether to show totals.



☑ Add to Filter Panel

This filter works best with Checkbox fields

Checked message: Checked

Unchecked message: Unchecked

Show Totals

● None ○ Count ○ Min ○ Max

Preview

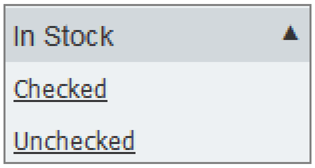
Checked

Unchecked

□ Show collapsed initially

Examples of how filter is displayed in browser:

1. Show totals - "None". Default checked/unchecked messages.

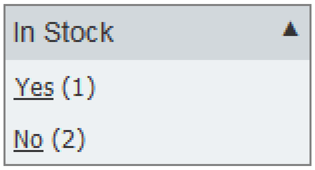


In Stock ▲

Checked

Unchecked

2. Show totals - "Count". Checked/unchecked messages - "Yes"/"No".



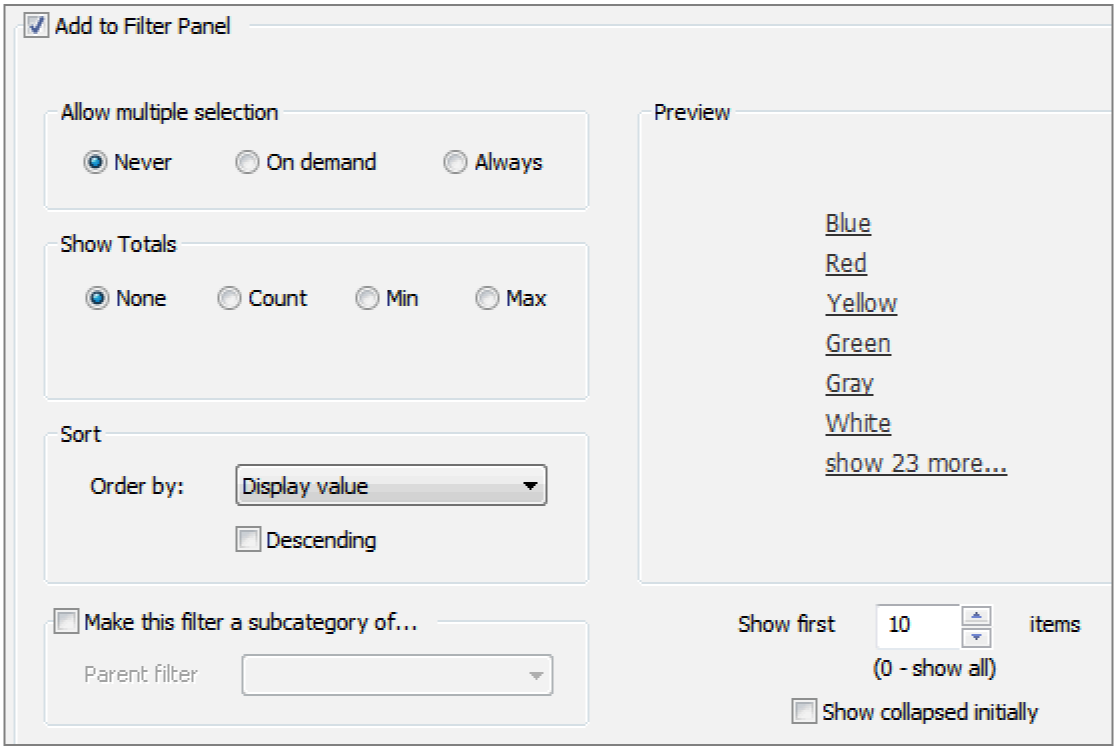
In Stock ▲

Yes (1)

No (2)

2.16.13.3 Values list

Displays filter as list of values. You can select whether to allow multiple selection and show totals. Sort filter options by display or database values. Set the number of first X filter options on initial load (useful when you have lots of categories).



☑ Add to Filter Panel

Allow multiple selection

● Never ○ On demand ○ Always

Show Totals

● None ○ Count ○ Min ○ Max

Sort

Order by: Display value

□ Descending

□ Make this filter a subcategory of...

Parent filter

Preview

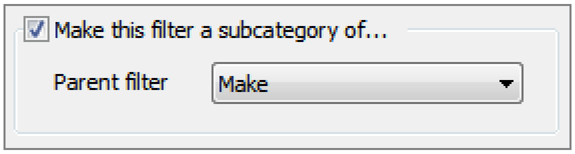
Blue
Red
Yellow
Green
Gray
White
show 23 more...

Show first 10 items

(0 - show all)

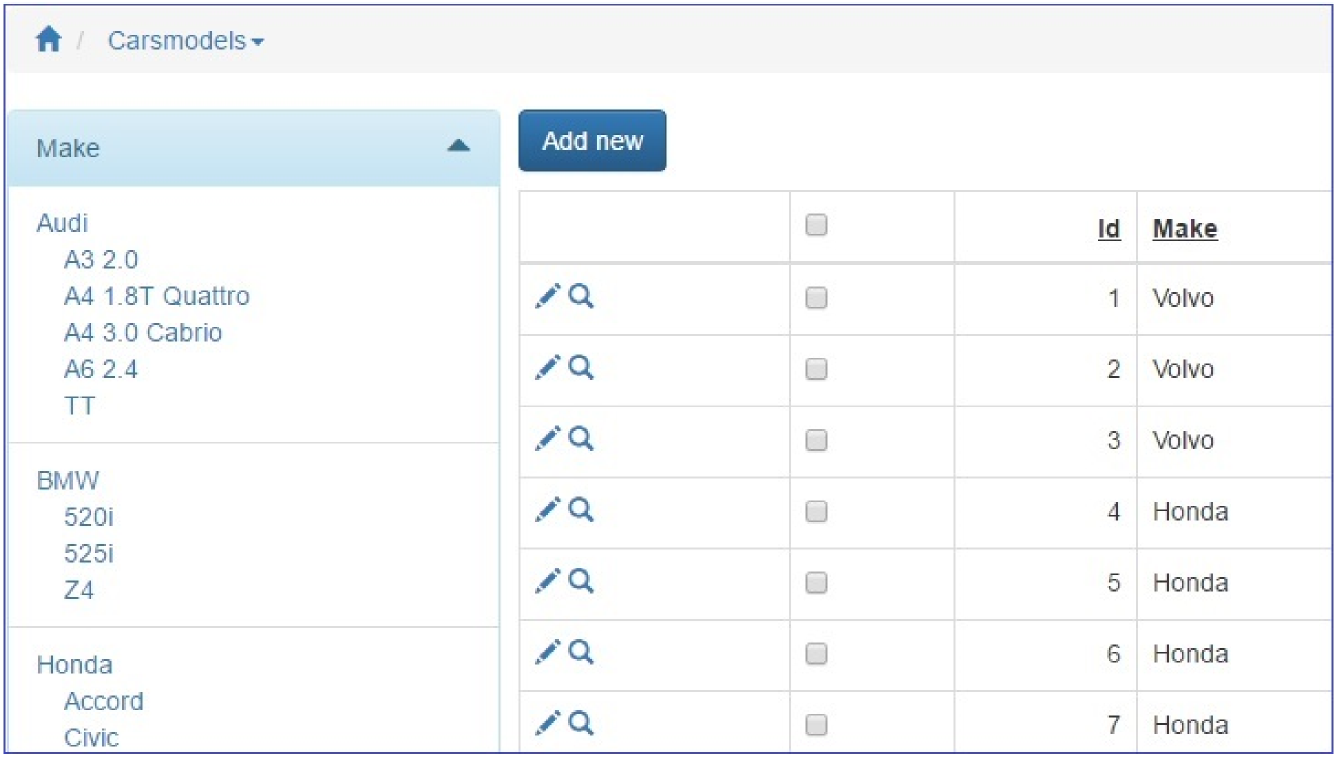
□ Show collapsed initially

Select parent filter and filter data by categories and subcategories in the browser:



☑ Make this filter a subcategory of...

Parent filter Make



Audi
A3 2.0
A4 1.8T Quattro
A4 3.0 Cabrio
A6 2.4
TT

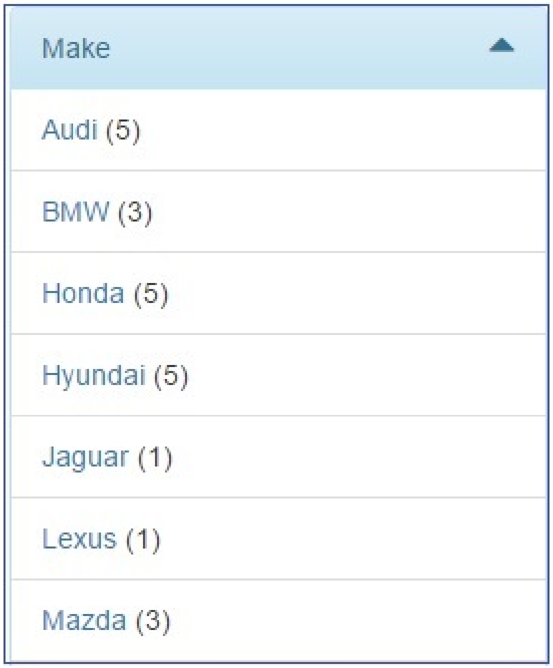
BMW
520i
525i
Z4

Honda
Accord
Civic

	□	Id	Make
	□	1	Volvo
	□	2	Volvo
	□	3	Volvo
	□	4	Honda
	□	5	Honda
	□	6	Honda
	□	7	Honda

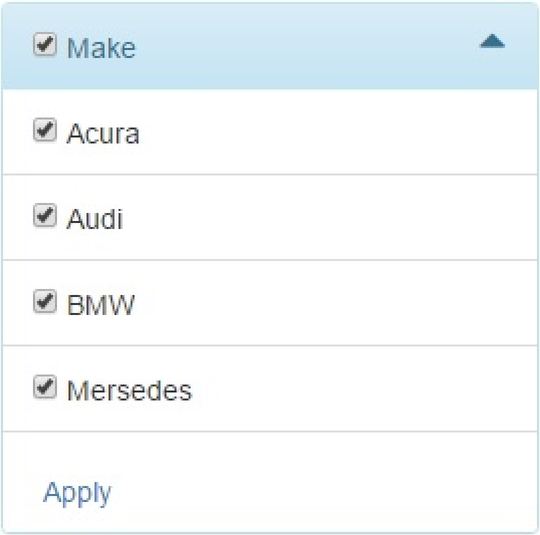
Examples of how filter is displayed in browser:

1. Allow multiple selection - "Never", Show totals - "Count".

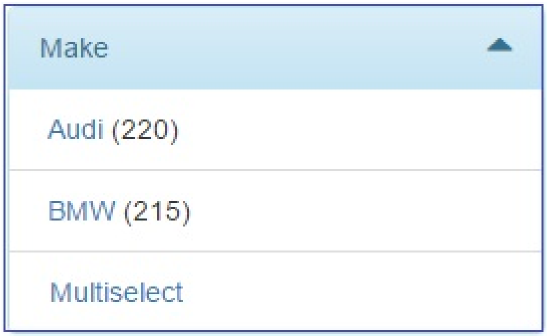


Audi (5)
BMW (3)
Honda (5)
Hyundai (5)
Jaguar (1)
Lexus (1)
Mazda (3)

2. Allow multiple selection - "Always", Show totals - "None".

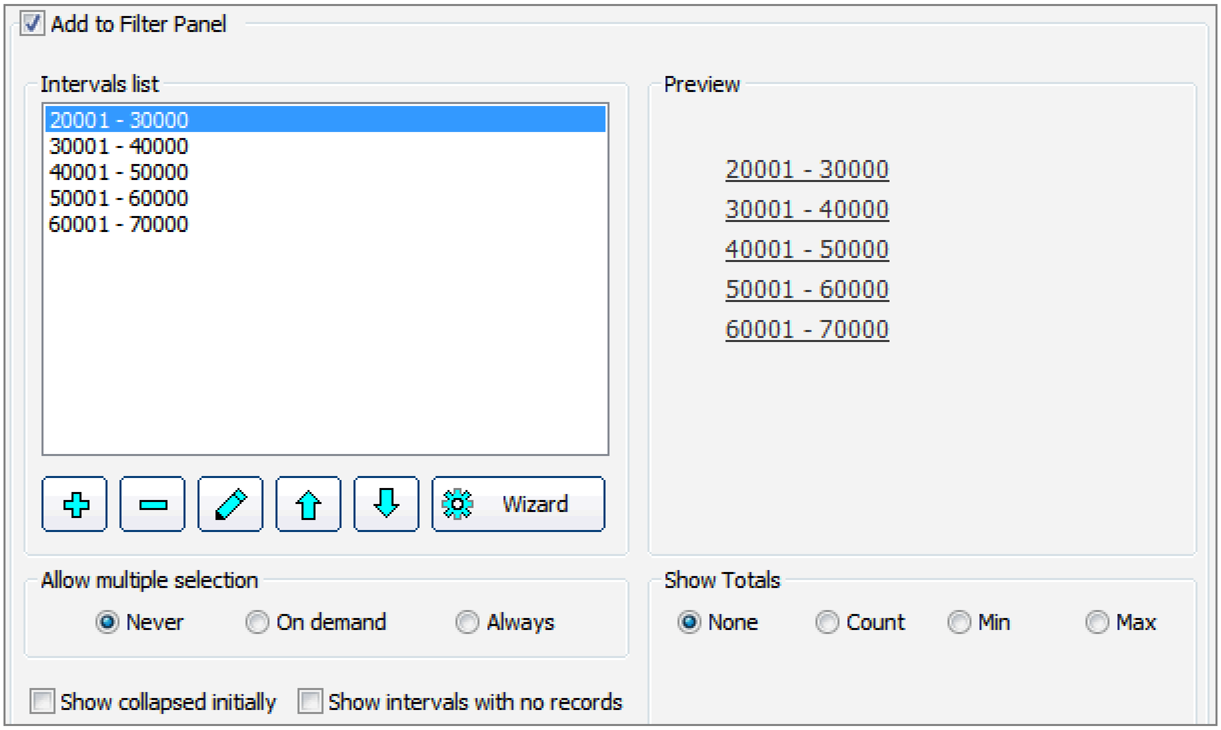


3. Allow multiple selection - "On demand", Show totals - "Max" (*Horsepower* field).



2.16.13.4 Interval list

Displays filter as interval list. You can create intervals list manually or using Wizard, select whether to allow multiple selection and show totals.



☑ Add to Filter Panel

Intervals list

20001 - 30000
30001 - 40000
40001 - 50000
50001 - 60000
60001 - 70000

Preview

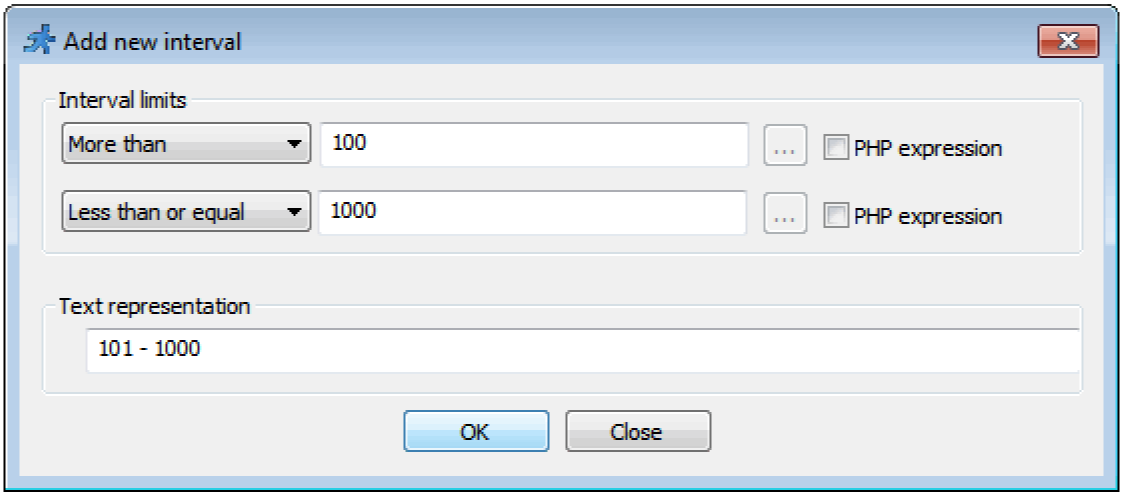
20001 - 30000
30001 - 40000
40001 - 50000
50001 - 60000
60001 - 70000

Allow multiple selection: ● Never ○ On demand ○ Always

Show Totals: ● None ○ Count ○ Min ○ Max

□ Show collapsed initially □ Show intervals with no records

Use 'plus'/'minus' buttons to add/delete intervals and buttons with arrows to order them. Also you can edit interval using button with pensil.



Add new interval

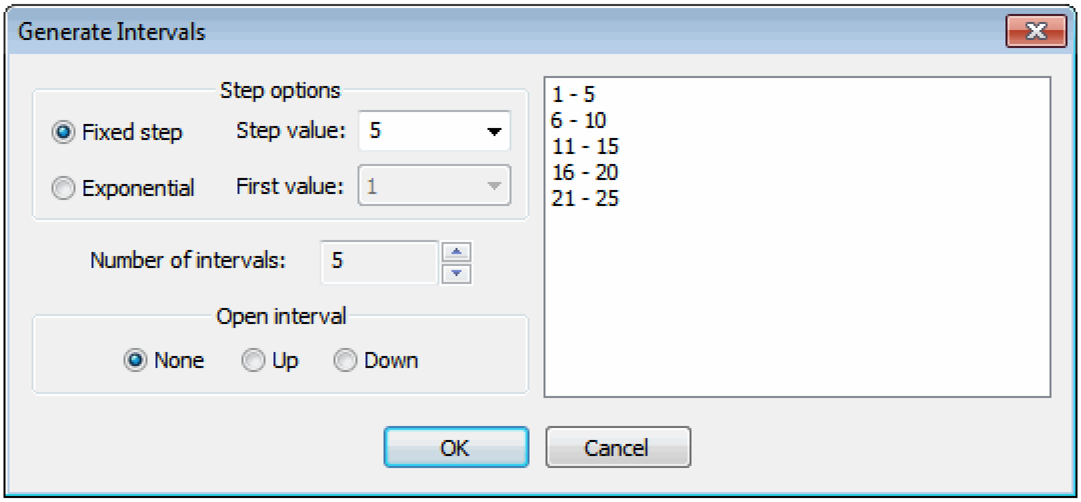
Interval limits

More than: 100 □ PHP expression

Less than or equal: 1000 □ PHP expression

Text representation: 101 - 1000

Use Wizard to generate multiple intervals at once. For numeric fields you can define the intervals' step options and number of intervals.



Step options

● Fixed step Step value: 5

○ Exponential First value: 1

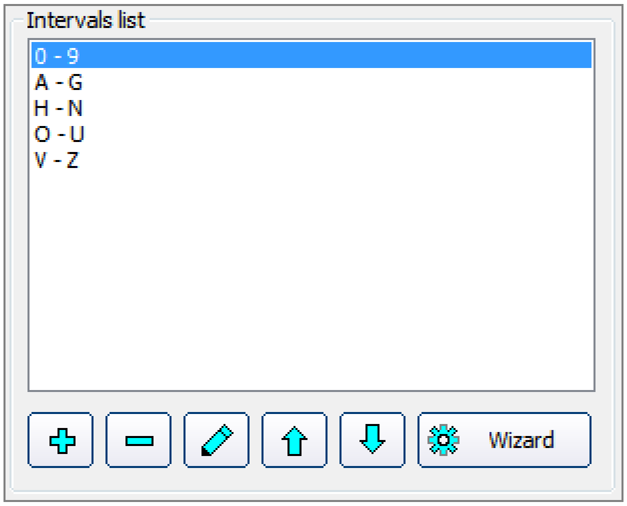
Number of intervals: 5

Open interval

● None ○ Up ○ Down

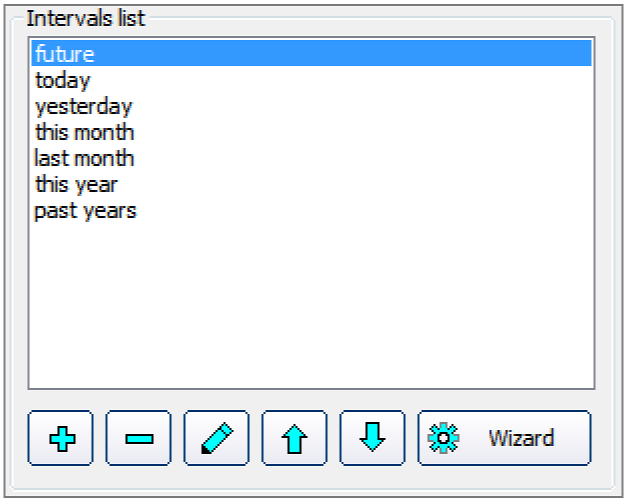
```
1 - 5
6 - 10
11 - 15
16 - 20
21 - 25
```

Wizard will generate the following intervals for text fields:



```
0 - 9
A - G
H - N
O - U
V - Z
```

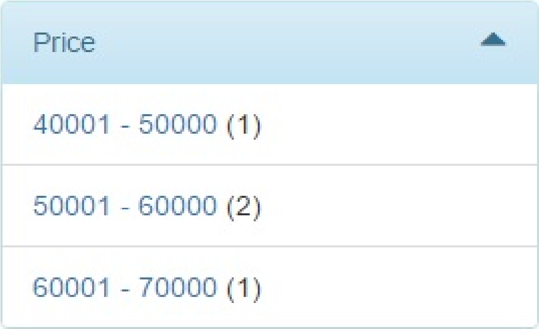
and the following intervals for date fields :



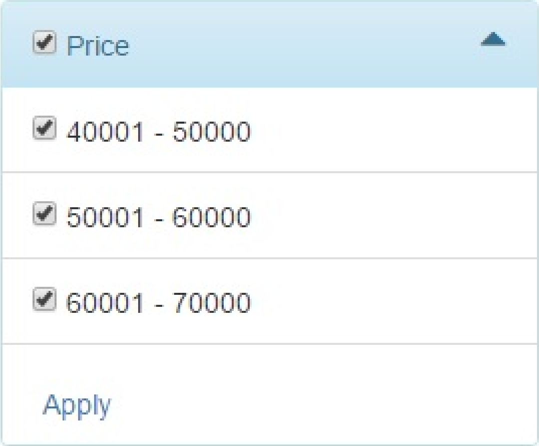
```
future
today
yesterday
this month
last month
this year
past years
```

Examples of how filter is displayed in browser:

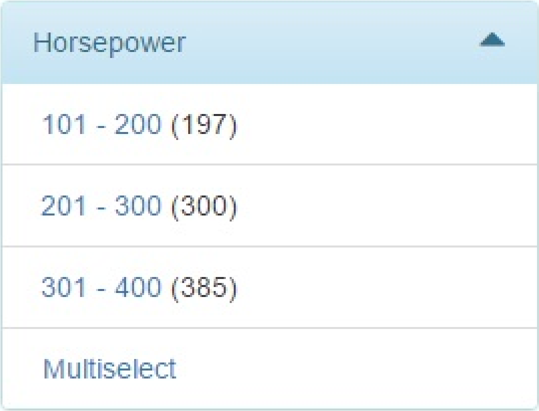
1. Allow multiple selection - "Never", Show totals - "Count".



2. Allow multiple selection - "Always", Show totals - "None".

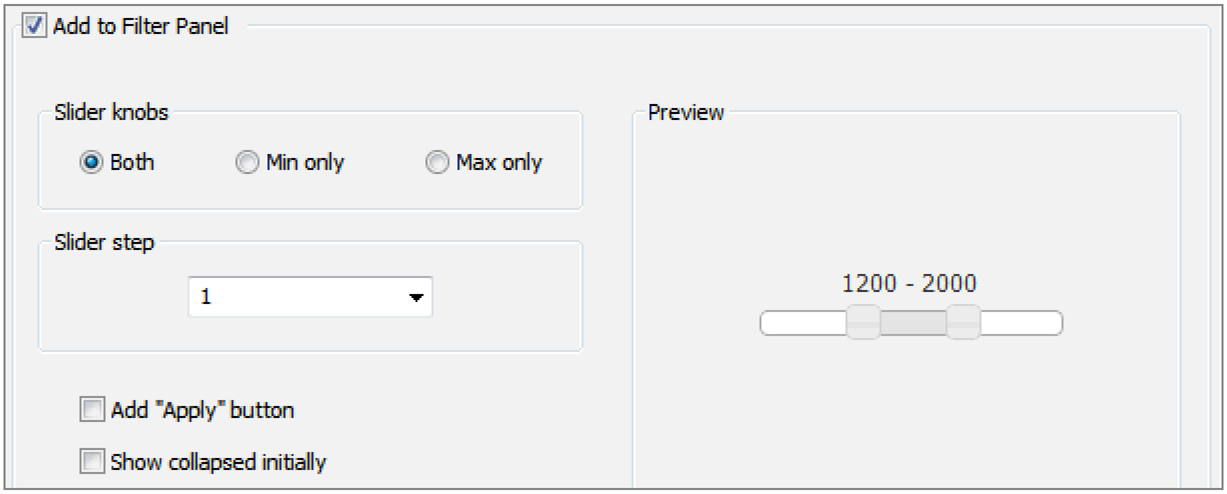


3. Allow multiple selection - "On demand", Show totals - "Max" (*Horsepower* field).



2.16.13.5 Interval slider

Displays filter as interval slider. You can select the appearance of slider knobs and slider step.



☑ Add to Filter Panel

Slider knobs

● Both ○ Min only ○ Max only

Slider step

1

□ Add "Apply" button

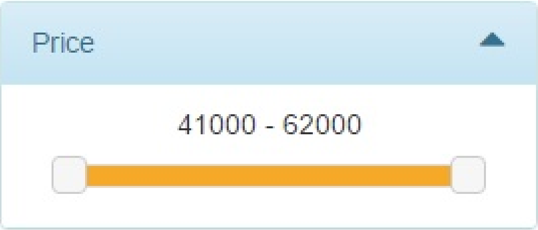
□ Show collapsed initially

Preview

1200 - 2000

Examples of how filter is displayed in browser:

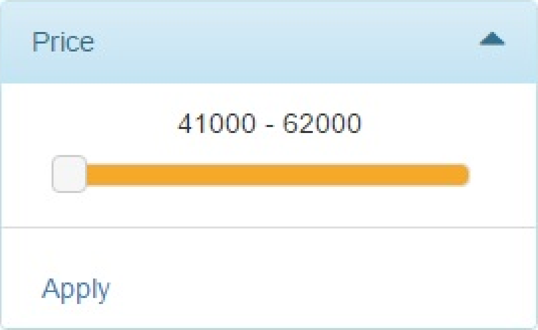
1. Slider knobs - "Both".



Price

41000 - 62000

2. Slider knobs - "Min only", added "Apply" button.



Price

41000 - 62000

Apply

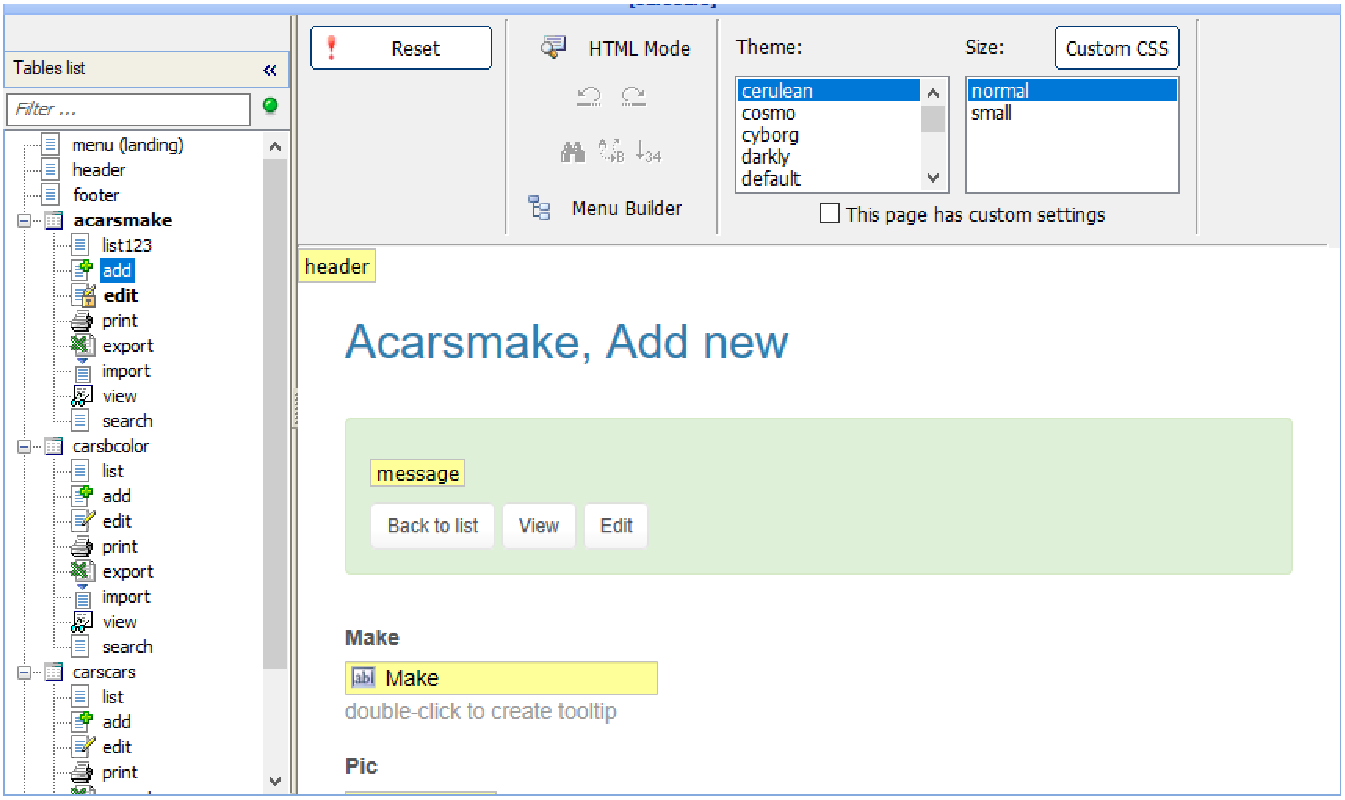
2.17 Editor

2.17.1 About Editor

Quick jump

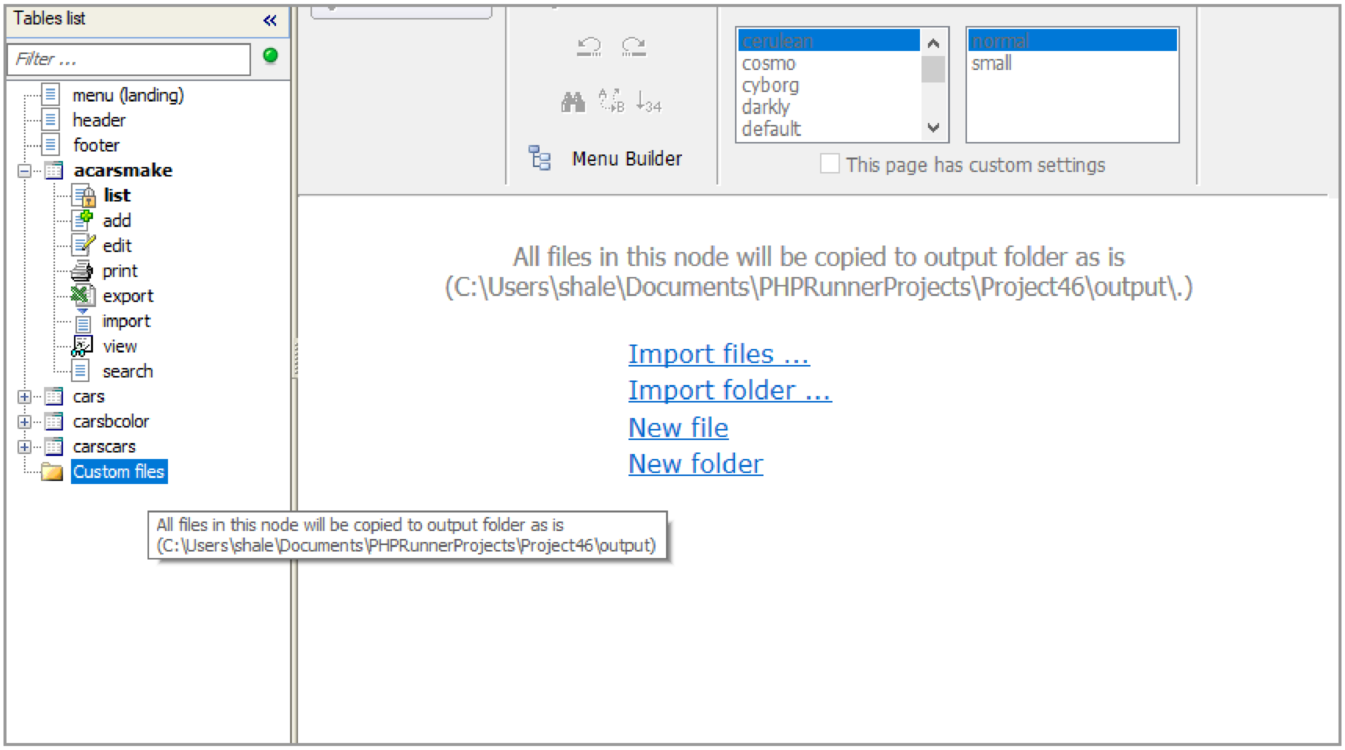
Adding custom pages to your application
Editing field names
Setting landing page
Locking of custom pages

Visual Editor built-in in PHPRunner is an editor control that allows the user to edit HTML contents in a more user friendly way. The editor control is very flexible and enables the advanced user to change the appearance of the pages in design or html mode.



Adding custom pages to your application

You can add a new page to your application or import existing one. You can import the whole folder with HTML, images, CSS files and edit/preview them in Visual Editor.



All files in this node will be copied to output folder as is (C:\Users\shale\Documents\PHPRunnerProjects\Project46\output\.)

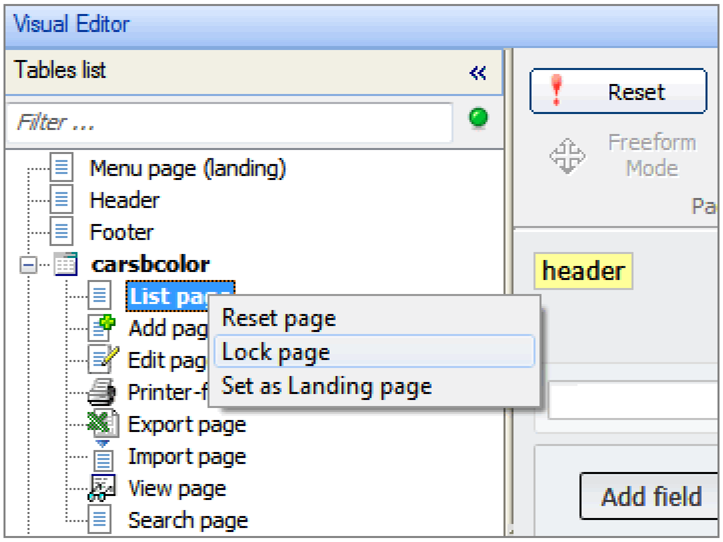
Import files ...
Import folder ...
New file
New folder

All files in this node will be copied to output folder as is (C:\Users\shale\Documents\PHPRunnerProjects\Project46\output)

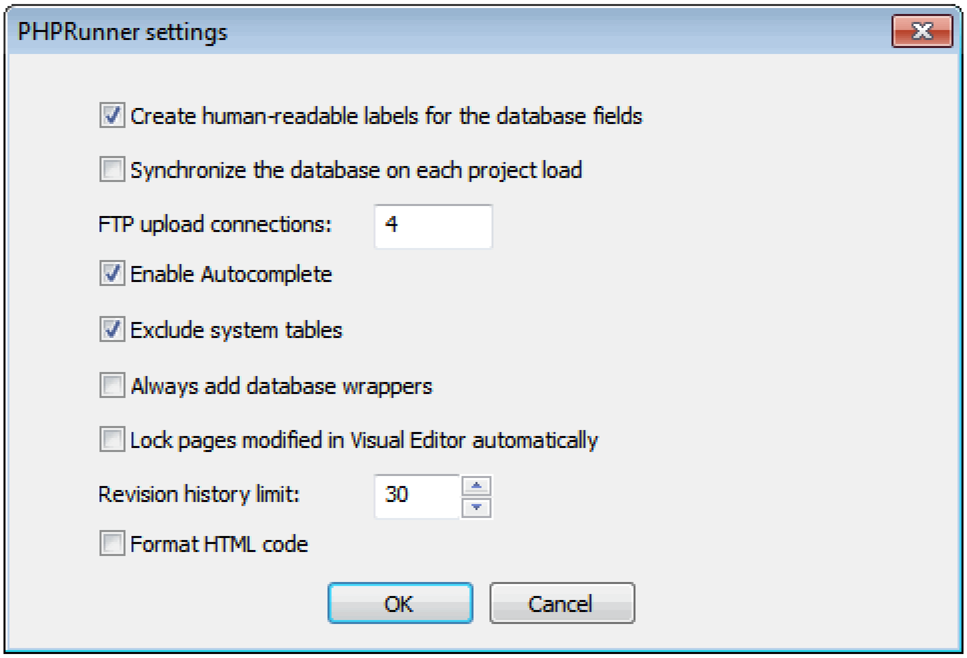
☐ This page has custom settings

Locking of custom pages

After you modified a page in Visual Editor, you can lock it from further automatic modifications. To lock a page, right click the page name and select **Lock page**. The lock icon will appear before the name of locked page. Note that you can modify locked pages manually. To unlock the page, right click the page name and select **Unlock page**.



Also you can enable **Lock pages modified in Visual Editor automatically** option in **Project -> Settings**. After that all pages that you manually modify will be locked from further automatic modifications.



PHPRunner settings

☑ Create human-readable labels for the database fields

□ Synchronize the database on each project load

FTP upload connections: 4

☑ Enable Autocomplete

☑ Exclude system tables

□ Always add database wrappers

□ Lock pages modified in Visual Editor automatically

Revision history limit: 30

□ Format HTML code

Editing field names

Now you can edit field names directly Page Designer. But if multiple project languages are selected on **Miscellaneous** page, you will not be able to do that. In this case double-click the field name and edit it in the opened Label Editor.

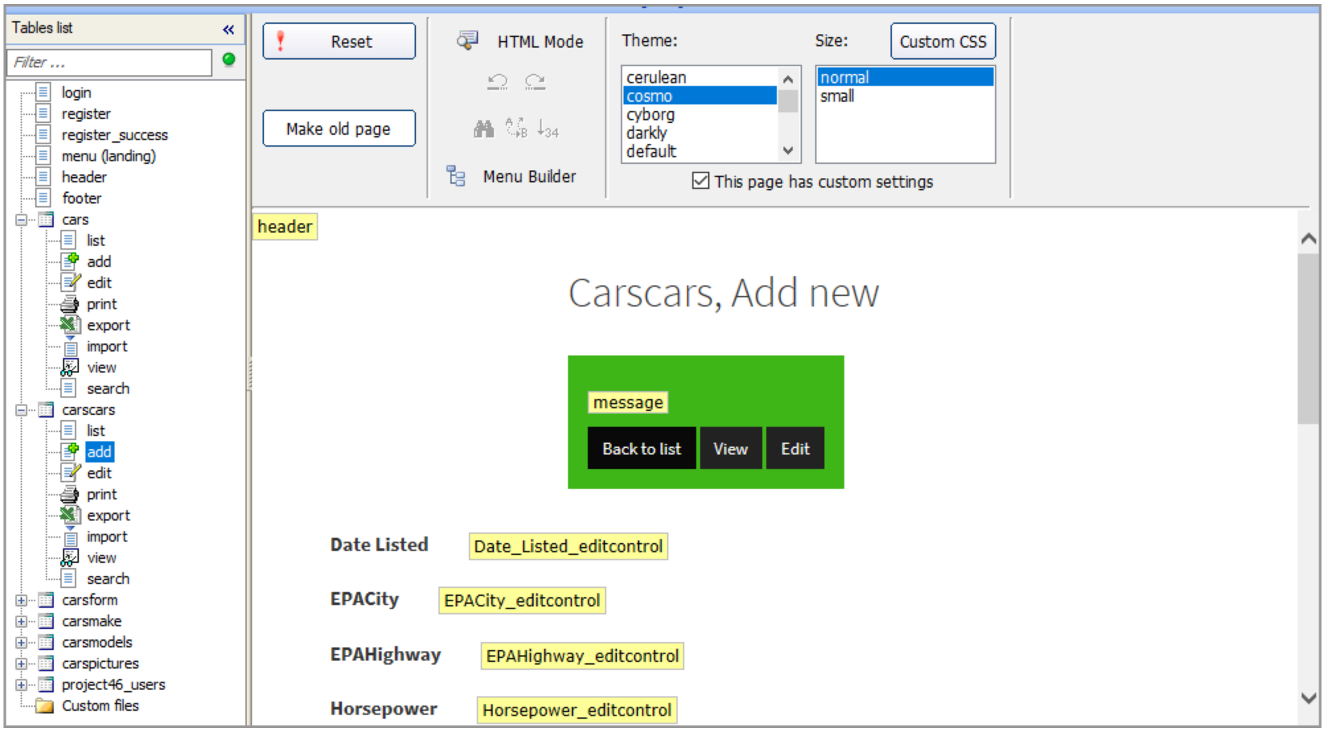
Setting landing page

You can set any page as landing page - the page to open when the user enters the site (if login is disabled or login and guest user login are enabled) or just after login. To do that right click the page name and select **Set as Landing page**.

The **Visual Editor** page allows you to modify the visual appearance of your pages and instantly preview the results.
You can see two sections on that page:

- **Tables list** on the left. Moving across tables and their related pages, you can select a page to view/modify. To filter pages by the page name, use the text field above the events tree.
- **Editor** on the right where you can alter certain aspects of your page, such as the project theme, fonts size and custom CSS, and preview how the selected page will look like in a browser.

This page has custom settings is an option that allows you to select the appearance for pages that differ from the project ones.



Theme: cosmo

Size: normal

☑ This page has custom settings

Definition of layout, style, color scheme and theme

Layout

A layout is a page template (for an Add page, Edit page etc). Each template defines corresponding page structure and stored in the *.ly* file in the folder *C:\Program Files\PHPRunner10.0\templates\layouts*.

You can choose the layouts for all pages by selecting the project scheme, for all List/Add/Edit/etc. pages by selecting layout from the Layouts list and for the currently selected page by enabling **This page has custom settings** option and then selecting layout from the Layouts list.

Styles

Style is a set of parameters that includes font, font size, margins, borders, alignment and more. Styles are stored as the *.style* files in the folder *C:\Program Files\PHPRunner10.0\styles*.

Color schemes

The color scheme describes the color palette of layout elements. Color schemes are stored as the *.color* files in the folder *C:\Program Files\PHPRunner10.0\styles\colors*.

Themes

Themes combine the properties of the Styles and the Color schemes.

2.17.2 Customizing CSS examples

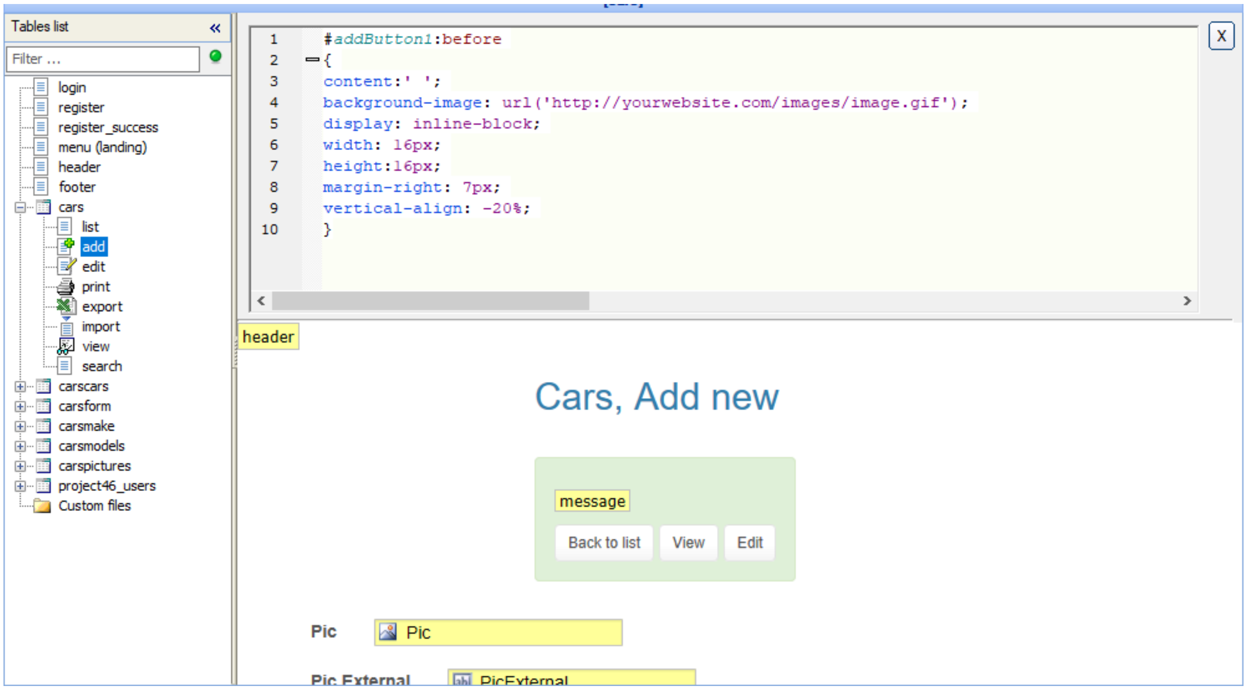
Quick jump	
How to change font in grid on the List page	How to make Bootstrap1 layout grid 100% width
How to hide "required" icon	How to change cursor to a hand icon while hovering over a dashboard element
How to change the color of menu items and project name	How to turn on word wrap for all table cells
How to change the color of a regular button	How to turn off word wrap for all table cells
How to change the color of a main button	Making the 'View details' icon bigger
How to change the color of the menu bar	How to change 'View details' icon mouseover tooltip
How to change the text color of breadcrumb item	How to change icon to the word "Details"
How to change the text color of breadcrumb container	How to change the width, font size and color of search suggest window
How to change color of dashboard panel headers	How to change the color of the info on the Welcome page panels
How to set the image as the page background	How to make the login form semi-transparent

See also:
Building a nice looking login page with custom CSS

For example, let's say you want to add a background image only to one List page in the project. Select this List page in Visual Editor. Enable **This page has custom settings**. Click **Custom CSS** and add the following code:

```
body.function-list {
height:100%;
background:transparent url("http://mywebsite.com/images/some_pic.jpg")
no-repeat center center fixed;
background-size:cover;
}
```

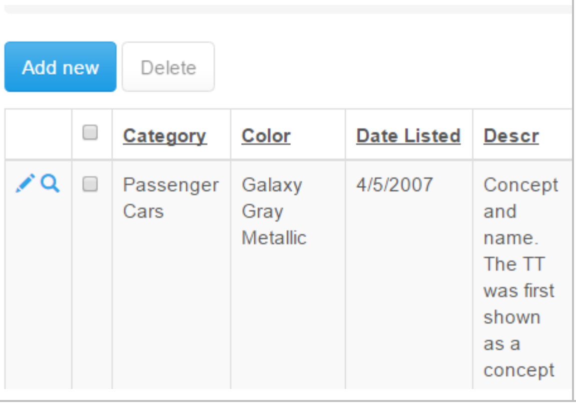
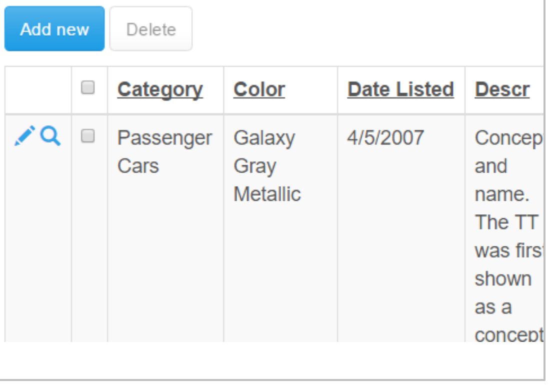
When you have finished CSS customization, click the **Close** X button to return to the view mode.
You can also add some custom CSS in the Page Designer Cell/Element Properties.



```
#addButton1:before
{
content:' ';
background-image: url('http://yourwebsite.com/images/image.gif');
display: inline-block;
width: 16px;
height:16px;
margin-right: 7px;
vertical-align: -20%;
}
```

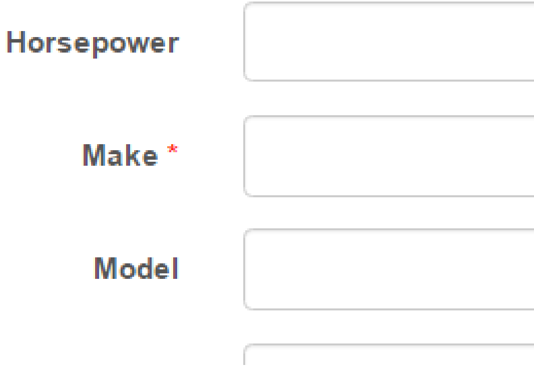
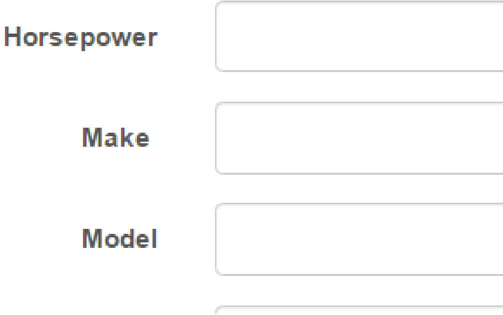
Example 1. How to change font in grid on the List page.

```
[data-brick=grid] {
font-size: 20px;
}
```

Before custom CSS was applied	After custom CSS was applied
 Add new Delete Category Color Date Listed Descr Passenger Cars Galaxy Gray Metallic 4/5/2007 Concept and name. The TT was first shown as a concept	 Add new Delete Category Color Date Listed Descr Passenger Cars Galaxy Gray Metallic 4/5/2007 Concep and name. The TT was firs shown as a concept

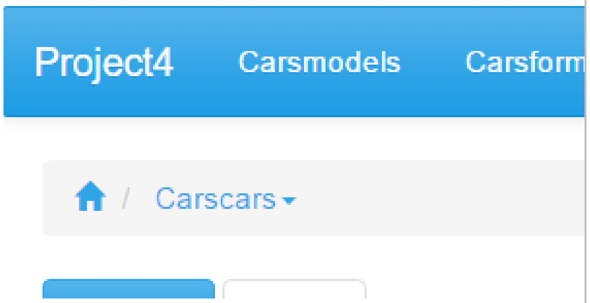
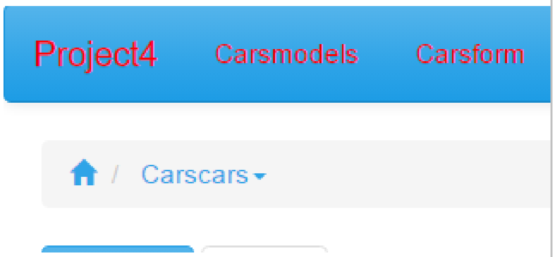
Example 2. How to hide "required" icon

```
.icon-required {
display: none;
}
```

Before custom CSS was applied	After custom CSS was applied
	

Example 3. How to change the color of menu items and project name

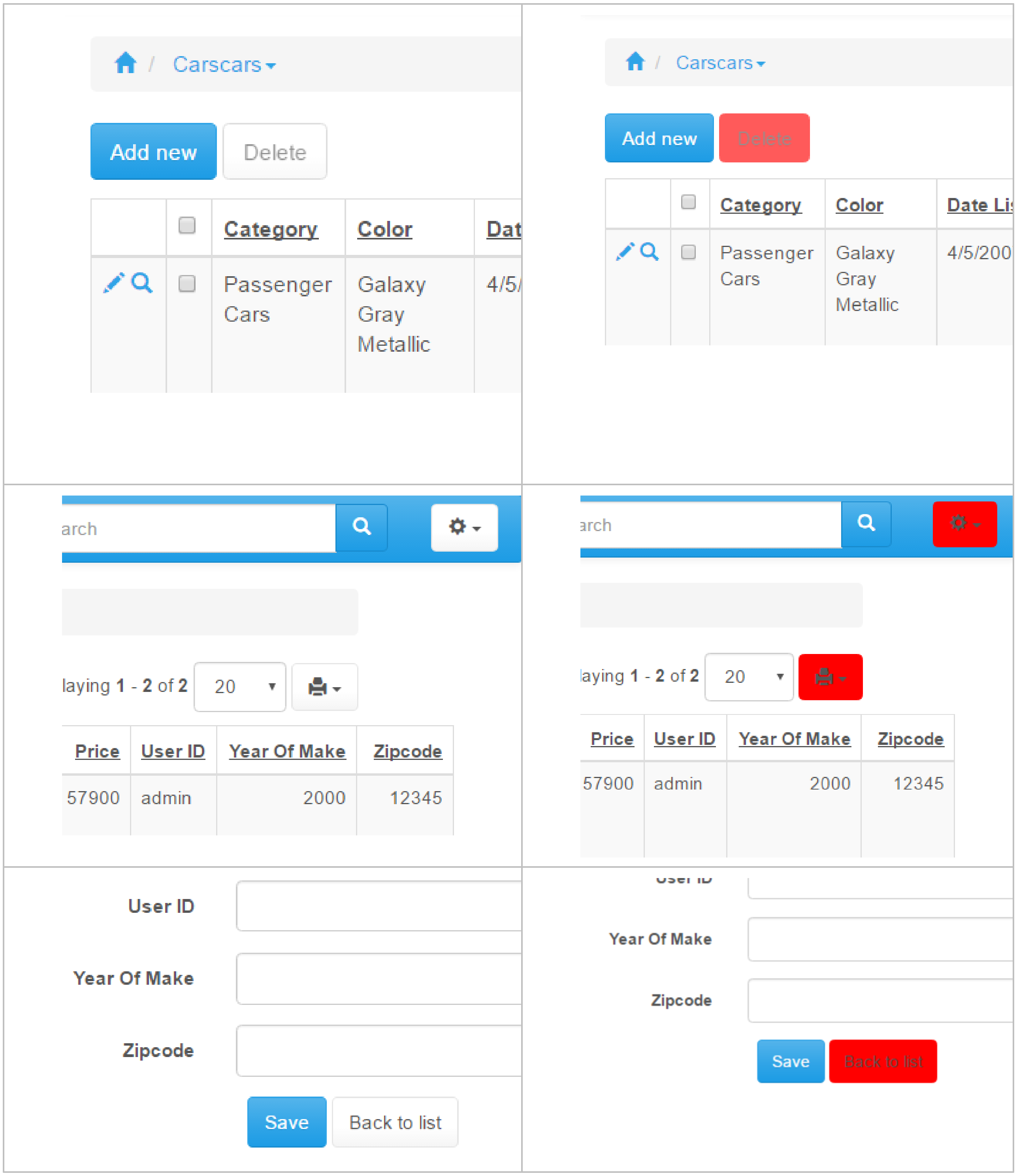
```
.navbar-nav>li>a, a.navbar-brand {
color: red !important;
}
```

Before custom CSS was applied	After custom CSS was applied
	

Example 4. How to change the color of a regular button

```
.btn.btn-default {
background: red;
border-color: red;
}
```

Before custom CSS was applied	After custom CSS was applied



	□	Category	Color	Dat
	□	Passenger Cars	Galaxy Gray Metallic	4/5/

	□	Category	Color	Date Li
	□	Passenger Cars	Galaxy Gray Metallic	4/5/200

laying **1** - **2** of **2** 20

Price	User ID	Year Of Make	Zipcode
57900	admin	2000	12345

laying **1** - **2** of **2** 20

Price	User ID	Year Of Make	Zipcode
57900	admin	2000	12345

User ID

Year Of Make

Zipcode

User ID

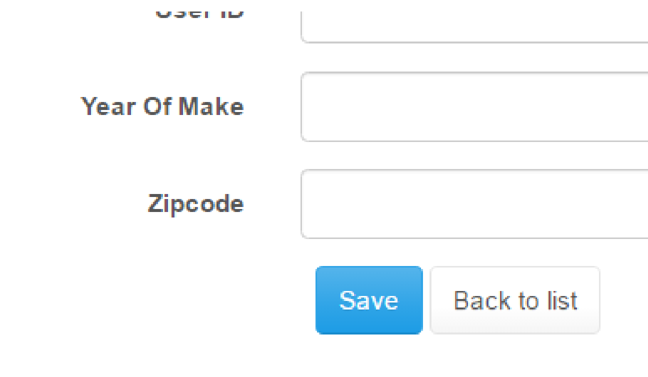
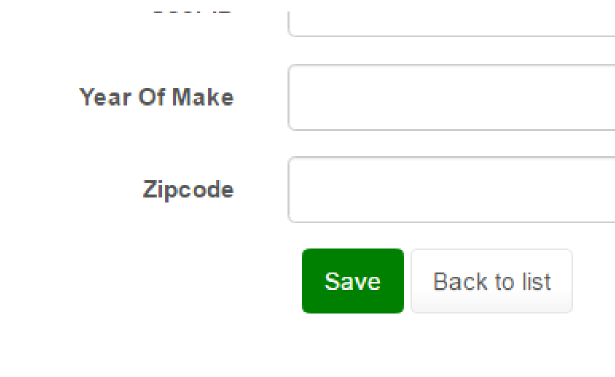
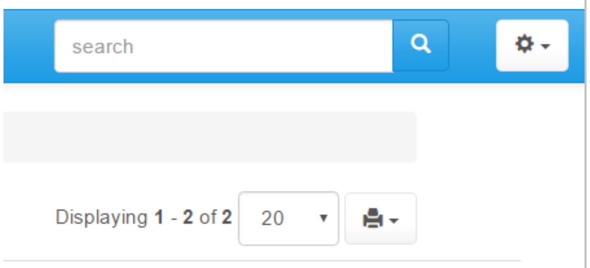
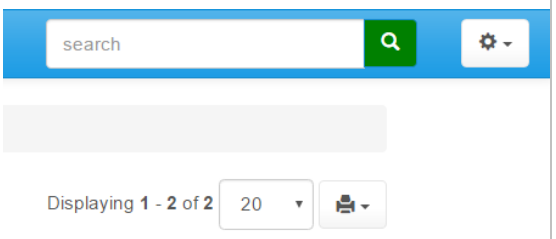
Year Of Make

Zipcode

Example 5. How to change the color of a main button

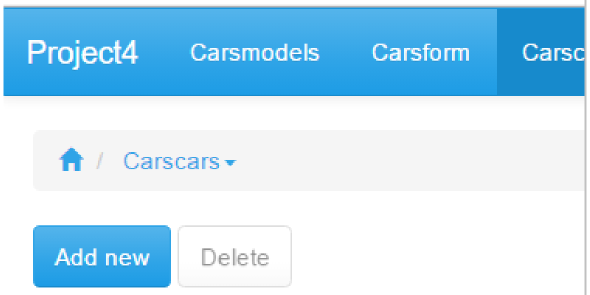
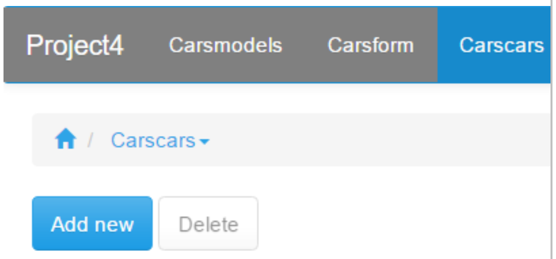
```
    .btn.btn-primary {
```

```
    background: green;
    border-color: green;
    }
```

Before custom CSS was applied	After custom CSS was applied
	
	

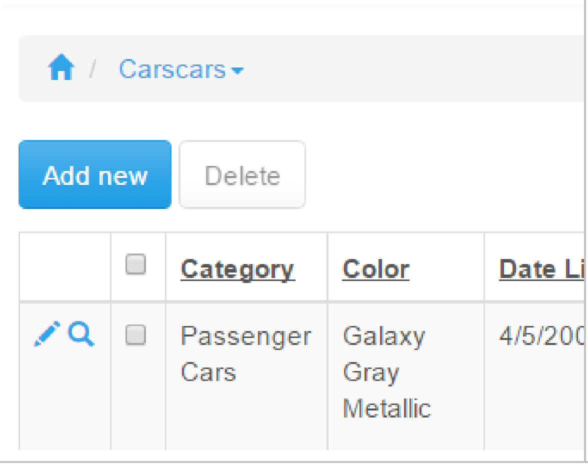
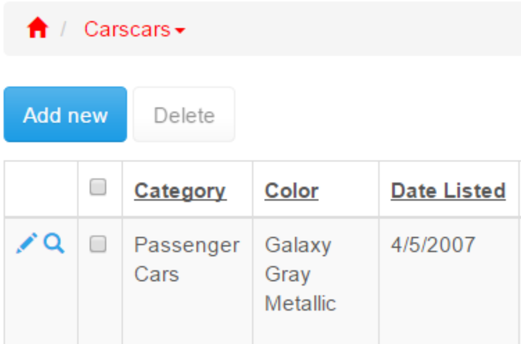
Example 6. How to change the color of the menu bar

```
    .bs-navform.bs-navform.bs-navform {
    background: grey;
    }
```

Before custom CSS was applied	After custom CSS was applied
 Project4 Carsmodels Carsform Carsc Carscars Add new Delete	 Project4 Carsmodels Carsform Carscars Carscars Add new Delete

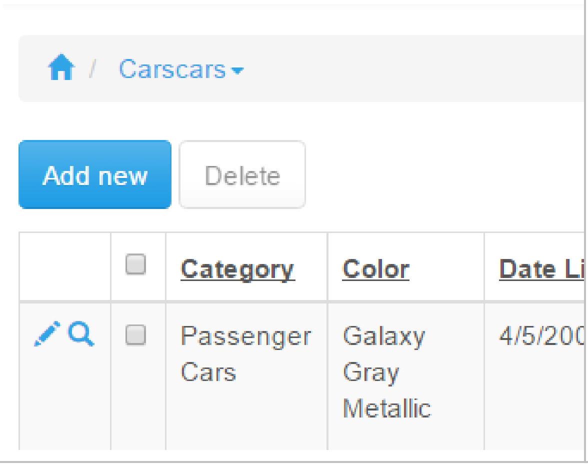
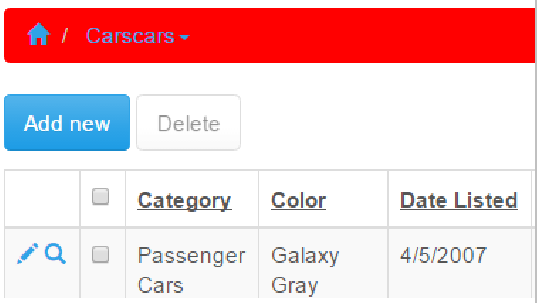
Example 7. How to change the text color of breadcrumb item

```
ol.breadcrumb a {
color: red;
}
```

Before custom CSS was applied	After custom CSS was applied
	

Example 8. How to change the text color of breadcrumb container

```
ol.breadcrumb {
background: red;
}
```

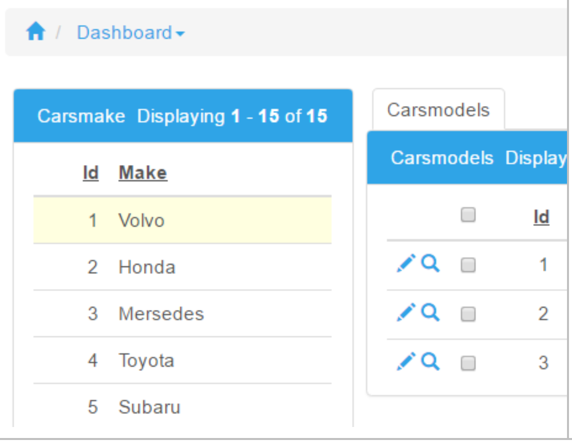
Sample screenshot before	After custom CSS was applied
	

Example 9. How to change color of dashboard panel headers

```
.panel-primary > div.panel-heading {
```

```
background: #808080;
}
```

Before custom CSS was applied



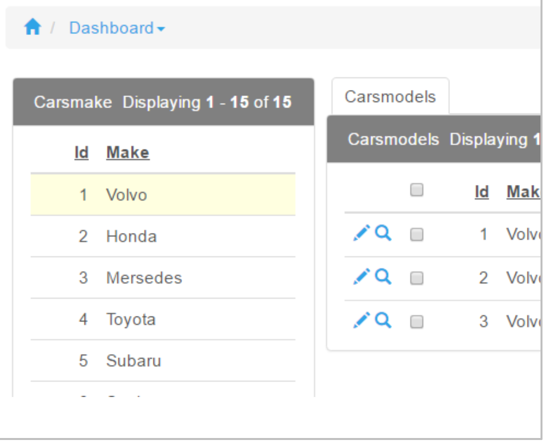
Carsmake Displaying 1 - 15 of 15

Id	Make
1	Volvo
2	Honda
3	Mersedes
4	Toyota
5	Subaru

Carsmodels Display

Id
1
2
3

After custom CSS was applied



Carsmake Displaying 1 - 15 of 15

Id	Make
1	Volvo
2	Honda
3	Mersedes
4	Toyota
5	Subaru

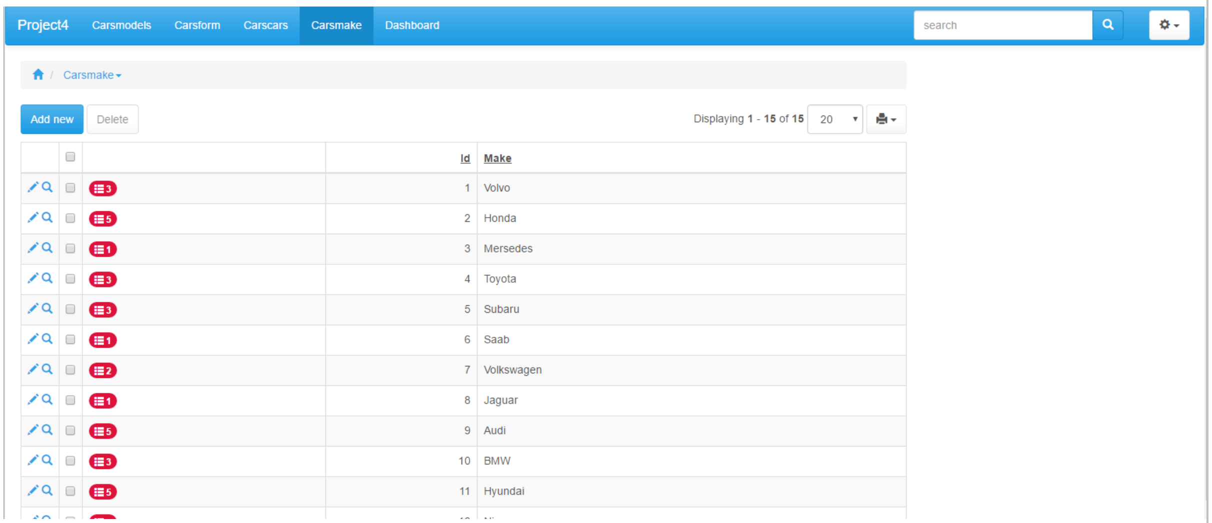
Carsmodels Displaying 1

Id	Mak
1	Volv
2	Volv
3	Volv

Example 10. How to make Bootstrap1 layout grid 100% width

```
.bs-center, .bs-middle, .bs-grid, .bs-grid > table {
width: 100% !important;
}
```

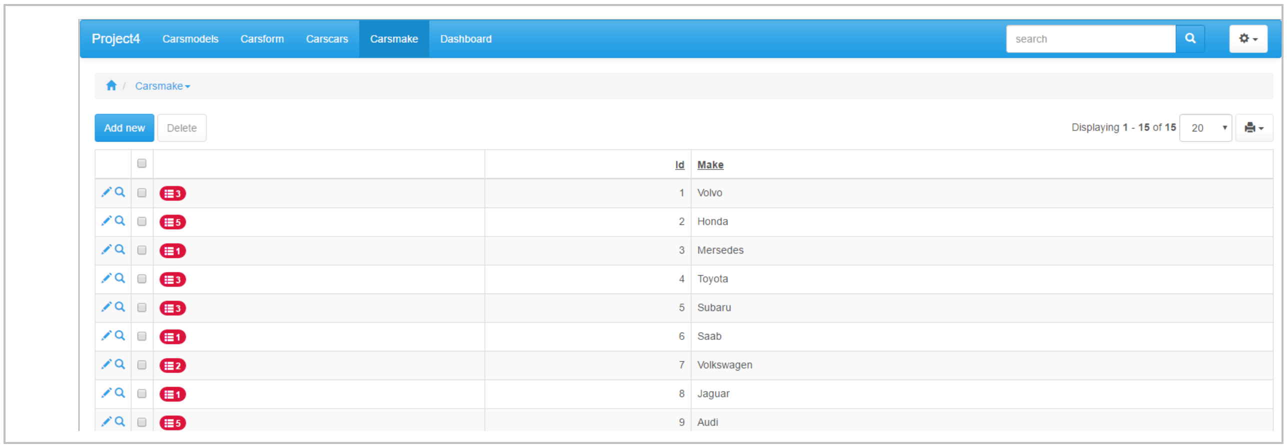
Before custom CSS was applied



Displaying 1 - 15 of 15

	Id	Make
3	1	Volvo
5	2	Honda
1	3	Mersedes
3	4	Toyota
3	5	Subaru
1	6	Saab
2	7	Volkswagen
1	8	Jaguar
5	9	Audi
3	10	BMW
5	11	Hyundai

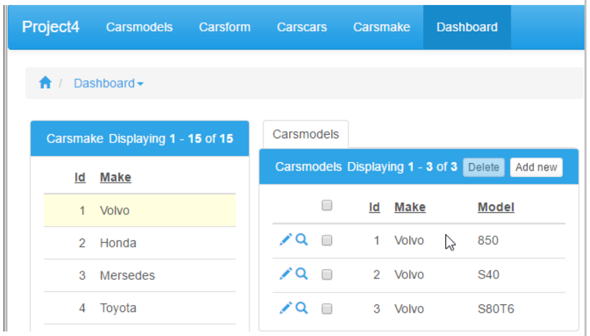
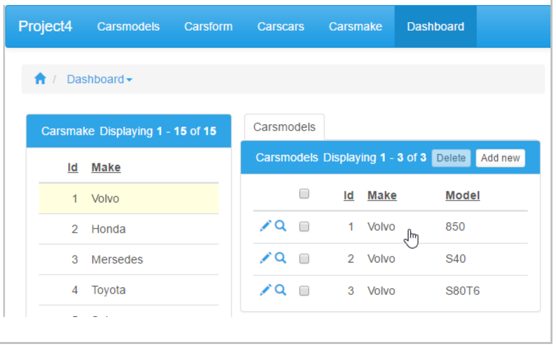
After custom CSS was applied



	Id	Make
3	1	Volvo
5	2	Honda
1	3	Mersedes
3	4	Toyota
3	5	Subaru
1	6	Saab
2	7	Volkswagen
1	8	Jaguar
5	9	Audi

Example 11. How to change cursor to a hand icon while hovering over a dashboard element

```
.bs-dbelement {
     cursor: pointer;
}
```

Before custom CSS was applied	After custom CSS was applied
	

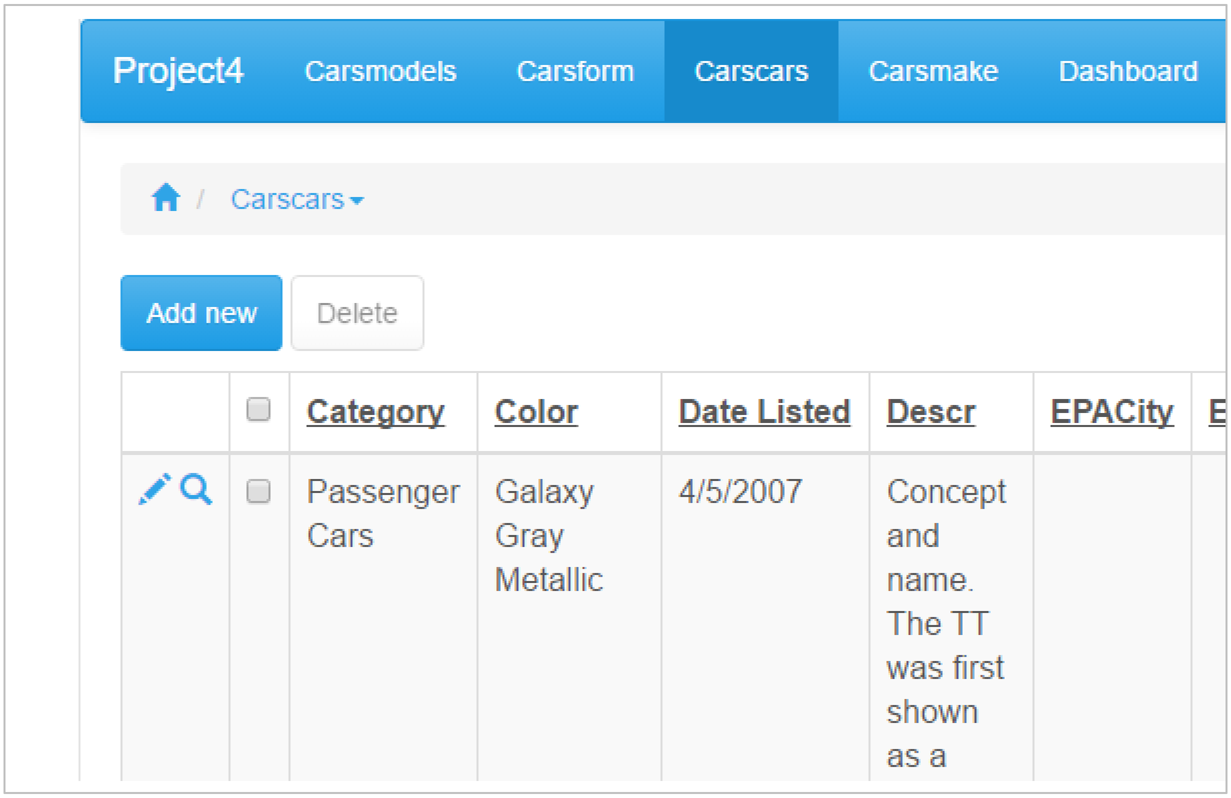
Id	Make
1	Volvo
2	Honda
3	Mersedes
4	Toyota

Id	Make	Model
1	Volvo	850
2	Volvo	S40
3	Volvo	S80T6

Example 12. How to turn on word wrap for all table cells

```
td.bs-gridcell {
white-space: normal;
}
```

After custom CSS was applied

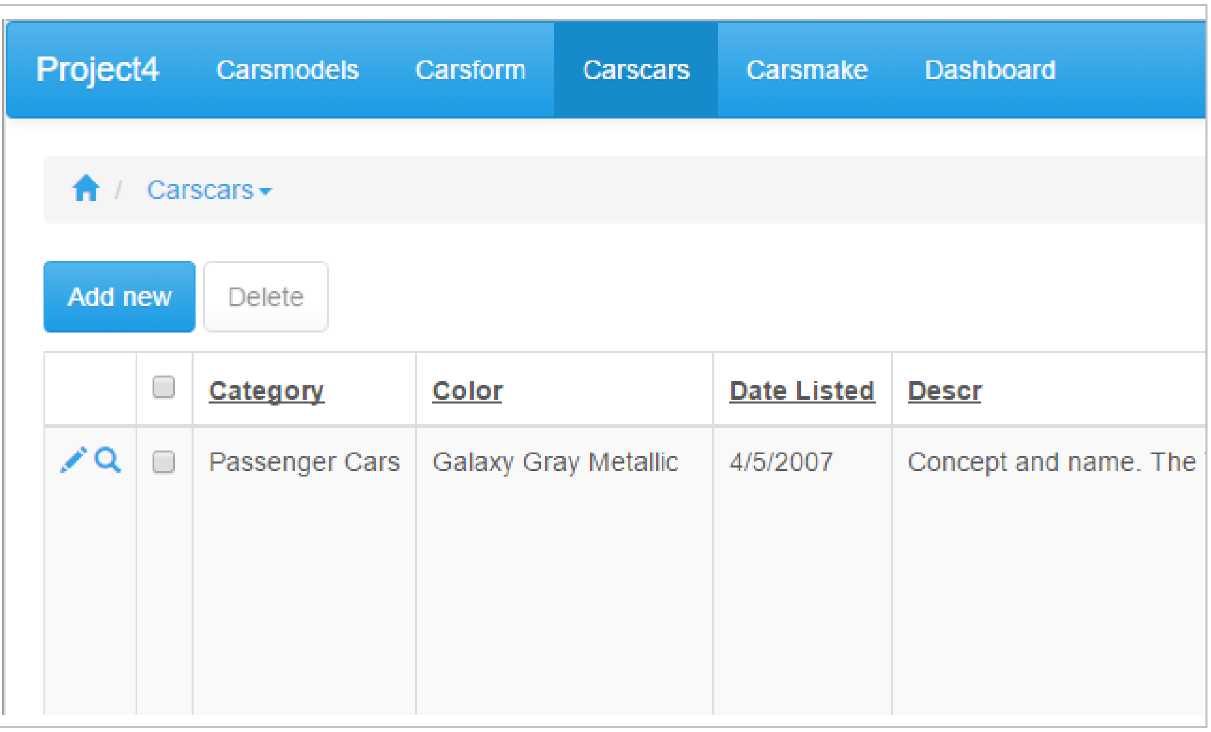


	□	Category	Color	Date Listed	Descr	EPACity	E
	□	Passenger Cars	Galaxy Gray Metallic	4/5/2007	Concept and name. The TT was first shown as a		

Example 13. How to turn off word wrap for all table cells

```
td.bs-gridcell {
white-space: nowrap;
}
```

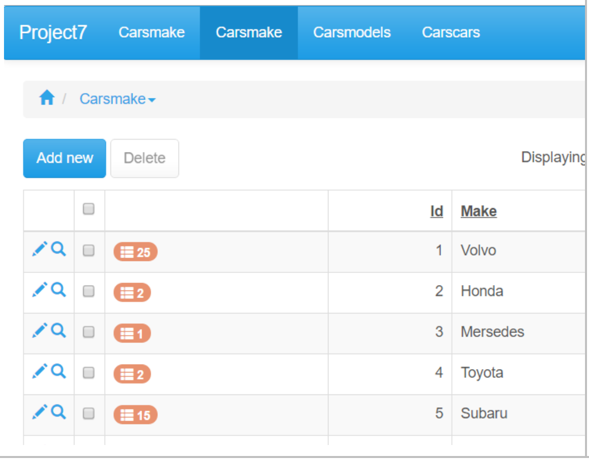
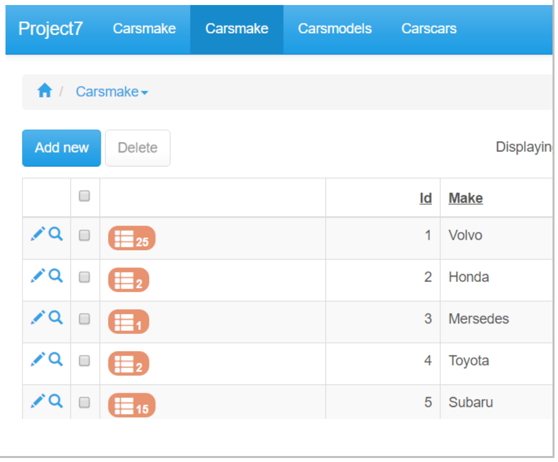
After custom CSS was applied



	□	Category	Color	Date Listed	Descr
	□	Passenger Cars	Galaxy Gray Metallic	4/5/2007	Concept and name. The

Example 14. How to make the 'View details' icon bigger

```
span.glyphicon.glyphicon-th-list {
font-size: 150%;
}
```

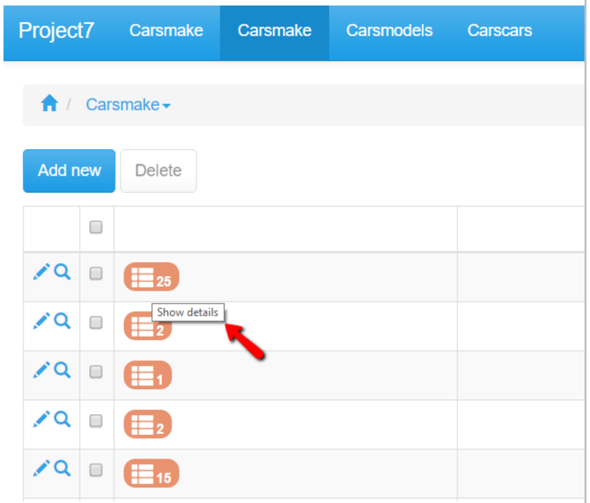
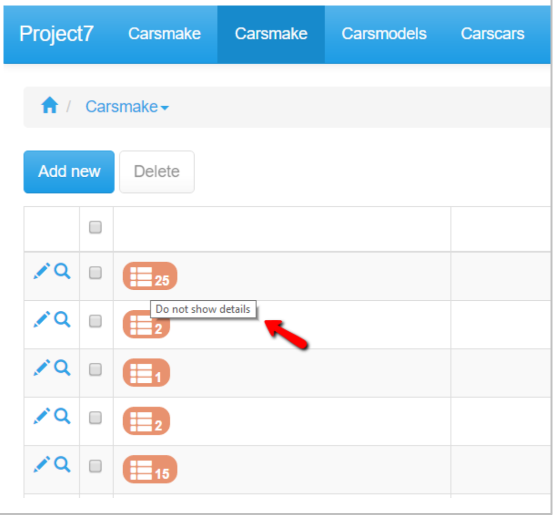
Before custom CSS was applied	After custom CSS was applied
	

	□		Id	Make
	□	25	1	Volvo
	□	2	2	Honda
	□	1	3	Mersedes
	□	2	4	Toyota
	□	15	5	Subaru

Example 15. How to change 'View details' icon mouseover tooltip.

Add the following to the Javascript OnLoad event of the List page.

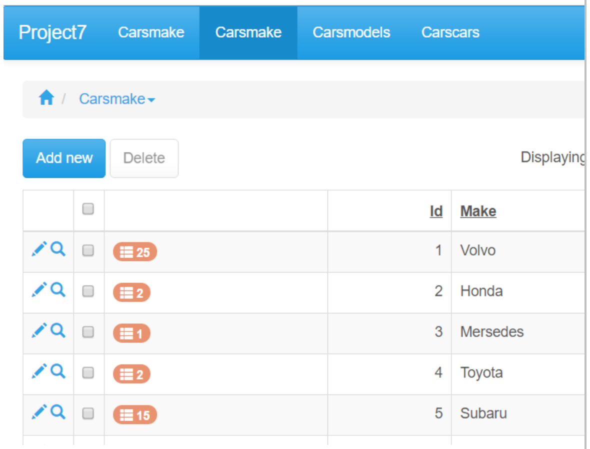
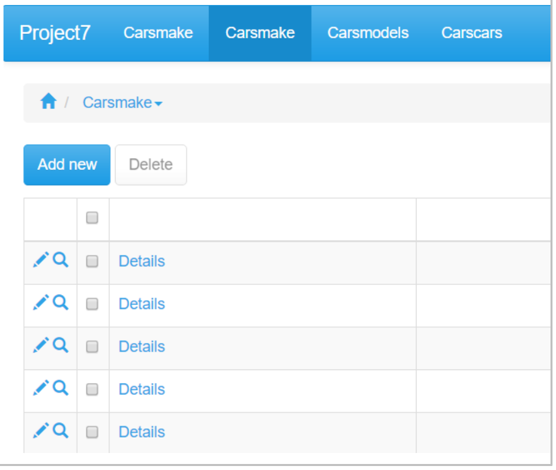
```
$("a[id^=details_]").attr('title', 'Do not show details');
```

Before custom CSS was applied	After custom CSS was applied
	

Example 16. How to change icon to the word "Details".

Add the following to Javascript OnLoad event of the List page.

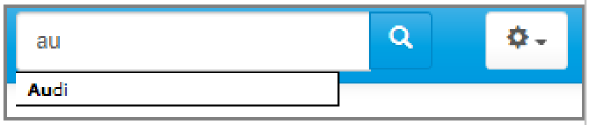
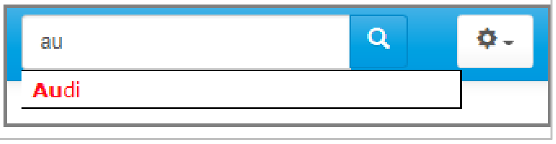
```
$("a[id^=details_]").find("span").attr("class","").text("Details");
```

Before custom CSS was applied	After custom CSS was applied
	

	Id	Make
25	1	Volvo
2	2	Honda
1	3	Mersedes
2	4	Toyota
15	5	Subaru

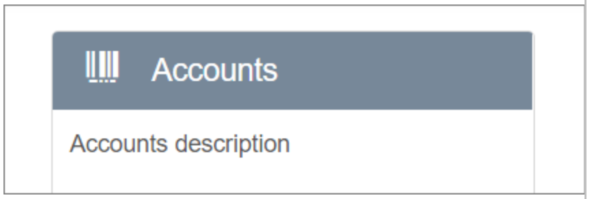
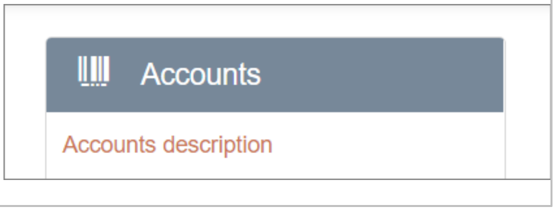
Example 17. How to change the width, font size and color of search suggest window.

```
.suggest_link,  .suggest_link_over  {
  color: red;
  font-size: 14px;
}

.search_suggest {
 width: 300px;
}
```

Before custom CSS was applied	After custom CSS was applied
	

Example 18. How to change the color of the info on the Welcome page panels

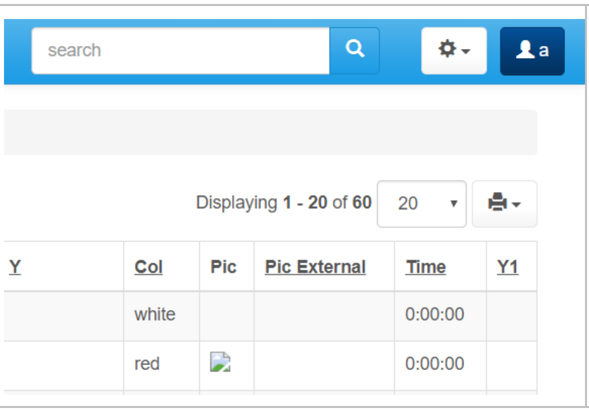
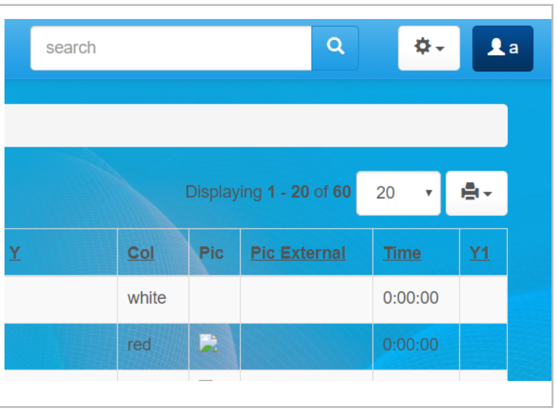
```
bs-welcome-content {
 color: #CC7755;

}
```

Before custom CSS was applied	After custom CSS was applied
	

Example 19. How to set the image as the page background

```
body {
height:100%;
background:transparent url("/images/some_image.jpg") no-repeat center center fixed;
background-size:cover;
}
```

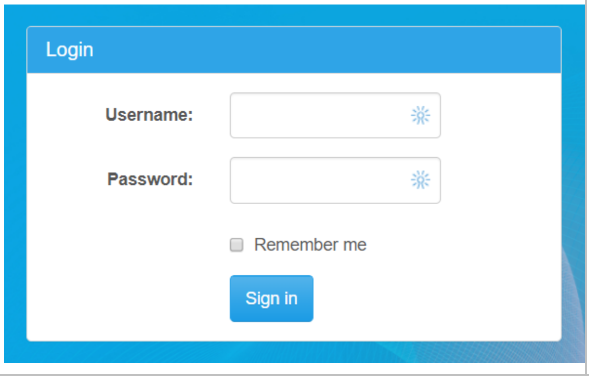
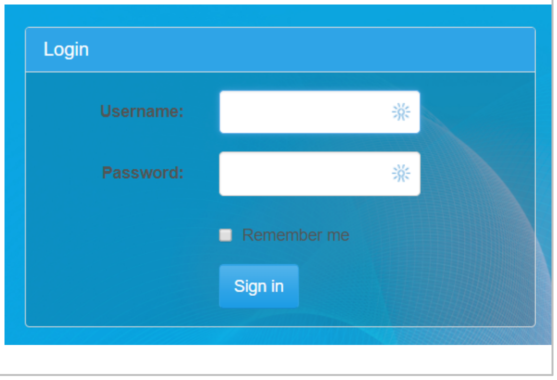
Before custom CSS was applied	After custom CSS was applied

Example 20. How to make the login form semi-transparent

```
.bs-pagepanel {
background: rgba(17,17,17,0.15) !important;
}d: rgba(17,17,17,0.15) !important;
}
```

Sample screenshot of the Login page with the background image set in Example 19:

Before custom CSS was applied	After custom CSS was applied
	

Quick jump	
How to change font in grid on the List page	How to make Bootstrap1 layout grid 100% width
How to hide "required" icon	How to change cursor to a hand icon while hovering over a dashboard element
How to change the color of menu items and project name	How to turn on word wrap for all table cells

Quick jump	
How to change the color of a regular button	How to turn off word wrap for all table cells
How to change the color of a main button	Making the 'View details' icon bigger
How to change the color of the menu bar	How to change 'View details' icon mouseover tooltip
How to change the text color of breadcrumb item	How to change icon to the word "Details"
How to change the text color of breadcrumb container	How to change the width, font size and color of search suggest window
How to change color of dashboard panel headers	How to change the color of the info on the Welcome page panels
How to set the image as the page background	How to make the login form semi-transparent

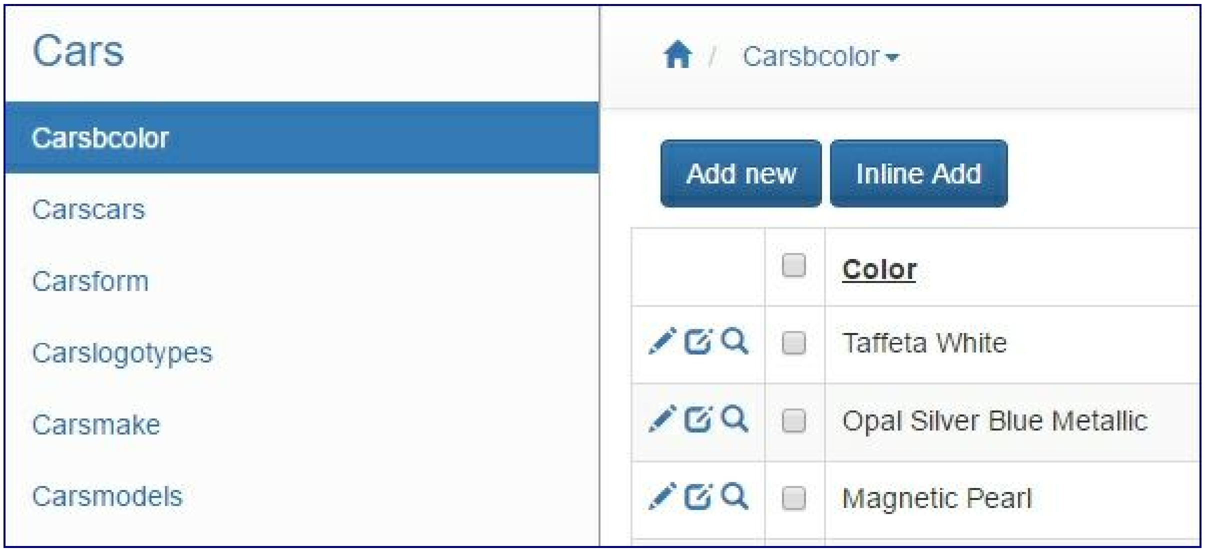
See also:
Building a nice looking login page with custom CSS

2.17.3 Menu builder

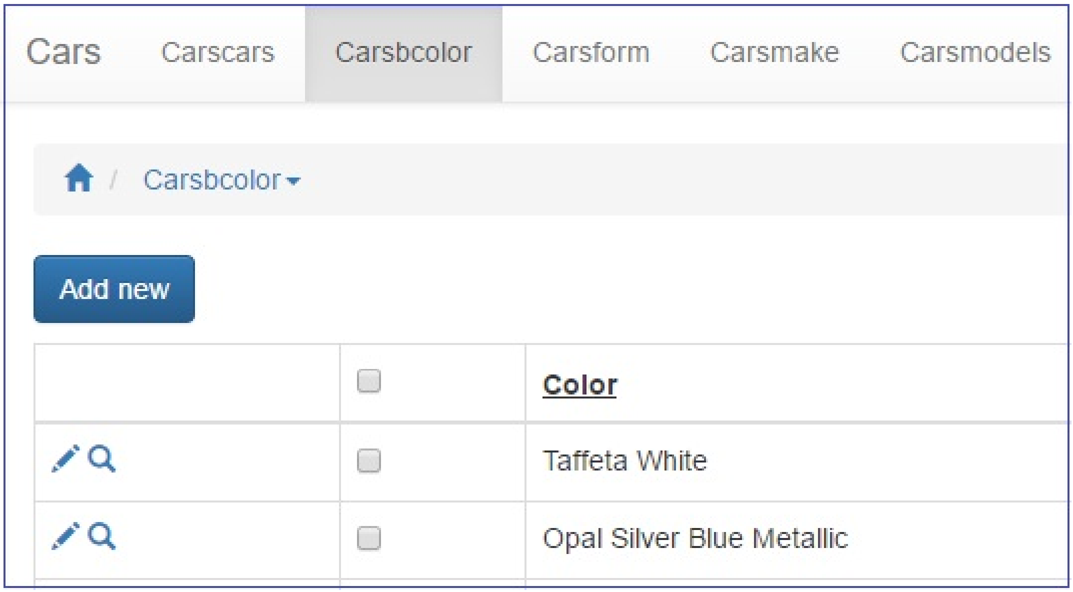
Quick jump
Link attributes
Tree-like menu
Drilldown menu
Welcome page

The Menu Builder lets you organize your tables and views into multi-level cascading menu for quicker navigation. This feature is particularly useful if you have a large number of tables.

Once a project is created, menu consists of tables selected for the current project. Depending on the selected layout menu will appear horizontally or vertically. On this stage menu is single-level. More info on selecting layout.

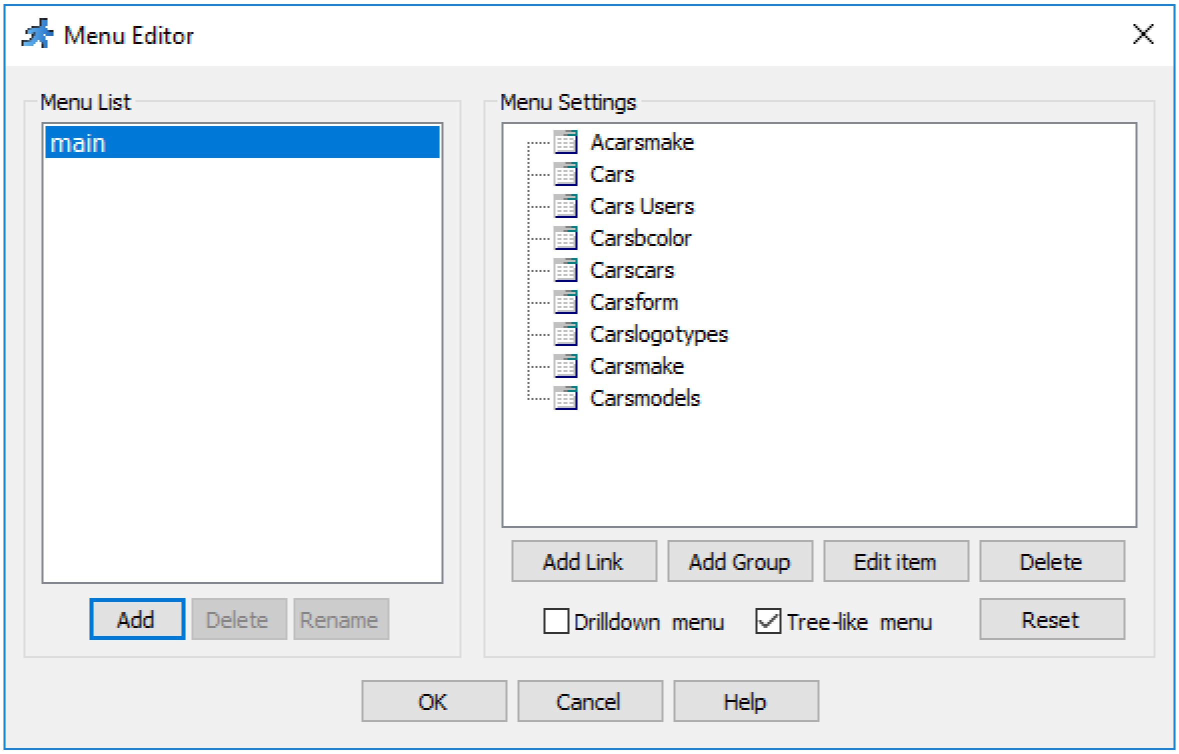


	□	Color
	□	Taffeta White
	□	Opal Silver Blue Metallic
	□	Magnetic Pearl



	□	Color
	□	Taffeta White
	□	Opal Silver Blue Metallic

To open Menu builder, go to the **Tables** page and click Menu Builder.



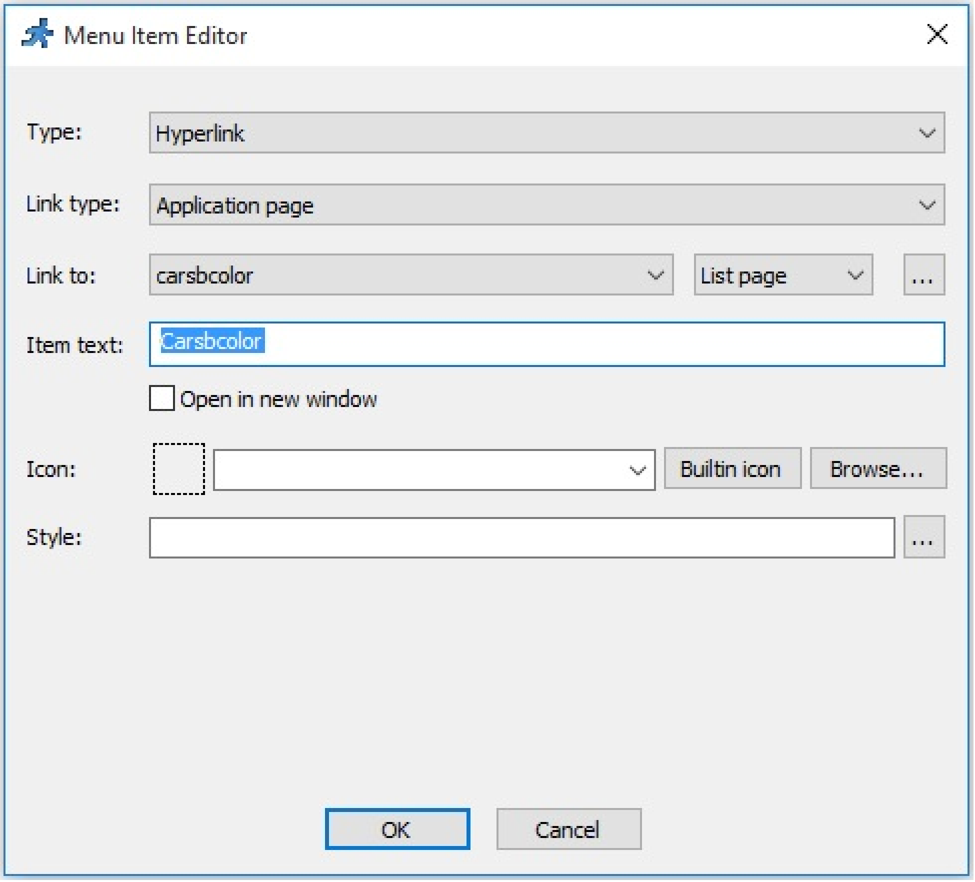
☐ Drilldown menu ☑ Tree-like menu

You can change already created menu entries by double-clicking them, using **Edit item** button or selecting **Properties** with right-mouse button. You can add new groups and links. Notice that group can also include the link or serve just as folder for other groups and links.

Note: You can always change the type of any menu entry from link to group and vice versa.

Delete button on the right panel deletes currently selected menu item. **Reset** button resets menu to its default state - as if project and menu was just created.

Link attributes



Menu Item Editor

Type: Hyperlink

Link type: Application page

Link to: carsbcolor | List page

Item text: Carsbcolor

□ Open in new window

Icon:

Style:

While adding or editing a link you are able to specify the following link attributes:

Type - Select between a hyperlink or a group.

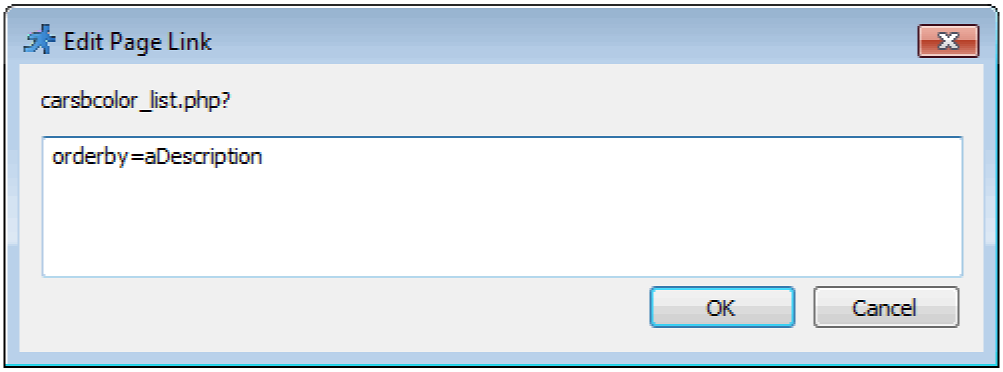
Link type - Select between PHPRunner page or External page.

Link to - If **Link type** is set to External page, then add link to any web page. If **Link type** is set to PHPRunner page, then firstly select a table within your project and then one of pages that are available for selected table. Usually, you can choose between:

- **List** page - displays data from the table;
- **Add** page - allows to add new record to the table;
- **Search** page - provides advanced search for a data in the table;
- **Print** page - prepares user-friendly page with data from the table for printing.

More info about what kind of PHPRunner pages you can create and use.

Click '...' button next to the page type list box to set link parameters. Example of link parameter: *orderby=aDescription.* The link with parameter is *carsbcolor_list.php?orderby=aDescription*. If you want to display the Country Sales Report order by the Sales figures you can set the parameter to orderby=dSales. If you have more complex parameters the best way to proceed would be to copy the parameter string from the application URL and paste it into the parameter dialog.



carsbcolor_list.php?

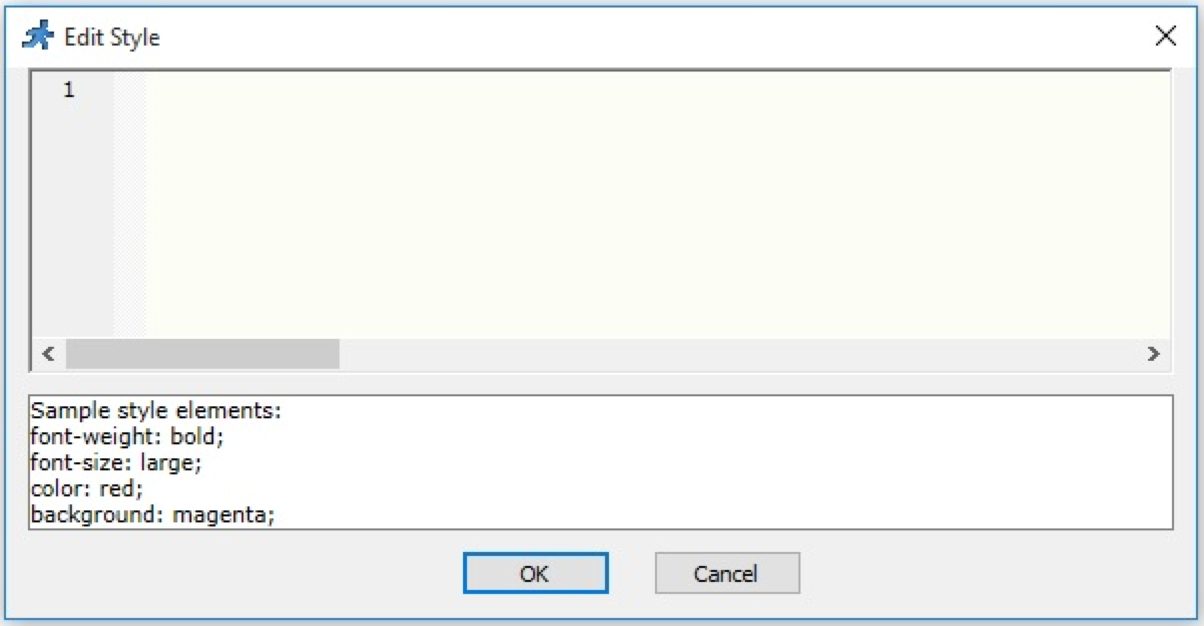
orderby=aDescription

Note that user group permissions are applied to the internal links (PHPRunner pages) and not applied to the external ones.

Link text - Link text will be displayed as menu item title. If multilanguage support is turned on in your project, you will see **Multilanguage** button that allows translating link text into several languages. More info on multiple language support.

Icon - select an image that will displayed before menu item title.

Style - Set menu item text style by adding custom CSS style description. You can add several CSS properties using semicolon between them.



Sample style elements:
font-weight: bold;
font-size: large;
color: red;
background: magenta;

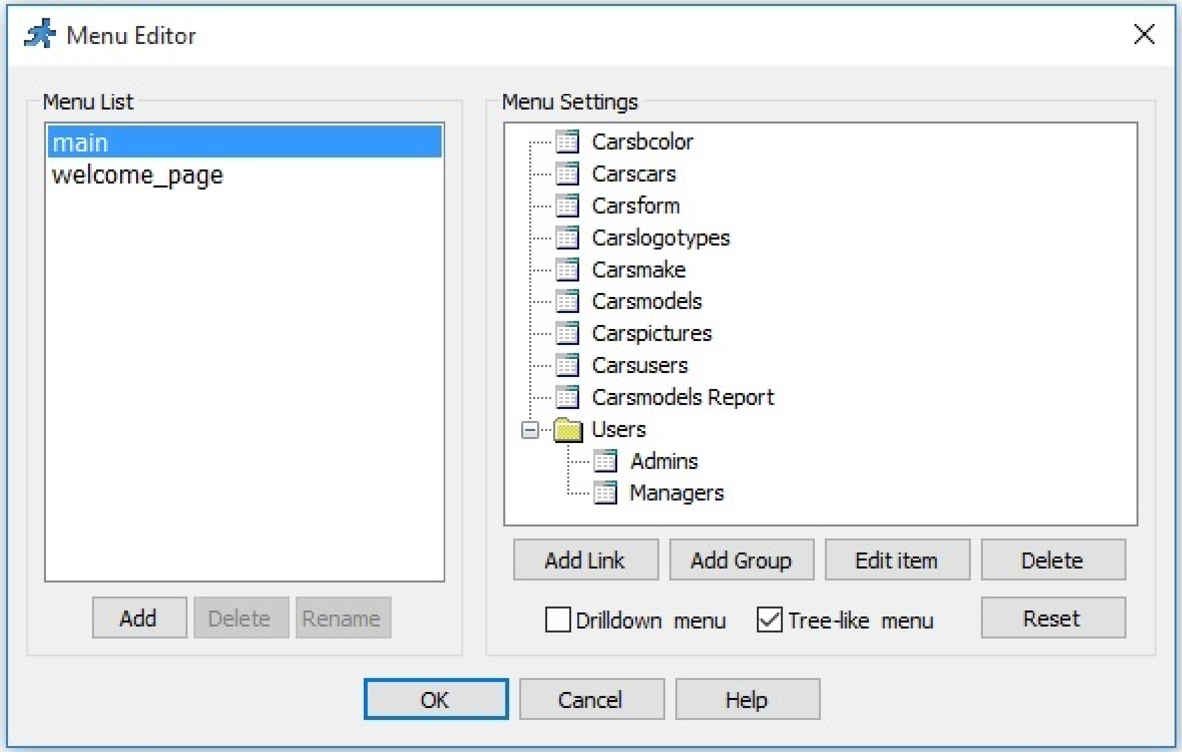
Here are several examples of how you can modify menu item text style:

- font-size:12px; color:red
- font-size:200%; font-weight:bold
- font: 12px italic; border: solid
- color:rgb(255,0,0); background-color:black
- border: dotted red 2 px

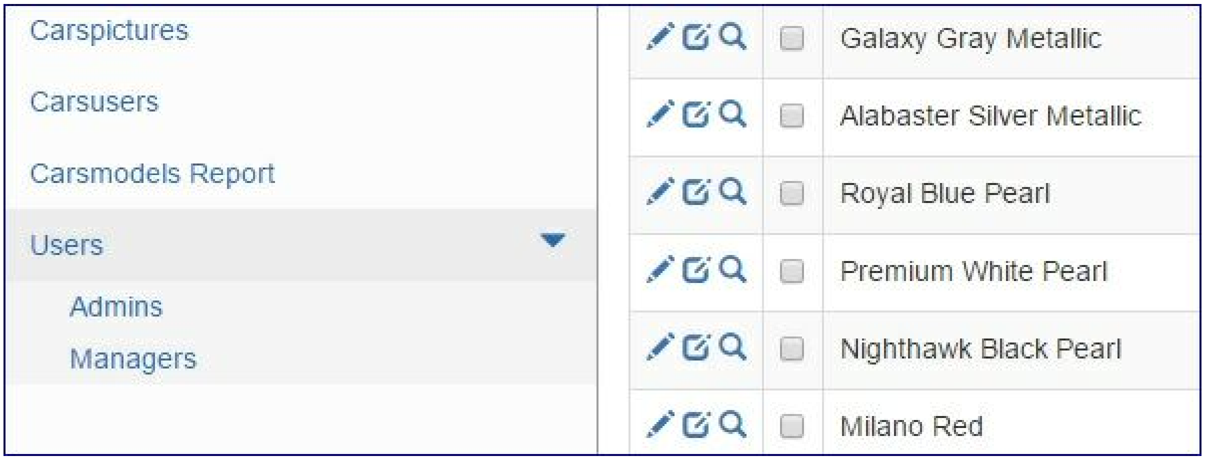
Open in new window option allows opening the links that compose a menu in new browser window.

Tree-like menu

To create a cascading menu, create new group(s) and drag and drop already created menu entries into that groups building cascading structure or create new ones.



For layout with vertical menu the **Tree-like menu** check box allows to display cascade menu that expands/collapses menu folders.



Galaxy Gray Metallic
Alabaster Silver Metallic
Royal Blue Pearl
Premium White Pearl
Nighthawk Black Pearl
Milano Red

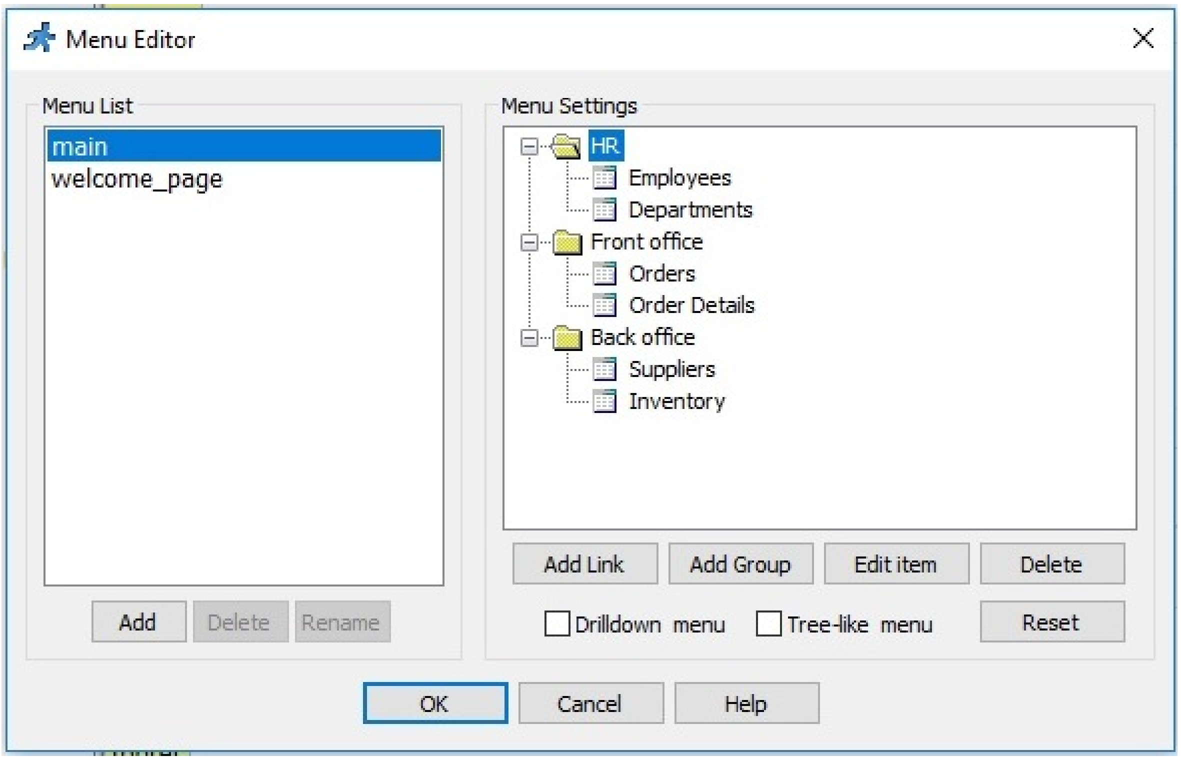
Drilldown menu

The Drilldown menu option might be handy with a multi-level hierarchical menu.

Instead of displaying the whole menu structure on each page, the drilldown menu shows only the current submenu and its children.

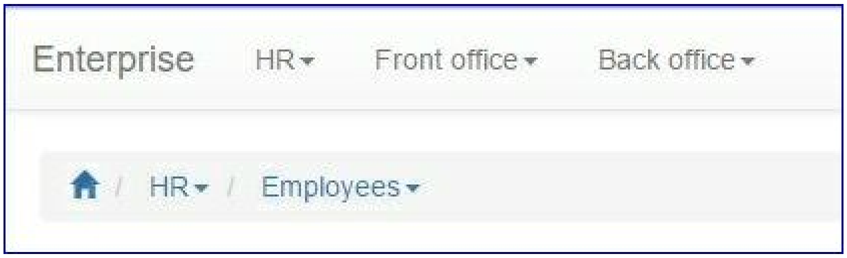
The Breadcrumbs control helps the user to determine the position of the page in the application hierarchy.

For example, with the following menu structure:

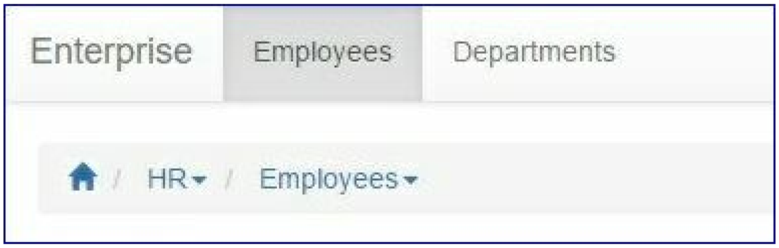


□ Drilldown menu □ Tree-like menu

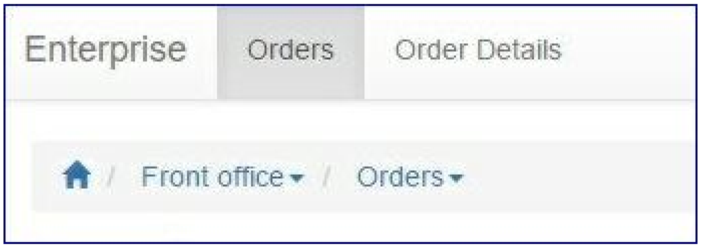
Without the Drilldown menu option you'll see the same menu on each page:



With the Drilldown menu option you'll see this menu on the Employees page:

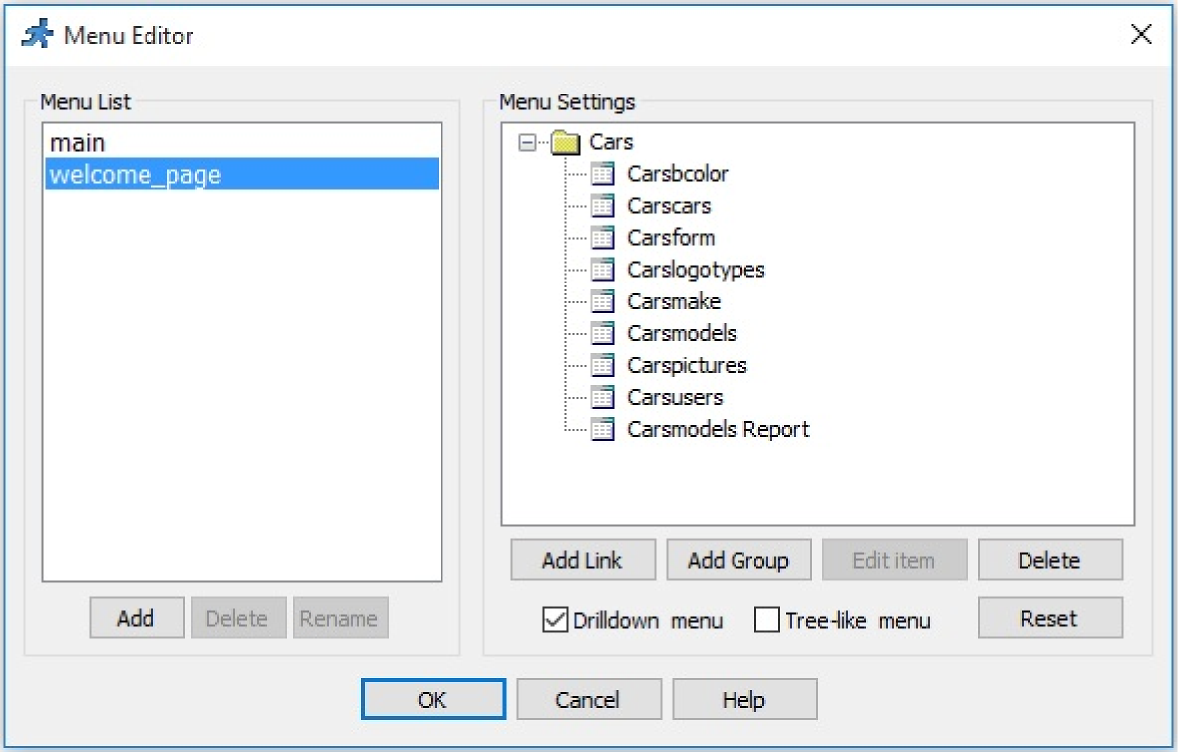


And this one on the Orders page:



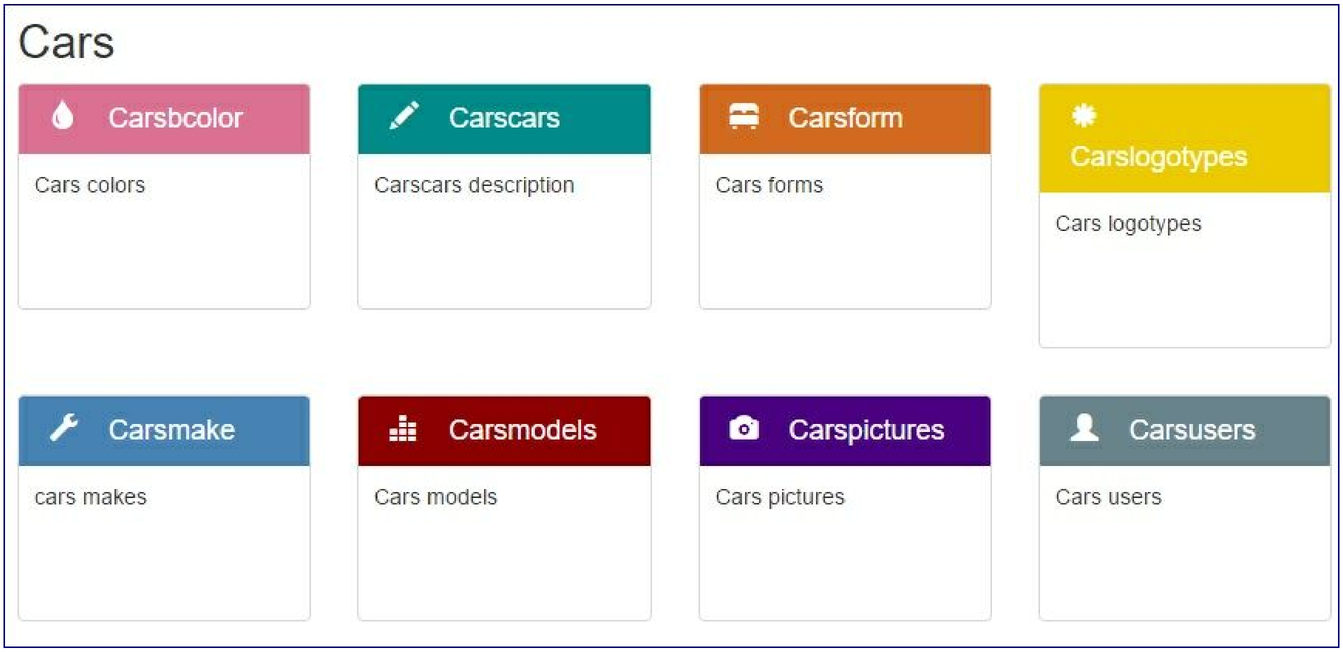
Welcome page

Welcome page is the first page you see in browser. You are able to choose a several most important menu items and display them on start page along with short description. To customize its appearance, choose "welcome_page" in the Menu Builder.

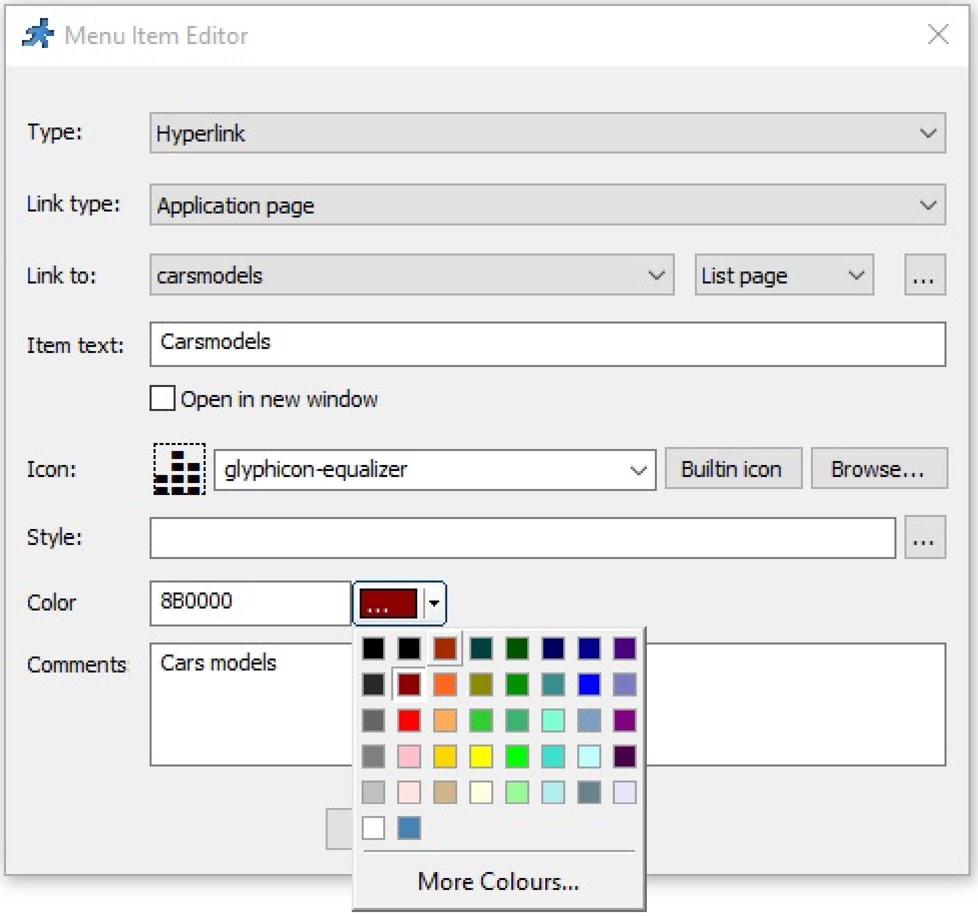


☑ Drilldown menu ☐ Tree-like menu

If you use the bootstrap layout, it will look like this:



Besides the possibilities to add links, groups, create cascading menu, you can alter icons, color, description etc. of your welcome page.



Menu Item Editor

Type: Hyperlink

Link type: Application page

Link to: carsmodels List page

Item text: Carsmodels

□ Open in new window

Icon: glyphicon-equalizer

Style:

Color 8B0000

Comments Cars models

2.18 Event Editor

Quick jump

Revisions

Intellisense

Events are the fragments of PHP code executed before or after record was added, edited, deleted, when new user registers etc. Therefore events allow you to define some actions that will be activated when certain conditions are met.

Here are a few examples of some common events:

- send an email which contains the data from a new record, or shows changes to an existing record;

- save data in another table;

- check record uniqueness;
- show related to current record info.

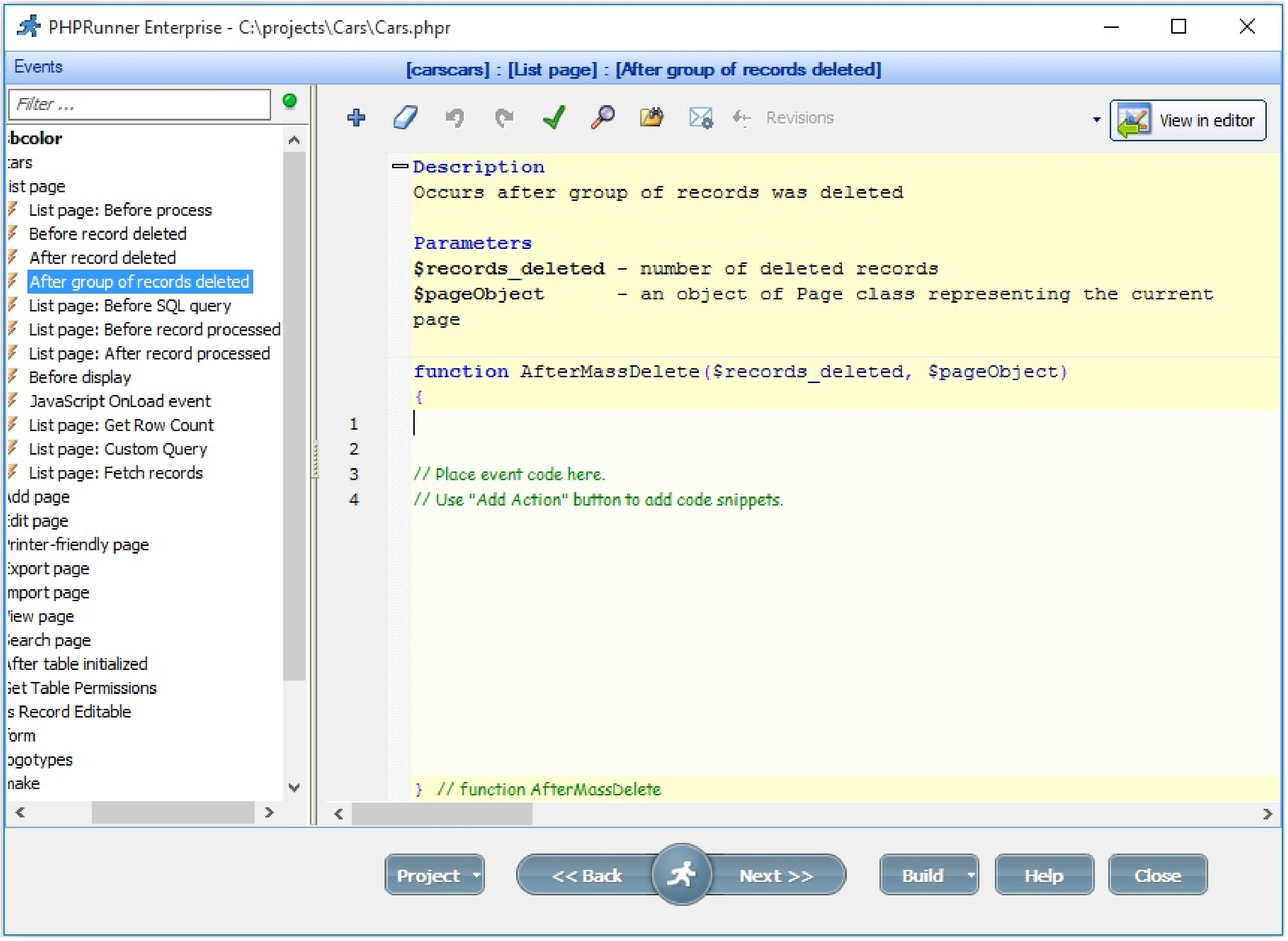
Events can be divided into global and table events. Global events are not specific to any table or view. Table events belong to a table or a view.

Add an event

1. Go to the **Events** page.

2. Select an event from the tree in the left pane. To show events with the actual code only, click the ● button. To filter events by the event name, use text field above the events tree.

After you selected the event, on the right pane you will see definition of the function that implements selected event and description of the function parameters. To hide the parameters description click ⊟ button.



```
Description
Occurs after group of records was deleted

Parameters
$records_deleted - number of deleted records
$pageObject      - an object of Page class representing the current
page

function AfterMassDelete($records_deleted, $pageObject)
{

// Place event code here.
// Use "Add Action" button to add code snippets.

} // function AfterMassDelete
```

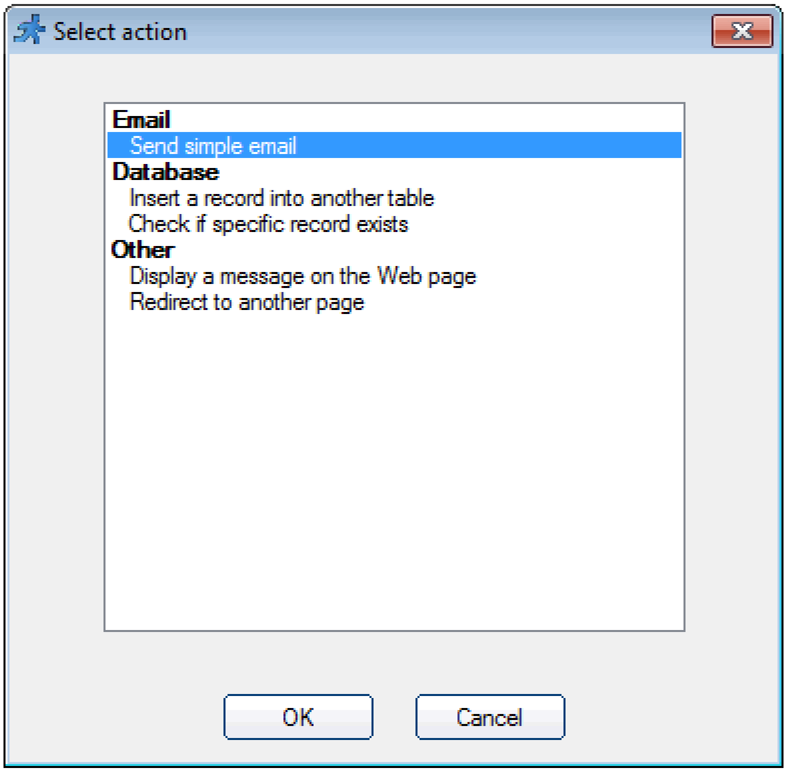
While you are in Event Editor or in Visual Editor press and hold CTRL button and scroll your mouse wheel to increase/decrease font size.

3. Add your own PHP code or click ✚ button to choose one of the predefined actions. More info about predefined actions and sample events with code snippets. Read what common parameters you can use in the events.

Event editor's toolbar description:

Button	Description
✚	Allows to add one of the predefined actions.
⌫	Erases the event code.
↶	Reverses the last operation.
↷	Reverses the last "Undo" operation.
✓	Checks the event code syntax.
🔍	Opens *Search and Replace* dialog that allows to search and replace within an event or within all events.
📂	Opens *Find All* dialog.
Revisions	Displays revision history panel.

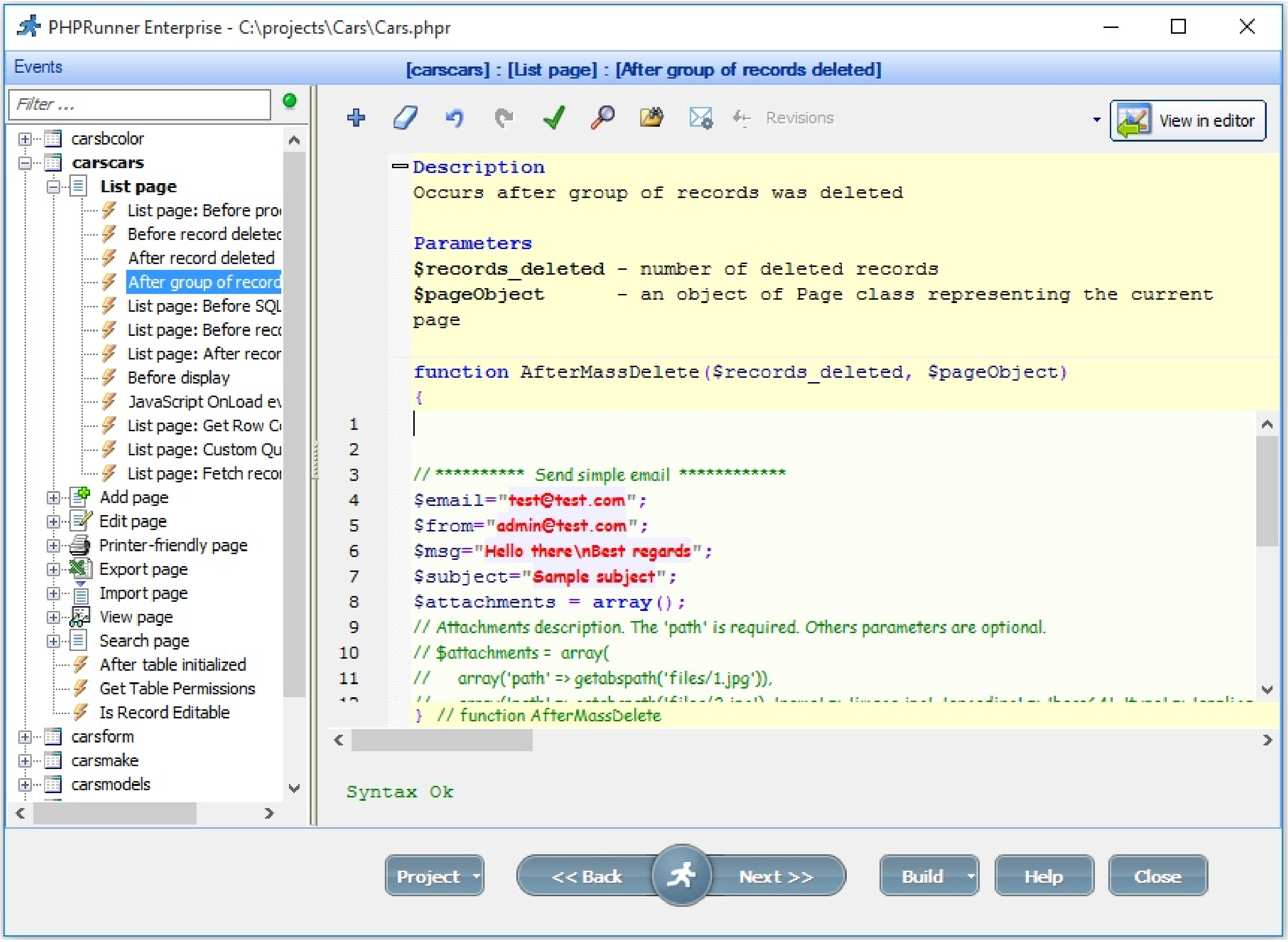
4. Choose one of the predefined actions.


Select action
Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

5. Modify sample event code by putting the correct values instead of the values colored in red.

Example: in the sample code shown below insert:

- email address instead of "test@test.com" and "admin@test.com";
- email message instead of "Hello there\nBest regards";
- email subject instead of "Sample subject".



```
Description
Occurs after group of records was deleted

Parameters
$records_deleted - number of deleted records
$pageObject      - an object of Page class representing the current
page

function AfterMassDelete($records_deleted, $pageObject)
{


// **********  Send simple email ************
$email="test@test.com";
$from="admin@test.com";
$msg="Hello there\nBest regards";
$subject="Sample subject";
$attachments = array();
// Attachments description. The 'path' is required. Others parameters are optional.
// $attachments = array(
//     array('path' => getabspath('files/1.jpg')),
} // function AfterMassDelete
```

```
Syntax Ok
```

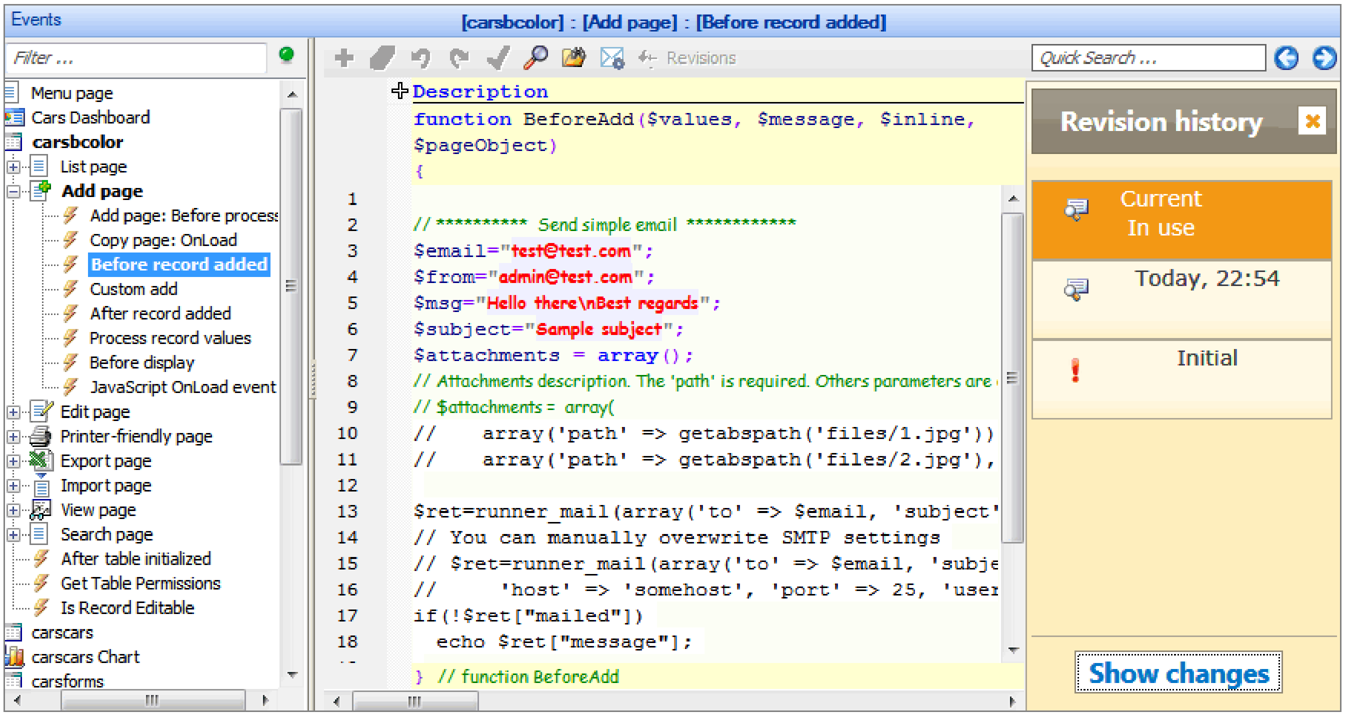
Note: You can add more than one action to the event.

To edit the event, select event from the tree and modify PHP code. To delete the event, select event from the tree and click ![] button.

Revisions

You can track all changes made in Event Editor, review and restore any revision. To see revision history, click **Revisions** button.

Click **Show changes** at the bottom of revision panel to see changes between current and previous revisions. To restore any revision, select that revision and click **Restore revision**.



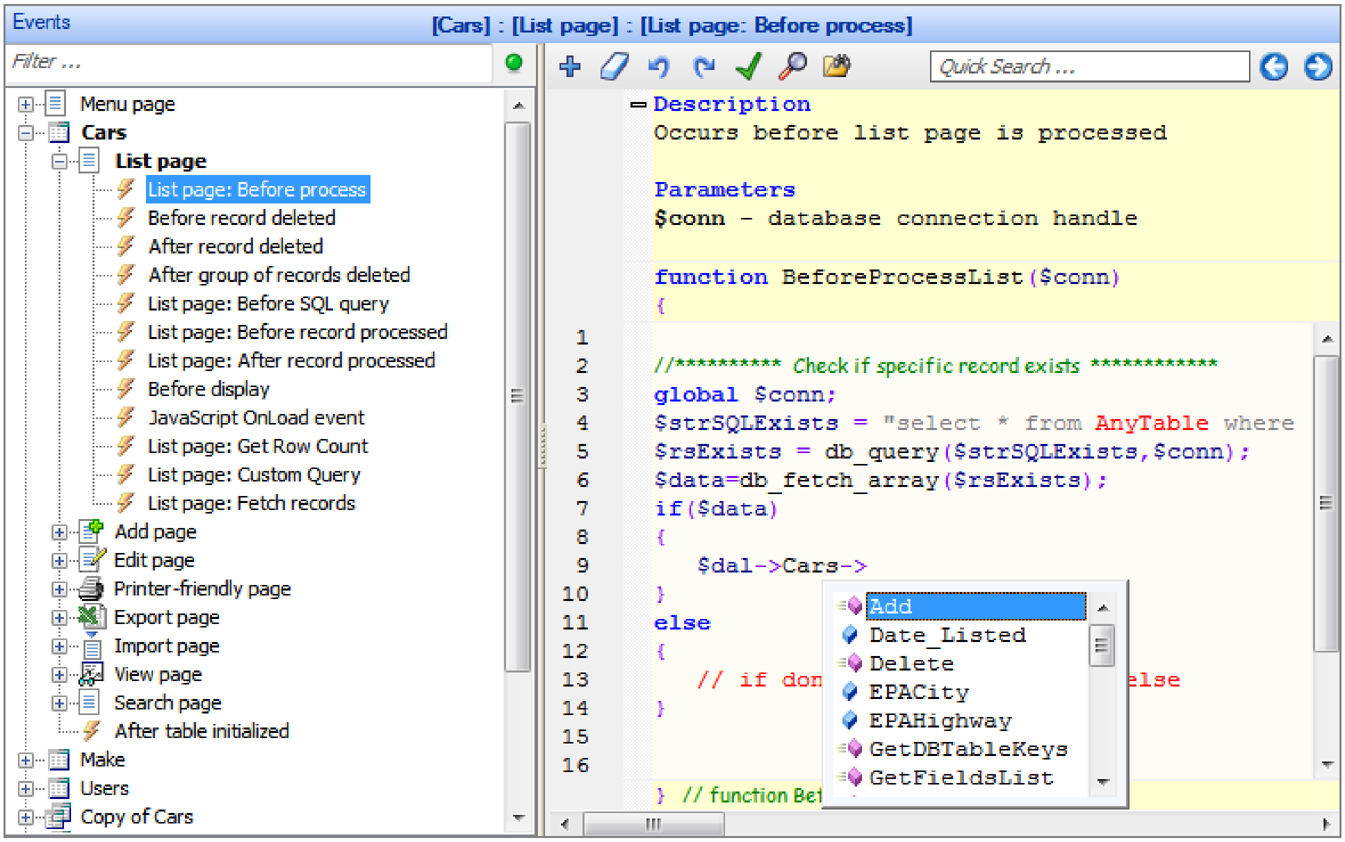
```
Description
function BeforeAdd($values, $message, $inline,
$pageObject)
{

// ********** Send simple email ************
$email="test@test.com";
$from="admin@test.com";
$msg="Hello there\nBest regards";
$subject="Sample subject";
$attachments = array();
// Attachments description. The 'path' is required. Others parameters are
// $attachments = array(
//     array('path' => getabspath('files/1.jpg'))
//     array('path' => getabspath('files/2.jpg'),

$ret=runner_mail(array('to' => $email, 'subject'
// You can manually overwrite SMTP settings
// $ret=runner_mail(array('to' => $email, 'subje
//       'host' => 'somehost', 'port' => 25, 'user
if(!$ret["mailed"])
  echo $ret["message"];
} // function BeforeAdd
```

Intellisense

Intellisense is a convenient way to access the functions' descriptions and variables. The Events Editor recognizes the functions and variables and will pop up the function description or the list of available variables to choose from.

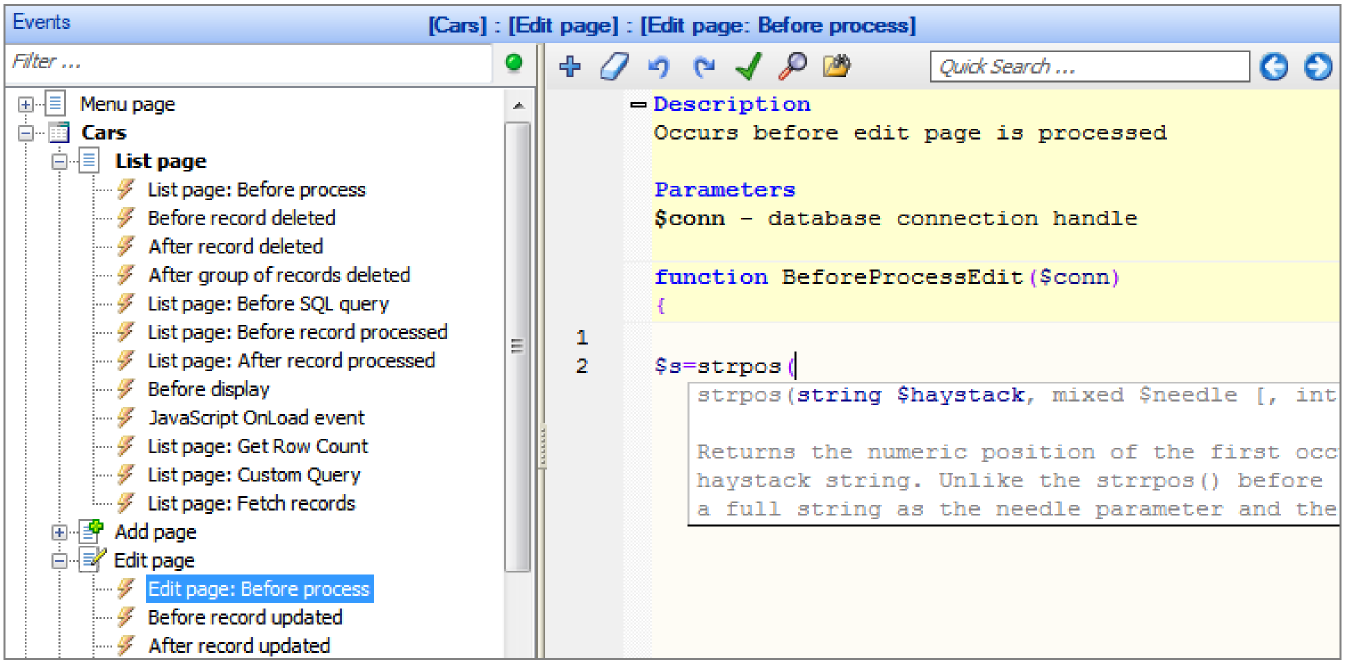
Here are a few examples:

Data Access Layer, database field names



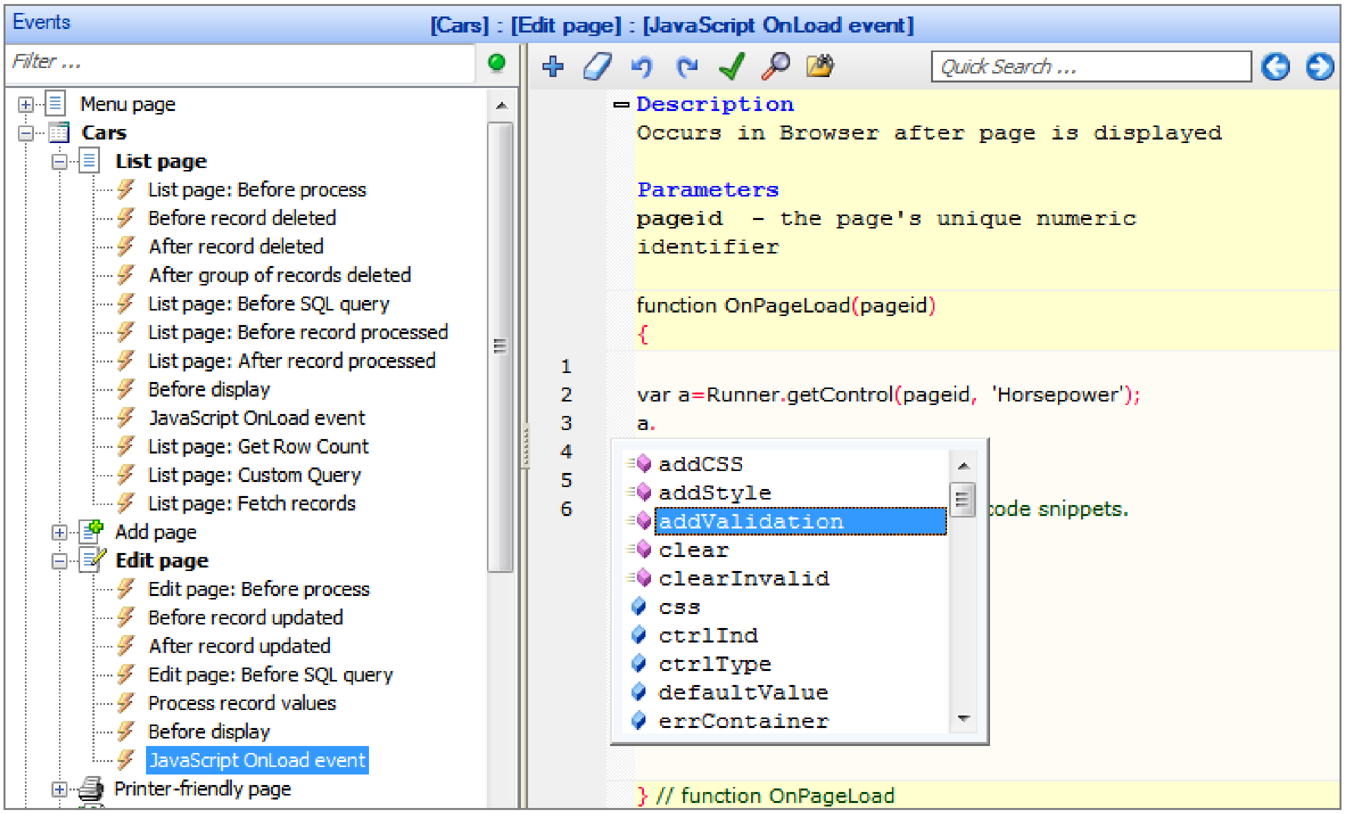
```
Description
Occurs before list page is processed

Parameters
$conn - database connection handle

function BeforeProcessList($conn)
{

//********** Check if specific record exists ************
global $conn;
$strSQLExists = "select * from AnyTable where
$rsExists = db_query($strSQLExists,$conn);
$data=db_fetch_array($rsExists);
if($data)
{
    $dal->Cars->
}
else
{
    // if don                                   else
}


} // function Bef
```

PHP functions



```
Description
Occurs before edit page is processed

Parameters
$conn - database connection handle

function BeforeProcessEdit($conn)
{

$s=strpos(
 strpos(string $haystack, mixed $needle [, int

 Returns the numeric position of the first occ
 haystack string. Unlike the strrpos() before
 a full string as the needle parameter and the
```

Javascript API functions

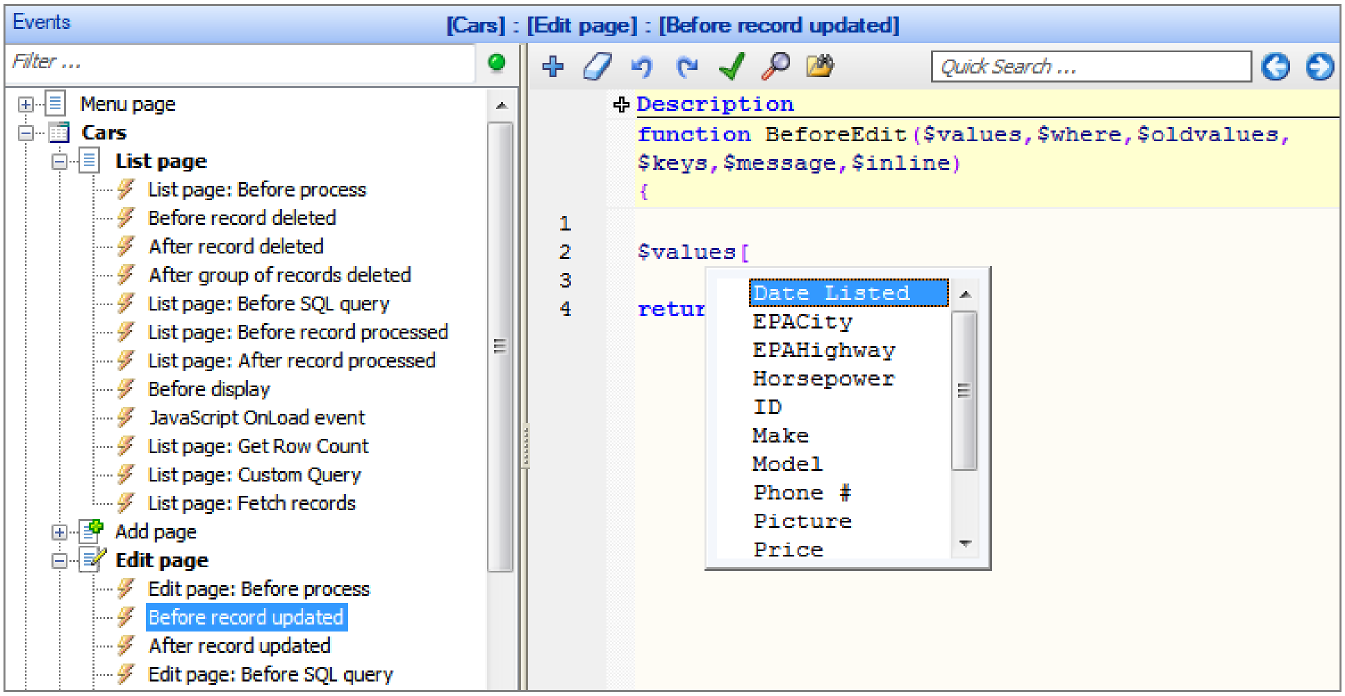


Menu page
Cars
List page
List page: Before process
Before record deleted
After record deleted
After group of records deleted
List page: Before SQL query
List page: Before record processed
List page: After record processed
Before display
JavaScript OnLoad event
List page: Get Row Count
List page: Custom Query
List page: Fetch records
Add page
Edit page
Edit page: Before process
Before record updated
After record updated
Edit page: Before SQL query
Process record values
Before display
JavaScript OnLoad event
Printer-friendly page

```
Description
Occurs in Browser after page is displayed

Parameters
pageid  - the page's unique numeric
identifier

function OnPageLoad(pageid)
{

var a=Runner.getControl(pageid, 'Horsepower');
a.

ode snippets.

} // function OnPageLoad
```

```
addCSS
addStyle
addValidation
clear
clearInvalid
css
ctrlInd
ctrlType
defaultValue
errContainer
```

Field names



Menu page
Cars
List page
List page: Before process
Before record deleted
After record deleted
After group of records deleted
List page: Before SQL query
List page: Before record processed
List page: After record processed
Before display
JavaScript OnLoad event
List page: Get Row Count
List page: Custom Query
List page: Fetch records
Add page
Edit page
Edit page: Before process
Before record updated
After record updated
Edit page: Before SQL query

```
Description
function BeforeEdit($values,$where,$oldvalues,
$keys,$message,$inline)
{

$values[

retur
```

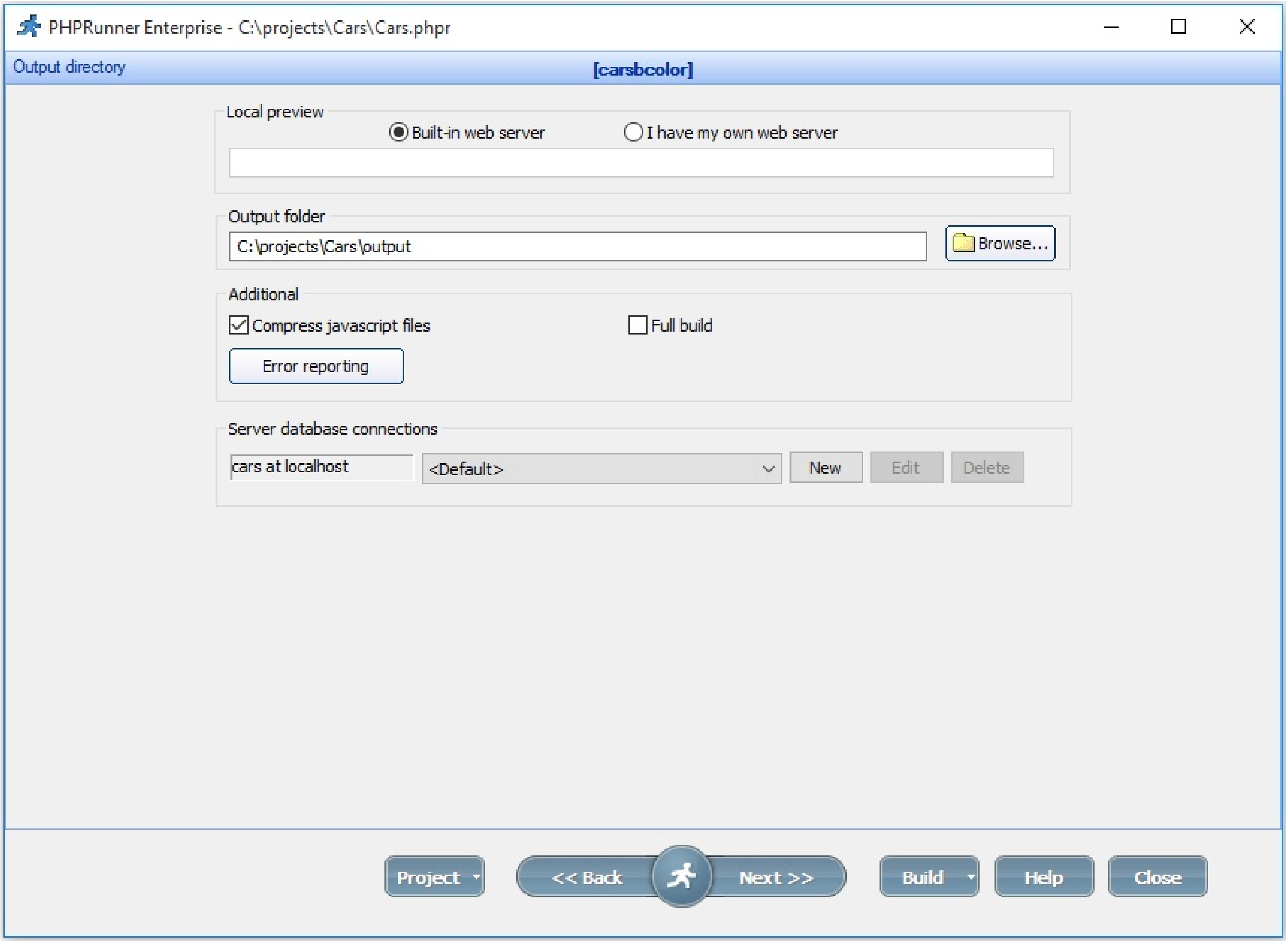
```
Date Listed
EPACity
EPAHighway
Horsepower
ID
Make
Model
Phone #
Picture
Price
```

2.19 Output directory settings

On the **Output directory** screen you can choose output directory where all generated files will be saved.

Preview

PHPRunner comes with built-in web server (Apache). In most cases you can leave all settings on this page "as is", click **Build** and **View in browser** on the next screen. Built-in web server won't interfere with your existing web server if you have one.



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Output directory [carsbcolor]

Local preview

● Built-in web server ○ I have my own web server

Output folder

C:\projects\Cars\output

Additional

☑ Compress javascript files □ Full build

Server database connections

cars at localhost <Default>

In some cases you may want to view generated application using your own web server. In this case switch to **I have my own web server** option and enter URL manually. You should change output directory as well to one of web server subdirectories (i.e. *C:\xampp\htdocs\project1* if you use XAMPP).

Note: If you don't have a web server and want to install one, check How to install local server.

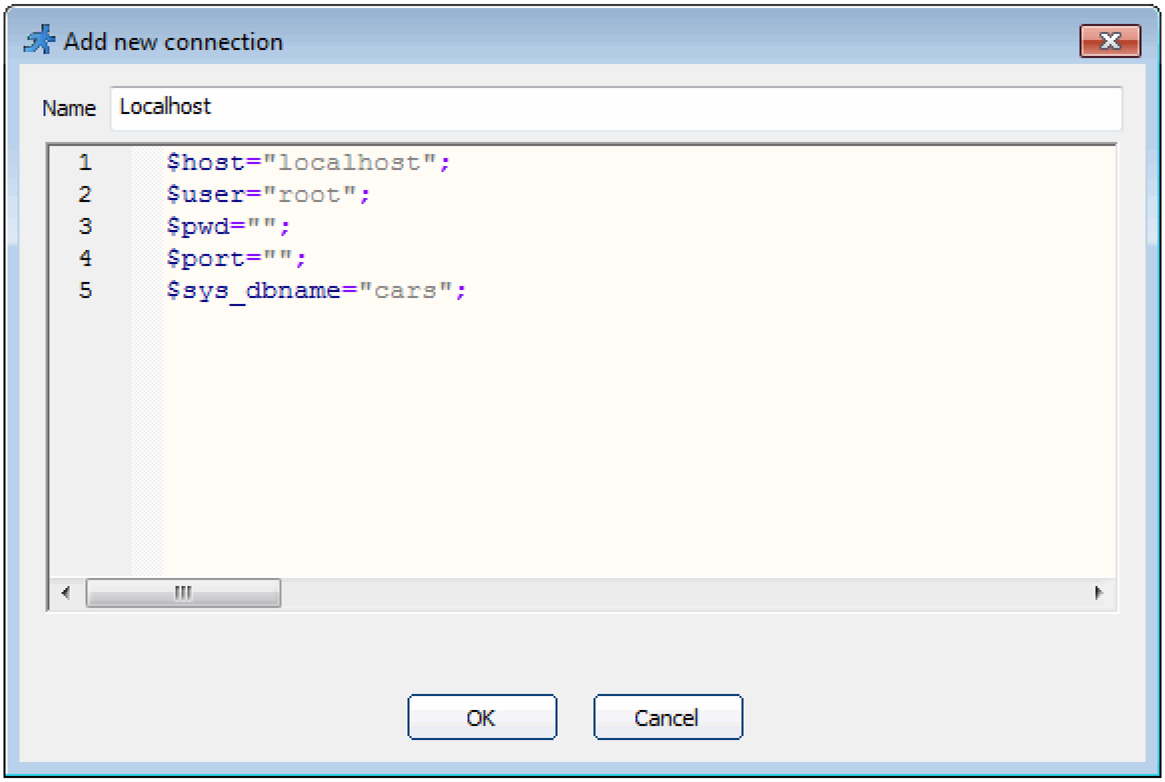
Select the **Full build** check box to perform full build of the project. Otherwise, partial (faster) build will be performed.

Compress javascript files option allows to compress generated Javascript files leading to faster load times. Unless you need to debug Javascript code leave this option on.

Server database connections

You can create several database connections and select required one before building the project. For example, you can use <Default> database connection for the local testing and create new database connection for uploading files to the production server.

To create new database connection, click the **New** button. Then specify connection name and settings. Click **OK**.



Name: Localhost

```
$host="localhost";
$user="root";
$pwd="";
$port="";
$sys_dbname="cars";
```

If you are using MySQL locally, database connection settings may be as follows:

```
$host="localhost";
$user="root";
$pwd="";
$port="";
$sys_dbname="cars";
```

When uploading files to the server, you may use the following settings:

```
$host="localhost";
$user="mike375";
$pwd="dcHd*eS2";
```

```
$port="";
$sys_dbname="mike375_cars";
```

How to set local time on built-in web server

To set the time zone on your local machine when using the built-in web server, you need to adjust *timezone* parameter in *C:\Program Files\PHPRunner10.0\EmbeddedServer\php\php.ini.template* file.

1. Open the file *C:\Program Files\PHPRunner10.0\EmbeddedServer\php\php.ini.template*.

2. Find the line

```
date.timezone = 'UTC'
```

and modify the time zone, for example:

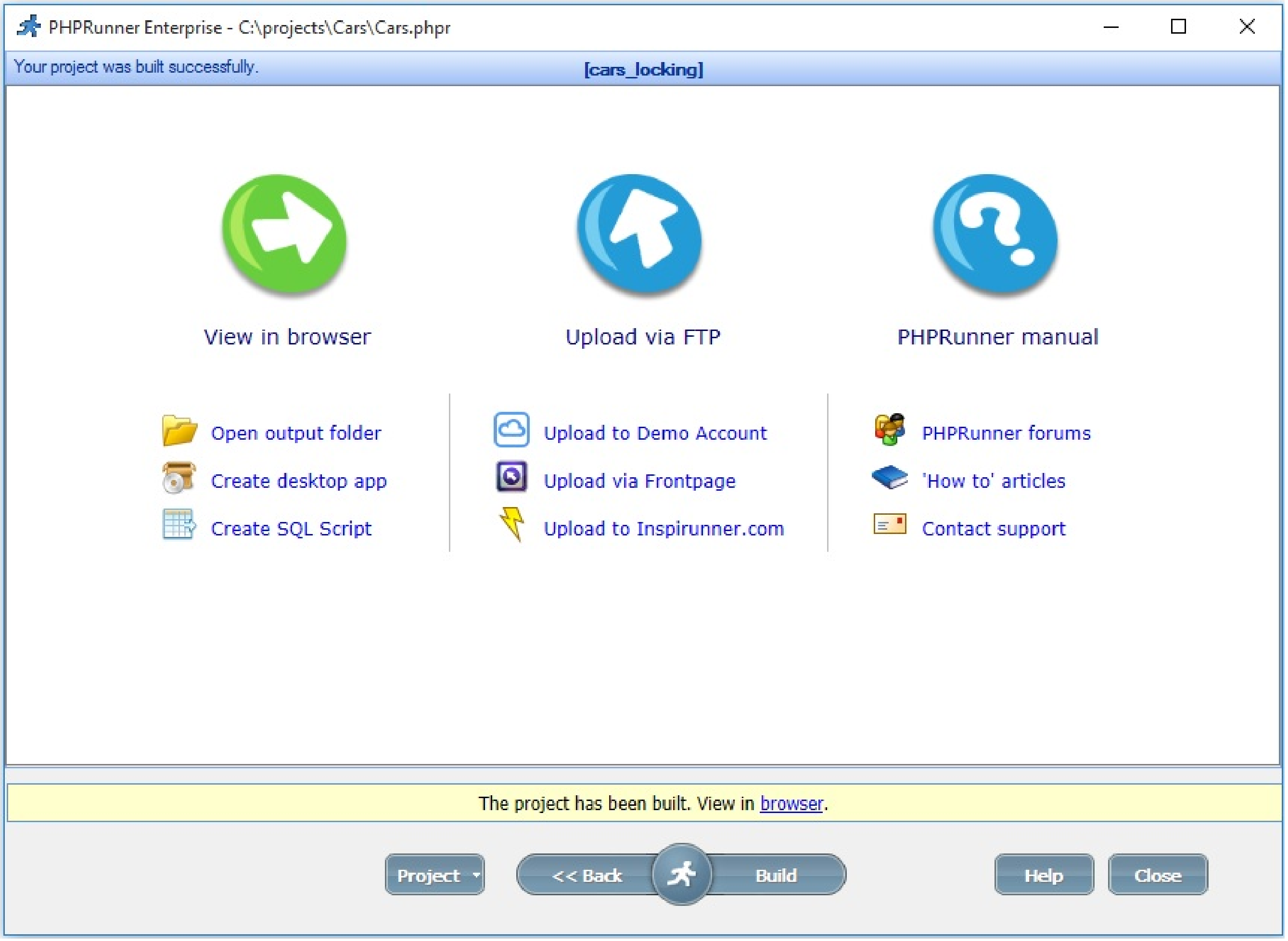
```
date.timezone = 'EDT'
```

3. Save the file and restart PHPRunner.

You can find the time zone abbreviation around the world at http://www.timeanddate.com/library/abbreviations/timezones/.

2.20 After you are done

After you successfully built PHP files you have the following options:



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Your project was built successfully.

[cars_locking]

The project has been built. View in browser.

Testing PHP pages locally

- **View in browser** - run generated application in web browser. Note that this will not work if you enabled the 'Connect using PHP' option while connecting to your database.

If for for some reason built-in web server doesn't start (nothing happens after you click 'View in browser' button) make sure your antivirus or firewall software doesn't block the web server. Turn it off for a minute and try 'View in browser' one more time. Built-in web server uses ports 8085-8090 so you may want to open them in your firewall.

- **Open output folder** - open Windows Explorer to browse the folder with generated PHP files.
- **Create desktop application** - build and package desktop applications.
- **Create SQL script** - to create SQL script for tables/data transfer to another server.

Testing PHP pages on the remote Web server

- **Publish via FTP** - to upload files to the remote Web server you can use built-in FTP client.

- **Publish via FrontPage** - if your website supports FrontPage Extensions use Frontpage publishing feature.
- **Demo Account** - also you can open a free Demo Account with us and publish your project to our demo web server with a single mouse click.
- **Quick upload to InspiRunner.com** - if you have account at InspiRunner.com, use this option to publish your project.

If you prefer to use third-party tools to upload generated applications check the following topics:

- Uploading generated application using third-party FTP client
- Uploading generated application using Frontpage

If you having problems using PHPRunner or like to learn some advanced techniques you can check PHPRunner articles or ask your questions in support forum.

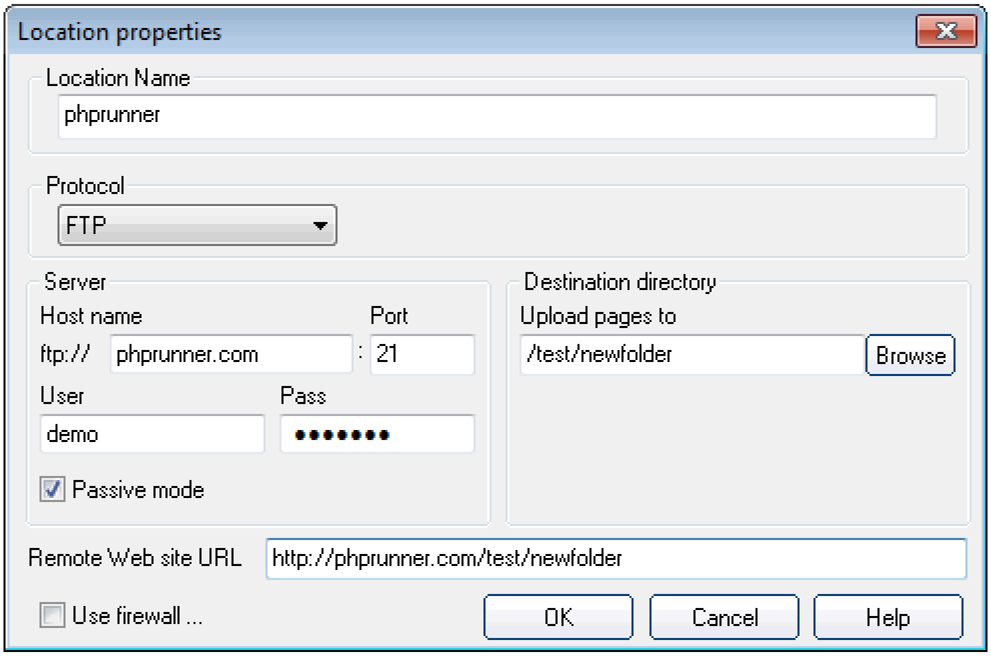
Also you can contact our support team.

2.21 FTP upload

To upload files to FTP server, you need to setup **FTP location properties**.

Enter Location name, Host name, User name and Password to enable **Browse** button. Click **Browse** to choose directory to upload generated files. Choose between FTP, SFTP and FTPS protocols.

If your web hosting provider has a designated folder for databases, you need to move database file to that directory. Connection string will be updated to match new database folder automatically.


Location properties

Location Name: phprunner

Protocol: FTP

Server

Host name: ftp:// phprunner.com

Port: 21

User: demo

Pass: •••••••

☑ Passive mode

Destination directory

Upload pages to: /test/newfolder

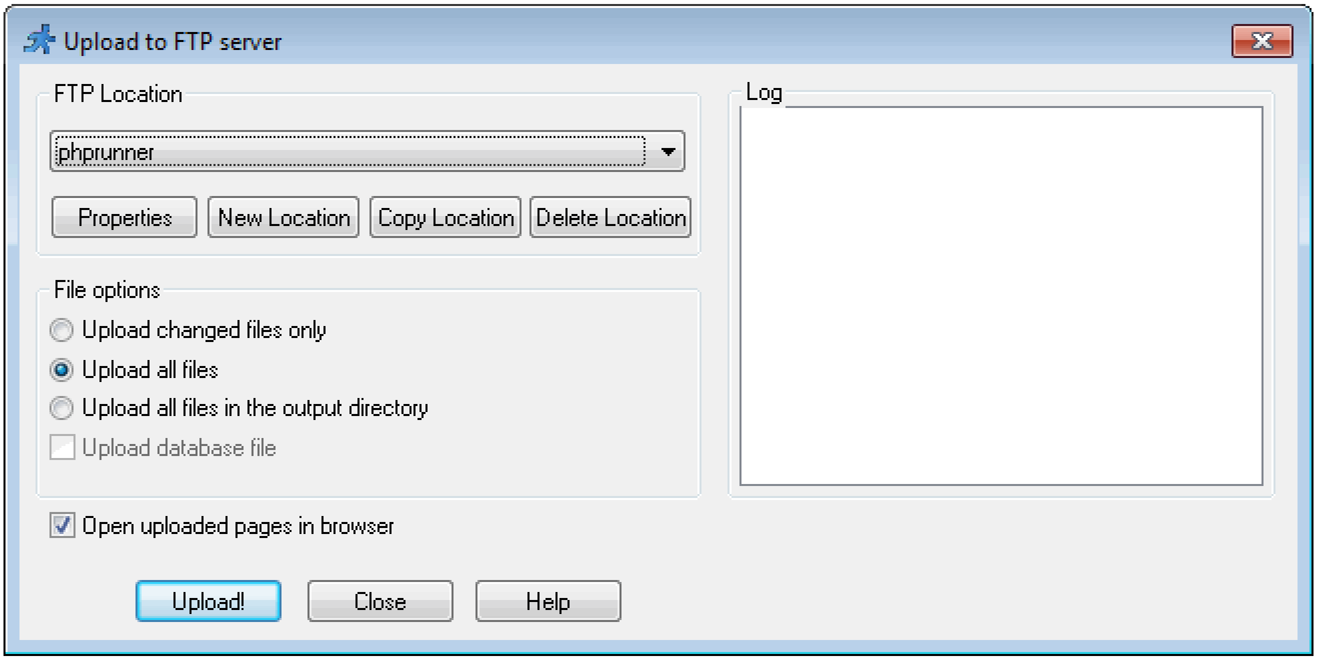
Remote Web site URL: http://phprunner.com/test/newfolder

☐ Use firewall ...

If you have connected to database successfully but can't create folder or upload files you can use **Passive mode**.

Fill in **Remote Web site URL** textbox to open downloaded pages in browser for test purposes.

To upload files choose FTP location and click **Upload** button. You can stop upload at any time by clicking **Stop** button.



FTP Location
phprunner

File options
○ Upload changed files only
● Upload all files
○ Upload all files in the output directory
□ Upload database file

☑ Open uploaded pages in browser

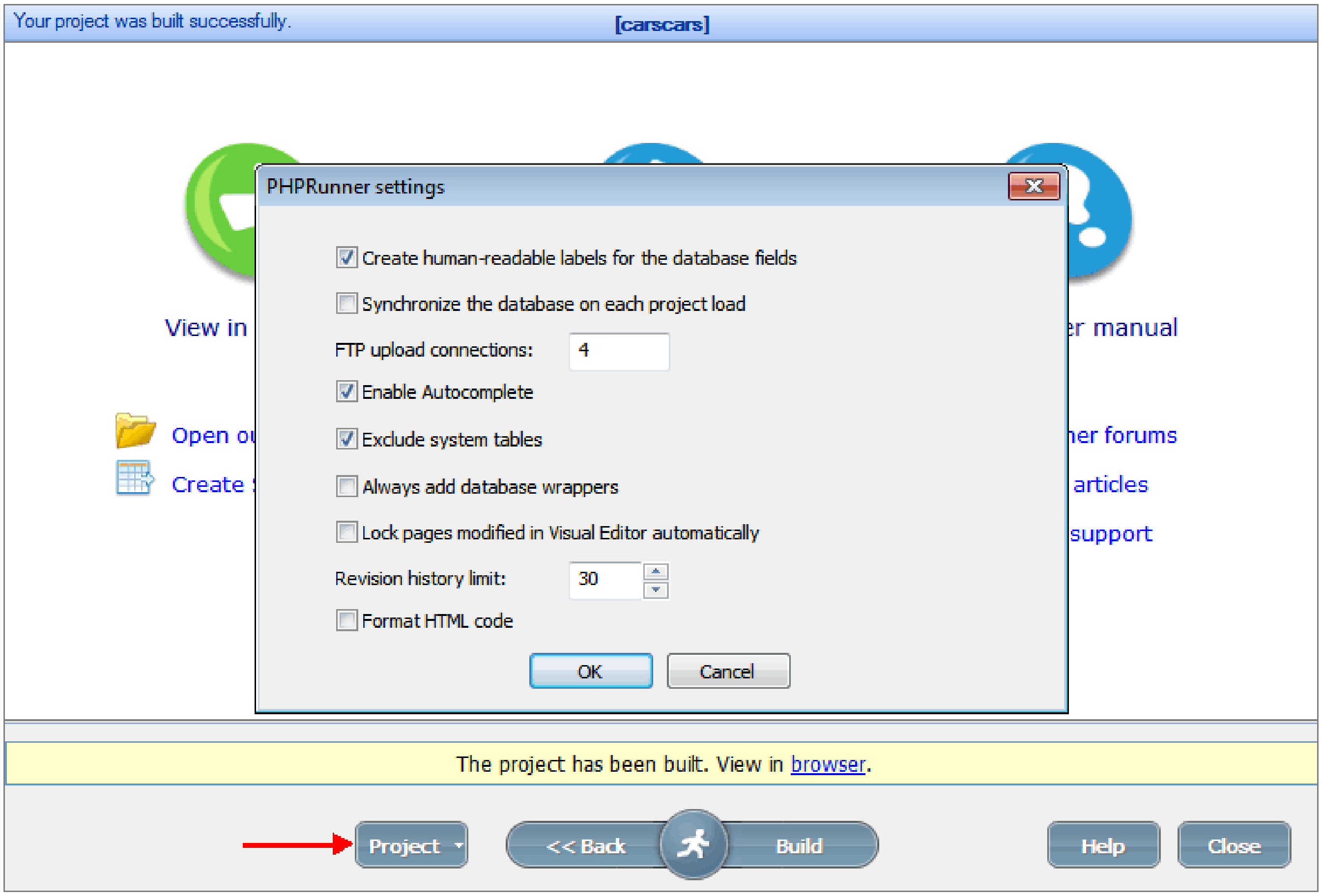
You can choose between the following **File options**:

- **Upload changed files only** - to upload files that were changed since previous upload.

- **Upload all files** - to upload all generated files.

- **Upload all files in the output directory** - to upload all files including files that were created not by PHPRunner.

Note: you can increase the upload speed by changing the number of FTP threads in **Project** -> **Settings**.



PHPRunner settings

☑ Create human-readable labels for the database fields

□ Synchronize the database on each project load

FTP upload connections: 4

☑ Enable Autocomplete

☑ Exclude system tables

□ Always add database wrappers

□ Lock pages modified in Visual Editor automatically

Revision history limit: 30

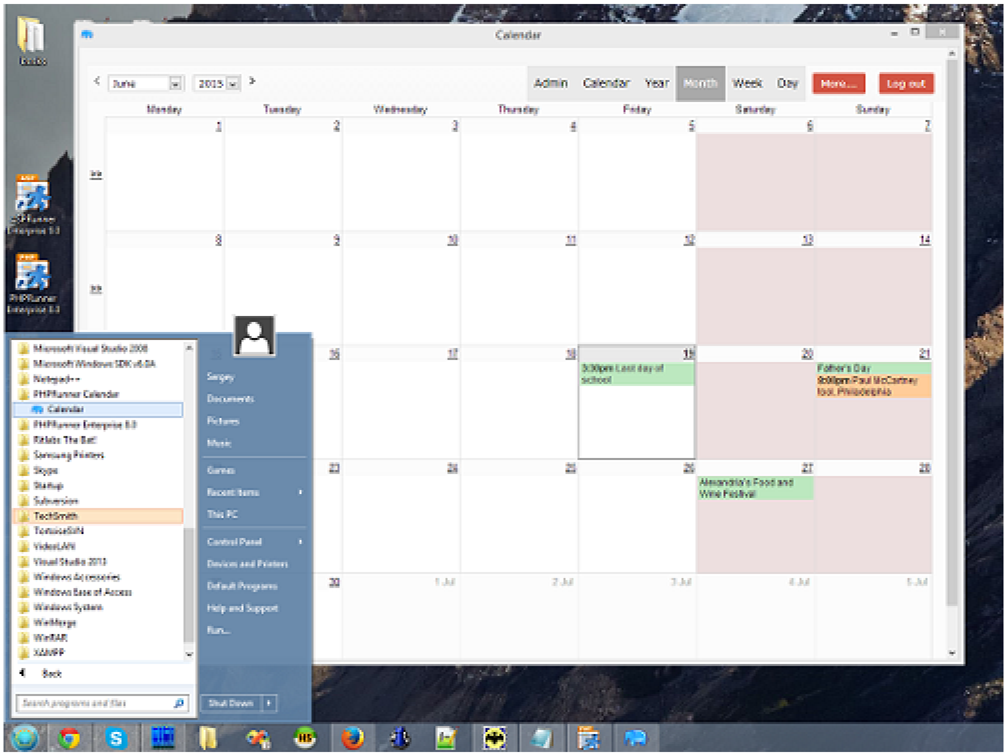
□ Format HTML code

OK Cancel

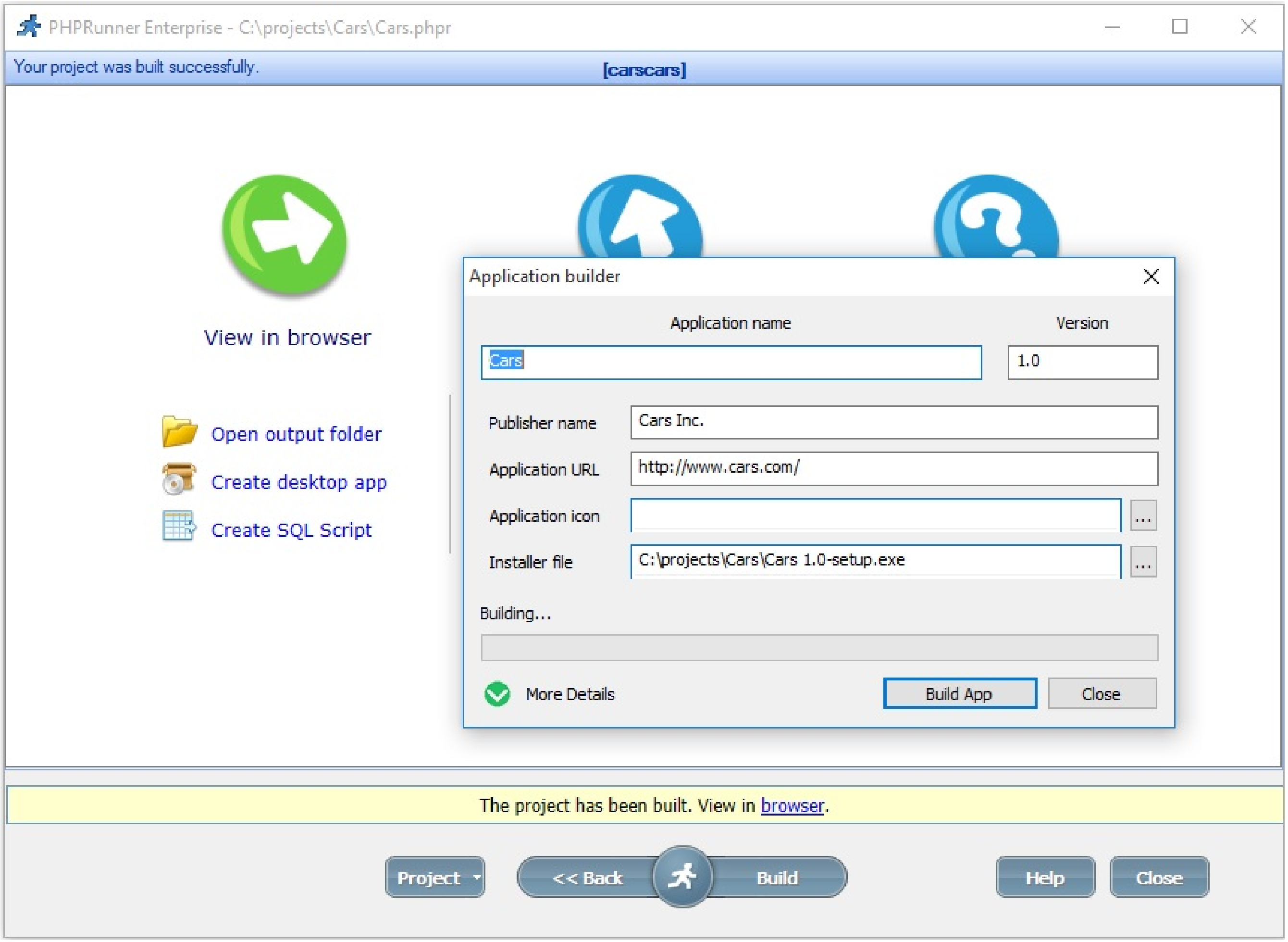
The project has been built. View in browser.

2.22 Desktop Applications

PHPRunner can also build and package desktop applications as well as web applications. Applications like Word or Excel, that can be installed on any Windows machine, require no Internet connection and zero setup, like a screenshot below:



To make a new application, click "Create desktop app" after building the project.



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Your project was built successfully. [carscars]

Application builder

Application name: Cars

Version: 1.0

Publisher name: Cars Inc.

Application URL: http://www.cars.com/

Application icon:

Installer file: C:\projects\Cars\Cars 1.0-setup.exe

Building...

The project has been built. View in browser.

When you build a desktop version of your application 'Publisher name', 'Application URL' and 'Application icon' fields are optional.

If your desktop application shows an error message, check the PHP Desktop knowledge base for additional info. Look up your error message there and take the steps suggested.

How it works

First of all, Desktop apps functionality is built on the top of the open source PHP Desktop project. PHP Desktop packages together the following software components:

- web server (Mongoose)
- PHP
- Chrome browser

When you start PHP Desktop application, it starts the web server first specifying 'www' directory as website root folder. Then it starts Chrome browser pointing to web server's home page. Browser application is modified to hide all menus, navigation buttons and address bar. You just see the application itself.

PHPRunner takes one extra step packaging the whole application into a single installable EXE file. Inno Setup is used for this purpose. The whole packaging process is described with great details in the 'Create Your First Desktop Application With PHP And PHP Desktop' tutorial.

Limitations

- Windows only

PHP Desktop runs on Windows only.

- PHP only

PHP Desktop, apparently, is for PHP only. However, since a standalone IIS Express browser exists we can expect at some point that

- Installer is not signed

If you want to make desktop version of your app available worldwide you definitely need to sign the installer.

- Not an offline version

Desktop application is no different from the web application except it sits on your desktop. If your application uses a remote database, you do need Internet connection to connect to that database.

Some typical use cases of desktop version

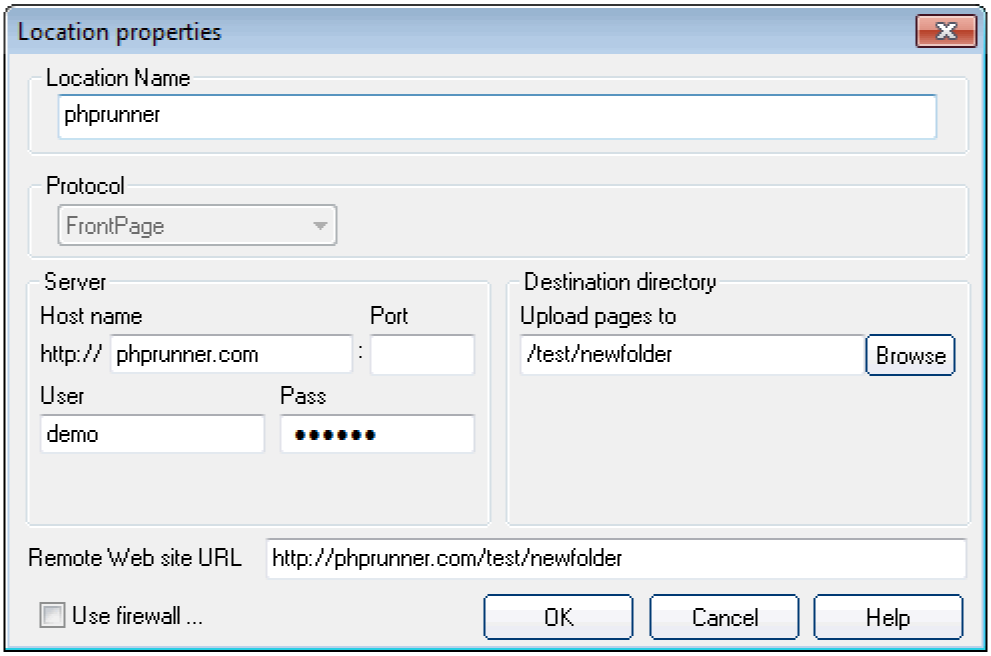
- You need to create a desktop application. You are a PHP developer who needs to create a desktop application. PHPRunner can help you with this. The natural choice is to use SQLite or MS Access database in this case.
- You need to create an application that doesn't use database at all. An example - Euro 2016 desktop application. It connects to football-data.org API, retrieves data and displays it. You don't need a website or a database in this scenario.
- When you need an access to hardware or to file system. Using a desktop app you can access devices connected to end user machine, work with file system, use COM objects etc. You can do pretty much everything the typical desktop application can.
- You can use your desktop application to connect to remote database. This is the less typical scenario but still plausible. Probably you do not have a website or do not want to maintain a website, take case of security etc. All you have is a database like MySQL or Amazon RDS with remote access. In this case desktop application can be useful as well.

2.23 FrontPage Publishing

To upload files with FrontPage Server Extensions, you need to setup **FrontPage location properties**.

Enter Host name, User, Password to enable **Browse** button. Click **Browse** to choose directory to upload generated files.

If your web hosting provider has a designated folder for databases, you need to move database file to that directory. Connection string will be updated to match new database folder automatically.


Location properties

Location Name: phprunner

Protocol: FrontPage

Server
Host name: http:// phprunner.com
Port:
User: demo
Pass: ••••••

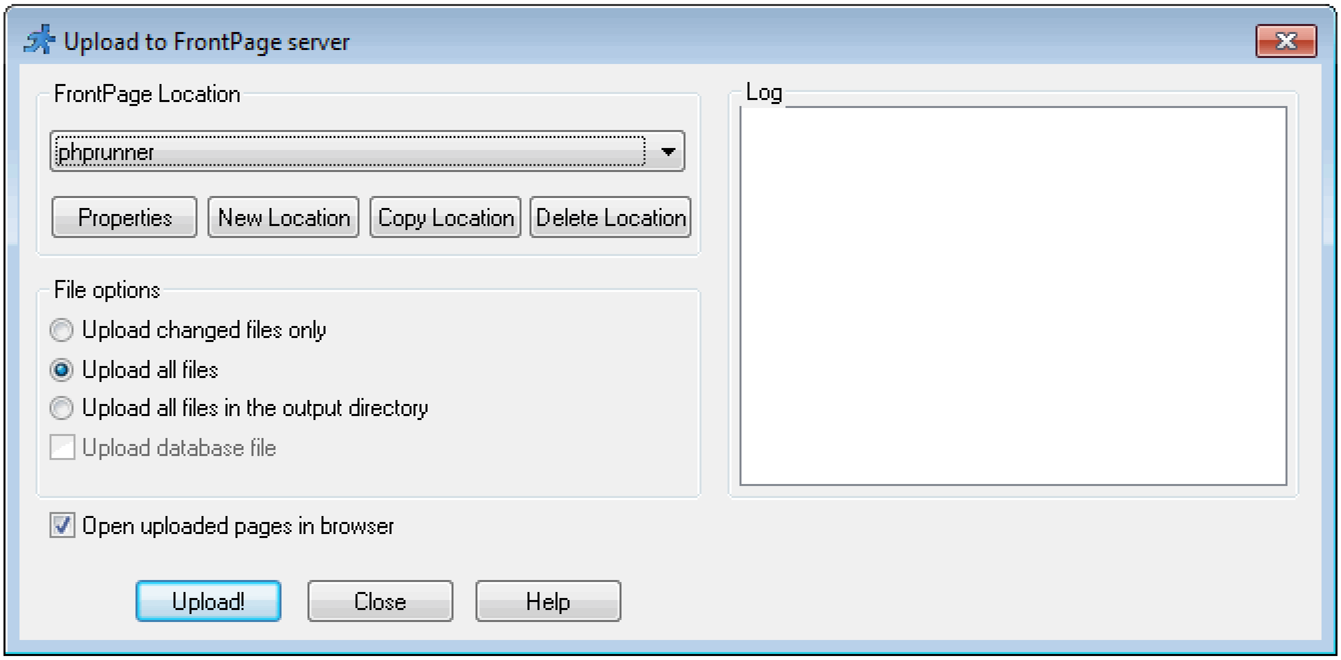
Destination directory
Upload pages to: /test/newfolder

Remote Web site URL: http://phprunner.com/test/newfolder

□ Use firewall ...

Fill in **Remote Web site URL** textbox to open downloaded pages in browser for test purposes.

To upload files choose FrontPage location and click **Upload** button. You can stop upload at any time by clicking **Stop** button.


Upload to FrontPage server

FrontPage Location: phprunner

File options
○ Upload changed files only
● Upload all files
○ Upload all files in the output directory
□ Upload database file

☑ Open uploaded pages in browser

You can choose between the following **File options**:

- **Upload changed files only** - to upload files that were changed since previous upload.
- **Upload all files** - to upload all generated files.
- **Upload all files in the output directory** - to upload all files including files that were created not by PHPRunner.

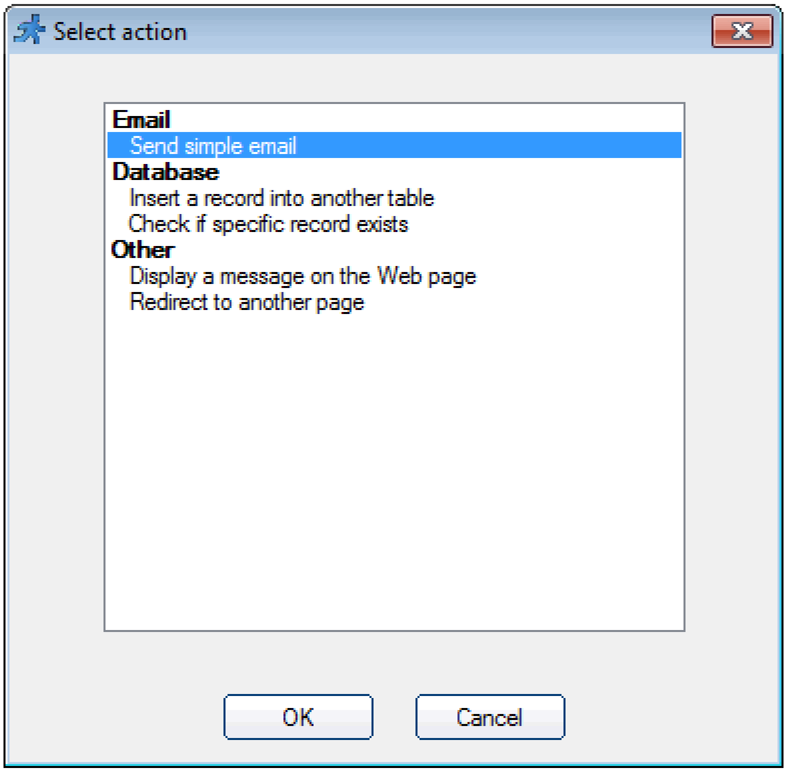
3 Advanced topics

3.1 Events

3.1.1 Predefined actions

3.1.1.1 Send simple email

To send a simple email use **Send simple email** action.


Select action
Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

```
// ********** Send simple email ************
$email="test@test.com";
$from="admin@test.com";
$msg="Hello there\nBest regards";
```

```
$subject="Sample subject";
$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, 'from'=>$from));
if(!$ret["mailed"])
   echo $ret["message"];
```

You can send HTML email as well:

```
// ********** Send HTML email ************
$email="test@test.com";
$from="admin@test.com";
$msg="Hello there\nBest regards";
$subject="Sample subject";
$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'htmlbody' =>
$msg, 'charset' => 'UTF-8', 'from'=>$from));
if(!$ret["mailed"])
   echo $ret["message"];
```

If you need to customize email templates that will be send to the users read this article: Email templates

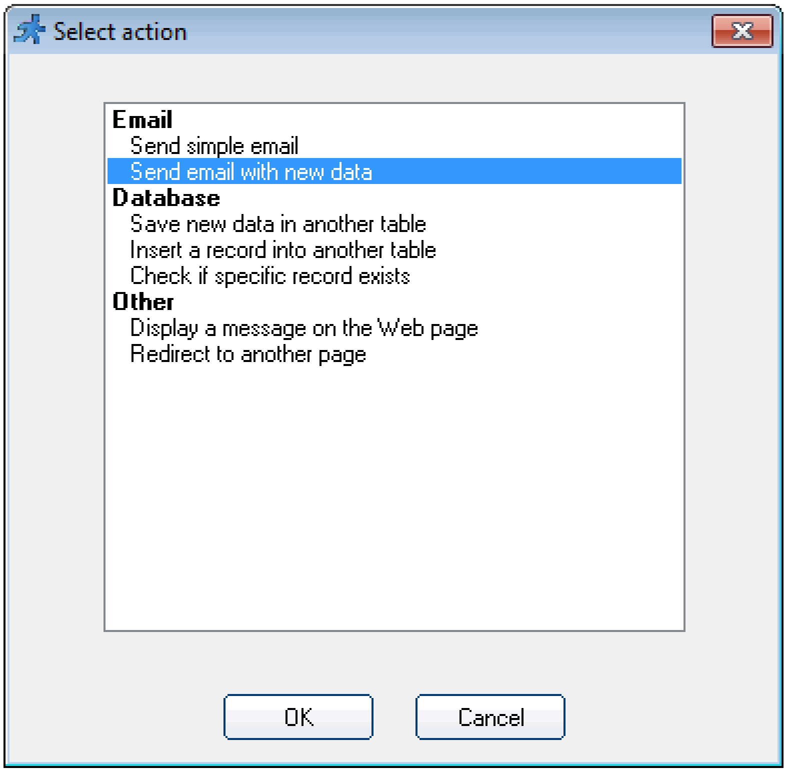
More info on **runner_mail** function.

3.1.1.2 Send email with new data

To send an email with new data use **Send email with new data** action.

Available in following events:

- Add page: Before record added;
- Add page: After record added;
- Edit page: Before record updated;
- Edit page: After record updated.


Select action
Email
Send simple email
Send email with new data
Database
Save new data in another table
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

```
//**********  Send email with new data  ************
$email="test@test.com";
$from="admin@test.com";
$msg="";
$subject="New data record";

$msg.= "Name: ".$values["name"]."\r\n";
$msg.= "Email: ".$values["email"]."\r\n";
$msg.= "Age: ".$values["age"]."\r\n";

$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, 'from'=>$from));
if(!$ret["mailed"])
  echo $ret["message"];
```

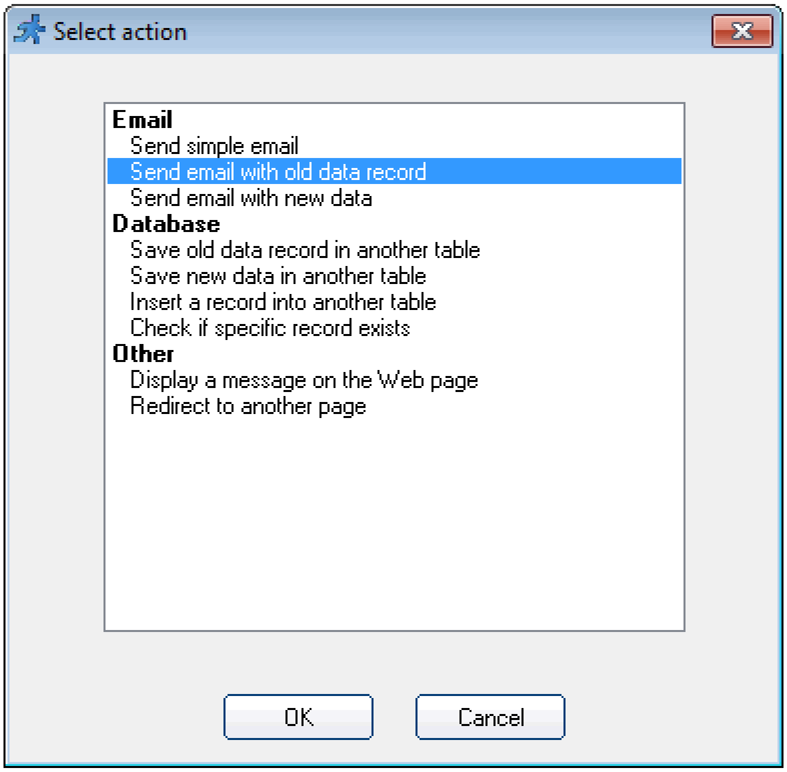
3.1.1.3 Send email with old data

To send an email with old data use **Send email with old data record** action.

Available in following events:

- Edit page: Before record updated;

- Edit page: After record updated;
- List page: Before record deleted.


Select action
Email
Send simple email
Send email with old data record
Send email with new data
Database
Save old data record in another table
Save new data in another table
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

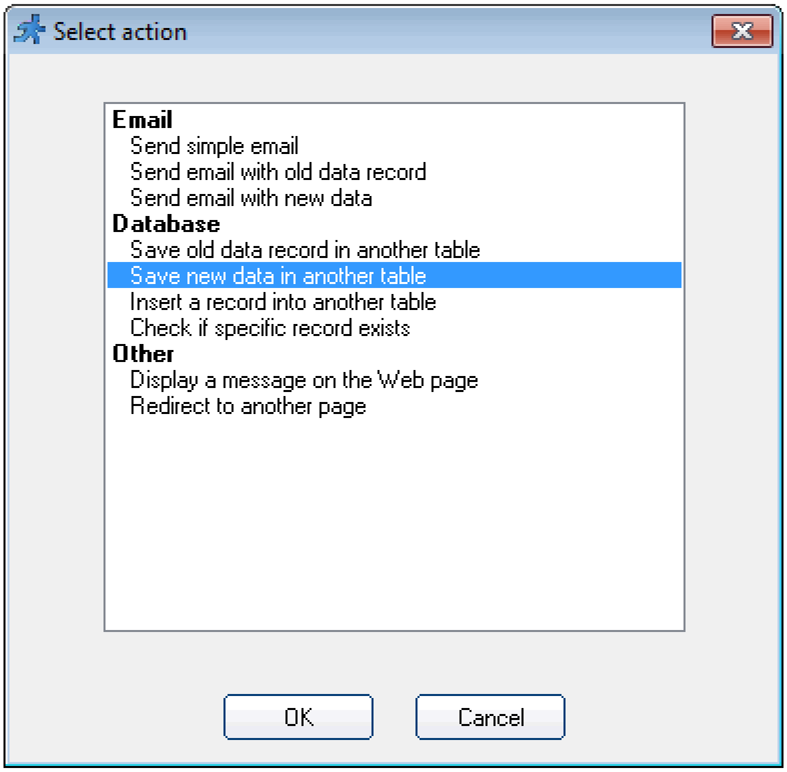
```
//**********  Send email with old data  ************
$email="test@test.com";
$from="admin@test.com";
$msg="";
$subject="New data record";

$msg.= "Name: ".$oldvalues["name"]."\r\n";
$msg.= "Email: ".$oldvalues["email"]."\r\n";
$msg.= "Age: ".$oldvalues["age"]."\r\n";

$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, 'from'=>$from));
if(!$ret["mailed"])
  echo $ret["message"];
```

3.1.1.4 Save new data in another table

To save new data in another table use **Save new data in another table** action.

Available in following events:

- Add page: Before record added;
- Edit page: Before record updated.


Select action
Email
Send simple email
Send email with old data record
Send email with new data
Database
Save old data record in another table
Save new data in another table
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

Example 1. Direct SQL query

```
//**********  Save new data in another table  ************
// note: text field values need to be wrapped by single quotes
$sql = "INSERT INTO AnotherTable (name, email, age) values (";
$sql .=  "'".$values["name"]."',";
$sql .=  "'".$values["email"]."',";
$sql .=  $values["age"];
$sql .= ")";
CustomQuery($sql);
```

Example 2. Insert record using Data Access Layer (DAL)

Another table needs to be part of the project.

```
//**********  Save new data in another table  ************
global $dal;
$tblEvents = $dal->Table("AnotherTable");
$tblEvents->Value["name"]=$values["name"];
```

```
$tblEvents->Value["email"]=$values["email"];
$tblEvents->Value["age"]=$values["age"];
$tblEvents->Add();
```

See also

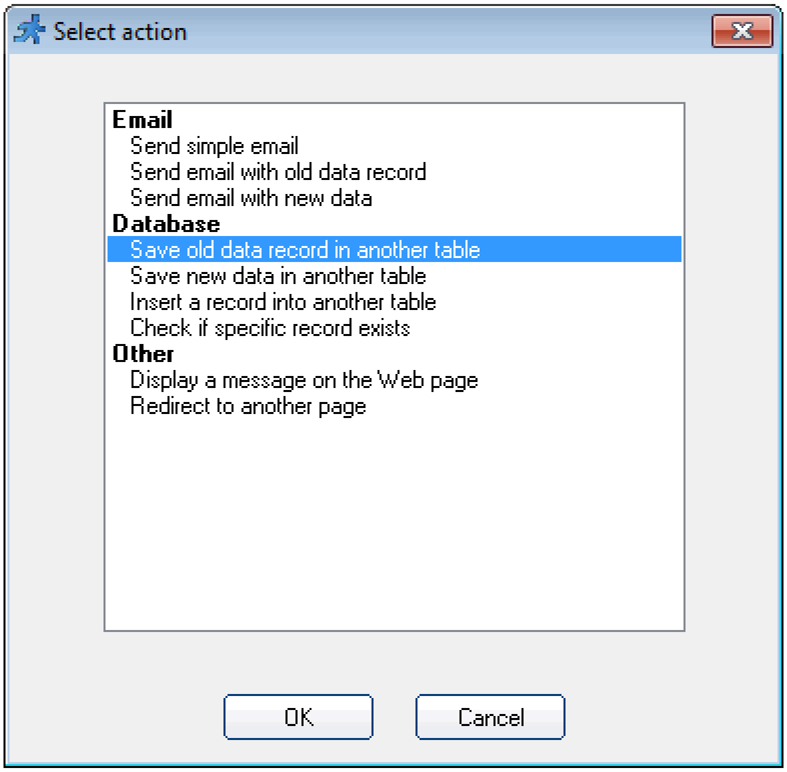
- Data_Access_Layer

3.1.1.5 Save old data in another table

To save old data in another table use **Save old data record in another table** action.

Available in following events:

- Edit page: Before record updated;
- List page: Before record deleted.



Select action

Email
Send simple email
Send email with old data record
Send email with new data
Database
Save old data record in another table
Save new data in another table
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page

Note: Change the values listed in red to match your specific needs.

Example 1. Direct SQL query

```
//**********  Save old data in another table  ************
// note: text field values need to be wrapped by single quotes
$sql = "INSERT INTO AnotherTable (name, email, age) values (";
$sql .=  "'".$oldvalues["name"]."',";
$sql .=  "'".$oldvalues["email"]."',";
$sql .=  $oldvalues["age"];
$sql .= ")";
CustomQuery($sql);
```

Example 2. Insert record using Data Access Layer (DAL)

Another table needs to be part of the project.

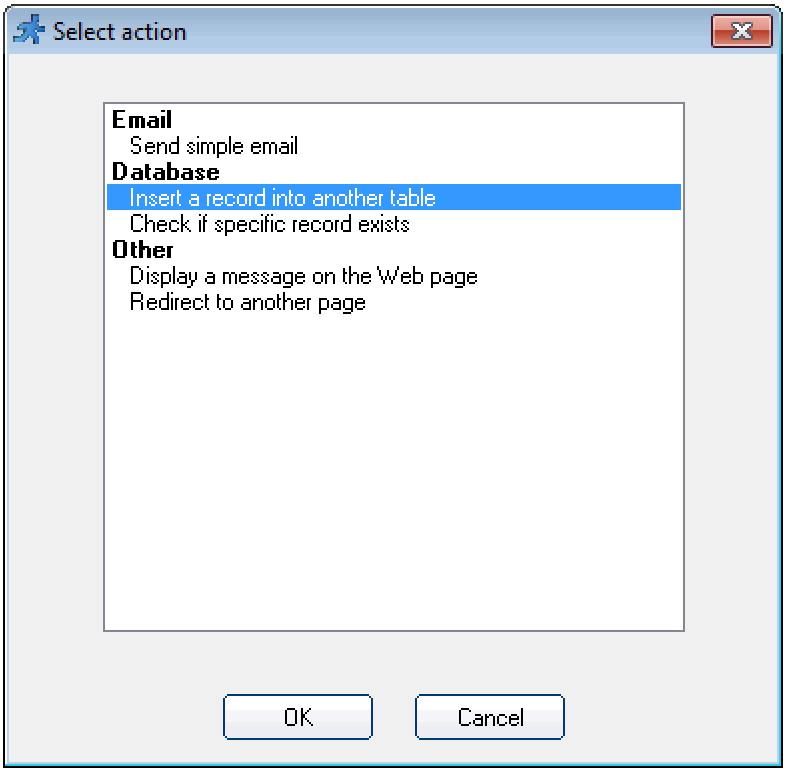
```
//**********  Save old data in another table  ************
global $dal;
$tblEvents = $dal->Table("AnotherTable");
$tblEvents->Value["name"]=$oldvalues["name"];
$tblEvents->Value["email"]=$oldvalues["email"];
$tblEvents->Value["age"]=$oldvalues["age"];
$tblEvents->Add();
```

See also

- Data_Access_Layer

3.1.1.6 Insert a record into another table

To insert a record into another table use **Insert a record into another table** action.


Select action
Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

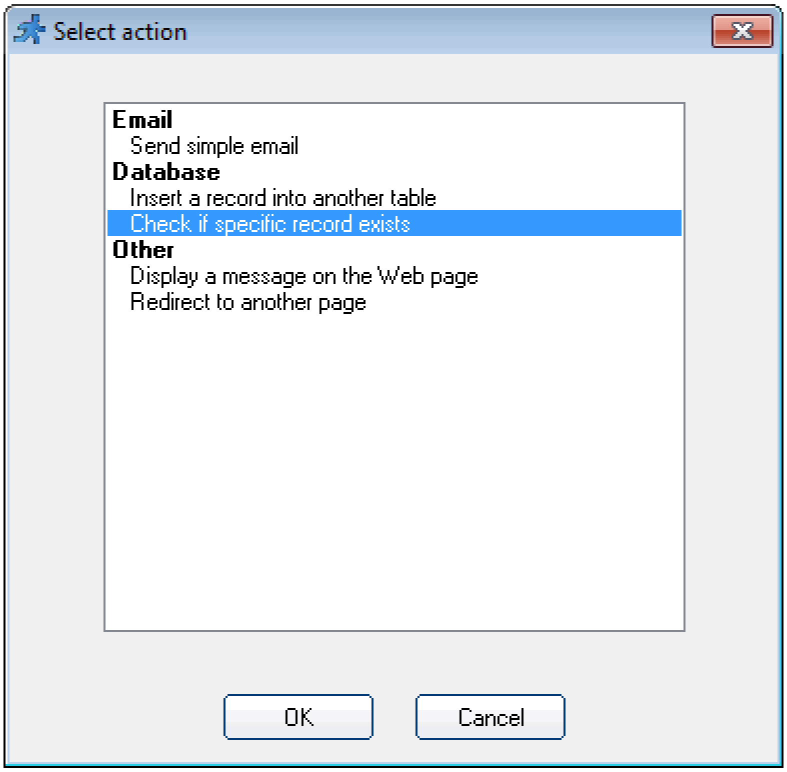
```
//**********  Insert a record into another table  ************
$sql = "INSERT INTO AnotherTable (name, email, age) values
   ('Bob Smith', 'bobsmith@gmail.com', 32)";
CustomQuery($sql);
```

See also

- DAL: CustomQuery()

3.1.1.7 Check if specific record exists

To check if a specific record exists use **Check if specific record exists** action.


Select action
Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

```
//**********  Check if specific record exists  ************
$strSQLExists = "select * from AnyTable where AnyColumn='SomeValue'";
$rsExists = db_query($strSQLExists);
$data=db_fetch_array($rsExists);
if($data)
{
  // if record exists do something
}
else
{
  // if dont exist do something else
}
```

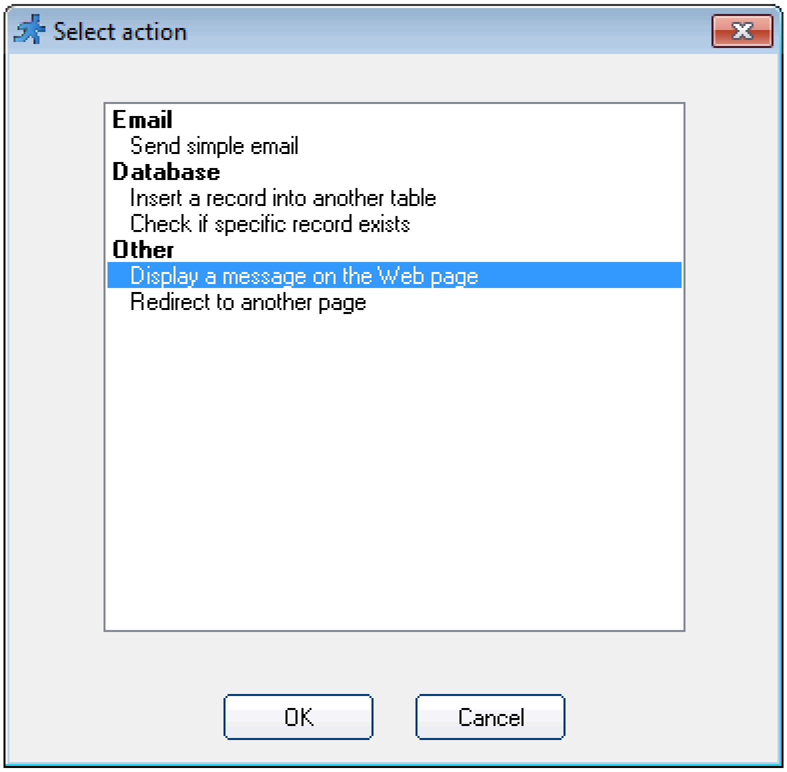
Example

Let's assume we need to check the uniqueness of the entered SSN number. Use the following code in the Before record added event:

```
//**********  Check if specific record exists  ************
$strSQLExists = "select * from Clients where SSN='" . $values["SSN"] . "'";
$rsExists = db_query($strSQLExists);
$data=db_fetch_array($rsExists);
if($data)
{
  // if record exists do something
  $message = "SSN " . $values["SSN"] . " already exists."
  return false;
}
else
{
  // if dont exist do something else
}
```

3.1.1.8 Display a message on the Web page

To display a message on the Web page use **Display a message on the Web page** action.

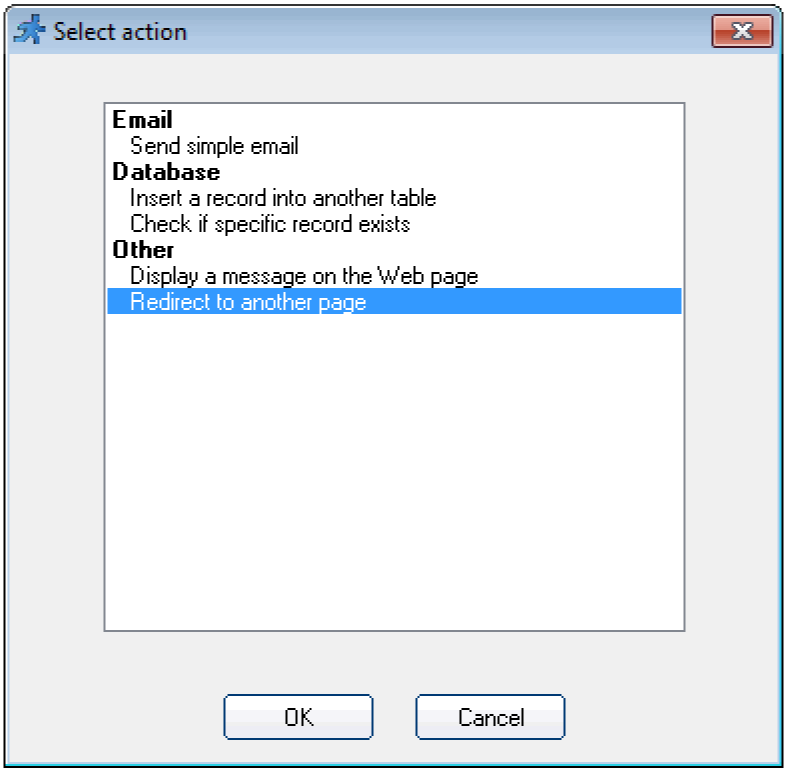

Select action
Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page
OK
Cancel

Note: Change the values listed in red to match your specific needs.

```
//**********  Display a message on the Web page  ************
echo "Your message here";
```

3.1.1.9 Redirect to another page

To redirect to another page use **Redirect to another page** action.



Email
Send simple email
Database
Insert a record into another table
Check if specific record exists
Other
Display a message on the Web page
Redirect to another page

Note: Change the values listed in red to match your specific needs.

```
//**********  Redirect to another page  ************
header("Location: anypage.php");
exit();
```

3.1.2 Sample events

3.1.2.1 Appearance

Add custom field to form

To add custom field to the form follow the instructions below.

Note: Change the values listed in red to match your specific needs.

1. Proceed to the **Visual Editor** page, switch to HTML mode and add a custom input field to your form. Make sure you specify field ID.

```
<INPUT id=test type=text>
```

2. Add the following code to Javascript OnLoad event of the page where custom field was added:

```
this.on('beforeSave', function(formObj, fieldControlsArr, pageObj){
   var val = $("#test").val();
   formObj.baseParams['test'] = val;
});
```

3. In any event like Before record added or Before process use *$_REQUEST["test"]* to access the custom field value.

Add dropdown list box with values for search

Add dropdown list box with values for search. E.g. when you select a car make name from dropdown list box, only the data concerning selected car make are displayed.

Insert PHP code snippet on the **Page Designer** page.

Note: Change the values listed in red to match your specific needs.

```
$str= "<select style='width: 150px; display: inline-block;' class='form-
control' onchange=\"window.location.href=this.options[this.".
  "selectedIndex].value;\"><option value=\"\">Please select</option>";
//select values from database
$strSQL = "select Make from Cars";
$rs = db_query($strSQL);
while ($data = db_fetch_array($rs))
  $str.="<option value='cars_list.php?q=(Make~equals~".
  $data["Make"].")'>".$data["Make"]."</option>";
$str.="</select>";
echo $str;
```

Add link to user profile to the menu

Lets say you want to give each user a quick access to her own data in users table. It would be nice to add 'My profile' link right to the main menu.

Link to user profile page looks like *users_edit.php?editid1=XXXX*. We assume that login table name is *users* and *XXXX* is the value of primary key field in users table.

Note: Change the values listed in red to match your specific needs.

1. Save ID of user account in session variable. For this purpose add the following code to AfterSuccessfulLogin event:

```
$_SESSION["user_id"]=$data["id"];
```

In this example *id* is a primary key column name in login table.

2. Create a new menu item via Menu Builder.

- Link type: Application page.

- Link to: Users (login table) List page.
- Link text: My profile.

3. Now add the following code to MenuItem: Modify event:

```
if ($menuItem->getTitle()=="My profile") {
   $menuItem->setUrl("users_edit.php?editid1=".$_SESSION["user_id"]);
}
return true;
```

Add new button to Add/Edit pages

Lets say we need to add a new button to the Add or Edit page that will save the record and redirect user back to the List page.

Note: Change the values listed in red to match your specific needs.

1. Add a new button to the Add or Edit page via Insert button function in Visual Editor.
2. Add the following code to the ClientBefore event of the button:

```
pageObj.on('beforeSave', function(formObj, fieldControlsArr, pageObj){
  formObj.baseParams['golist'] = "1";
});

$("#saveButton1").click();

return false;
```

3. Add the following code to the AfterAdd event of Add page or AfterEdit event of Edit page:

```
if($_REQUEST["golist"])
{
  header("Location: ..._list.php");
  exit();
}
```

Change cell background color

To change any cell background color use the following code in After Record Processed event.

Note: Change the values listed in red to match your specific needs.

```
$record["FieldName_css"]='background:red;';
```

Also you can change background color of a cell when the mouse hovers over it:

```
$record["FieldName_hovercss"]='background:yellow;';
```

See also

- Change row background color
- Conditional formatting

Change font size in text box

To change font size in all text boxes placed on a page use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
$("input[type=text]").css('fontSize', '120%');
```

See also

- Javascript API

Change 'Logged on as' message

The "Logged on as" message can be changed in the AfterSuccessfulLogin event. Here is how you can display user full name instead of username:

Note: Change the values listed in red to match your specific needs.

```
$_SESSION["UserName"] = $data["FirstName"].' '.$data["LastName"]
```

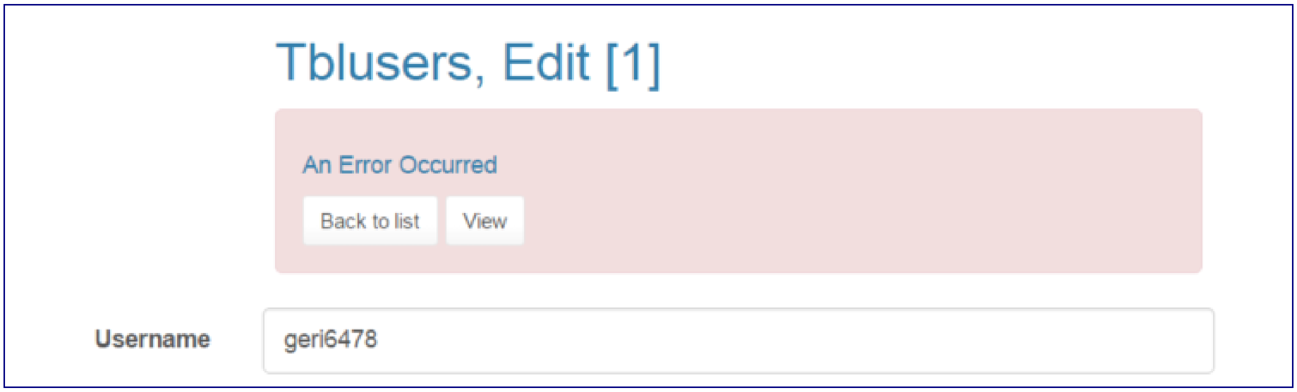
Change message after record was added or saved

To make a change message use the following code in AfterAdd or AfterEdit events.

Example 1.

```
$pageObject->setMessageType(MESSAGE_ERROR);
$pageObject->setMessage("An Error Occurred");
```

This is how the result will look:



Tblusers, Edit [1]

An Error Occurred

Username geri6478

Example 2.

```
$pageObject->setMessageType(MESSAGE_INFO);
$pageObject->setMessage("Data was saved, all good");
```

💡 **Note**: These codes will always work for Edit page. And will not work for Add page if you choose to display master data on details page.

Change message after record was added or saved

Change row background color

To change any row background color use the following code in After Record Processed event.

💡 **Note**: Change the values listed in red to match your specific needs.

```
$record["css"]='background:blue;';
```

Also you can change background color of a row when the mouse hovers over it:

```
$record["hovercss"]='background:yellow;';
```

See also

- Change cell background color
- Conditional formatting

Change width of edit box with AJAX popup

To change width of edit box with AJAX popup use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.getDispElem().css("width", "200px");
```

See also

- Javascript API

Change width of text field on Quick Search panel

To change width of text field on Quick Search panel use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
$("input[name^='value_make'], select[name^='value_make']").width(150);
```

See also

- Javascript API

Disable record editing

To hide the **Edit** link when the record status is "processed", use the following code in the List page: After record processed event.

Note: Change the values listed in red to match your specific needs.

```
if ($data["Status"]=="processed")
{
   $record["edit_link"] = false;
}
```

To be sure that record editing is disabled, also perform a checkup on the **Edit** page.

Hide buttons in some rows of datagrid

Let's say you added a button to the datagrid to export a single record. And if the record has been already exported (value of field *exported* equals 1), you want to hide the export button in this row.

Note: Change the values listed in red to match your specific needs.

1. Switch to HTML mode in Visual Editor and add wrappers around button's code. Here is the sample code:

```
{BEGIN MyButton}
   <A class="rnr-button button" id="MyButton" href="#" typeid="ib">
      MyButton
   </A>
{END MyButton}
```

2. Now in the List page: AfterRecordProcessed event add the following code:

```
if ($data['exported']!=1)
$record["MyButton"]=true;
```

This code will only display button if value of field *exported* equals 1.

Hide controls on Add/Edit pages, based on logged user name

To hide controls on Add/Edit pages, based on logged user name, use the following code in Add Page: BeforeDisplay and/or Edit Page: BeforeDisplay event.

Note: Change the values listed in red to match your specific needs.

1. To show the *Horsepower* field edit control only if user name equals "admin", use the following code:

```
if (Security::getUserName()!="admin")
   $pageObject->hideField("Horsepower");
```

2. To show the *Horsepower* field edit control only if current user belongs to the admin group, use the following code:

```
if (Security::isAdmin())
   $pageObject->hideField("Horsepower");
```

Note that this code sample will work only with static permissions.

3. To hide the *Horsepower* field edit control placed on tab or folding section on the Edit/Add/View page, use the following code:

```
$pageObject->hideField("Horsepower");
```

More info about PHPRunner Security API.

Hide empty fields on View page

Let's say you have a list of fields on the View page and you want to hide those that are empty. For this purpose use the following code in the [View page: Before display]() event for each field you need to hide. This example uses field named *EPAHighway*.

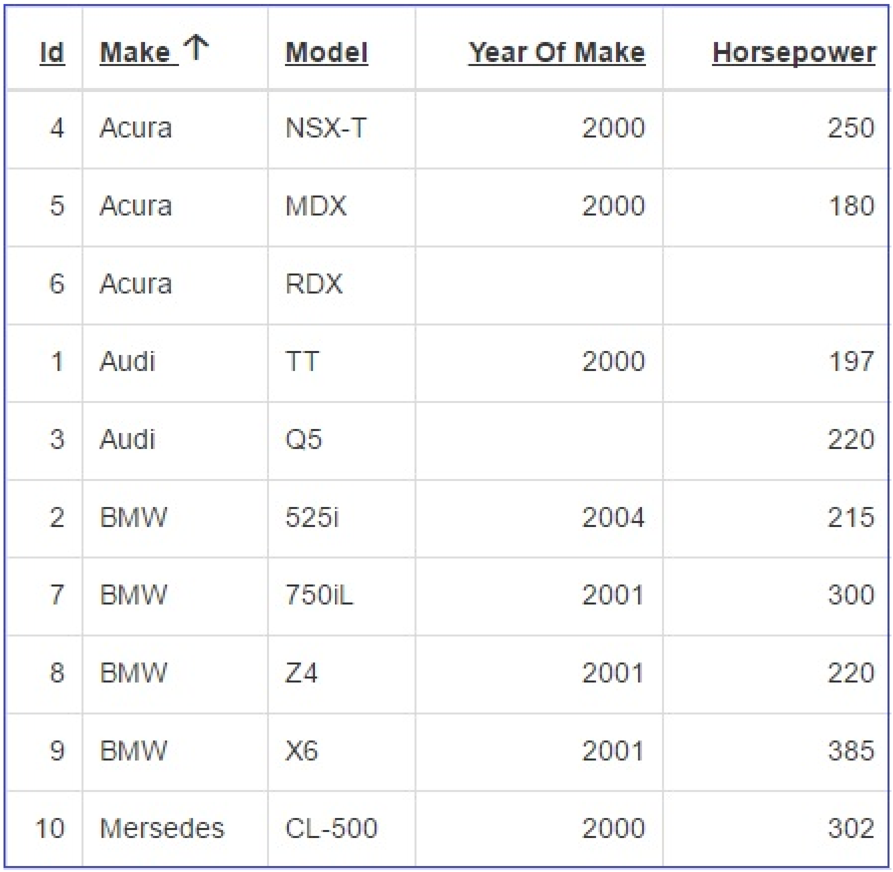
Note: Change the values listed in red to match your specific needs.

```
if (!$values["EPAHighway"])
   $pageObject->hideField("EPAHighway");
```

There is an option to hide empty fields on the View page automatically. To enable it, go to the **Choose pages** screen, click the **Settings** button near the **View record** checkbox and then select **Hide empty fields** checkbox.

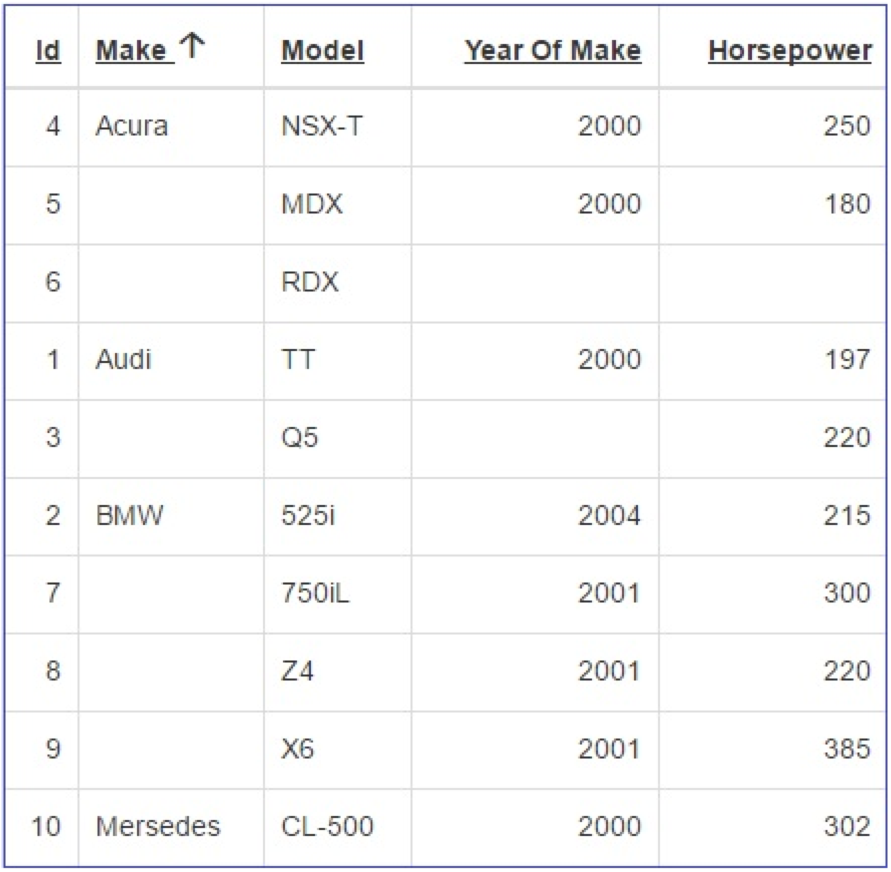
Hide repeating values on List page

Here is the typical list page with the list of cars sorted by make.



Id	Make ↑	Model	Year Of Make	Horsepower
4	Acura	NSX-T	2000	250
5	Acura	MDX	2000	180
6	Acura	RDX		
1	Audi	TT	2000	197
3	Audi	Q5		220
2	BMW	525i	2004	215
7	BMW	750iL	2001	300
8	BMW	Z4	2001	220
9	BMW	X6	2001	385
10	Mersedes	CL-500	2000	302

Sometimes you may need to make this screen less cluttered removing repeating values in the *Make* column. Something like this:



Id	Make ↑	Model	Year Of Make	Horsepower
4	Acura	NSX-T	2000	250
5		MDX	2000	180
6		RDX		
1	Audi	TT	2000	197
3		Q5		220
2	BMW	525i	2004	215
7		750iL	2001	300
8		Z4	2001	220
9		X6	2001	385
10	Mersedes	CL-500	2000	302

Here is how this can be done:

1. Add the following code to List page: BeforeProcess event:

```
$_SESSION["Make"]="";
```

2. Set 'View as' type of *Make* field to 'Custom' and paste the following code there:

```
if ($value==$_SESSION["Make"])
$value="";
else
$_SESSION["Make"]=$value;
```

Print search parameters on List page

If you print Advanced search or Search panel parameters on the List page, add PHP code snippet with following code:

```
global $strTableName;
if (isset($_SESSION[$strTableName.'_advsearch']))
{
   $searchClauseObj = unserialize($_SESSION[$strTableName.'_advsearch']);
}
```

```
$fieldSearchData = $searchClauseObj->_where[$strTableName."_srchFields"];

echo "Search was completed."."<br>";

for($i=0; $i<count($fieldSearchData); $i++){
   $fName = $fieldSearchData[$i]['fName'];
   $val1 = $fieldSearchData[$i]['value1'];
   $val2 = $fieldSearchData[$i]['value2'];
   $srchCat = $fieldSearchData[$i]['opt'];
   $srchNot = $fieldSearchData[$i]['not'];
   echo "Search Field: ".$fName."<br>";
   echo "Search Option: ".$val1."<br>";
}
```

Printing basic search parameters:

```
global $strTableName;
if (isset($_SESSION[$strTableName.'_advsearch']))
{
   $searchClauseObj = unserialize($_SESSION[$strTableName.'_advsearch']);
}

echo $searchClauseObj->_where[$searchClauseObj-
>sessionPrefix."_simpleSrch"];
```

Redirect to details page after master record was added

Sometimes you need to redirect visitor to the details table (i.e Order details) right after new master record was added (Orders table). The trick is to pass correct master key to the details table. Once user redirected there she can start adding details records using Inline or regular Add.

The following code needs to be added to AfterAdd event of the master table.

Note: Change the values listed in red to match your specific needs.

```
header("Location: order_details_list.php?mastertable=orders&masterkey1=" .
   $values["OrderID"]);
exit();
```

In this example:

- *order_details_list.php* - URL of the details list page;
- *orders* - master table name;
- *OrderID* - name of the master key field.

Show data from master table on detail view/edit/add page

Show data from master table on details view/edit/add page.

To add this event insert PHP code snippet on the **Page Designer** page.

Note: Change the values listed in red to match your specific needs.

```
global $pageObject;
echo "Master Info<br>";
if ($data = $pageObject->getMasterRecord())
  {
    echo "Field1: ".$data["Field1"]."<br>";
    echo "Field2: ".$data["Field2"]."<br>";
  }
```

For example, **Orders** is a master table, while **Order details** is a details.

```
global $pageObject;
echo "Order Info<br>";
if ($data = $pageObject->getMasterRecord())
  {
    echo "Customer: ".$data["CustomerID"]."<br>";
    echo "Employee: ".$data["EmployeeID"]."<br>";
  }
```

For more information see getMasterRecord.

Show dropdown list of US states if US was selected in country list

To show dropdown list of US states if US was selected in the country list or hide it otherwise, use the following code in Javascript OnLoad event on Add/Edit pager in popup and inline.

Note: Change the values listed in red to match your specific needs.

```
var ctrlCountry = Runner.getControl(pageid, 'country');
var ctrlState = Runner.getControl(pageid, 'state');

ctrlCountry.on('change', function(e) {
    if (this.getValue() == 'US') {
        ctrlState.show();
    } else {
        ctrlState.hide();
    }
});
```

You may want to hide field label as well. Use the following code to hide or show the whole table row with state edit control based on country field selection. It will work in popup and inline.

```
var ctrlCountry = Runner.getControl(pageid, 'country');

ctrlCountry.on('change', function(e) {
    if (this.getValue() == 'US') {
        pageObj.showField("state");
    } else {
        pageObj.hideField("state");
    }
});
```

See also

- [Javascript API]

3.1.2.2 Database

Before deleting a record check for related records

Before deleting a record in the Orders table check for related items in the OrderDetails table. Add the following code to the [List page: Before record deleted] event.

Note: Change the values listed in red to match your specific needs.

```
// Parameters:
// where - string with WHERE clause pointing to record to be deleted.
global $dal;
$tblOrder = $dal->Table("OrderDetails");
$rs = $tblOrder->Query("OrderID=".$deleted_values["OrderID"],"");
$data = db_fetch_array($rs);
if($data)
  return false;
else
  return true;
```

For more information about using Data Access Layer (DAL), see [Data Access Layer].

Dynamic SQL query

Sometimes you need to present different data to different groups of users. Let's say you run a classified ads board and needs to display last seven days data to regular users while admin should be able to see all adds.

Add the following code to the [After table initialized] event.

Note: Change the values listed in red to match your specific needs.

1. For MS SQL Server database:

```
if (Security::getUserName()!="admin")
   $query->addWhere("DATEDIFF(day,DateColumn, GETDATE()) <= 7");
```

2. For MySQL database:

```
if (Security::getUserName()!="admin")
  $query->addWhere("DATE_SUB(CURDATE(), INTERVAL 7 DAY) <= DateColumn");
```

3. For MS Access database:

```
if (Security::getUserName()!="admin")
   $query->addWhere("DATEDIFF('d',DateColumn, Now()) <= 7");
```

4. When we use Dynamic Permissions code will be slightly different (MySQL example). If user doesn't belong to admin group we only show last seven days data.

```
if (!Security::isAdmin())
    $query->addWhere("DATE_SUB(CURDATE(), INTERVAL 7 DAY) <= DateColumn");
```

See also

- Method: addWhere
- About SQLQuery class

Limit number of records users can add

Lets say you want to limit the number of records users can add to the certain table. For example, if you run classified ads website you want free users to be able to post up to 3 ads, basic plan users can add up to 10 ads etc.

Sample database structure

Table1: users, userid, password, limit.

Table2: ads, id, userid, adtext.

Tables are linked via *userid* field in both. It would also make sense to turn on Advanced security mode 'Users can see all data, can edit their own data only'.

Use the following code in Add page: Before Record Added event.

Scenario 1. All users can add the same number of records

```
$limit = 3;

$rs = DB::Query("select count(*) as c from ads where userid = " .
$_SESSION["UserID"]);
$data = $rs->fetchAssoc();
$count = $data["c"];

```

```
if ($count>= $limit)
{
   echo "Limit reached: $count records added already";
   exit();
}
```

Scenario 2. Each user has it's own limit. Limits are stored in the *userlimit* field of users table

```
$rs = DB::Query("select count(*) as c from ads where userid = " .
$_SESSION["UserID"]);
$data = $rs.fetchAssoc();
$count = $data["c"];

$rs2 = DB::Query("select ".AddFieldWrappers( "userlimit" )." from users
where userid = " . $_SESSION["UserID"]);
$data2 = $rs2.fetchAssoc();
$limit = $data2["userlimit"];

if ($count>= $limit)
{
   echo "Limit reached: $count records added already";
   exit();
}
```

Select multiple values from checkboxes or a list field and have them appear as individual database entrie

To select multiple values from check boxes or a list field and have them appear as individual database entries use the following code in Add page: Before Record Added event.

Note: Change the values listed in red to match your specific needs.

```
if ($values["fieldname"])
{
  $arr = explode(",",$values["fieldname"]);
  // This is the name of the multi check box or
  // list select field, its value becomes $arr
for ($i=0;$i<count($arr);$i++)
{
  $strInsert = "insert into TableName (field) values ('".$arr[$i]."')";
  // add more fields from the add page to be inserted into database

  db_exec($strInsert);
}
header("Location: MyPage_list.php");
```

```
  // Exit and Redirect to the list page after updating database
exit();
}
```

Show list of customer orders

Show list of all orders placed by this customer on Order Edit page.

Insert PHP code snippet on the **Page Designer** page.

Note: Change the values listed in red to match your specific needs.

```
global $dal;
$tblOrders = $dal->Table("Orders");
$rs = $tblOrders->Query("OrderID=".$_REQUEST["editid1"],"");
$data = db_fetch_array($rs);
$CustomerID = $data["CustomerID"];
echo "Orders placed by " .$CustomerID. "<br>";
$rsOrders = $tblOrders->Query("customerid='".$CustomerID."'","");
while($data=db_fetch_array($rsOrders))
{
  echo "<a target=_blank href=Orders_edit.php?editid1=".$data["OrderID"].
  ">". $data["OrderID"]."</a> ".$data["OrderDate"]."<br>";
}
```

For more information about using Data Access Layer (DAL), see Data Access Layer.

Store the date and time when a record is modified

To keep track of when records in a table are last modified or who made the modification you have two options:

1. Set default value under 'Edit as' settings

To store login name of the user who made the change, set default value to:

```
$_SESSION["UserID"]
```

To store modification date and time, set default value to:

```
now()
```

Make sure to clear **AutoUpdate Value** field.

2. Modify BeforeAdd/BeforeEdit events

Add the following code to the BeforeAdd/BeforeEdit events:

```
$values["DateUpdated"]=now();
$values["UpdateBy"]=$_SESSION["UserID"];
```

Update multiple records on the List page

To update multiple records on the List page at the same time you can create new button **Update selected**, then choose records on the List page and click the **Update selected** button.

Note: Change the values listed in red to match your specific needs.

1. Proceed to the **Visual Editor** page.

2. Create the custom button **Update selected** and add the following code

to the **Server** tab:

```
global $dal;

while ( $data = $button->getNextSelectedRecord() ) {
    // set ReportsTo field to 'Bob Smith'
    $sql = "Update employees set ReportsTo='Bob Smith' where
KeyField=".$data["KeyField"];
    CustomQuery($sql);
}
$result["txt"] = "Records were updated.";
```

and to the **Client After** tab:

```
var message = result["txt"];
ctrl.setMessage(message);
```

The **Client Before** tab should be blank (delete sample code there if any).

For more information about using Data Access Layer (DAL), see Data Access Layer.

Update multiple tables

To update two joined tables use the following code in Add page: Before record added and/or Edit page: Before record updated events.

Note: Change the values listed in red to match your specific needs.

```
$sql = "update othertable set joinedfield=".$values["joinedfield"]." ... ";
db_exec($sql);
unset($values["joinedfield"]);
```

To update master and details tables use the following code for details table in Add page: Before record added and/or Edit page: Before record updated events.

```
global $dal;
$tblDetail = $dal->Table("DetailTableName");
$tblDetail->Value["Field1"] = $values["Field1"];
$tblDetail->Param["OrderID"] = $values["OrderID"];
$tblDetail->Update();
unset($values["Field1"]);
```

Field1 is a field in the details table and **OrderID** is the linked field in the master table.

For more information about using Data Access Layer (DAL), see Data Access Layer.

Search Master and Details tables together

Here is how you can **search master and details tables together**. For instance you have Orders and OrderDetails tables and need to find orders that contain a certain product.

1. Modify Orders SQL Query to add a dummy field named 'product'. Make sure this field is searchable.

```
SELECT
        OrderID,
        CustomerID,
        EmployeeID,
        OrderDate,
        ShipAddress,
        ShipCity,
        ShipRegion,
        ShipPostalCode,
        ShipCountry,
        '' as product
FROM orders
```

2. Orders table, AfterTableInit event:

```
$srchObj = SearchClause::getSearchObject("orders");

$value = $srchObj->getFieldValue("product");

if( $value != null ) {
$srchObj->setSearchSQL("product", "OrderID in (select OrderID from OrderDetails wh
}
```

In this event we do a subquery to find all orders that contain the product in question.

More information:
About Search API
setSearchSQL

3.1.2.3 Email

Email selected records

To send an email with several selected records on the List page, you need to create custom button.

Note: Change the values listed in red to match your specific needs.

1. Proceed to the **Visual Editor** page.

2. Create the custom button **Update selected** and add the following code

to the **Server** tab:

```
$body = "";
while( $data = $button->getNextSelectedRecord() )
{
    $body .= "OrderID: " . $data['OrderID'] . "\n";
    $body .= "Customer: " . $data['CustomerID'] . "\n";
    $body .= "Employee: " . $data['EmployeeID'] . "\n-----------\n\n";
}

// send the email
$email = "test@test.com";
$subject = "Sample subject";
$arr = runner_mail(array('to' => $email, 'subject' => $subject,'body' =>
$body));

$result["txt"] = "Emails were sent.";

// if error happened print a message on the web page
if( !$arr["mailed"] )
{
    $errmsg = "Error happened: <br>";
    $errmsg.= "File: " . $arr["errors"][0]["file"] . "<br>";
    $errmsg.= "Line: " . $arr["errors"][0]["line"] . "<br>";
    $errmsg.= "Description: " . $arr["errors"][0]["description"] . "<br>";
    $result["txt"] = $errmsg;
}
```

and to the **Client After** tab:

```
var message = result["txt"];
ctrl.setMessage(message);
```

The **Client Before** tab should be blank (delete sample code there if any).

Sample email message:

```
OrderID: 10249

Customer: TRADH

Employee: 6

------------------

OrderID: 10251

Customer: VICTE

Employee: 3

------------------

OrderID: 10253

Customer: HANAR

Employee: 3

------------------
```

Note: How to send email to the current logged user

Instead of the hardcoded email address you can send it to the current logged user. If you use email address as a username use the following:

```
$email=$_SESSION["UserID"];
```

If email address is stored in another field of users table you need to save it to the session variable in AfterSuccessfulLogin event:

```
$_SESSION["email"]=$data["email"];
```

Then in BeforeProcess event code use the following:

```
$email=$_SESSION["email"];
```

Send an email to selected users

Sometimes you need to send an email to selected users. Let's say one of the fields on a page *EmailField* contains user email. The key field on the page is *KeyColumn*, table name - *TableName*.

Let's consider two situations:

Subject and text of the email are hard-coded and determined directly in the event

Note: Change the values listed in red to match your specific needs.

1. Proceed to the **Visual Editor** page.

2. Create the custom button (e.g. "Email selected") and add the following code to the **Server** tab:

```
$emails = array();

while( $data = $button->getNextSelectedRecord() )
{
   if( $data["EmailField"] )
       $emails[] = $data["EmailField"];
}

// send the email
$email = implode(", ", $emails);
$subject = "Sample subject";
$body = "Your email text here";
$arr = runner_mail(array('to' => $email, 'subject' => $subject, 'body'
=> $body));

$result["txt"] = "Emails were sent.";

// if error happened print a message on the web page
if( !$arr["mailed"] )
{
   $errmsg = "Error happened: <br>";
   $errmsg.= "File: " . $arr["errors"][0]["file"] . "<br>";
   $errmsg.= "Line: " . $arr["errors"][0]["line"] . "<br>";
   $errmsg.= "Description: " . $arr["errors"][0]["description"] .
"<br>";
   $result["txt"] = $errmsg;
}
```

and to the **Client After** tab:

```
var message = result["txt"];
ctrl.setMessage(message);
```

The **Client Before** tab should be blank (delete sample code there if any).

User enters subject, email body, "from" email address

Note: Change the values listed in red to match your specific needs.

1. Proceed to the **Visual Editor** page and select List page.

2. Create the custom button (e.g. "Email selected"). Add the following code to the **Client Before tab** tab when you are using **Bootstrap layout**:

```
if ( proxy["emailUsers"] === true ) {
    params["emailFrom"] = proxy["emailFrom"];
    params["emailBody"] = proxy["emailBody"];
    params["emailSubject"] = proxy["emailSubject"];

    proxy["emailUsers"] = false;
    return true;
}

var selBoxes = pageObj.getSelBoxes( pageid ),
    args = {
        modal: true,
        header: "Input from, subject and body",
        html: '<div>'
           + '<div>From: <input type="text" id="emailFrom"s
style="margin: 5px 0"></div>'
           + '<div>Subject: <input type="text" id="emailSubject"s
style="margin: 5px 0"></div>'
           + '<div>Body: <textarea id="emailBody"></textarea></div>'
           + '</div>',
        footer: '<a href="#" id="emailUsersSave" class="btn btn-
primary">' + Runner.lang.constants.TEXT_SAVE + '</a>'
            + '<a href="#" id="emailUsersCancel" class="btn btn-
default">' + Runner.lang.constants.TEXT_CANCEL + '</a>',
        afterCreate: function( win ) {
            $("#emailUsersSave").on("click", function( e ) {
                var context = win.body();

                proxy["emailUsers"] = true;
                proxy["emailFrom"] = $("#emailFrom", context).val();
                proxy["emailBody"] = $("#emailBody", context).val();
                proxy["emailSubject"] = $("#emailSubject",
context).val();
                $('[id="' + ctrl.id + '"]').click();

                e.preventDefault();
                win.destroy();
            });

            $("#emailUsersCancel").on("click", function( e ) {
                e.preventDefault();
                win.destroy();
            });
        }
    };

if ( selBoxes.length == 0 || !confirm('Do you really want to email
these records?') ) {
    return false;
}

Runner.displayPopup( args );
return false;
```

Add the following code to the **Client Before tab** tab when you are using simple **non-Bootstrap layout**:

```
if ( proxy["emailUsers"] === true ) {
    params["emailFrom"] = proxy["emailFrom"];
    params["emailBody"] = proxy["emailBody"];
    params["emailSubject"] = proxy["emailSubject"];

    proxy["emailUsers"] = false;
    return true;
}

var selBoxes = pageObj.getSelBoxes( pageid ),
    args = {
        modal: true,
        header: "Input from, subject and body",
        html: '<div>'
           + '<div>From: <input type="text" id="emailFrom"s
style="margin: 5px 0"></div>'
           + '<div>Subject: <input type="text" id="emailSubject"s
style="margin: 5px 0"></div>'
           + '<div>Body: <textarea id="emailBody"></textarea></div>'
           + '</div>',
        afterCreate: function( win ) {
            win._obj.bodyNode.setStyle("textAlign", "center");

            win._obj.addButton({
                label : Runner.lang.constants.TEXT_SAVE,
                name : "saveButton",
                template: "<a />",
                classNames : "rnr-button main",
                section: Runner.Y.WidgetStdMod.FOOTER,
                action : function (e) {
                    var context = win._obj.bodyNode.getDOMNode();
                    proxy["emailUsers"] = true;
                    proxy["emailFrom"] = $("#emailFrom",
context).val();
                    proxy["emailBody"] = $("#emailBody",
context).val();
                    proxy["emailSubject"] = $("#emailSubject",
context).val();
                    $('[id="' + ctrl.id + '"]').click();
                    e.preventDefault();
                    win.destroy();
                }
            });

            win._obj.addButton({
                label : Runner.lang.constants.TEXT_CANCEL,
                template: "<a />",
                classNames : "rnr-button",
                section: Runner.Y.WidgetStdMod.FOOTER,
                action : function (e) {
                    e.preventDefault();
                    win.destroy();
                }
            });
        }
    };

if ( selBoxes.length == 0 || !confirm('Do you really want to email
these records?') ) {
    return false;
}

Runner.displayPopup( args );
return false;
```

and to the **Server tab** tab:

```
if( $params["emailBody"] || $params["emailSubject"] ||
$params["emailFrom"] )
{
   $emails = array();
   while ( $data = $button->getNextSelectedRecord() )
   {
      if ( $data["EmailField"] )
         $emails[] = $data["EmailField"];
   }

   // send the email
   $from = $params["emailFrom"];
   $email = implode(",", $emails);
   $body = $params["emailBody"];
   $subject = $params["emailSubject"];
   $arr = runner_mail(array('to' => $email, 'subject' => $subject,
'body' => $body, 'from' => $from));

   $result["txt"] = "Emails were sent.";
   if (!$arr["mailed"])
   {
      $errmsg = "Error happened: <br>";
      $errmsg.= "File: " . $arr["errors"][0]["file"] . "<br>";
      $errmsg.= "Line: " . $arr["errors"][0]["line"] . "<br>";
      $errmsg.= "Description: " . $arr["errors"][0]["description"] .
"<br>";
      $result["txt"] = $errmsg;
   }
}
```

and to the **Client After tab** tab:

```
var message = result["txt"] + " !!!";
ctrl.setMessage(message);
```

After selecting record(s) on the List page and clicking the "Email selected" button, you will see a popup window where you can enter From, Subject and Body parameters of your letter. Click "Save" button to send an email to selected user(s).

Send an email with updated fields only

To send an email with updated fields only use the following code in Edit page: Before record updated event.

Note: Change the values listed in red to match your specific needs.

```
$body = "Changed fields \n";
foreach ($values as $key => $value)
{
   if ($value!=$oldvalues[$key])
      $body .= $key . ": " . $value . "\n";
}
$email="test@test.com";
$subject="Sample subject";
runner_mail(array('to' => $email, 'subject' => $subject, 'body' => $body));
```

Send mass email to all users

To send an email with new data to email addresses from user table use the following code in one of the following events:

- Add page: Before record added;
- Add page: After record added;
- Edit page: Before record updated;

- Edit page: After record updated.

Note: Change the values listed in red to match your specific needs.

```
global $dal;
//select emails from Users table
$tblUsers = $dal->Table("UsersTableName");
$rs = $tblUsers->QueryAll();
while ($data = db_fetch_array($rs))
{
  $email=$data["EmailAddress"];
  $from="admin@test.com";
  $msg="Check what's hot this season";
  $subject="Monthly newsletter";
  $ret=runner_mail(array('to' => $email, 'subject' => $subject,
    'body' => $msg, 'from'=>$from));
  if(!$ret["mailed"])
    echo $ret["message"]."<br>";
}
```

For more information about using Data Access Layer (DAL), see Data Access Layer.

Send an email with attachment from the database

To send an email with attachments stored in database field use the following code in events like BeforeAdd/AfterAdd/BeforeEdit/AfterEdit. Make sure to replace "Field that stores attachments" with the actual field name.

Note: Change the values listed in red to match your specific needs.

Please note that attachments will only work when you selected "Use custom mailer server settings" under "Email settings".

```
$email= "test@gmail.com" ;
$msg="File Attached";
$subject="Attached some files";

$fileArray = my_json_decode($values["Field that stores attachments"]);
$attachments = array();
$msg = "";
foreach($fileArray as $f)
{
   $attachments[] = array("path" => $f["name"]);
}

$msg.= "Some field: ".$values["Some field"]."\r";
$msg.= "Some other field: ".$values["Some other field"]."\r";
```

```
$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, "attachments" => $attachments));
if(!$ret["mailed"])
    echo $ret["message"];
```

3.1.2.4 Upload

Rename uploaded files

Each uploaded to the disk file has two names: display name, that you can see on the site, and physical file name on the disk. Let's study how to rename both display and physical file names. Use the following code in Add page: Before record added and/or Edit page: Before record updated events.

Note: Change the values listed in red to match your specific needs.

1. Rename display name of uploaded file

```
// get information about uploaded files
$fileArray = my_json_decode($values["fieldname"]);

// rename each file. In this example - convert to lowercase.
for($i = 0; $i < count($fileArray); $i++)
{
    $fileArray[$i]["usrName"] = strtolower($fileArray[$i]["usrName"]);
}

// update values of the field that stores file names
$values["fieldname"] = my_json_encode($fileArray);
```

In this example *fieldname* stores names of uploaded files.

2. Rename physical name of uploaded file

This code snippet changes the file names to the *LastName* field values.

```
// get information about uploaded files
$fileArray = my_json_decode($values["fieldname"]);

// rename files
for($i = 0; $i < count($fileArray); $i++)
{
    $fileName = $fileArray[$i]["name"];
    $newFileName = "files/".$values["LastName"].$i.".jpg";
    rename($fileName, getabspath($newFileName));
    $fileArray[$i]["name"] = $newFileName;
}

// update values of the field that stores file names
```

```
$values["fieldname"] = my_json_encode($fileArray);
```

In this example *fieldname* stores names of uploaded files.

3.1.2.5 Misc

Check if start date is earlier than end date

On the Edit page verify that the start date is earlier than the end date. Use the following code in the Edit page: Before record updated event.

Note: Change the values listed in red to match your specific needs.

```
if (toPHPTime($values["start_date"]) > toPHPTime($values["end_date"]))
{
   $message = "Start date can not be later than End date.";
   return false;
}
else
{
   return true;
}
```

Redirect to user info edit page

To redirect to the user info edit page use the following code in Login page: After successful login event.

Note: Change the values listed in red to match your specific needs.

```
global $dal;
$tblUsers = $dal->Table("UserTableName");
$rs = $tblUsers->Query("UserNameField='".$username."' and
  PasswordField='".$password."'","");
$data = db_fetch_array($rs);
header("Location: tablename_edit.php?editid1=".$data["IDField"]);
exit();
```

For more information about using Data Access Layer (DAL), see Data Access Layer.

Restrict access to PHPrunner application by IP address

Restricting or allowing access by IP address is an easy task. Here are a few examples. This code needs to be added to the beginning of AfterAppInit event. Please note that this tutorial uses IPv4 addresses.

1. Allow local access only:

```
if ($_SERVER['REMOTE_ADDR'] != '127.0.0.1') exit();
```

2. Allow access from the list of approved IP addresses:

```
$array = array('127.0.0.1', '96.24.12.122', '96.24.12.123');
if (!in_array($_SERVER['REMOTE_ADDR'], $array)) exit();
```

3. Restrict access from a certain IP address:

```
if ($_SERVER['REMOTE_ADDR'] == '123.123.123.123') exit();
```

4. Restrict access from the list of addresses:

```
$array = array('127.0.0.1', '96.24.12.122', '96.24.12.123');
if (in_array($_SERVER['REMOTE_ADDR'], $array)) exit();
```

Save user data in session variables

Often you need to save data from the login table in session variables for later use. For this purpose you can use AfterSuccessfulLogin event.

Note: Change the values listed in red to match your specific needs.

```
function AfterSuccessfulLogin($username, $password, $data)
{
  $_SESSION["FirstName"] = $data["FirstName"];
}
```

Then you can use `$_SESSION["FirstName"]` as default value of any field or in other events.

Speed up data entry using events

On the Add page you might want to populate some edit boxes with previously used values. This can be done using a combination of Default values and AfterAdd Event.

For example, at the end of day you need to enter several data records for each Employee. To populate the Date and Employee fields with the most recent values, you need to complete the following steps:

1. Set Default values of those fields to **$_SESSION["Date"]** and **$_SESSION["Employee"]** respectively. This can be done in "Edit as" settings dialog.

2. Create an AfterAdd event and use the following code to save Date and Employee values in Session variables:

```
$_SESSION["Date"] = $values["Date"];
$_SESSION["Employee"] = $values["Employee"];
```

3.1.3 Global events

3.1.3.1 Login page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before login

Description

Function **BeforeLogin** is executed on the login page before verifying username and password.

Syntax

```
BeforeLogin($username,$password,$message,$pageObject)
```

Arguments

$username - user-entered login name.

$password - user-entered password.

$message - if the function return false, place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: login will be validated against the login table.

False: invalid login.

Applies to pages

Login

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After successful login

Description

Function **AfterSuccessfulLogin** is executed on the login page after username and password successfully verified.

Syntax

```
AfterSuccessfulLogin($username,$password,$data)
```

Arguments

$username - user-entered login name.

$password - user-entered password.

$data - array with existing user record in the login table. Access fields by *$data["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work.

In case of Active Directory based security data array contains attributes from user record in AD. For instance, this is how you can retrieve user's email and store it in a session variable:

```
$_SESSION["UserEmail"]=$data["email"];
```

Applies to pages

Login

Recommended sample events

Add link to user profile to the menu

Change 'Logged on as' message

Redirect to user info edit page

Save user data in session variables

After unsuccessful login

Description

Function **AfterUnsuccessfulLogin** is executed on the login page after failed authentication.

Syntax

```
AfterUnsuccessfulLogin($username,$password,$message,$pageObject)
```

Arguments

$username - user-entered login name.

$password - user-entered password.

$message - place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Login

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After Logout

Description

The event fires when user presses the Logout button.

This event handler will not be called if the user closes the browser or the sessions times out.

Syntax

```
AfterLogout($username)
```

Arguments

$username - the login name of the user having logged out.

Return value

no return value

Applies to pages

Login

See also

Redirect to another page

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.3.2 Menu page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.3.3 Register page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before registration

Description

Function **BeforeRegister** is executed on the registration page.

Syntax

```
BeforeRegister($userdata,$message,$pageObject)
```

Arguments

$userdata - array that stores values entered on Registration page. To access specific field value use *$userdata["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$userdata["PlayerId"].* Note that *$userdata["playerid"]* or *$userdata["PlayerID"]* will not work*.*

$message - place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: continue with registration.

False: prohibit registration.

Applies to pages

Registration

Recommended predefined actions and sample events

Send a simple email

Send an email with new data

Save new data in another table

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After successful registration

Description

Function **AfterSuccessfulRegistration** is executed on the registration page after new user data was added to the database.

Syntax

```
AfterSuccessfulRegistration($userdata,$pageObject)
```

Arguments

$userdata - array that stores values entered on Registration page. To access specific field value use *$userdata["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$userdata["PlayerId"].* Note that *$userdata["playerid"]* or *$userdata["PlayerID"]* will not work*.*

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Registration

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After unsuccessful registration

Description

Function **AfterUnsuccessfulRegistration** is executed on the registration page if new user record wasn't created.

Syntax

```
AfterUnsuccessfulRegistration($userdata,$message,$pageObject)
```

Arguments

$userdata - array that stores values entered on Registration page. To access specific field value use *$userdata["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$userdata["PlayerId"].* Note that *$userdata["playerid"]* or *$userdata["PlayerID"]* will not work*.*

$message - place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Registration

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.3.4 Change password page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before change password

Description

Function **BeforeChangePassword** is executed on the change password page after password was changed.

Syntax

```
BeforeChangePassword($oldpassword,$newpassword,$pageObject)
```

Arguments

$oldpassword - old password.

$newpassword - new password.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: allow password to be changed.

False: do not allow password to be changed.

Applies to pages

Change Password

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After password changed

Description

Function **AfterChangePassword** is executed on the change password page after password was changed.

Syntax

```
AfterChangePassword($oldpassword,$newpassword,$pageObject)
```

Arguments

$oldpassword - old password.

$newpassword - new password.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Change Password

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.3.5 Remind password page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before password reminder sent

Description

Function **BeforeRemindPassword** is executed on the password reminder page before password reminder is sent to user email.

Syntax

```
BeforeRemindPassword($username,$email,$pageObject)
```

Arguments

$username - username entered by the user.

$email - email entered by the user.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: continue with password reminder.

False: cancel password reminder procedure.

Applies to pages

Password Reminder

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After password reminder sent

Description

Function **AfterRemindPassword** is executed on the password reminder page after password reminder is sent to user email.

Syntax

```
AfterRemindPassword($username,$email,$pageObject)
```

Arguments

$username - username entered by the user.

$email - email entered by the user.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Password Reminder

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.3.6 After application initialized

Description

Function **AfterAppInit** is executed in the beginning of each page before any processing takes place. Use this event to override any global PHPRunner variables. See help area in Event editor for the list of available global variables.

Note: It's not recommended to display anything on the web page from this event. This may break your application.

Syntax

```
AfterAppInit()
```

Applies to pages

All pages.

Example

Lets say you need to troubleshoot your application displaying executed SQL query on the top of the page. Usually you proceed to *include/appsettings.php* file and set **$dDebug** variable to **true** though changing this variable back and forth is a little or no joy.

Using AfterAppInit event you can display debug info simply adding **debug=true** to the URL.

AfterAppInit code:

```
if ($_REQUEST["debug"]=="true")
  $dDebug=true;
```

Sample URL: categories_list.php?debug=true

Note

Database connection is not yet open in this event. If you need to perform any database operations open database connection manually.

Example:

```
$myconn=db_connect();
$rs=db_query("select * from users where session_id='" .
$_SESSION["sessionid"]. "'",$myconn);
$data=db_fetch_array($rs);
if($data)
{
   $_SESSION["Avatar"] = $data["avatar"];
}
```

Recommended sample events

Restrict access to PHPRunner application by IP address

3.1.3.7 Menu item: Modify

Description

Function **ModifyMenuItem** is executed for each Menu item before a page is displayed in the browser. Use this event to modify or hide menu items.

Syntax

```
ModifyMenuItem($menuItem)
```

Arguments

$menuItem - menu item object.

Return value

True: menu item is shown.

False: menu item is hidden.

Methods

- *getLinkType()* - get the link type. Link types are: **Internal** (link to a page generated by PHPRunner, e.g. List, Print etc.), **External** (link to any external web page), **None** (menu item is not a link, e.g. group or separator).
- *getUrl()* - get the URL of external link.
- *setUrl($url)* - set the link URL and make it external.
- *getParams()* - get parameters of internal link.
- *setParams($params)* - set parameters of internal link. These parameters may be also set on the **Choose page** screen using '...' button next to the **List page**. Parameters are concatenated with the link. E.g. if parameters are *foo=bar&bar=foo*, the link will be *..._list.php?foo=bar&bar=foo*.
- *setTitle($title)* - set the link title.
- *getTitle()* - get the link title.
- *getTable()* - get the table name, internal link points to.
- *setTable($table)* - set the table name.
- *getPageType()* - get the page type (List, Add, etc.).
- *setPageType($pType)* - set the page type (List, Add, Search, Print, Report, Chart).

Applies to pages

All pages with *Menu* element. Insert your code into the *Menu Item: Modify* event.

Adding parameters to menu item

```
                                                    Menu Item: Modify event
```

```
if ($menuItem->getLinkType() == 'External')
   {
      $menuItem->setUrl('http://localhost/mn1/carsmodels_list.php');
   }
   else if($menuItem->getLinkType() == 'Internal')
   {
      $menuItem->setParams('id=30');
      if ($menuItem->getTable() == 'carsmake')
      {
         $menuItem->setTable('carsmodels');
      }
   }
   else
   {
      return false;
   }
   return true;
```

Hide some menu items based on the logged user

Hide some menu items based on the application users. If the menu item is the link to internal application page, you can assign the table permissions. However, if the menu item is an external link the permissions will have to be set in *Menu Item: Modify* event.

Menu Item: Modify event

```
if ($_SESSION["GroupID"]!="manager")
{
   $title = $menuItem->getTitle();
   if ($title=="Yahoo Finance")
      return false;
}
return true;
```

Display record counter next to each menu item

Display the record counter for each menu item. The code will validate if the menu item is an internal table or view and will concatenate the number of records with the menu item name.

Menu Item: Modify event

```
if($menuItem->getLinkType() == 'Internal')
{
  global $tables_data;
  $table=$menuItem->getTable();
  include_once(getabspath("include/".$table."_settings.php"));
  $ps = new ProjectSettings($table);
  $table= $ps->getOriginalTableName();
```

```
    $rs=DB::Query("select count(*) as c from " . AddTableWrappers($table));
    $data = $rs.fetchAssoc();
    $menuItem->setTitle($menuItem->getTitle() . " (". $data["c"] . ")");
}
return true;
```

Hide some menu items in Mobile mode

```
    if ($menuItem->getTable() == 'Cars' && MobileDetected()) {
        return false;
    }
    return true;
```

Recommended predefined actions and sample events

- Check to see if a specific record exists

3.1.3.8 Before audit log

Description

Function **OnAuditLog** is executed before a record is added to the log.

Syntax

```
OnAuditLog($action,$params,$table,$keys,$newvalues,$oldvalues)
```

Arguments

$action - Action occured.

$params - $params[0]: user IP address, $params[1]: username.

$table - Table being modified.

$keys - Array of key column values pointing to the current record.

$newvalues - Array of field values being added to the database.

$oldvalues - Array of previous field values. Applies to the Edit and Delete functions.

Return value

True: to save the action in the log.

False: to not save the action in the log.

Applies to pages

All pages. Insert your code into the *Before audit log* event.

Example

If you do not want to record actions done by the admin in the audit log, you can use the following code:

Before audit log event

```
if ($params[1]=="admin")
   return false;

return true;
```

3.1.4 Table events

3.1.4.1 Add page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Copy page: OnLoad

Description

Function **CopyOnLoad** is executed when Add page is loaded in Copy mode.

Syntax

```
CopyOnLoad($values,$where,$pageObject)
```

Arguments

$values - array of values to be displayed on the page. You can modify the values of this array. To access specific field value use *$values["FieldName"]*. Example: to clear the *PassportNumber* field value before copying, use *$values["PassportNumber"]="";*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$where - WHERE clause that points to the record to be copied. Example: ID=19.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Add (in Copy mode)

Recommended predefined actions and sample events

Send a simple email

Send an email with old data

Save old data in another table

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

Before record added

Description

Function **BeforeAdd** is executed before a record is physically added to the database. Will work in all add modes: Inline Add, Regular Add and Add page in popup.

Syntax

```
BeforeAdd($values,$message,$inline,$pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$message - place the message to be displayed into this variable.

$inline - equals to *true* when the Inline Edit in process, *false* otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: continue with adding a record.

False: record would not be added.

Applies to pages

Add, Inline Add

Recommended sample events

Add custom field to form

Limit number of records users can add

Select multiple values from checkboxes or a list field and have them appear as individual database entries

Update multiple tables

Send mass email to al users

Rename uploaded files

Speed up data entry using events

Custom add

Description

Function **CustomAdd** is executed before a record is physically added to the database. It is designed to replace the standard Add procedure. Use it when you do not want record to be added to the table in question.

Syntax

```
CustomAdd($values, $keys, $error, $inline, $pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$keys - array of key column values that point to the edited record. To access specific key column use *$keys["KeyFieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$keys["PlayerId"].* Note that *$keys["playerid"]* or *$keys["PlayerID"]* will not work*.*

$error - place the message to be displayed into this variable.

$inline - equals to *true* when the Inline Add in process, *false* - otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: if you want the application to add the record for you.

False: if you have added the record in your code.

Applies to

Add, Inline Add.

Example

Lets consider the situation when records are never added directly to the main table (e.g. *Cars*). Instead, records are added to the temporary *TempCars* table and then moved to the main *Cars* table once approved by admin. In this case the following code in **CustomAdd** event will do the job:

```
global $dal;
$tblTempCars = $dal->Table("TempCars");
$tblTempCars->Value["make"]=$values["make"];
$tblTempCars->Value["model"]=$values["model"];
$tblTempCars->Value["yearOfMake"]=$values["yearOfMake"];
$tblTempCars->Add();
return false;
```

You may notice that BeforeAdd event does the similar job. The main difference is that when you return *false* in **BeforeAdd** event this situation is considered as error and user will understand that something went wrong. On the other side, returning *false* in **CustomAdd** event is perfectly legitimate and application execution continues after that.

After record added

Description

Function **AfterAdd** is executed after a record is physically added to the database. Will work in all add modes: Inline Add, Regular Add and Add page in popup.

Syntax

```
AfterAdd($values,$keys,$inline,$pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$keys - array of key column values that point to the new record. To access specific key column use *$keys["KeyFieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$keys["PlayerId"].* Note that *$keys["playerid"]* or *$keys["PlayerID"]* will not work.

$inline - equals to *true* when the Inline Edit in process, *false* otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

If you need to display a message on the page or pass a variable value to JavaScript you need to add the following to the end of your event code:

```
$pageObject->stopPRG = true;
```

A complete code example. Passing the value of true to JavaScript variable named "saved":

```
$pageObject->setProxyValue('saved', true);
$pageObject->stopPRG = true;
```

Applies to pages

Add, Inline Add

Recommended sample events

Add new button to Add/Edit pages

Change message after record was added or saved

Redirect to details page after master record was added

Send mass email to al users

Speed up data entry using events

Process record values

Description

Function **ProcessValues<PageName>** is executed before the record is displayed. Use this event to modify the displayed field values.

Syntax

```
ProcessValues<PageName>($values,$pageObject)
```

Arguments

$values - array of values to be displayed on the page. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

No return value.

Applies to pages

View, Edit

Example

Display empty *"Comment"* field when user edits a record:

```
$values["Comment"]="";
```

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.2 Edit page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before record updated

Description

Function **BeforeEdit** is executed before a data record is updated in the database. Will work in all edit modes: Inline Edit, Regular Edit and Edit page in popup.

Syntax

```
BeforeEdit($values,$where,$oldvalues,$keys,$message,$inline,$pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$where -WHERE clause that points to the edited record. Example: ID=19.

$oldvalues - array with existing field values. To access specific column value use *$oldvalues["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$oldvalues["PlayerId"].* Note that *$oldvalues["playerid"]* or *$oldvalues["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$oldvalues* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$oldvalues["Inv_Salesrep"]*.

$keys - array of key column values that point to the edited record. To access specific key column use *$keys["KeyFieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$keys["PlayerId"].* Note that *$keys["playerid"]* or *$keys["PlayerID"]* will not work*.*

$message - place the message to be displayed into this variable.

$inline - equals to *true* when the Inline Edit in process, *false* otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: changes will be saved.

False: changes would not be saved.

Applies to pages

Edit, Inline Edit

Recommended sample events

Update multiple tables

Send an email with updated fields only

Send mass email to al users

Rename uploaded files

Check if start date is ealier than end date

Custom record update

Description

Function **CustomEdit** is executed before a data record is updated in the database. It is designed to replace the standard Edit procedure. Use it when you do not want record to be updated in the table in question.

Syntax

```
CustomEdit($values, $where, $oldvalues, $keys, $error, $inline,
$pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$where -WHERE clause that points to the edited record. Example: ID=19.

$oldvalues - array with existing field values. To access specific column value use *$oldvalues["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$oldvalues["PlayerId"].* Note that *$oldvalues["playerid"]* or *$oldvalues["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$oldvalues* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$oldvalues["Inv_Salesrep"]*.

$keys - array of key column values that point to the edited record. To access specific key column use *$keys["KeyFieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$keys["PlayerId"].* Note that *$keys["playerid"]* or *$keys["PlayerID"]* will not work*.*

$error - place the message to be displayed into this variable.

$inline - equals to *true* when the Inline Edit in process, *false* otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True: if you want the application to update the record for you.

False: if you have updated the record in your code.

Applies to

Edit, Inline Edit.

Example

Lets consider the situation when records are never edited directly in the main table (e.g. *Cars*). Instead, records are added to the temporary *TempCars* table and then moved to the main *Cars* table once approved by admin. In this case the following code in **CustomAdd** event will do the job:

```
global $dal;
$tblTempCars = $dal->Table("TempCars");
$tblTempCars->Value["make"]=$values["make"];
$tblTempCars->Value["model"]=$values["model"];
$tblTempCars->Value["yearOfMake"]=$values["yearOfMake"];
$tblTempCars->Add();
return false;
```

Process record values

Description

Function **ProcessValues<PageName>** is executed before the record is displayed. Use this event to modify the displayed field values.

Syntax

```
ProcessValues<PageName>($values,$pageObject)
```

Arguments

$values - array of values to be displayed on the page. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

No return value.

Applies to pages

View, Edit

Example

Display empty *"Comment"* field when user edits a record:

```
$values["Comment"]="";
```

After record updated

Description

Function **AfterEdit** is executed after a data record was updated in the database. Will work in all edit modes: Inline Edit, Regular Edit and Edit page in popup.

Syntax

```
AfterEdit($values,$where,$oldvalues,$keys,$inline,$pageObject)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$where -WHERE clause that points to the edited record. Example: ID=19.

$oldvalues - array with existing field values. To access specific column value use *$oldvalues["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$oldvalues["PlayerId"].* Note that *$oldvalues["playerid"]* or *$oldvalues["PlayerID"]* will not work.

If the field was assigned an alias in the SQL query, then the *$oldvalues* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$oldvalues["Inv_Salesrep"]*.

$keys - array of key column values that point to the edited record. To access specific key column use *$keys["KeyFieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$keys["PlayerId"].* Note that *$keys["playerid"]* or *$keys["PlayerID"]* will not work.

$inline - equals to true when the Inline Edit in process, false otherwise.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

If you need to display a message on the page or pass a variable value to JavaScript you need to add the following to the end of your event code:

```
$pageObject->stopPRG = true;
```

A complete code example. Passing the value of true to JavaScript variable named "saved":

```
$pageObject->setProxyValue('saved', true);
$pageObject->stopPRG = true;
```

Applies to pages

Edit, Inline Edit

Recommended sample events

Add new button to Add/Edit pages

Change message after record was added or saved

Send mass email to al users

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$values,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$values - array of displayed record contents. You can read and write to the array. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

View, Edit

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

Hide empty fields on View page

Before display event for all pages except View/Edit

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.3 List page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before record deleted

Description

Function **BeforeDelete** is executed before a record is deleted.

Syntax

```
BeforeDelete($where,$deleted_values,$message,$pageObject)
```

Arguments

$where - WHERE clause that points to the record to be deleted. Example: ID=19.

$deleted_values - array with the field values from the record to be deleted. To access specific field value use *$deleted_values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$deleted_values["PlayerId"].* Note that *$deleted_values["playerid"]* or *$deleted_values["PlayerID"]* will not work*.*

$message - place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True - record will be deleted.

False - record would not be deleted.

Applies to pages

List

Recommended sample events

Before deleting a record check if related records exist

Update multiple records on the List page

After record deleted

Description

Function **AfterDelete** is executed after a record was deleted.

Syntax

```
AfterDelete($where,$deleted_values,$message,$pageObject)
```

Arguments

$where - WHERE clause that points to the record has been deleted. Example: ID=19.

$deleted_values - array with the field values from the record has been deleted. To access specific field value use *$deleted_values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$deleted_values["PlayerId"].* Note that *$deleted_values["playerid"]* or *$deleted_values["PlayerID"]* will not work*.*

$message - place the message to be displayed into this variable.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After group of records deleted

Description

Function **AfterMassDelete** is executed after bulk delete operation.

Syntax

```
AfterMassDelete($records_deleted,$pageObject)
```

Arguments

$records_deleted - number of records that were deleted.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

Before record processed

Description

Function **BeforeProcessRow<PageName>** is executed right after a database record was retrieved from the database and before formatting is applied.

Syntax

```
BeforeProcessRow<PageName>($data,$pageObject)
```

Arguments

$data - array of field values of the record being processed. To access specific field value use *$data["FieldName"]*. Note that you can read the *$data* array and also write to it changing the values before showing them on the page.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work*.*

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True - record will be shown on the page.

False - record will be skipped.

Applies to pages

List, Print

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After record processed

Description

Function **BeforeMoveNext<PageName>** is executed after record was processed and formatting applied.

Syntax

```
BeforeMoveNext<PageName>($data,$row,$record,$pageObject)
```

Arguments

$data - array of field values of the record being processed. To access specific field value use *$data["FieldName"]*. Note that you can only read the *$data* array.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work.

$row - array representing a row on the page.

$record - array representing a table record on the page.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Print

Recommended sample events

Change cell background color

Change row background color

Disable record editing

Hide buttons in some rows of datagrid

See also

- Conditional formatting

Before SQL query

Description

Function **BeforeQuery<PageName>** is executed before SELECT SQL query is processed. Use this event if you like to modify default SQL query, add a WHERE clause etc.

Syntax

```
BeforeQuery<PageName>($strSQL,$strWhereClause,$strOrderBy,$pageObject)
```

Arguments

$strSQL - SQL query to be executed.

$strWhereClause - WHERE clause applied to the SQL query.

$strOrderBy - ORDER BY query applied to the SQL query.

$pageObject - an object representing the current page.

Applies to pages

List, Print, Export, Chart

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

Get Row Count

Description

ListGetRowCount event is executed before List page is processed. Use this function when your database doesn't support record count.
You need to use this event together with ListFetchArray and ListQuery events. Read about it in this article How to display data returned by stored procedure

Syntax

```
ListGetRowCount($searchObj,$masterTable,$masterKeysReq,$selectedRecords,$
pageObject)
```

Arguments

$searchObj - Instance of a class which performes search.

$masterTable - Master table name.

$masterKeysReq - Array of keys.

$selectedRecords - Array of selected records on Print Page, null on List Page.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

Function returns number of records or *false*.

Applies to pages

List, Print

Examples

1. Return hardcoded number of records

```
return 10;
```

2. Calculate and return number of records

```
return DBLookup("select count(*) from test");
```

Custom Query

Description

ListQuery event is executed before List page is processed. Use this event when data set is more complicated than result of SQL query.

Here is the article that shows how to use ListFetchArray and ListQuery events together: How to display data returned by stored procedure

Syntax

```
ListQuery($searchObj,$orderBy,$howOrderBy,$masterTable,$masterKeysReq,$se
lectedRecord,$pageSize,$myPage,$pageObject)
```

Arguments

$searchObj - Instance of a class which performes search.

$orderBy - Array with field order.

$howOrderBy - Array with sort type for 'orderBy' array.

$masterTable - Master table name.

$masterKeysReq - Array of keys.

$selectedRecord - Array of selected records on Print Page, null on List Page.

$pageSize - Number of records per page.

$myPage - Current page number.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

Function returns data array or *false*.

Applies to pages

List, Print

Custom record fetch

Description

Function **ListFetchArray** fetches records from the given array and returns them as array.

Use this function in conjunction with Custom query function.

Here is the article that shows how to use ListFetchArray and ListQuery events together:

How to display data returned by stored procedure

Syntax

```
ListFetchArray($rs,$pageObject)
```

Arguments

$rs - array of values returned from Custom query event.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Print

3.1.4.4 Report page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

Before SQL query

Description

Function **BeforeQuery<PageName>** is executed before SELECT SQL query is processed. Use this event if you like to modify default SQL query, add a WHERE clause etc.

Syntax

```
BeforeQuery<PageName>($strSQL,$strWhereClause,$strOrderBy,$pageObject)
```

Arguments

$strSQL - SQL query to be executed.

$strWhereClause - WHERE clause applied to the SQL query.

$strOrderBy - ORDER BY query applied to the SQL query.

$pageObject - an object representing the current page.

Applies to pages

List, Print, Export, Chart

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.5 Chart page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

Before SQL query

Description

Function **BeforeQuery<PageName>** is executed before SELECT SQL query is processed. Use this event if you like to modify default SQL query, add a WHERE clause etc.

Syntax

```
BeforeQuery<PageName>($strSQL,$strWhereClause,$strOrderBy,$pageObject)
```

Arguments

$strSQL - SQL query to be executed.

$strWhereClause - WHERE clause applied to the SQL query.

$strOrderBy - ORDER BY query applied to the SQL query. Example:

$pageObject - an object representing the current page.

Applies to pages

List, Print, Export, Chart

ChartModify

Quick jump

- Add horizontal scroller
- Add horizontal scroller and set initial zoom to 50%
- Change labels color and font size
- Series appearance
- Change colors in pie/doughnut chart
- Customize chart title
- Values formatting
- Multiple Y-axis
- Adding range markers
- Create two series chart with two Y-Axis

Description

Function **ChartModify** occurs before chart is displayed. Use this event to modify chart settings.

Syntax

```
ChartModify(chart,proxy,pageObj)
```

Arguments

chart - Chart object.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

PHP (BeforeDisplay event):

$pageObject->setProxyValue("name", $_SESSION["UserID"]);

Javascript (ChartModify event)

alert(proxy.name);

pageObj - RunnerPage object

Applies to pages

Chart

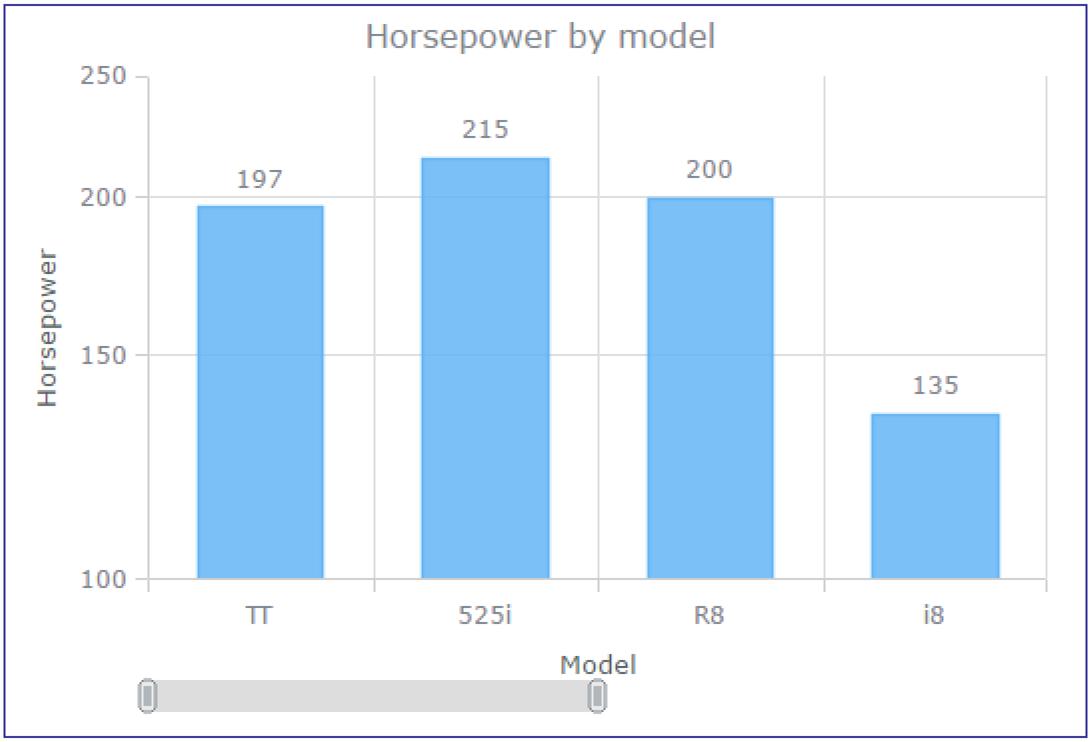
Official AnyChart documentation: https://api.anychart.com/

Below are a few examples of how you can use **ChartModify** event.

Example 1. Add horizontal scroller.

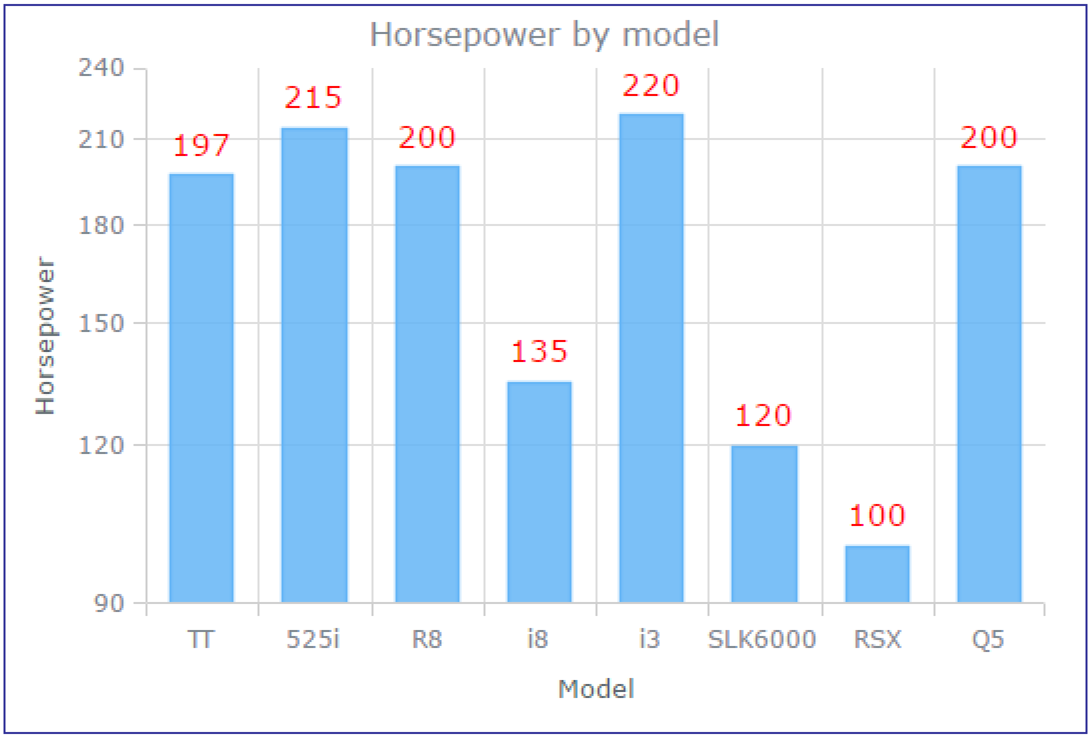
```
var currentScroller = chart.xScroller();
currentScroller.enabled(true);
```

Example 2. Add horizontal scroller and set initial zoom to 50%.



```
var zoom = chart.xZoom();
zoom.setTo(0, 0.5);

var currentScroller = chart.xScroller();
currentScroller.enabled(true);
```

Example 3. Change labels color and font size.



Using separate API calls to set series color and font size:

```
// Gets series by index, 0 - first series, 1 -second series etc
var series = chart.getSeriesAt(0);
series.labels(true);
series.labels().fontSize(15);
series.labels().fontColor("#ff0000");
```

Setting several labels parameters in one go:

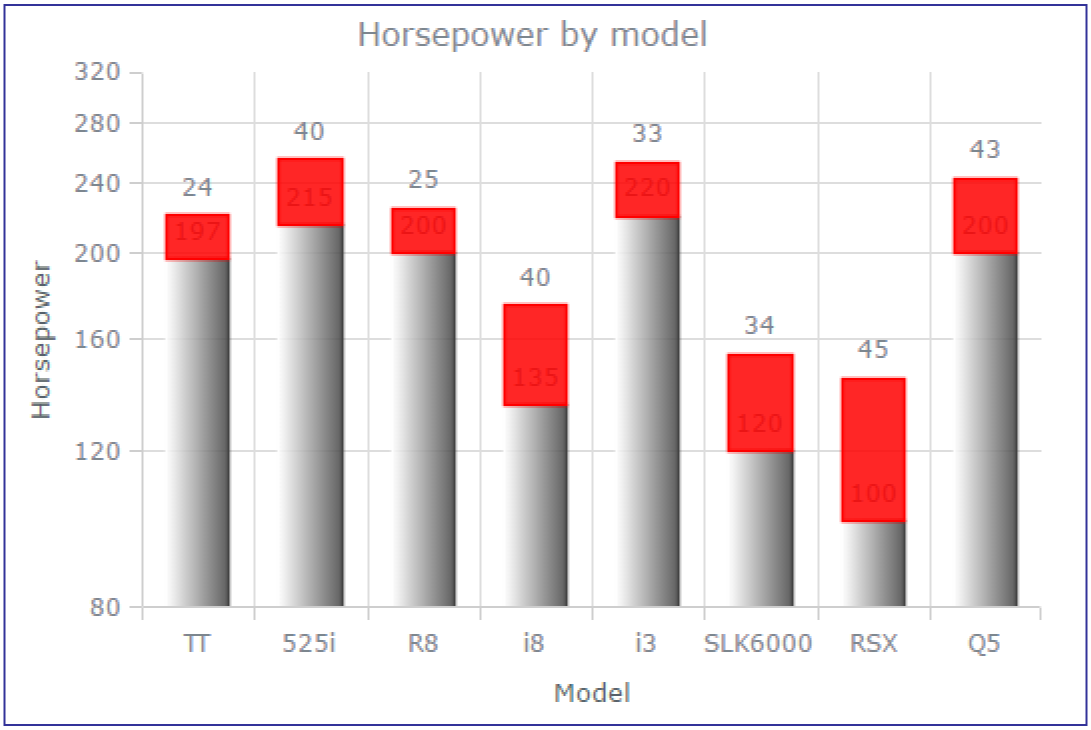
```
// Gets series by index, 0 - first series, 1 -second series etc
var series = chart.getSeriesAt(0);
series.labels(true);
series.labels.{fontSize: 15, fontColor: "#ff0000"});
```

Example 4. Series appearance.

Single color:

```
// Gets series by index, 0 - first series, 1 -second series etc
var series = chart.getSeriesAt(0);
series.color("#FF0000", 0.25);
```

Gradient fill:



```
// Gets series by index, 0 - first series, 1 -second series etc
var series = chart.getSeriesAt(0);
series.color(["#FEFEFE", "#424242"], 0.69, 0.59);
```

Example 5. Customize chart title.

Change title color:

```
chart.title().fontColor("#FF0000");
```

Change title and color:

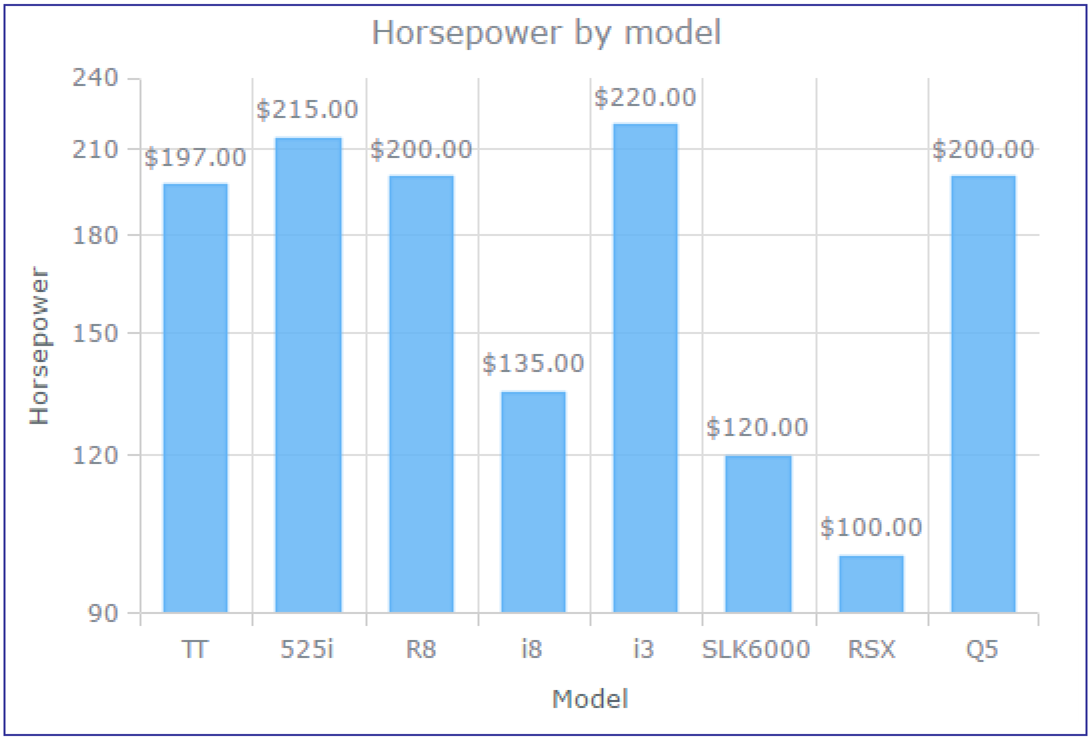
```
chart.title({text: "Custom title", fontColor: "#F44336"});
```

Disable title:

```
chart.title(false);
```

Example 6. Values formatting.

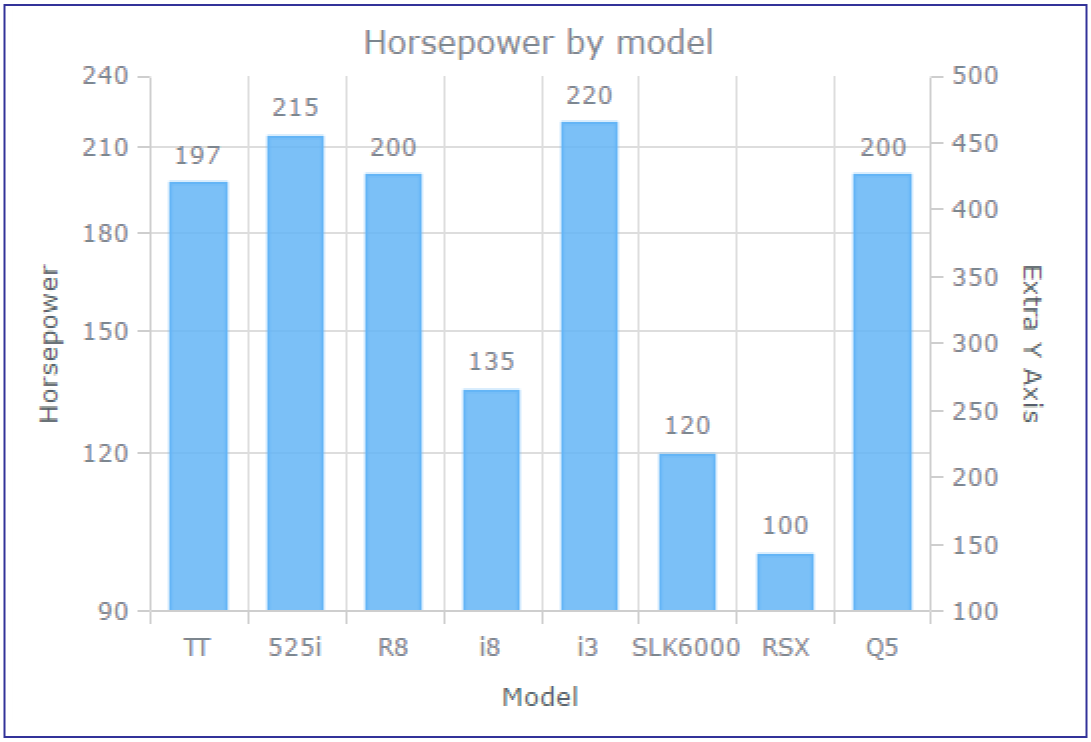
Formatting values as currency. You can use any Javascript code to format values.



```
var series = chart.getSeriesAt(0);
series.labels().textFormatter(function(){
        var num = Number(this.value);
    return(("$"+num.toFixed(2)));
});
```

Example 7. Multiple Y-axis.

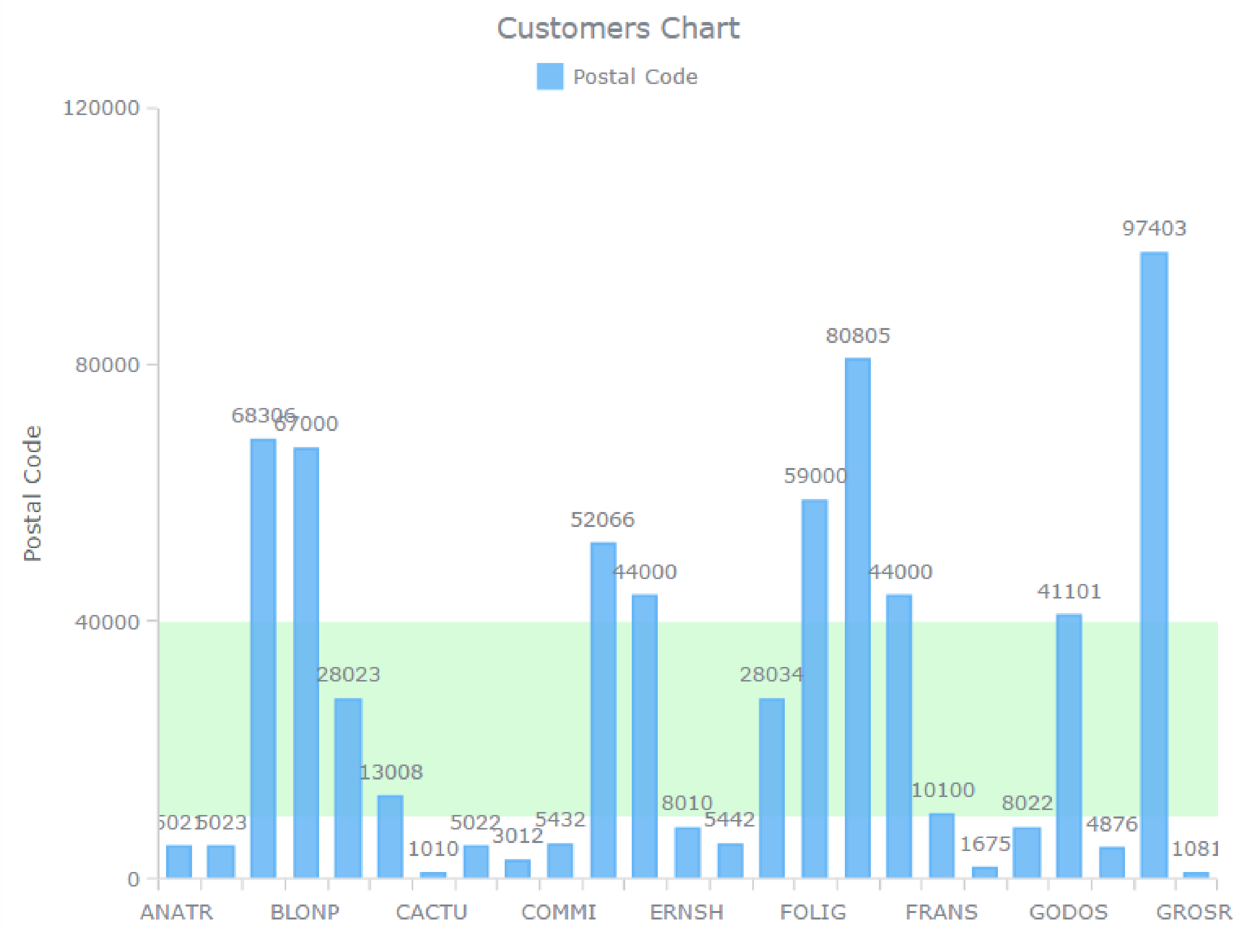
Here is how to add an extra Y-axis on the right ranging from 100 to 500 with ticks after each 25 points:



```
var extraYScale = anychart.scales.linear();
extraYScale.minimum(100);
extraYScale.maximum(500);
var extraTicks = extraYScale.ticks();
extraTicks.interval(25);

// Create and tune additional y axis
var extraYAxis = chart.yAxis(1);
extraYAxis.orientation("right");
extraYAxis.scale(extraYScale);
extraYAxis.title("Extra Y Axis");
```

Example 8. Adding range markers.

Here is how to add range markers:



```
var vMarker = chart.rangeMarker(0);
vMarker.from("10000");
vMarker.to("40000");
vMarker.axis(chart.yAxis());
vMarker.fill("#d7fcda");
```

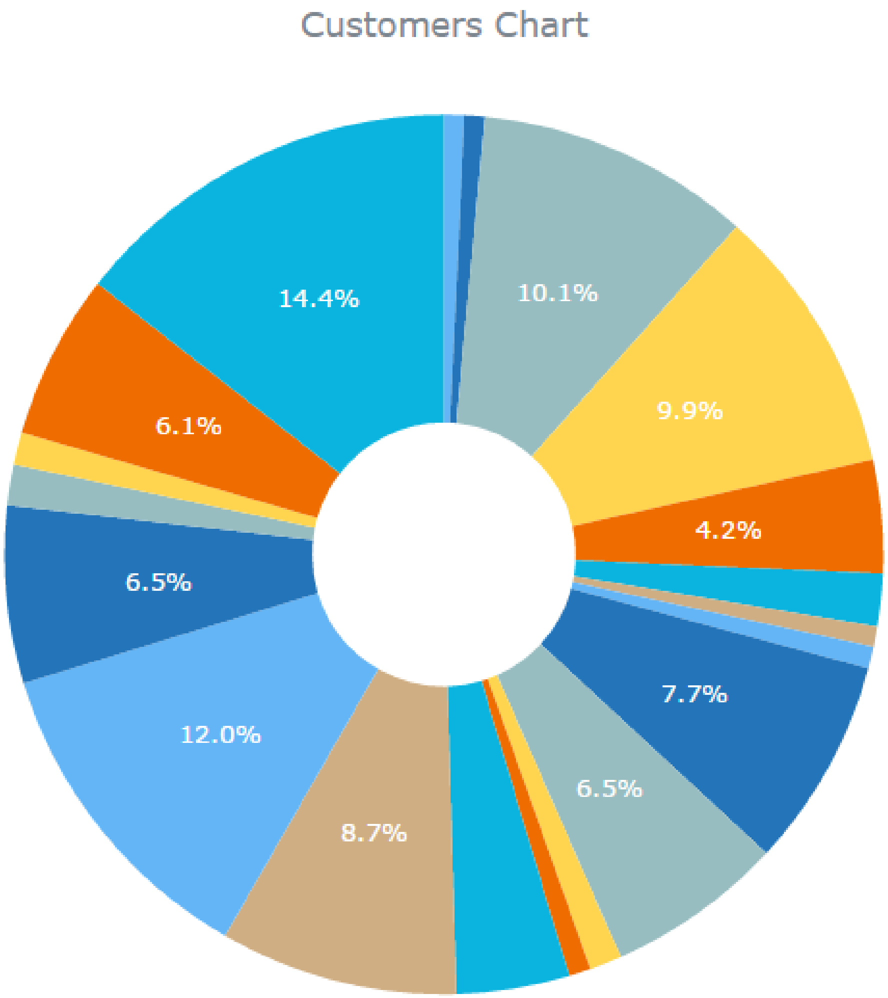
More info:

http://playground.anychart.com/docs/7.13.0/samples/AGST_Range_Marker_04-plain

Example 9. How to change colors of individual slices in pie/doughnut chart.

```
var palette = anychart.palettes.distinctColors();
palette.items(["#64B5F6", "#2374B8", "#97BDC1", "#FFD54F", "#EF6C00", "#0AB4DE",
"#CFAE83"]);
chart.palette(palette);
```

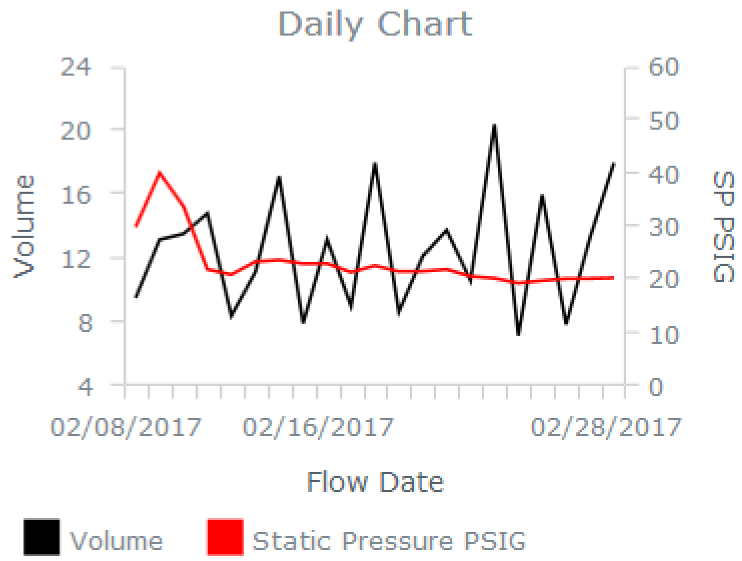
If your pie chart has more slices than colors you specify colors will be repeated.



Example 10. How to create two series chart with two Y-Axis.

An example of two series chart with two Y-Axis. The idea is to choose scales the way that makes both series clearly visible.

```
var series2 = chart.getSeriesAt(1);

var extraYScale = anychart.scales.linear();
extraYScale.minimum(0);
extraYScale.maximum(60);
var extraTicks = extraYScale.ticks();
extraTicks.interval(10);

var extraYAxis = chart.yAxis(1);
extraYAxis.orientation("right");
extraYAxis.scale(extraYScale);
extraYAxis.title("Title right axis");

series2.yScale(extraYScale);
```



JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.6 Printer-friendly page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before record processed

Description

Function **BeforeProcessRow<PageName>** is executed right after a database record was retrieved from the database and before formatting is applied.

Syntax

```
BeforeProcessRow<PageName>($data,$pageObject)
```

Arguments

$data - array of field values of the record being processed. To access specific field value use *$data["FieldName"]*. Note that you can read the *$data* array and also write to it changing the values before showing them on the page.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True - record will be shown on the page.

False - record will be skipped.

Applies to pages

List, Print

Recommended predefined actions and sample events

Send a simple email

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Redirect to another page

After record processed

Description

Function **BeforeMoveNext<PageName>** is executed after record was processed and formatting applied.

Syntax

```
BeforeMoveNext<PageName>($data,$row,$record,$pageObject)
```

Arguments

$data - array of field values of the record being processed. To access specific field value use *$data["FieldName"]*. Note that you can only read the *$data* array.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work.

$row - array representing a row on the page.

$record - array representing a table record on the page.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Print

Recommended sample events

Change cell background color

Change row background color

Disable record editing

Hide buttons in some rows of datagrid

See also

- Conditional formatting

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

Before SQL query

Description

Function **BeforeQuery<PageName>** is executed before SELECT SQL query is processed. Use this event if you like to modify default SQL query, add a WHERE clause etc.

Syntax

```
BeforeQuery<PageName>($strSQL,$strWhereClause,$strOrderBy,$pageObject)
```

Arguments

$strSQL - SQL query to be executed.

$strWhereClause - WHERE clause applied to the SQL query.

$strOrderBy - ORDER BY query applied to the SQL query.

$pageObject - an object representing the current page.

Applies to pages

List, Print, Export, Chart

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.7 View page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Process record values

Description

Function **ProcessValues<PageName>** is executed before the record is displayed. Use this event to modify the displayed field values.

Syntax

```
ProcessValues<PageName>($values,$pageObject)
```

Arguments

$values - array of values to be displayed on the page. To access specific field value use *$values["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work*.*

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

No return value.

Applies to pages

View, Edit

Example

Display empty *"Comment"* field when user edits a record:

```
$values["Comment"]="";
```

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$values,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$values - array of displayed record contents. You can read and write to the array. To access specific field value use *$values["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

View, Edit

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

Hide empty fields on View page

Before display event for all pages except View/Edit

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.8 Search page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

Before display

Description

Function **BeforeShow<PageName>** is executed right before a page is displayed in the browser. Use this event to modify the value of any template variable or to define a new one.

Syntax

```
BeforeShow<PageName>($xt,$templatefile,$pageObject)
```

Arguments

$xt - template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

$templatefile - name of the template file being displayed.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, Add, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Example

To display some text on the List page:

1. Proceed to the **Visual Editor** page.

2. Switch to HTML mode and find the line {END container_recordcontrols}. Add the following code right before it:

```
<DIV>{$new_variable}</DIV>
```

3. Add the following code to the **BeforeShow** event.

Note: Change the values listed in red to match your specific needs.

```
$new_variable = "test value";
$xt->assign("new_variable",$new_variable);
```

Recommended sample events

Hide controls on Add/Edit pages, based on logged user name

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

3.1.4.9 Import page

Before import started

Description

Function **BeforeImport** is executed before import is started.

Syntax

```
BeforeImport($pageObject, $message)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

$message - this message will be displayed on the page if import were cancelled

Applies to pages

Import

Recommended predefined actions and sample events

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

Before record inserted

Description

Function **BeforeInsert** is executed before a record is inserted. Use this event to modify the record before it is inserted.

Syntax

```
BeforeInsert($rawvalues, $values, $pageObject, $message)
```

Arguments

$rawvalues - raw field values from the imported file.

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

$message - this message will be added to the Import Log if record wasn't imported

Return value

True - record will be inserted.

False - record would not be inserted.

Applies to pages

Import

Recommended predefined actions and sample events

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

After import finished

Description

Function **AfterImport** is executed after import is finished.

Syntax

```
AfterImport($count,$skipCount,$pageObject)
```

Arguments

$count - number of records imported.

$skipCount - number of lines not imported.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

Import

Recommended predefined actions and sample events

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

3.1.4.10 Export page

Before process

Description

Function **BeforeProcess<PageName>** is executed before any processing takes places. Use this event to redirect user to another page, send an email or log user action in the database.

Syntax

```
BeforeProcess<PageName>($pageObject)
```

Arguments

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password, Menu

Recommended sample events

Add custom field to form

Hide repeating values on View page

Email selected records

JavaScript OnLoad

Description

Function **OnPageLoad** occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Syntax

```
OnPageLoad(pageObj,pageid,proxy,inlineRow)
```

Arguments

pageObj - RunnerPage object that represents a current page object.

pageid - page's unique numeric identifier.

proxy - data transferred from PHP code using $pageObject->setProxyValue function.

inlineRow - InlineRow object available in Add/Edit page events in the inline mode.

Applies to pages

List, View, Add, Edit, Print, Search, Export, Report, Chart, Login, Register, Password reminder, Change password

Examples

- **List page behavior**

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to refresh List page after Edit in popup

How to hide 'Edit selected'/'Delete selected' buttons

How to display all Options on Search panel

- **Add/Edit lists behavior**

How to ask for confirmation before saving the record

How to control multi-step pages

How to work with tabs

- **Form and controls**

Add custom field to form

How to show dropdown list of US states if US was selected in country list

How to enable/disable a button

- **Tweaking control appearance**

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

Change font size in text box

Change font in "edit" controls

How to convert input into upper case

Before record exported

Description

Function **BeforeOut** is executed before a record is exported. Use this event to modify the record before it is exported.

Syntax

```
BeforeOut($data,$values,$pageObject)
```

Arguments

$data - In Data. To access specific field value use *$data["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$data["PlayerId"].* Note that *$data["playerid"]* or *$data["PlayerID"]* will not work.

$values - Out Data. Array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$pageObject - an object representing the current page. For more information, see RunnerPage class.

Return value

True - record will be exported.

False - record would not be exported.

Applies to pages

Export

Recommended predefined actions and sample events

Insert a record into another table

Check to see if a specific record exists

Display a message on the Web page

3.1.4.11 After table initialized

Description

Function **AfterTableInit** is executed in the beginning of each page before any processing takes place, right after AfterAppInit. Use this event to override any table-specific PHPRunner variables. See help area in Event editor for the list of available table variables.

Note: It's not recommended to display anything on the web page from this event. This may break your application.

Syntax

```
AfterTableInit()
```

Applies to pages

All table specific pages like List, Print, Edit, Add, Export etc.

Example

To set default records per page value to 10:

```
$tdata<tableName>[".pageSize"] = 10;
```

Recommended sample events

[Dynamic SQL query]

[Add WHERE clause to the current SQL query]

[Replace WHERE clause of the current SQL query]

[Add a field name to the end of SELECT clause of the current SQL query]

[Remove a field name from the SELECT clause of the current SQL query]

[Replace field name in the SELECT clause of the current SQL query with new one]

3.1.4.12 Get Table Permissions

Description

Function **GetTablePermissions** occurs after table initialized. Use this event to return an access mask for the table:

A - Add;

D - Delete;

E - Edit;

S - List/View/Search;

P - Print/Export;

I - Import;

M - Admin option. User can see all records in the table.

Syntax

```
GetTablePermissions($permissions)
```

Arguments

$permissions - string containing the permissions calculated for a given user and table.

Return value

$permissions - string containing the permissions calculated for a given user and table.

Applies to pages

All table specific pages like List, Print, Edit, List, Add, Export etc.

Example

Prohibit the editing of table data on the weekend:

```
$dw = date( "w", $timestamp)
if ($dw==0 || $dw==6)
        $permissions="S";

return $permissions;
```

3.1.4.13 Is Record Editable

Description

Function **IsRecordEditable** occurs after table initialized. Use this event to implement custom edit permissions. This event is executed once for each record on the List page as well as on the Edit page. This event will be also called before record is deleted.

Syntax

```
IsRecordEditable($values,$isEditable)
```

Arguments

$values - array of values to be written to the database. To access specific field value use *$values["FieldName"]*.

> Field names are case-sensitive. If the field name is *PlayerId*, you should use *$values["PlayerId"].* Note that *$values["playerid"]* or *$values["PlayerID"]* will not work.
>
> If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep ...,* you should use *$values["Inv_Salesrep"]*.

$isEditable - editing flag calculated for the current record.

Return value

True - record is editable.

False - record is not editable.

Applies to pages

List, Edit

Example 1

Disable editing of data in certain table on weekends:

```
$dw = date("w", time());
$isEditable = $dw!=6 && $dw!=0;
return $isEditable;
```

Example 2

Enable editing of records only with odd IDs (1,3,5 ...):

```
if ($values["ID"] & 1)
  return false;
else
  return true;
```

3.1.5 Page life cycle overview

Quick jump	
Global Events	Login/Registration pages
List/Print pages	View/Search/Report/Chart pages
Edit/Add pages	Import page

When a PHPRunner-generated page runs, the page goes through a life cycle in which it performs a series of processing steps. These include initialization, retrieving data, instantiating controls and rendering. It is important to understand the page life cycle so that you can write code at the appropriate life-cycle stage for the effect you intend.

Global Events

AfterAppInit	Occurs in the beginning of each page before any processing takes place. Use this event to override any global PHPRunner variables. See help area in Event editor for the list of available global variables.

AfterTableInit	Occurs in the beginning of each page before any processing takes place, right after AfterAppInit. Use this event to override any table-specific PHPRunner variables. See help area in Event editor for the list of available table variables.
ModifyMenuItem	This event will be executed for each Menu item before a page is displayed in the browser. Use this event to modify or hide menu items.

List/Print pages

BeforeProcessList BeforeProcessPrint	Occurs when page processing starts and database connection is established. Use this event for the following: - change database connection to point to another database - read request data and populate session variables - redirect to another page
BeforeDelete	This event will be executed once for each record to be deleted. Use this event for the following: - prevent a record from being deleted - save deleted record in another table
AfterDelete	Occurs once for each record after it was deleted
AfterMassDelete	Occurs after bulk delete operation
BeforeProcessRowList BeforeProcessRowPrint	Occurs after database record is retrieved from the database before formatting is applied. Use this event for the following: - modify value of any column - prevent certain records from being displayed on the page - calculate/display your own totals and subtotals
BeforeMoveNextList BeforeMoveNextPrint	Occurs after database record is retrieved from the database and formatting is applied. Use this event for the following: - display an empty row or a header between group of records. - apply additional formatting to any column
BeforeShowList BeforeShowPrint	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following: - define a new template variable or change the value of existing one

	- display different template
OnPageLoad	Occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Edit/Add pages

BeforeProcessEdit BeforeProcessAdd	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following: - define a new template variable or change the value of existing one - display different template
BeforeEdit BeforeAdd	Occurs before new data is written to the database. Will work in all add/edit modes: Inline Add/Edit, Regular Add/Edit and Add/Edit page in popup. Use this event for the following: - prevent data record from being added or edited - send an email - save old record in another table - add a record to the log table
AfterEdit AfterAdd	Occurs after data was written to the database. Will work in all add/edit modes: Inline Add/Edit, Regular Add/Edit and Add/Edit page in popup. Use this event for the following: - send an email - add a record to the log table
ProcessValuesEdit	Occurs before the record is displayed (before the BeforeShowEdit event).
BeforeShowEdit BeforeShowAdd	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following: - define a new template variable or change the value of existing one - display different template
OnPageLoad	Occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Login/Registration pages

BeforeProcessLogin BeforeProcessRegister	Occurs after page is processed and ready to be displayed in the browser.

	Use this event for the following: - define a new template variable or change the value of existing one - display different template
BeforeLogin BeforeRegister	Occurs before user logs in or new user data is written to the database. Use this event for the following: - prevent user from being logged in or registered - send an email - add a record to the log table
AfterSuccessfulLogin AfterSuccessfulRegistration	Occurs after user is logged in or registered successfully. Use this event for the following: - redirect user to another page - send an email - add a record to the log table
AfterUnsuccessfulLogin AfterUnsuccessfulRegistration	Occurs if user was not logged in or was not registered.
BeforeShowLogin BeforeShowRegister	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following: - define a new template variable or change the value of existing one - display different template
OnPageLoad	Occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

View/Search/Report/Chart pages

BeforeProcess<PageName>	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following: - define a new template variable or change the value of existing one - display different template
ProcessValuesView	Occurs before the record is displayed (before the BeforeShowView event).
BeforeShow<PageName>	Occurs after page is processed and ready to be displayed in the browser. Use this event for the following:

	- define a new template variable or change the value of existing one - display different template
OnPageLoad	Occurs after page is displayed in browser. Use this event to work with the "edit" controls using the Javascript API.

Import page

BeforeImport	This event is executed before import is started.
BeforeInsert	Occurs before a record is inserted. Use this event to modify the record before it is inserted.
AfterImport	This event is executed after import is finished.

3.1.6 Common event parameters

You can use the following common parameters in your event code:

$pageObject - an object of RunnerPage class that represents a current page. For more information, see RunnerPage class.

Example:

Before record updated event

```
// Get the current record and display the Make and Model fields' values
$data = $pageObject->getCurrentRecord();
echo $data["Make"] ." ".$data["Model"];
```

$strTableName - name of the currently selected table.

$values - array with the fields values from Add/Edit form.

Example:

Before record updated event

```
echo $values["Field1"];
```

If the field was assigned an alias in the SQL query, then the *$values* array will get the alias instead of field name from the database. E.g. if you have SQL query *select salesrep_id AS Inv_Salesrep* ..., you should use *$values["Inv_Salesrep"]*.

$keys - array with the key columns.

Example:

After record added event

```
echo $keys["ID"];
```

$templatefile - name of the template file being displayed.

**$xt -** template engine object. Use *$xt->assign($name, $val)* to assign a value *$val* to the variable *$name*.

Example:

Before display event

```
$message = "This message";
$xt->assign("message",$message);
```

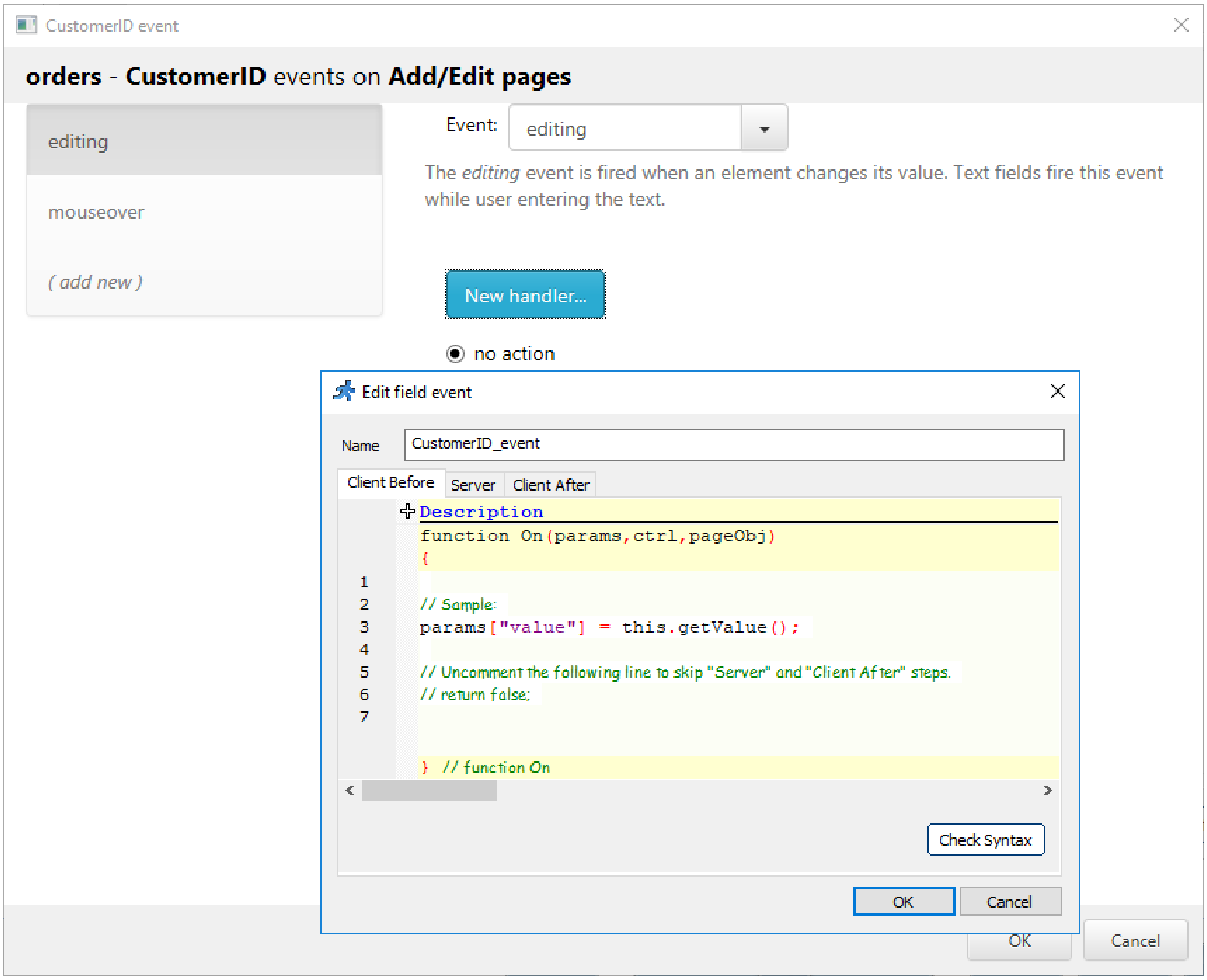
3.1.7 Field events

Field events option allows to perform an action when cursor enters edit field or leaves it or mouse is over a field. Perform any sort of validation, make other fields hidden or required etc. Designed to work on Add, Edit and Register pages.

For example, the *editing* event is fired when an element changes its value. Text fields fire this event while user entering the text.

See also How to access fields in the field events

Passing data between events



Event: editing

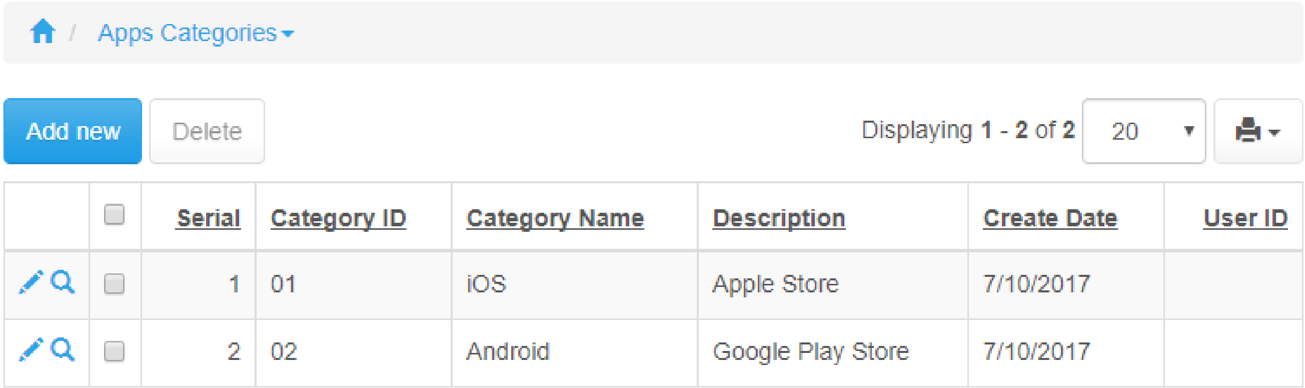
The *editing* event is fired when an element changes its value. Text fields fire this event while user entering the text.

Name: CustomerID_event

```
Description
function On(params,ctrl,pageObj)
{

// Sample:
params["value"] = this.getValue();

// Uncomment the following line to skip "Server" and "Client After" steps.
// return false;


} // function On
```

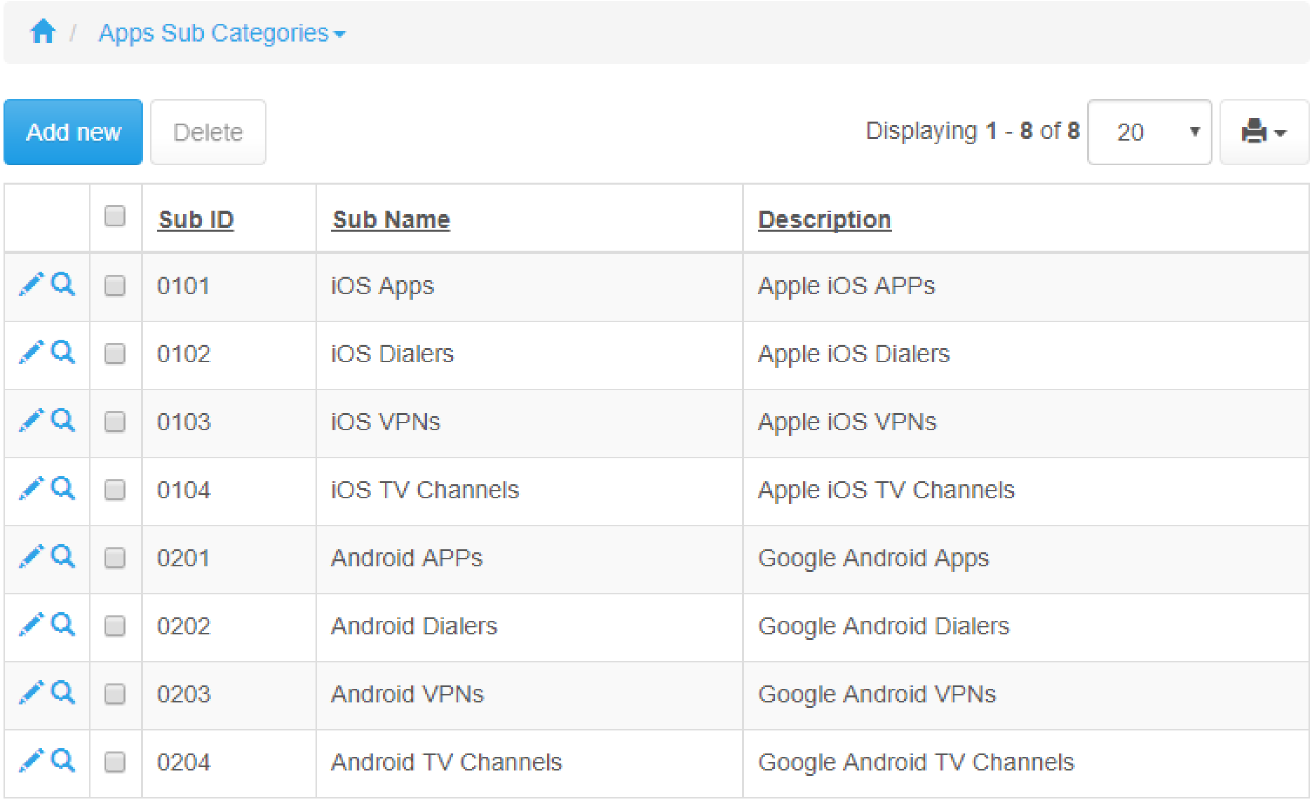
Using field events

We have two tables: categories and sub_categories. CategoryID is a two-digit category code.



	□	Serial	Category ID	Category Name	Description	Create Date	User ID
	□	1	01	iOS	Apple Store	7/10/2017	
	□	2	02	Android	Google Play Store	7/10/2017	

In sub_categories table SubID consists of two-digit category code and and two-digit subcategory code.

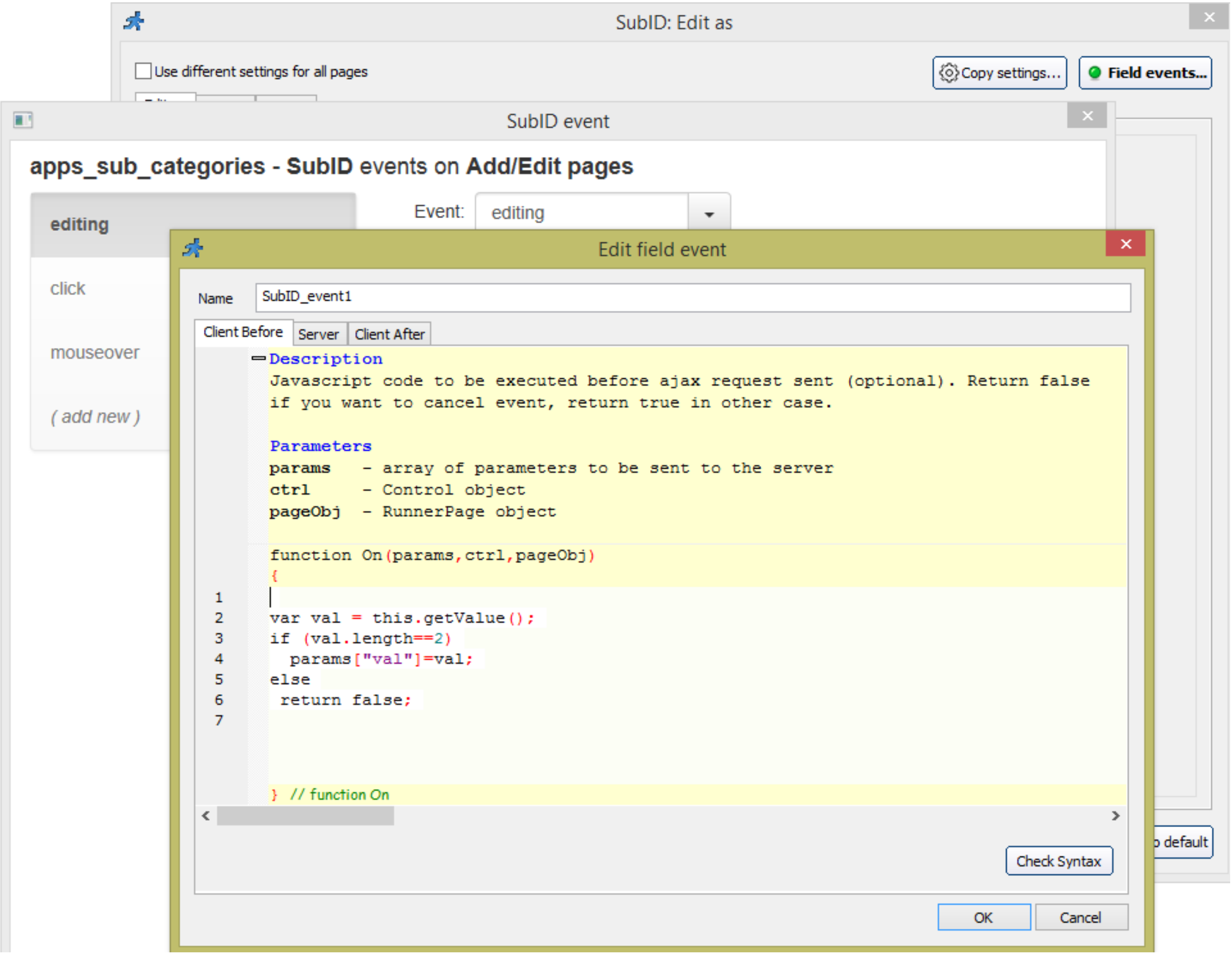


	□	Sub ID	Sub Name	Description
	□	0101	iOS Apps	Apple iOS APPs
	□	0102	iOS Dialers	Apple iOS Dialers
	□	0103	iOS VPNs	Apple iOS VPNs
	□	0104	iOS TV Channels	Apple iOS TV Channels
	□	0201	Android APPs	Google Android Apps
	□	0202	Android Dialers	Google Android Dialers
	□	0203	Android VPNs	Google Android VPNs
	□	0204	Android TV Channels	Google Android TV Channels

What we need to do is to simplify adding a large number of subcategories. When user types in first two digit (category) we want to show category name underneath and also calculate the next subcategory number. We will do that using field events.

Code

First we need to create Field Event for SubID field. We will use 'editing' event here which will be called every time the content of field is changed.



☐ Use different settings for all pages

apps_sub_categories - SubID events on Add/Edit pages

Event: editing

Name: SubID_event1

```
Description
Javascript code to be executed before ajax request sent (optional). Return false
if you want to cancel event, return true in other case.

Parameters
params   - array of parameters to be sent to the server
ctrl     - Control object
pageObj  - RunnerPage object

function On(params,ctrl,pageObj)
{

var val = this.getValue();
if (val.length==2)
  params["val"]=val;
else
 return false;

} // function On
```

ClientBefore:

```
var val = this.getValue();
if (val.length==2)
  params["val"]=val;
else
 return false;
```

In ClientBefore code we check the length of entered code and only send it to server side when code is two digits long.

Server:

```
$result["catName"] = DBLookup("select CategoryName from categories where
CategoryID='".$params["val"]."'");
$sub = DBLookup("select max(substring(SubID,3,2))+1 from sub_categories where
left(SubID,2)='".$params["val"]."'");
$result["newCat"] = $params["val"].str_pad($sub,2,"0",STR_PAD_LEFT);
```

On Server side we pull CategoryName from categories table and also calculate next subcategory ID. We pass both category name and new subcategory code back to ClientAfter event.

ClientAfter:

```
$("#sub_tip").remove();
$("input[id=value_SubID_1]").after("<div id='sub_tip' style='margin-top: 10px;
color: blue;'>"+result["catName"]+"</div>");
ctrl.setValue(result["newCat"]);
```

In this event we remove previous tooltip with category name and add a new one. We also make it appear in blue color. Then we set the value of SubID with new subcategory code we received from the Server event.

Here you can see how it works in generated application.

See also:

How to access fields in the field events

Passing data between events

3.2 Programming topics

3.2.1 Buttons

3.2.1.1 Button object

Button object

The *$button* object is used in the *OnServer* event (**Server** tab) of the inserted into the grid buttons. See Inserting button.

Methods

Method	Description
getCurrentRecord()	Gets the current record.
getNextSelectedRecord()	Returns associative arrays with values of the records selected on List page.

Methods

getCurrentRecord

Gets the current record.

Syntax

```
getCurrentRecord()
```

Arguments

No arguments.

Return value

If the button is inserted in the grid on the List page or on the Edit/View page, it returns an associative array (field name => value). Otherwise it returns false.

Example

Let's assume we have a button on the Classified Ad View page, which allows writing a letter to the person who posted the announcement. The page from which the message will be sent is *reply_add.php* and we need to pass the *email* parameter to this page.

OnServer event:

```
$record = $button->getCurrentRecord();
$result["email"]=$record["email"];
```

ClientAfter event:

```
location.href='reply_add.php?email='+result["email"];
```

getNextSelectedRecord

Returns associative arrays with values of the records selected on List page (marked with checkboxes). If nothing is selected or all selected records are processed, returns false. It is available to any buttons on the List page.

Syntax

```
getNextSelectedRecord()
```

Arguments

No arguments.

Return value

Returns associative arrays with values of the records selected on List page (marked with checkboxes). If nothing is selected or all selected records are processed, returns false. It is available to any buttons on the List page.

Example

Send emails to selected customers:

```
while($record = $button->getNextSelectedRecord())
{
   $message = "Dear ".$record["FirstName"];
   runner_mail(array('to' => $record["email"], 'subject' => "Greetings",
 'body' => $message));
}
```

3.2.1.2 rowData object

rowData object

After you inserted button to a datagrid, you can use *rowData* object to manipulate records. *rowData* object is available in *OnBefore* event (**Client Before** tab) and *OnAfter* event (**Client After** tab) of the inserted button and on Edit/View pages. See Inserting button.

rowData.fields - object, where index is the field name and value - JQuery container object that contains the field.

Example1. Make a red box around the value of ID field

```
rowData.fields['ID'].css('border', '1px solid red');
```

Example2. Get the HTML code of ID field

```
alert(rowData.fields['ID'].html());
```

Example3. Add a prefix to the value of ID field

```
rowData.fields['ID'].html("some prefix" + rowData.fields['ID'].html());
```

rowData.keys - array of values of key fields in the record.

```
for (var i = 0; i < rowData.keys.length; i ++) {
   // Do something
}
```

rowData.id - ID of the record. If you are running Inline Edit, you can use the Runner.getControl function.

Example:

```
for(var fName in rowData.fields){
   // get the control object, will work only if you open "inline"
   var ctrl = Runner.getControl(rowData.id, fName);
   if ( !ctrl ) {
      continue;
   }

   // get the control value
   var val = ctrl.getValue();
   if( val == "audi" )
      rowData.fields[fName].hide();
}
```

3.2.2 Data Access Layer (DAL)

3.2.2.1 About Data Access Layer

Data Access Layer (DAL) is now deprecated. While existing code will keep working we recommend to use Database API

Each table in DAL is presented as PHP class, all fields are PHP variables declared in this class.

Variables/Arrays

Variable/Array	Description

Table("TableName")	Provides access to a table.
	Example:
	`global $dal;`
	`$tblCars = $dal->Table("Cars");`

Value["FieldName"]	Provides access to the field values. *Value["FieldName"]* refers to the field to be updated or added. Example: `global $dal;`

	`$tblCars = $dal->Table("Cars");`
	`$tblCars->Value["Make"]="Volvo";`
	If the field name contains only English alphabet letters, digits and underscores (doesn't contain spaces and special characters), starts with a letter and doesn't coincide with PHP reserved words, you can also use the following notation to access the field:
	`global $dal;`
	`$tblCars = $dal->Table("Cars");`
	`$tblCars->Make="Volvo";`

Param["FieldName"]	Provides access to the field values.
	Param["FieldName"] is used in the WHERE clause of update query. This allows you to avoid the confusion when the same field appears in the field list and in WHERE clause.
	Example:
	`global $dal;`
	`$tblUsers = $dal->Table("UsersTable");`
	`$tblUsers->Param["FirstName"]="Bob";`
	`$tblUsers->Value["FirstName"]="Jim";`
	`$tblUsers->Update();`
	The corresponding SQL query:
	`Update UsersTable set 'FirstName'='Jim' where 'FirstName'='Bob'`

Methods

Method	Description
Add()	Inserts a new record into the database.
CustomQuery()	Runs custom SQL query.
Delete()	Deletes one or more records from the database.
DBLookup()	Executes an SQL query passed as a parameter and returns the first value of the first entry or null if nothing is found.
FetchByID()	Selects one or more records matching the condition.
Query()	Selects records from database sorting data by *orderby* field or fields and return recordset.

QueryAll()	Selects all records.
TableName()	Returns table name. This function is used for complex query with calculated fields or joined tables.
Update()	Updates one or more records in the database.
UsersTableName()	Returns properly formatted login table name.
whereAdd()	Adds new AND condition to the existing WHERE clause.

Examples

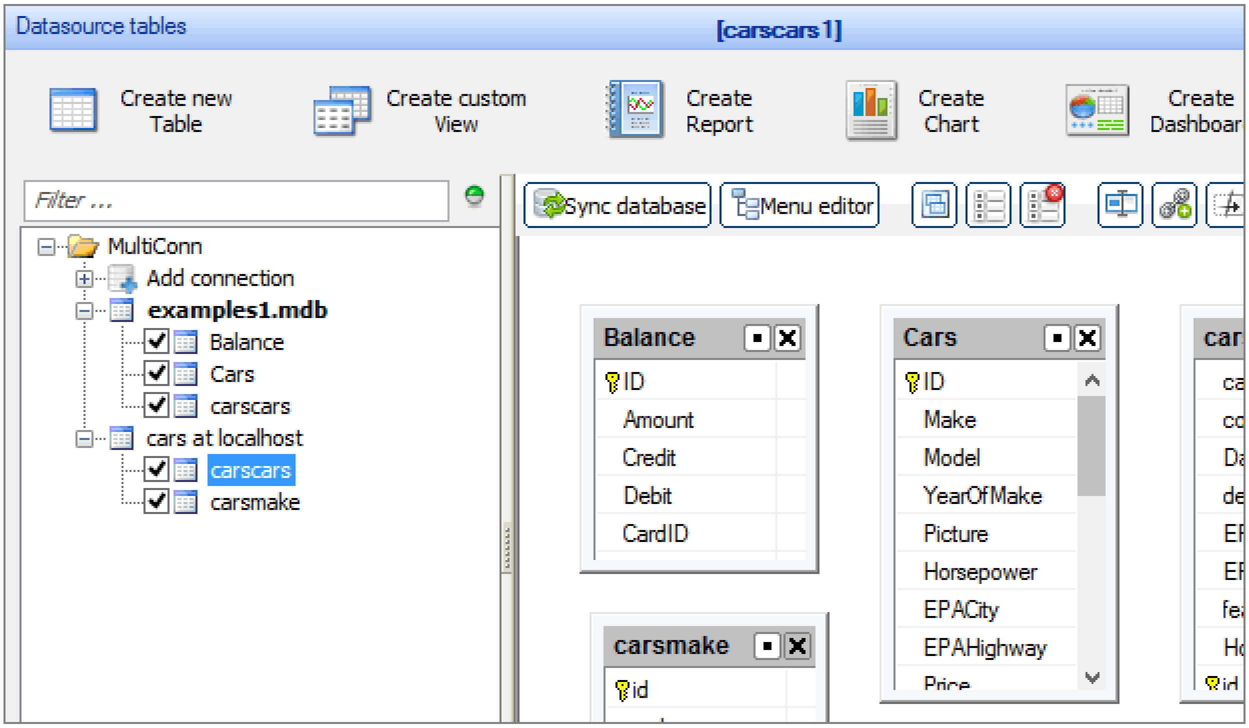
- Before deleting a record check if related records exist
- Dynamic SQL Query
- Redirect to user info edit page
- Show data from master table on details view/edit/add page
- Show list of customer orders
- Update multiple records on the List page
- Update multiple tables

See also

- Using DAL functions in projects with multiple database connections

3.2.2.2 Using DAL functions in projects with multiple database connections

In Enterprise Edition of PHPRunner you can use multiple database connections in a single project. This article explains how you can access data from multiple databases in your events.



Method 1: using DAL functions

There are three options to refer to a table:

1. Finds first matching table by it's name in all connections, starting with the primary connection.

```
global $dal;
$dal->Table("table")
```

2. Schema name helps to identify tables with identical names located in different schemas in databases like Oracle, Postgre and SQL Server.

```
global $dal;
$dal->Table("table","schema")
```

3. Here schema name can be left empty. Last parameter is the connection name as it appears on **Datasource tables** screen.

```
global $dal;
$dal->Table("table","schema","connection")
```

A complete code example:

```
global $dal;
$table = $dal->Table("cars", "", "cars at localhost");
```

```
$rs = $table->QueryAll();
while ($data = db_fetch_array($rs)) {
    echo $data["Make"].
    ", ".$data["Model"].
    "<br>";
}
```

Method 2: using free form SQL Queries

1. Example how to update all cars where make is *Audi* making *YearOfMake* equals *2002:*

```
global $cman;
$cman->byName("cars at localhost")->exec("update carscars set
yearofmake=2002 where make='Audi'");
```

2. Example of a query returning data:

```
global $cman;
$connection = $cman->byName("cars at localhost");
$rs = $connection->query("select count(*) as c  from cars where make='Audi'
");
$data = $rs->fetchAssoc();
echo "Number of Audi cars listed: ".$data["c"];
```

3.2.2.3 Methods

Add

Inserts new record into the database. This function is deprecated and we recommend use DB:Insert() function from Database API.

Syntax

```
Add()
```

Arguments

No arguments.

Return value

No return value.

Example

Insert new record into the database:

```
global $dal;
```

```
$tblEvents = $dal->Table("EventsTable");
$tblEvents->Value["event"]="First event";
$tblEvents->Value["public"]="yes";
$tblEvents->Add();
```

The corresponding SQL query:

```
Insert into EventsTable (event,public) values ('First event','yes')
```

See also

- Method: Delete

CustomQuery

Runs custom SQL query. This function is deprecated and we recommend use DB:Query() function from Database API.

Syntax

```
CustomQuery($sql)
```

Arguments

$sql - SELECT clause. Example: *"select * from UsersTable where ID=32"*.

Return value

Returns the recordset.

Example 1

Query that doesn't return data:

```
$sql = "update Users set active=0 where id=32";
CustomQuery($sql);
```

Example 2

Query that returns data:

```
$sql = "select count(*) as c from orders group by customerid";
$rs = CustomQuery($sql);
$data = db_fetch_array($rs);
echo "Number of customers: " . $data["c"];
```

See also

- Example: Update multiple records on the List page

- Method: Query
- Method: QueryAll

Delete

Deletes one or more records from the database.This function is deprecated and we recommend use DB:Delete() function from Database API.

Syntax

```
Delete()
```

Arguments

No arguments.

Return value

No return value.

Example 1

Delete all records where ID is '32':

```
global $dal;
$tblUsers = $dal->Table("UsersTable");
$tblUsers->Param["ID"]=32;
$tblUsers->Delete();
```

The corresponding SQL query:

```
Delete from UsersTable where ID=32
```

Example 2

Delete all records where FirstName is 'Bob' and Email is 'test@test.com':

```
global $dal;
$tblUsers = $dal->Table("UsersTable");
$tblUsers->Param["FirstName"]="Bob";
$tblUsers->Param["Email"]="test@test.com";
$tblUsers->Delete();
```

The corresponding SQL query:

```
Delete from UsersTable where FirstName='Bob' and Email='test@test.com'
```

See also

- Method: Add

DBLookup

Executes an SQL query passed as a parameter and returns the first value of the first entry or null if nothing is found.

Syntax

```
DBLookup($sql)
```

Arguments

$sql - SELECT clause. Example: *"select * from UsersTable where ID=32"*.

Return value

Returns the first value of the first entry or null if nothing is found.

Example

Returns the *zip* value where *userid* is "25":

```
$zip = DBLookup("select zip from users where userid=25");
```

FetchByID

Selects one or more records from the database matching the condition. This function is deprecated and we recommend use DB:Select() function from Database API.

Syntax

```
FetchByID()
```

Arguments

No arguments.

Return value

Returns the recordset on success or FALSE on error.

Example

Select all records where ID is '32'. To fetch a returned row as an associative array use *db_fetch_array* function.

```
global $dal;
$tblUsers = $dal->Table("UsersTable");
$tblUsers->Param["ID"]=32;
$rs = $tblUsers->FetchByID();
```

```
while( $data = db_fetch_array($rs) )
{
   echo $data["UsersName"];
}
```

The corresponding SQL query:

```
select * from UsersTable where ID=32
```

Query

Selects records from the database sorting data by the *orderby* field or fields and return the recordset. This function is deprecated and we recommend use DB:Query() function from Database API.

Syntax

```
Query(where, orderby)
```

Arguments

where - WHERE clause. Example: *ID=19*.

orderby - one or more fields used to sort the recordset.

Return value

Returns the recordset on success or FALSE on error.

Example 1

Select all records where name contains 'Jim'. To fetch a returned row as an associative array use *db_fetch_array* function.

```
global $dal;
$tblUsers = $dal->Table("UsersTable");
$rs = $tblUsers->Query("Name like '%Jim%'","Email DESC");
while( $data = db_fetch_array($rs) )
{
   echo $data["fieldName"]."<br>";
}
```

The corresponding SQL query:

```
select * from UsersTable where Name like '%Jim%' order by Email DESC
```

Example 2

Select and print all orders for John Sample. To fetch a returned row as an associative array use *db_fetch_array* function.

```
global $dal;
$tblOrders = $dal->Table("OrdersTable");
$rs = $tblOrders->Query("Customer='John Sample'","OrderID DESC");
while ($data = db_fetch_array($rs))
{
   echo "Order ".$data["OrderID"]." was placed ".$data["OrderDate"]." by
".$data["Customer"]."<br>";
}
```

The corresponding SQL query:

```
select * from OrdersTable where Customer like 'John Sample'
  order by OrderID DESC
```

See also

- Example: Show data from master table on details view/edit/add page
- Example: Before deleting a record check if related records exist
- Example: Redirect to user info edit page
- Example: Show list of customer orders
- Method: QueryAll
- Method: CustomQuery

QueryAll

Selects all records from the table. This function is deprecated and we recommend use DB::Query() function from Database API.

Syntax

```
QueryAll()
```

Arguments

No arguments.

Return value

Returns the recordset on success or FALSE on error.

Example

Send mass email to all users:

```
global $dal;
//select emails from Users table
```

```
$tblUsers = $dal->Table("UsersTableName");
$rs = $tblUsers->QueryAll();
while ($data = db_fetch_array($rs))
{
  $email.=$data["EmailAddress"].", ";
  $from="admin@test.com";
  $msg="Check what's hot this season";
  $subject="Monthly newsletter";
  $ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, 'from'=>$from));
  if(!$ret["mailed"])
    echo $ret["message"]."<br>";
}
```

See also

- Method: Query
- Method: CustomQuery

TableName

Returns the table name. This function is used for complex query with calculated fields or joined tables. This function is deprecated.

Syntax

```
TableName()
```

Arguments

No arguments.

Return value

Returns the table name.

Example

Using complex query:

```
global $dal;
$tblProducts = $dal->Table("Products");
$sql = "select sum(UnitsInStock) as total, concat(Category,' ',
  ProductName) as FullProductName from ";
$sql .= $tblProducts->TableName() . " group by country";
```

See also

- Method: UsersTableName

Update

Updates one or more records in the database. This function is deprecated and we recommend use DB:Exec() function from Database API.

Syntax

```
Update()
```

Arguments

No arguments.

Return value

No return value.

Example

Update record where ID is 32 making **FirstName** field value 'Jim' and **LastName** field value 'Morrison':

```
global $dal;
$tblUsers = $dal->Table("UsersTable");
$tblUsers->Param["ID"]=32;
$tblUsers->Value["FirstName"]="Jim";
$tblUsers->Value["LastName"]="Morrison";
$tblUsers->Update();
```

The corresponding SQL query:

```
Update UsersTable set FirstName='Jim', LastName='Morrison' where ID=32
```

See also

- Example: Update multiple tables

UsersTableName

Returns properly formatted login table name. This function is deprecated and we recommend using Security API

Syntax

```
UsersTableName()
```

Arguments

No arguments.

Return value

Returns properly formatted login table name.

Example

Print all email addresses from the Users table:

```
$sql = "select * from " . UsersTableName();
$rs = CustomQuery($sql);
while ($emailsData = db_fetch_array($rs))
   echo $emailsData["EmailAddress"]."<br>";
```

See also

- Method: TableName

whereAdd

Adds new AND condition to the existing WHERE clause. This function is deprecated and we recommend using Dynamic SQL query.

Syntax

```
whereAdd($where, $condition)
```

Arguments

$where - WHERE clause. Example: *"ID=19"*.

$condition - any condition clause. Example: *"CustomerID='123'"*.

Return value

Returns updated WHERE clause.

See also

- Example: Dynamic SQL Query

3.2.3 Database API

3.2.3.1 About Database API

Database API functions similarly to the Data Access Layer (DAL) but is more convenient to use and can work with multiple connections. We would recommend using Database API, but DAL will work as well.

Methods

Method	Description
DB::SetConnection(name)	Sets current database link when working with multiple database connections.

DB::Exec(SQL)	Executes SQL query using current active connection.
DB::Query(SQL)	Executes SQL Query that returns data.
DB::LastId()	Returns last inserted autoincremented field value in the current connection.
DB::LastError()	Returns last error message in the current connection.
DB::Insert()	Adds records to database.
DB::Delete()	Deletes records from database.
DB::Select()	Retrieves data from database.
DB::PrepareSQL()	Prepares SQL query to use SQL variables

3.2.3.2 Methods

SetConnection

Sets the current database connection. Recommended for use when working with multiple database connections.
All consequent API calls made via DB:: functions will be executed via this connection.
When working with the default (primary) connection the call of this function is not required.

To switch back to primary connection call this function without arguments.

Syntax

```
DB::SetConnection(name)
```

Arguments

name - connection name, as it appears in wizard

Return value

No return value.

Example

```
// switch to Northwind connection
DB::SetConnection("Northwind");

// execute query
DB::Exec("update orders set Status='shipped' where OrderDate=CURDATE()");
```

```
// switch back to primary connection
DB::SetConnection("");
```

See also

- DB::Exec(SQL)

Exec

Executes SQL query using current active connection.

Use for SQL Queries that do not return data.

Syntax

```
DB::Exec(SQL)
```

Arguments

SQL - query

Return value

true if executed successfully and *false* in all other cases

Example

```
DB::Exec("insert into orders (Name, Amount, OrderDate) values ('Jonh Smith', 1
```

Query

Executes an SQL Query that returns data.

Syntax

```
DB::Query(SQL)
```

Arguments

SQL - query

Return value

QueryResult object.

Example

```
$rs = DB::Query("select * from carsmake");

while( $data = $rs->fetchAssoc() )
{
   echo $data["id"];
   echo $data["make"];
}
```

LastId

Returns the last inserted autoincremented field value in the current connection.

Syntax

```
DB::LastId()
```

Arguments

no arguments

Return value

last inserted autoincremented field value in the current connection

LastError

Returns the last error message in the current connection.

Syntax

```
DB::LastError()
```

Arguments

no arguments required

Return value

last error message in the current connection

Insert

Inserts a record into a database.

Syntax

```
DB::Insert($table, array(field=>value))
```

Arguments

$table - the table in which the data will be inserted

`array(field=>value)` – the array with values that you wish to insert

Return value

No return value.

Example

```
// Insert a record into Cars table

$data = array();
$data["make"] = "Toyota";
$data["model"]  = "RAV4";
$data["price"] = 16000;
DB::Insert("cars", $data );
```

Delete

Deletes records from a database. The second parameter must be specified for the method to be executed successfully.

Syntax

```
DB::Delete(table, array(field=>value))
```

```
DB::Delete(table, $where)
```

Arguments

table - the table in which the data will be deleted from

array(field=>value) - array with values that define the condition of the delete query

`$where` – condition for the query; if unspecified, no action will be done

Return value

No return value.

Example

```
// delete from Cars where ID=50

$data = array();
$data["id"] = 50;
DB::Delete("cars", $data );
```

Alternative syntax

```
// delete from Cars where ID=50

DB::Delete("cars", "id=50" );
```

Select

Retrieves data from the database.

Syntax

```
DB::Select(table, array(field=>value))
```

```
DB::Select(table, $where)
```

Arguments

table - the table where the data will be selected from

array(field=>value) - array with values that define the condition of select query

`$where` – condition for the query

Return value

Array with the result of select query.

Example 1

Retrieve all data from Cars table where the make is Toyota and the model is RAV4.

```
$data = array();
$data["make"] = "Toyota";
$data["model"]  = "RAV4";
$rs = DB::Select("Cars", $data );
while( $record = $rs->fetchAssoc() )
{
     echo $record["id"];
     echo $record["make"];
}
```

Example 2

Retrieve all data from the table Cars where the price is less than 20,000.

```
$rs = DB::Select("Cars", "Price<20000" );
while( $record = $rs->fetchAssoc() )
{
    echo $record["id"];
    echo $record["make"];
}
```

PrepareSQL

This function prepares SQL query when you use SQL variables in it. SQL variables are case-insensitive.
Read more about using SQL variables in SQL and Lookup wizards

Syntax

```
DB::prepareSQL(SQL)
```

Arguments

SQL - query with variables

Return value

The prepared SQL query.

Example

- In “After Add” event you can use:

```
$sql = DB::prepareSQL("insert into log set lastid=:new.id");
DB::Exec( $sql );
```

- In button’s Server code or in field events for View/Edit pages:

```
$sql = DB::prepareSQL("insert into log set lastname=':name'");
DB::Exec( $sql );
```

3.2.3.3 QueryResult object

fetchAssoc

Syntax

```
fetchAssoc()
```

Arguments

no arguments

Return value

Returns record as an associative array:
"field" => value;
Null, if there are no matching records.

Example

```
$rs = DB::Query("select * from carsmake");
while( $data = $rs->fetchAssoc() )
{
   echo $data["id"];
   echo $data["make"];
}
```

fetchNumeric

Syntax

```
fetchNumeric()
```

Arguments

no arguments

Return value

returns record as array with numeric keys:
0 => value, 1=>value;
Null, if there are no more records.

Example

```
$rs = DB::Query("select * from carsmake");
while( $data = $rs->fetchNumeric() )
{
    echo $data[0];
    echo $data[1];
}
```

value

This is an alternative way of working with the query. Covenient to use when the query returns a single record.

This function should not be used together with **fetchAssoc** or **fetchNumeric**.

Syntax

```
value( fieldname or index )
```

Arguments

`fieldname or index`

Return value

Value.

Example

```
$rs = DB::Query("select * from Cars where id=20");
echo $rs->value("Make");

$rs = DB::Query("select count(*) from Cars");
echo $rs->value(0);
```

3.2.3.4 SQL variables in SQL and Lookup wizards

This feature can be used anywhere where you use SQL Queries. With SQL variables you can write cleaner code and easily implement custom dropdown boxes or advanced security. SQL variables are case-insensitive.

Where to use it:

- In regular SQL queries that you enter on “SQL Query” screen
- In WHERE Tabs on “SQL Query” screen
- In Lookup Wizard WHERE clause. It can be used in dropdowns that are dependent on any type of field, dependent on master dropdown, dependent with more complex rule than equality (i.e. age is more than selected).

```
WHERE CustomerID= ':user.CustomerID'
WHERE CustomerID= ':session.UserID'
```

- In events by using prepareSQL function. For example, in "After Add" event you can use:

```
$sql = DB::prepareSQL("insert into log set lastid=:new.id");
DB::Exec( $sql );
```

- In button's Server code or in field events for View/Edit pages:

```
$sql = DB::prepareSQL("insert into log set lastname=':name'");
DB::Exec( $sql );
```

A full list of SQL variables:

```
:field - current field value on Add, Edit or Register page
:master.field - any field from master record
:session.key - any session variable
:user.field - any field from login table
:old.field - old field value (before any changes were applied)
:new.field - new field value.
```

See also

Lookup wizards

3.2.4 Grid Row Javascript API

3.2.4.1 About Grid Row Javascript API

Grid Row Javascript API can be used to run Ajax event code on Click actions in the grid row/field. It can be also used on the List page for the buttons inserted into the grid.

Methods

Method	Description
row.fieldCell()	Returns jQuery object of table cell with this field.
row.getFieldValue()	Returns raw field value without formatting applied.
row.getFieldText()	Returns field value with formatting applied. In other words returns HTML code of this table cell.
row.setFieldText()	Sets field's HTML as it appears in grid.
row.setFieldValue()	Field value set this way will not appear in the grid and only applies to consequent getFieldValue calls.
row.record()	Returns Jquery object representing table row.

row.setMessage()	Prints a message in one of row cells. Useful to debug/troubleshoot your click action code.
row.getMessage()	Gets previous added message

See also

- See how to Insert button into datagrid

3.2.4.2 Methods

fieldCell

Returns jQuery object of table cell with this field.

Syntax

```
row.fieldCell(field);
```

Arguments

field - name of the field.

Return value

Returns jQuery object of table cell with this field.

Example

Change "make" field background to red:

```

row.fieldCell("make").css('background', 'red');
```

getFieldValue

Returns raw field value without formatting applied.

Syntax

```
row.getFieldValue(field);
```

Arguments

field - name of the field.

Return value

Returns raw field value without formatting applied.

getFieldText

Returns field value with formatting applied. In other words returns HTML code of this table cell.

Syntax

```
row.getFieldText(field);
```

Arguments

field - name of the field.

Return value

Returns field value with formatting applied.

For instance, we have a Lookup Wizard field that has different Link field and Display field selected. getFieldText(field) will return Display field value. getFieldValue(field) will return Link field value.

setFieldText

Sets field's HTML as it appears in grid.

Syntax

```
row.setFieldText(field, text);
```

Arguments

field - field name.

text - fields HTML.

Return value

No return value

setFieldValue

Sets the field value. Field value set this way will not appear in the grid and only applies to consequent getFieldValue calls.

Syntax

```
row.setFieldValue(field, value);
```

Arguments

field - field name.

value - field value.

Return value

No return value.

Remarks

Field value set this way will not appear in the grid and only applies to consequent getFieldValue calls.

record

Returns Jquery object representing table row.

Syntax

```
row.record()
```

Arguments

No arguments.

Return value

Returns Jquery object representing table row.

Example

Change row background:

```
row.record().css('background', 'red');
```

setMessage

Prints a message in one of row cells. Useful to debug/troubleshoot your click action code.

Syntax

```
row.setMessage(str)
```

Arguments

str - the string to be printed in row cell.

Return value

No return value.

Example

Add a message to first table cell:

```
row.setMessage('test message', row.record()[0]);
```

getMessage

Gets previous added message

Syntax

```
row.getMessage()
```

Arguments

No arguments.

Return value

No return value.

Example

```
row.setMessage(row.getMessage() + 'some extra text');
```

See also

- setMessage()

3.2.5 Javascript API

3.2.5.1 About Javascript API

Javascript API allows you to work with the "edit" controls, manage search panel. Javascript API objects are available in the Javascript OnLoad event of the appropriate page.

Objects

Object	Description
Control	Allows to work with the "edit" controls.
InlineRow	Allows to process clicking the Cancel button in the Inline mode on Add/Edit pages.
RunnerPage	Represents a current page object.
SearchController	Allows to manage the search panel.
SearchField	Allows to manage search fields on the search panel.

Javascript API works for Add/Edit/View/Register pages.

For example, use the following code to make the field red on the View page.

```
var ctrl = Runner.getControl(pageid,'action');
ctrl.addStyle('color: red');
```

Examples

Ask for confirmation before saving a record

Change font size in text box

Change font in "edit" controls

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to enable/disable a button

How to hide 'Edit selected'/'Delete selected' buttons

How to refresh List page after Edit in popup

How to control Inline Add/Edit functionality from script

How to work with tabs

Show dropdown list of US states if US was selected in country list

3.2.5.2 Control object

About Control object

Control object allows you to work with the "edit" controls. Control object is available in the Javascript OnLoad event of the appropriate page.

Before you start working with the "edit" controls, you need to get that controls. Use *.getControl()* method to get controls by the field name and page id:

```
var ctrl = Runner.getControl(pageid, fieldname);
```

Example:

```
var ctrl = Runner.getControl(pageid, 'Make');
```

To get all the controls of the table, you need to pass only the table name as the argument:

```
var ctrlArr = Runner.controls.ControlManager.getAt(tableName);
```

Example:

```
// Get all the controls for the table 'Cars'
var recCtrlsArr = Runner.controls.ControlManager.getAt('Cars');
// loop through all controls on the page making them all required
for(var i=0;i<recCtrlsArr.length;i++)
{
    var ctrl = recCtrlsArr[i];
    ctrl.addValidation("IsRequired");
}
```

Methods

Method	Description

addClass()	Adds the class to the control with a value.
addStyle()	Adds the style to the control.
addValidation()	Adds validation to the control.
clear()	Sets "" as the control value and if the control was not previously validated removes the message that it was not validated.
clearEvent()	Deletes events.
getDispElem()	Returns jQuery object - the element (input, textarea, select etc.) that displays the value of the selected field.
getValue()	Reads the control value.
hide()	Hides the control.
invalid()	Gets the control status after the last validation.
isReadonly()	Gets the control readonly status.
makeReadonly()	Makes control readonly.
makeReadWrite()	Makes control writable.
on()	Adds an event to the control and transfer the array of arguments to the handler.
removeClass()	Removes the class from the control.
removeValidation()	Removes validation from the control.
reset()	Sets the control value to the original one.
setDisabled()	Makes control "disabled".
setEnabled()	Makes control "enabled".
setFocus()	Sets focus to the control.
setValue()	Sets the control value.
show()	Shows the control.
validate()	Validates the control.
validateAs()	Validates the control value for some validation type.

Events

Event name	Description
blur	An element loses focus.
change	The user changes the content of a field.
click	Mouse clicks an object.
dblclick	Mouse double-clicks an object.
focus	An element gets focus.
keydown	A keyboard key is pressed.
keypress	A keyboard key is pressed or held down.
keyup	A keyboard key is released.
mousedown	A mouse button is pressed.
mousemove	The mouse is moved.
mouseout	The mouse is moved off an element.
mouseover	The mouse is moved over an element.
mouseup	A mouse button is released.
resize	A window or frame is resized.
select	Text is selected.

Examples

Ask for confirmation before saving a record

Change font in dropdown list

Change font in "edit" controls

Change width of edit box with AJAX popup

Change width of text field on Quick Search panel

How to calculate values (totals) on the fly

How to control Inline Add/Edit functionality from script

How to hide 'Edit selected'/'Delete selected' buttons

How to work with tabs

Show dropdown list of US states if US was selected in country list

Methods

addClass

Adds the class to the control with a value.

Syntax

```
ctrl.addClass(className);
```

Arguments

className - class name. Example: 'highlight'.

Return value

No return value.

Example

Add class *highlight* to the control:

```
                                          JavaScript OnLoad event
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.addClass('highlight');
```

See also

- Method: removeClass

addStyle

Adds the style to the control.

Syntax

```
ctrl.addStyle(style);
```

Arguments

style - CSS string with style definition. Example: *'width: 200px;'*.

Return value

No return value.

Example

Add the style 'display: none;' to the control:

```
                                          JavaScript OnLoad event
```

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.addStyle('display: none;');
```

addValidation

Adds validation to the control.

Syntax

```
ctrl.addValidation(validation_type);
```

Arguments

validation_type - one of the available validation types:

Validation type	Description
IsRequired	Makes field required.
IsNumeric	A number.
IsPassword	Password cannot be blank, cannot be 'Password' and should be at least 4 characters long.
IsEmail	Valid email address.
IsMoney	Numeric value. Decimal point is allowed. Examples: 13, 24.95.
IsZipcode	Five or ten digit number. Valid formats: 12345, 12345-6789 or 123456789.
IsPhonenumber	Numbers, spaces, hyphens, and parentheses are allowed. Examples: (123) 456-7890, 123 456 7890, 123 4567.
IsState	Two letter US state abbreviation. Examples: AK, AL, CA, MN.
IsSSN	Nine digit US social security number. Valid formats: 123-45-6789 or 123 45 6789.
IsCC	Valid credit card number.
IsTime	Any valid time format that match regional settings.
IsDate	Any valid date format that match regional settings.

{
regex: "regular_expression",
message: "warning_message",
messagetype: "message_type"
}
	A regular expression (regexp). Note that regexp validation overwrites the previous validation, if any.

Return value

No return value.

Example 1

Make all controls on the page required:

JavaScript OnLoad event

```
// Get all the controls for the table 'Cars'
var recCtrlsArr = Runner.controls.ControlManager.getAt('Cars');
// loop through all controls on the page making them all required
for(var i=0;i<recCtrlsArr.length;i++)
{
    var ctrl = recCtrlsArr[i];
    ctrl.addValidation("IsRequired");
}
```

Example 2

Add regular expression validation to the control:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Number');
ctrl.addValidation(/[0-9]/);
ctrl.customValidationFailedMessages[ "RegExp" ] = {
    message: "The field should be a number from 0 to 9",
    messageType: "Text"
};
```

See also

- Method: removeValidation

clear

Sets "" as the control value and if the control was not previously validated removes the message that it was not validated.

Syntax

```
ctrl.clear();
```

Arguments

No arguments.

Return value

No return value.

Example

Clear the control value:

[JavaScript OnLoad](#) event

```
var ctrl = Runner.getControl(pageid, 'Model');
ctrl.clear();
```

clearEvent

Deletes events.

Syntax

```
ctrl.clearEvent(event);
```

Arguments

event - one of the available [event values](#).

Return value

No return value.

Example

Delete 'click' event:

[JavaScript OnLoad](#) event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.clearEvent('click');
```

See also

- [Method: on](#)

getDispElem

Returns jQuery object - the element (input, textarea, select etc.) that displays the value of the selected field with the prefix *value_* (e.g. *value_Make* for the field *Make*).

Syntax

```
ctrl.getDispElem();
```

Arguments

No arguments.

Return value

Returns jQuery object.

Example

Change width of edit box with AJAX popup:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.getDispElem().css("width", "200px");
```

getValue

Reads the control value.

Syntax

```
var value = ctrl.getValue();
```

Arguments

No arguments.

Return value

Returns current value of the control.

Example1

Read the control value:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
var value = ctrl.getValue();
```

Example2

Check box control hold the value of 'on' (checked off) and empty string if unchecked:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Discounted');
if (ctrl.getValue()=='on')
   alert('Checked');
// clear check box
ctrl.setValue('');
```

See also

- Example: Show dropdown list of US states if US was selected in country list
- Method: setValue

hide

Hides the control.

Syntax

```
ctrl.hide();
```

Arguments

No arguments.

Return value

No return value.

Example

Hide the control (not the label):

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.hide();
```

If you need to hide the control and the label, use pageObj.hideField().

See also

- Example: Show dropdown list of US states if US was selected in country list
- Method: show

invalid

Gets the control status after the last validation.

Syntax

```
var isInvalid = ctrl.invalid();
```

Arguments

No arguments.

Return value

Returns *true* if control is Invalid, returns *false* if otherwise.

Example

Get the control status:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'YearOfMake');
var isInvalid = ctrl.invalid();
```

isReadonly

Gets the control readonly status.

Syntax

```
var readonly = ctrl.isReadonly();
```

Arguments

No arguments.

Return value

Returns *true* if control is readonly, returns *false* if otherwise.

Example

Check if control is readonly:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
var readonly = ctrl.isReadonly();
```

Remarks

Readonly controls will be submitted with the form and saved in the database. Readonly controls will be filled by Autofill feature. You can apply default values to Readonly controls.

See also

- Method: makeReadonly

makeReadonly

Makes control readonly.

Syntax

```
ctrl.makeReadonly();
```

Arguments

No arguments.

Return value

No return value.

Example

Make control readonly:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.makeReadonly();
```

Remarks

Readonly controls will be submitted with the form and saved in the database. Readonly controls will be filled by Autofill feature. You can apply default values to Readonly controls.

See also

- Method: isReadonly

makeReadWrite

Makes control writable.

Syntax

```
ctrl.makeReadWrite();
```

Arguments

No arguments.

Return value

No return value.

Example

Make control writable:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.makeReadWrite();
```

See also

- Method: MakeReadonly

on

Adds an event to the control and transfer the array of arguments to the handler.

Syntax

```
ctrl.on(event,handler,arguments);
```

Arguments

event - one of the available event values.

handler - function performed after the event was fired.

arguments - parameters passed to the handler function.

Return value

No return value.

Example

Add the 'click' event to the control:

JavaScript OnLoad event

```
ctrl.on('click', function() {
// call 'setValue' method to set new value
```

```
    this.setValue('newValue');
});
```

See also

- [Example: Show dropdown list of US states if US was selected in country list]()
- [Example: Ask for confirmation before saving a record]()
- [Method: clearEvent]()

removeClass

Removes the class from the control.

Syntax

```
ctrl.removeClass(className);
```

Arguments

className - class name. Example: 'highlight'*.*

Return value

No return value.

Example

Remove class *highlight* from the control:

```
                                                    JavaScript OnLoad event
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.removeClass('highlight');
```

See also

- [Method: addClass]()

removeValidation

Removes validation from the control.

Syntax

```
ctrl.removeValidation(validation_type);
```

Arguments

validation_type - one of the available validation types:

Validation constant	Description
IsRequired	Makes field required.
IsNumeric	A number.
IsPassword	Password cannot be blank, cannot be 'Password' and should be at least 4 characters long.
IsEmail	Valid email address.
IsMoney	Numeric value. Decimal point is allowed. Examples: 13, 24.95.
IsZipcode	Five or ten digit number. Valid formats: 12345, 12345-6789 or 123456789.
IsPhonenumber	Numbers, spaces, hyphens, and parentheses are allowed. Examples: (123) 456-7890, 123 456 7890, 123 4567.
IsState	Two letter US state abbreviation. Examples: AK, AL, CA, MN.
IsSSN	Nine digit US social security number. Valid formats: 123-45-6789 or 123 45 6789.
IsCC	Valid credit card number.
IsTime	Any valid time format that match regional settings.
IsDate	Any valid date format that match regional settings.
RegExp	A regular expression (regexp).

Return value

No return value.

Example

Remove validation "IsRequired" from the control:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.removeValidation("IsRequired");
```

See also

- Method: addValidation

reset

Sets the control value to the original one.

Syntax

```
ctrl.reset();
```

Arguments

No arguments.

Return value

No return value.

Example

Set the control value to the original one:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.reset();
```

setDisabled

Makes control disabled.

Syntax

```
ctrl.setDisabled();
```

Arguments

No arguments.

Return value

No return value.

Example

Make control "disabled":

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.setDisabled();
```

Remarks

Disabled controls will NOT be submitted with the form and will not be saved in the database. Disabled controls will be filled by Autofill feature. You can apply default values to Disabled controls.

See also

- [Method: setEnabled]()

setEnabled

Makes control enabled.

Syntax

```
ctrl.setEnabled();
```

Arguments

No arguments.

Return value

No return value.

Example

Make control "enabled":

[JavaScript OnLoad]() event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.setEnabled();
```

See also

- [Method: setDisabled]()

setFocus

Sets focus to the control and, depending on the *triggerEvent* argument, raises/does not raise "focus" event.

Syntax

```
ctrl.setFocus(triggerEvent);
```

Arguments

triggerEvent - if *true*, the "focus" event, previously added to the control using *on* function, is raised. If *false* or not specified, event is not raised.

Return value

No return value.

Example

Set focus to the control:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.setFocus();
```

See also

- Method: on

setValue

Sets the control value.

Syntax

```
ctrl.setValue(value);
```

Arguments

value - control value.

Return value

No return value.

Example1

Calculate total value on the fly:

JavaScript OnLoad event

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotal = Runner.getControl(pageid, 'Total');

function func() {
  ctrlTotal.setValue(Number(ctrlPrice.getValue()) *
Number(ctrlQuantity.getValue()));
};

ctrlPrice.on('keyup', func);
ctrlQuantity.on('keyup', func);
```

Example2

Check box control hold the value of 'on' (checked off) and empty string if unchecked:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Discounted');
if (ctrl.getValue()=='on')
   alert('Checked');
// clear check box
ctrl.setValue('');
```

See also

- Method: getValue

show

Shows the control.

Syntax

```
ctrl.show();
```

Arguments

No arguments.

Return value

No return value.

Example

Show the control:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.show();
```

See also

- Example: Show dropdown list of US states if US was selected in country list
- Method: hide

validate

Validates the control value against all previously added validation types.

Syntax

```
var vRes = ctrl.validate();
```

Arguments

No arguments.

Return value

Returns an array with two values: *result* (validation result, values - *true* or *false*) and *messageArr* (the array of validation errors). If *result* is *true*, then the array *messageArr* is blank.

Example

Validate the control and display validation error if any:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'make');
var vRes = ctrl.validate();
if ( !vRes.result ) {
    var message = "";
    for (var i = 0; i < ctrl.validationArr.length ; i++) {
        if ( vRes.messagesData[ ctrl.validationArr[i] ] ) {
            message += vRes.messagesData[ ctrl.validationArr[i] ].join("
");
        }
    }

    alert( message );
}
```

See also

- Method: validateAs

validateAs

Validates the control value for certain validation type. The control will not be marked as not validated.

Syntax

```
var vRes = ctrl.validateAs(validation_type);
```

Arguments

validation_type - one of the available validation types:

Validation constant	Description
IsRequired	Makes field required.
IsNumeric	A number.
IsPassword	Password cannot be blank, cannot be 'Password' and should be at least 4 characters long.
IsEmail	Valid email address.
IsMoney	Numeric value. Decimal point is allowed. Examples: 13, 24.95.
IsZipcode	Five or ten digit number. Valid formats: 12345, 12345-6789 or 123456789.
IsPhonenumber	Numbers, spaces, hyphens, and parentheses are allowed. Examples: (123) 456-7890, 123 456 7890, 123 4567.
IsState	Two letter US state abbreviation. Examples: AK, AL, CA, MN.
IsSSN	Nine digit US social security number. Valid formats: 123-45-6789 or 123 45 6789.
IsCC	Valid credit card number.
IsTime	Any valid time format that match regional settings.
IsDate	Any valid date format that match regional settings.

Return value

Returns *true* or validation error.

Example

Validate the control value for "IsRequired" validation:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, 'make');
Runner.validation.control = ctrl;
var vRes = ctrl.validateAs("IsRequired");
```

See also

- Method: validate

3.2.5.3 InlineRow object

About InlineRow object

InlineRow object allows to process clicking the Cancel button in the Inline mode on Add/Edit pages.

inlineRow.data object contains values of the fields as they were before editing. Example:

```
inlineRow.data["Price"]
```

Methods

Method	Description
onBeforeCancel	Method is invoked by clicking the Cancel button during inline add/edit.
onCancel	Method is invoked after the cancellation of inline add/edit.
revokeCancel	Revokes the cancellation of inline add/edit (record stays in edit mode).

Examples

```
if(inlineRow){
    inlineRow.onCancel = function(){
        console.log('edit cancelled');
    };
}
```

```
if(inlineRow){
 inlineRow.onBeforeCancel = function(){
            inlineRow.revokeCancel = true;
    };
}
```

3.2.5.4 RunnerPage object

About RunnerPage object

RunnerPage object represents a current page object and is called *pageObj*. It is available in the Javascript OnLoad event of the appropriate page.

Methods

Method	Description

getSearchController()	Gets the object of the search panel (SearchController object).
hideField()	Hides the field and field label.
showField()	Shows the previously hidden field and field label.

Methods

getSearchController

Gets the object of the search panel (SearchController object). *getSearchController()* method is available in the JavaScript OnLoad event of the List/Chart/Report/Advanced Search pages.

Syntax

```
var srch = pageObj.getSearchController();
```

Arguments

No arguments.

Return value

Returns SearchController object.

See also

- About SearchController object

hideField

Hides the field and field label. *hideField()* method is available in the JavaScript OnLoad event of the Add/Edit/View/Register pages.

Syntax

```
pageObj.hideField(field);
```

Arguments

field - field name. Example: *"Make"*.

Return value

No return value.

Example 1

Hide the *Make* field:

JavaScript OnLoad event

```
pageObj.hideField("Make");
```

Example 2

Show the *State* field, if *Country* is *USA,* or otherwise hide:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, "Country");
if(ctrl.getValue() != "USA")
pageObj.hideField("State");
ctrl.on('change', function(){
 if (this.getValue() == "USA"){
    pageObj.showField("State");
 }
 else {
    pageObj.hideField("State");
 }
});
```

See also

- JavaScript API: showField()
- Method: showField()
- Method: hideField()

showField

Shows the previously hidden field and field label. *showField()* method is available in the JavaScript OnLoad event of the Add/Edit/View/Register pages.

Syntax

```
pageObj.showField(field);
```

Arguments

field - field name. Example: *"Make"*.

Return value

No return value.

Example 1

Show the *Make* field:

JavaScript OnLoad event

```
pageObj.showField("Make");
```

Example 2

Show the *State* field, if *Country* is *USA,* or otherwise hide:

JavaScript OnLoad event

```
var ctrl = Runner.getControl(pageid, "Country");
if(ctrl.getValue() != "USA")
pageObj.hideField("State");
ctrl.on('change', function(){
 if (this.getValue() == "USA"){
    pageObj.showField("State");
 }
 else {
    pageObj.hideField("State");
 }
});
```

See also

- JavaScript API: hideField()
- Method: showField()
- Method: hideField()

getTabs

Returns RunnerTabs object, which represents a single tab group. If there are several tab groups on the page, N is a zero-based tab group index. Call this function without parameters will return first tab group. If N is more than number of tab groups null will be returned.

Syntax

```
getTabs( [n] )
```

Arguments

n - zero based index of a tab group. If no argument specified, returns first tab group on the page.

Return value

Returns RunnerTabs object.

Example

Activate third tab in the first tab group.

```
var tabs = pageObj.getTabs(0);
tabs.activate(2);
```

See also

- Tabs/Sections API

3.2.5.5 SearchController object

About SearchController object

SearchController object allows you to manage the search panel. It is available in the Javascript OnLoad event on the List/Report/Chart pages (pages with Search panel) and Advanced Search page.

Use *getSearchController()* method to get the search panel object:

```
var srch = pageObj.getSearchController();
```

Methods

Method	Description
addField()	Adds a field to the search panel and calls the callback function.
clear()	Deletes all fields from the search panel.

deleteField()	Deletes a field from the search panel.
display()	Hides or shows the search panel, shows search panel as popup.
getSearchFields()	Returns search fields related to all fields or only specified one.
toggleCriteria()	Hides or shows criteria block on the search panel, sets values of the search criteria.
toggleOptions()	Hides or shows options on the search panel.

See also

- SearchField object

Methods

addField

Adds a field to search panel. Then calls the callback function, passing SearchField object of added field as the argument. In the callback function you can use all the functionality available for SearchField object.

Syntax

```
srch.addField(fieldName, callback(field))
```

Arguments

fieldName - field name. Example: '*Make*'.

callback(field) - call to callback function. *field* is the SearchField object of added field. This argument may be omitted.

Return value

No return value.

Example 1

Add the *Make* field to the search panel:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
srch.addField('Make');
```

Example 2

Add the *Price* field to the search panel and set search parameters:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();

srch.addField('Price', function( field ) {
    field.getControl().setValue("20000");
    field.setOption(Runner.search.options.LESS_THAN);
});
```

Example 3

Add five *Make* fields to the search panel:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields('Make');
var count = field.length;

for (i=count; i<5; i++)
    srch.addField('Make');
```

See also

- SearchField object

clear

Deletes all search fields from the search panel.

Syntax

```
srch.clear();
```

Arguments

No arguments.

Return value

No return value.

deleteField

Deletes all entries of specified field from the search panel.

Syntax

```
srch.deleteField(fieldName);
```

Arguments

fieldName - field name. Example: '*Make*'.

Return value

No return value.

Example

Removes the *Make* field from the search panel:

```
                                                  JavaScript OnLoad event
var srch = pageObj.getSearchController();
srch.deleteField('Make');
```

display

Hides or shows the search panel, shows search panel as popup.

Syntax

```
srch.display(value);
```

Arguments

value - one of the values listed below:

Value	Description
hide	Hides search panel.
popup	Shows search panel as popup.
show	Shows search panel.

Return value

No return value.

Example

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
srch.display("show");
```

getSearchFields

Returns search fields related to all fields or only specified one.

Syntax

```
var fields = srch.getSearchFields(fieldName);
```

Arguments

fieldName - field name. Example: '*Make*'. *fieldName* argument may be omitted.

Return value

If the argument is specified, then returns the array of search fields related to specified field. If the argument is not specified, then returns the array of search fields related to all fields.

Example

Get an array of SearchField objects for the *Make* field:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields('Make');
for(var i=0; i<field.length; i++){
  // do something with SearchField objects
}
```

toggleCriteria

Hides or shows criteria block on the search panel, sets values of the search criteria.

Syntax

```
srch.toggleCriteria(value);
```

Arguments

value - one of the values listed below:

Value	Description
all	Sets search criteria to the *"all"* value.
any	Sets search criteria to the *"any"* value.
hide	Hides criteria block.
show	Shows criteria block.

Return value

No return value.

Example

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
srch.toggleCriteria('show');
```

See also

- Method: toggleOptions()

toggleOptions

Hides or shows options on the search panel.

Syntax

```
srch.toggleOptions(value);
```

Arguments

value - one of the values listed below:

Value	Description
hide	Hides search panel.

show	Shows search panel.

Return value

No return value.

Example

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
srch.toggleOptions('show');
```

See also

- Method: toggleCriteria()

3.2.5.6 SearchField object

About SearchField object

SearchField object allows to manage search fields on the search panel. It is used along with SearchController object. It is available in the Javascript OnLoad event on the List/Report/Chart pages (pages with Search panel) and Advanced Search page.

Use *getSearchFields()* method to get the array of search fields:

```
var srch = pageObj.getSearchController();
var fields = srch.getSearchFields();
```

Methods

Method	Description
addOption()	Adds the option to the list of search options.
getControl()	Returns the control object of the search field.
getName()	Returns the field name.
getOption()	Returns the option for search field that is currently selected.
getOptions()	Returns the list of all options that are currently available on the search panel.
getSecondControl()	Returns the second control object of the search field.

remove()	Removes the field from the search panel.
removeOption()	Removes the option from the list of search options.
setOption()	Selects the option for search field.

Search options

Option name	Description
BETWEEN	Search option "Between".
CONTAINS	Search option "Contains".
EMPTY	Search option "Empty".
EQUALS	Search option "Equals".
LESS_THAN	Search option "Less than".
MORE_THAN	Search option "More than".
STARTS_WITH	Search option "Starts with".
NOT_BETWEEN	Search option "NOT Between".
NOT_CONTAINS	Search option "NOT Contains".
NOT_EMPTY	Search option "NOT Empty".
NOT_EQUALS	Search option "NOT Equals".
NOT_LESS_THAN	Search option "NOT Less than".
NOT_MORE_THAN	Search option "NOT More than".
NOT_STARTS_WITH	Search option "NOT Starts with".

See also

- SearchController object

Methods

addOption

Adds an option to the list of search options on the search panel (CONTAINS, EQUALS etc.).

Syntax

```
field.addOption(option);
```

Arguments

option - one of the available search options with the following prefix *Runner.search.options*. Example: *Runner.search.options.MORE_THAN*, *Runner.search.options.NOT_EMPTY*.

Return value

No return value.

Example

Add *Empty* option to the list of search options:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
field.addOption(Runner.search.options.EMPTY);
```

getControl

Returns control object of the search field.

Syntax

```
var ctrl = field.getControl();
```

Arguments

No arguments.

Return value

Returns the control object.

Example

Get control object of the search field:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
var ctrl = field.getControl();
```

getName

Returns the field name.

Syntax

```
var fName = field.getName();
```

Arguments

No arguments.

Return value

Returns name of the current field.

Example

Get field name of the search field:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
var fName = field.getName();
```

getOption

Returns the option that is currently selected on the search panel (CONTAINS, EQUALS etc.).

Syntax

```
var option = field.getOption();
```

Arguments

No arguments.

Return value

Returns the option that is currently selected on the search panel. See search options list.

Example

Get search option of the search field:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
var option = field.getOption();
```

getOptions

Returns the list of search options that are currently available for selection.

Syntax

```
var options = field.getOptions();
```

Arguments

No arguments.

Return value

Returns the array of available search options. See search options constants list.

Example

Get the list of search options:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
var options = field.getOptions();
```

getSecondControl

Returns the second control object of the search field (for *between* option).

Syntax

```
var ctrl = field.getSecondControl();
```

Arguments

No arguments.

Return value

Returns the second control object of the search field.

Example

Get the second control of the search field on the search panel:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
var ctrl = field.getSecondControl();
```

remove

Removes a field from the search panel.

Syntax

```
field.remove();
```

Arguments

No arguments.

Return value

No return value.

Example

Remove search field from the search panel:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
field.remove();
```

removeOption

Removes an option from the list of search options on the search panel (CONTAINS, EQUALS etc.).

Syntax

```
field.removeOption(option);
```

Arguments

option - one of the available search options with the following prefix *Runner.search.options*. Example: *Runner.search.options.MORE_THAN*, *Runner.search.options.NOT_EMPTY*.

Return value

No return value.

Example

Remove the *Equals* option from the list of search options:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
field.removeOption(Runner.search.options.EQUALS);
```

setOption

Selects option for search field on the search panel (CONTAINS, EQUALS etc.).

Syntax

```
field.setOption(option);
```

Arguments

option - one of the available search options with the following prefix *Runner.search.options*. Example: *Runner.search.options.MORE_THAN*, *Runner.search.options.NOT_EMPTY*.

Return value

No return value.

Example

Select *Between* option for search field on the search panel:

JavaScript OnLoad event

```
var srch = pageObj.getSearchController();
var field = srch.getSearchFields()[0];
field.setOption(Runner.search.options.BETWEEN);
```

3.2.5.7 Examples

How to ask for confirmation before saving record

To ask a user for confirmation before saving a record use the following code in Javascript OnLoad event for Edit or Add page.

Note: Change the values listed in red to match your specific needs.

```
this.on('beforeSave', function(formObj, fieldControlsArr, pageObj){
   if (confirm('Save record?')){
      return true;
   }else{
      Runner.delDisabledClass ( pageObj.saveButton );
   return false;
   }
});
```

See also

- Javascript API

How to calculate values on the fly

Let's say there are three fields on the add/edit page: Price, Quantity and Total. To calculate total value on the fly use Javascript code (add it to the Add page: JavaScript OnLoad event or Edit page: JavaScript OnLoad event on the Events tab).

Note: Change the values listed in red to match your specific needs.

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotal = Runner.getControl(pageid, 'Total');

function func() {
   ctrlTotal.setValue(Number(ctrlPrice.getValue()) *
Number(ctrlQuantity.getValue()));
};

ctrlPrice.on('keyup', func);
ctrlQuantity.on('keyup', func);
```

See also

- Javascript API

How to change font size in text box

To change font size in all text boxes placed on a page use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
$("input[type=text]").css('fontSize', '120%');
```

See also

- Javascript API

How to change font in "edit" controls

You can change the font in the "edit" controls in the following ways:

Note: Change the values listed in red to match your specific needs.

1. Change the font in one "edit" control

Use the following code in Javascript OnLoad event:

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.addStyle('font-size: 12px; color: red;');
```

To get more information about using Javascript API, see Javascript API.

2. Change the font for all "edit" controls on one page

Proceed to the Style Editor, open the page you need to modify, click **modify** above styles list, click on the **Custom CSS** tab and add the following code there:

```
input,textarea {
```

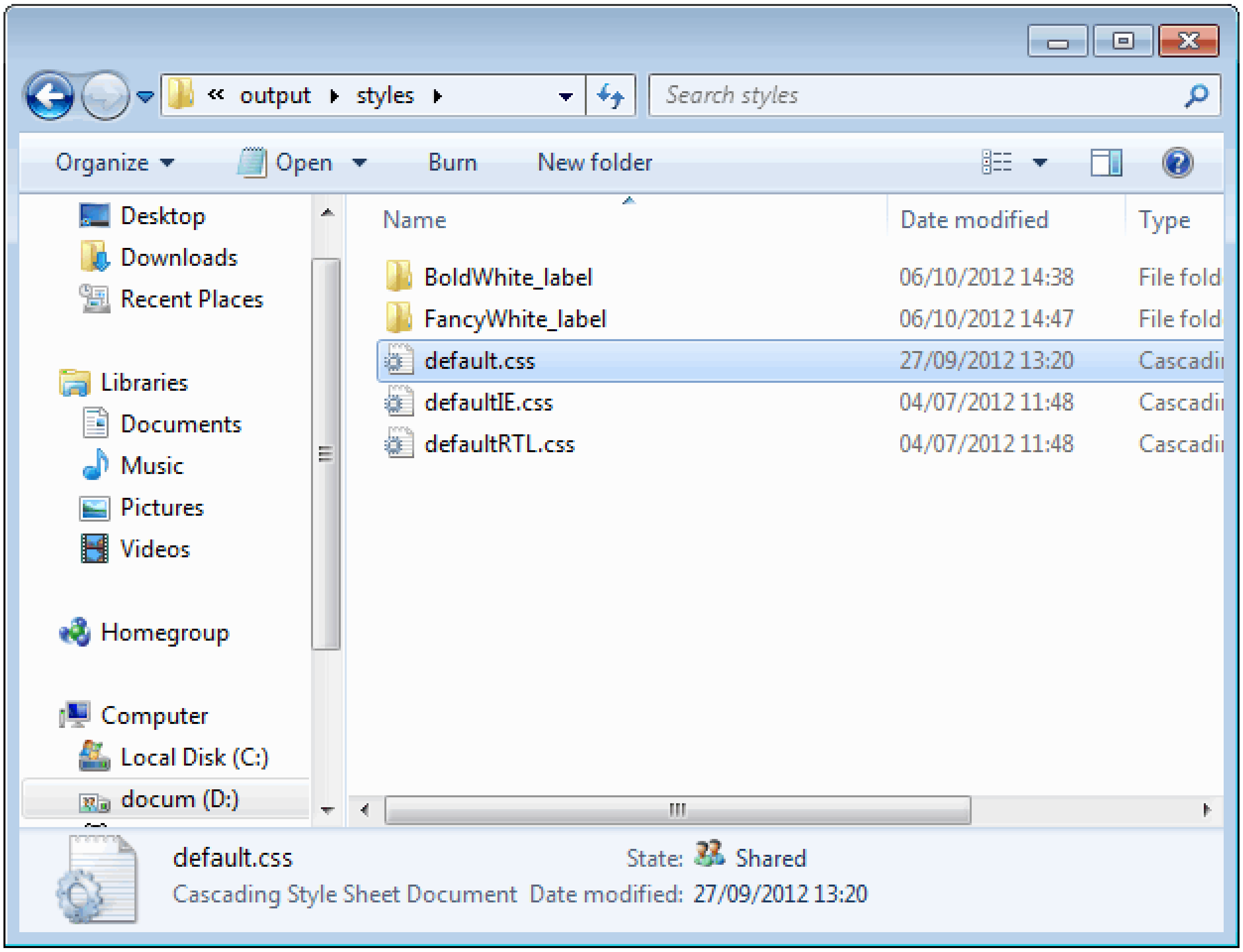
```
color:#003333;
font-family:Verdana,Arial,SunSans-Regular,Sans-Serif;
font-size:12pt;
}
```

or use the following code in Javascript OnLoad event:

```
$("input, textarea").css( {fontSize: '120%', color: 'red'} );
```

3. Change the font for all "edit" controls in the project

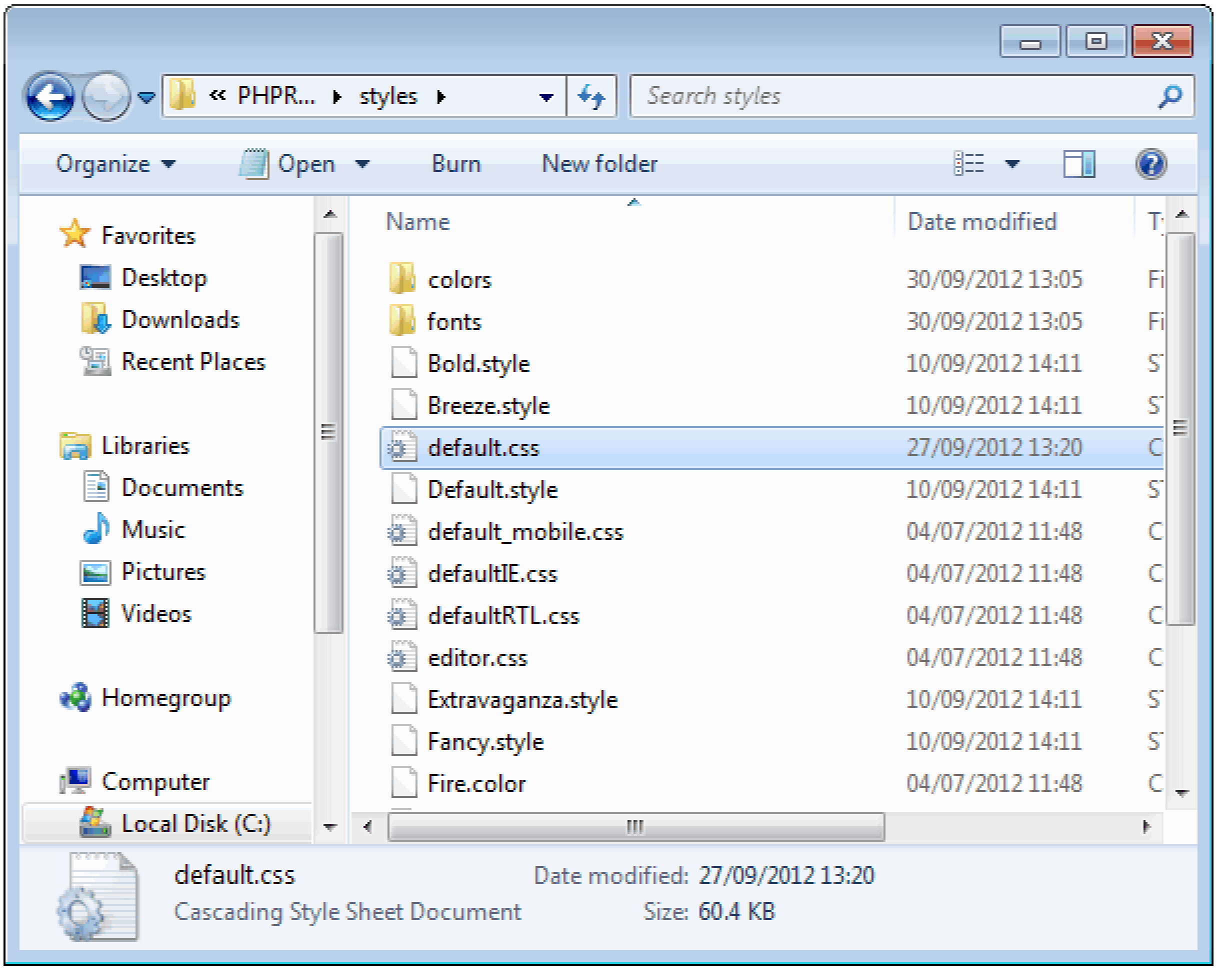
Change the description of the control style (INPUT, TEXTAREA) in the file *styles\default.css* in the generated project files



Name	Date modified	Type
BoldWhite_label	06/10/2012 14:38	File fold
FancyWhite_label	06/10/2012 14:47	File fold
default.css	27/09/2012 13:20	Cascadi
defaultIE.css	04/07/2012 11:48	Cascadi
defaultRTL.css	04/07/2012 11:48	Cascadi

default.css State: Shared
Cascading Style Sheet Document Date modified: 27/09/2012 13:20

or directly in the PHPRunner files - the file *PHPRunner10.0\styles\default.css*.



Name	Date modified	T
colors	30/09/2012 13:05	Fi
fonts	30/09/2012 13:05	Fi
Bold.style	10/09/2012 14:11	S
Breeze.style	10/09/2012 14:11	S
default.css	27/09/2012 13:20	C
Default.style	10/09/2012 14:11	S
default_mobile.css	04/07/2012 11:48	C
defaultIE.css	04/07/2012 11:48	C
defaultRTL.css	04/07/2012 11:48	C
editor.css	04/07/2012 11:48	C
Extravaganza.style	10/09/2012 14:11	S
Fancy.style	10/09/2012 14:11	S
Fire.color	04/07/2012 11:48	C

default.css Date modified: 27/09/2012 13:20
Cascading Style Sheet Document Size: 60.4 KB

How to change width of edit box with AJAX popup

To change width of edit box with AJAX popup use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
var ctrl = Runner.getControl(pageid, 'Make');
ctrl.getDispElem().css("width", "200px");
```

See also

- Javascript API

How to change width of text field on Quick Search panel

To change width of text field on Quick Search panel use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
$("input[name^='value_make'], select[name^='value_make']").width(150);
```

See also

- Javascript API

How to control Inline Add/Edit functionality from script

To control inline Add/Edit functionality from script, add one of the following code snippets to JavaScript OnLoad event of the List page:

Note: Change the values listed in red to match your specific needs.

1. Simulate "inline add" click:

```
pageObj.inlineAdd.inlineAdd();
```

2. Inline edit all records:

```
pageObj.inlineEdit.editAllRecs();
```

3. Inline edit record number X, where X ranges from 1 to number of records on the page:

```
// get list of record IDs
var recsId = pageObj.inlineEdit.getRecsId();

// click inline edit link
pageObj.inlineEdit.editRecById(recsId[X]);
```

4. Open details table inline preview:

```
// get list of record IDs
var recsId = pageObj.inlineEdit.getRecsId();

// details table name
var dTableName = 'DetailsTableName';
var dp = this.dpObjs[dTableName];
var row = dp.getRowById( recsId[X] );
dp.openDetailsTabs( row, $(null) );
```

5. Edit record number X in popup (when **Edit in popup** mode is turned on):

```
// get list of record IDs
var recsId = pageObj.inlineEdit.getRecsId();
```

```
// click Edit link
$("#editLink"+recsId[X]).click();
```

6. View record number X in popup (when **View in popup** mode is turned on):

```
// get list of record IDs
var recsId = pageObj.inlineEdit.getRecsId();

// click View link
$("#viewLink"+recsId[X]).click();
```

7. Open **Add** page in popup:

```
$("#addButton"+pageid).click();
```

8. Make new record appear at the end of the list (when **Inline** or '**Add in popup'** mode is enabled):

```
pageObj.addNewRecordsToBottom = true;
```

How to execute Javascript code after Inline Add or Edit

To refresh List page when inline adding or editing is finished, use the following code in Javascript OnLoad event of List page. This code can also work when Add or Edit page are displayed in popup.

1. Inline Add:

```
this.on('afterInlineAdd', function( fieldsData ) {
 location.reload();
} )
```

2. Inline Edit:

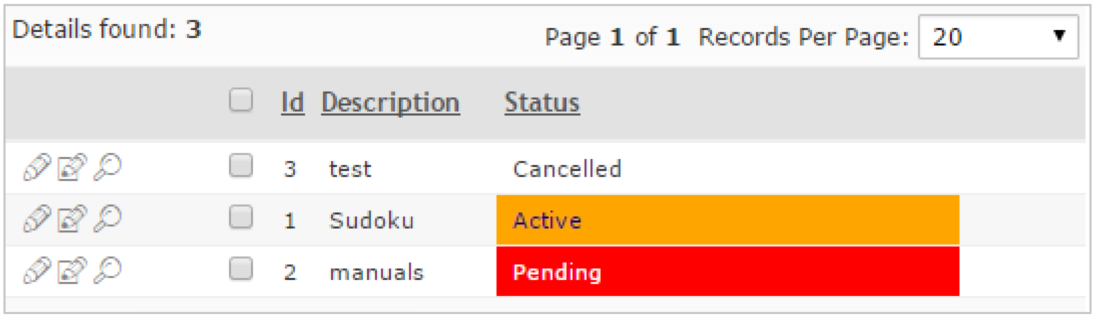
```
this.on('afterInlineEdit', function( fieldsData ) {
        location.reload();
} )
```

Description of *fieldsData* structure:

```
fieldsData = {
 name: field name,
 value: raw field value as it appears in the database,
 text: field value with formatting applied if any,
 container:  jQuery object, a container, where field value will be inserted
}
```

3. How to change the color of text and background of a field after Inline Add or Edit.

```
function funcAfter(fieldsData) {
    for (f in fieldsData) {
        var field = fieldsData[f];
        if (field.name == 'status') {
            if (field.value == 'Pending') {
                field.container.closest('td').css('background', 'red');
                field.container.closest('td').css('color', 'white');
            } else if (field.value == 'Active') {
                field.container.closest('td').css('background', 'orange');
                field.container.closest('td').css('color', 'darkblue');
            }
        }
    }

}

this.on('afterInlineEdit', funcAfter);
this.on('afterInlineAdd', funcAfter);
```



Details found: **3** Page **1** of **1** Records Per Page: 20 ▼

	□	Id	Description	Status
	□	3	test	Cancelled
	□	1	Sudoku	Active
	□	2	manuals	Pending

4. How to redirect user to View page after adding new record in popup.

Do not forget to replace "documents" with your table name.

```
function funcAfter(fieldsData) {
    for (f in fieldsData) {
        var field = fieldsData[f];
        if (field.name == 'id') {
            window.location.href = "documents_view.php?editid1=" +
field.value;
        }
    }

}

this.on('afterInlineAdd', funcAfter);
```

See also

- Javascript API

How to convert input into upper case

To convert user entered data into upper case, use the following code in Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
var ctrl = Runner.getControl(pageid, 'FieldName');
ctrl.addStyle('text-transform: uppercase;');
```

See also

- Javascript API

How to display all Options on Search panel

To display all Options on the Search panel use the following code in List page: Javascript OnLoad event.

Note: Change the values listed in red to match your specific needs.

```
$("[id^='showHideSearchType']").click();
```

See also

- Javascript API

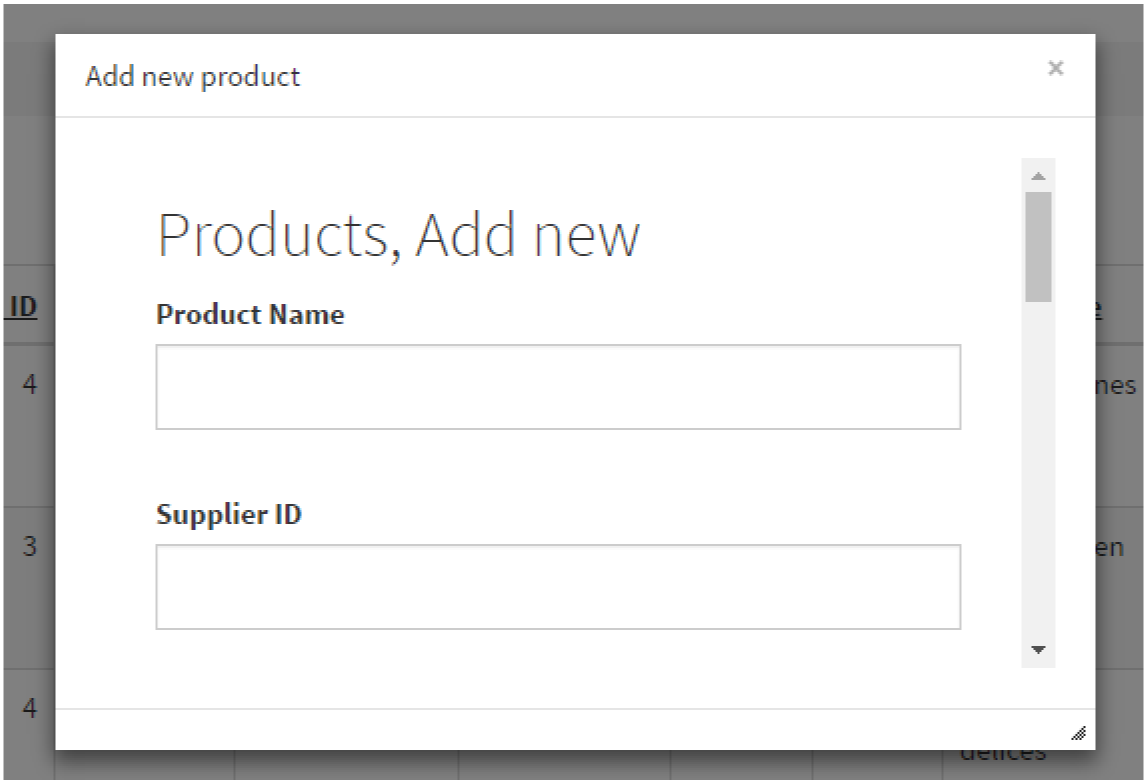
How to display any page in Bootstrap popup window.

There is an easy way to display any page in Bootstrap popup window, **Runner.displayPopup()** function.

There is only one mandatory parameter, URL of the page to be displayed. You can use this function anywhere you can use Javascript code i.e. in Javascript OnLoad event or ClientBefore event of custom button.

Here is how you can display Add page of Products table in popup:

```
Runner.displayPopup( {
       url: "products_add.php"
});
```



Here is the complete list of parameters:

```
url: URL of the page to be displayed

html: instead of specifying URL you can supply HTML code to be displayed in
popup window

header: to be displayed in popup header section

footer: to be displayed in popup footer section

afterCreate: function to be called after popup window is created

beforeClose: function to be called before popup window is closed.
Return false to prevent popup from being closed. Return true to proceed with
closing.

width: popup width in pixels

height: popup height in pixels
```

More examples.

Simple popup window

```
var win = Runner.displayPopup( {
      url: "products_add.php",
      width: 700,
      height: 500,
      header: 'Add new product'
});
```

Using HTML instead of URL

```
var win = Runner.displayPopup( {
            html: "<h1>Hello world!</h1><p>It works</p>",
            header: 'Custom popup text'
});
```

Use of close()

```
var win = Runner.displayPopup( {
      url: "products_add.php",
      width: 500,
      height: 300,
      header: 'Add new product'
});

alert('That was an example of popup window');

win.close();
```

Use of afterCreate() event

```
var win = Runner.displayPopup( {
      url: "products_add.php",
      header: 'Add new product',
      afterCreate: function(win) {
            win.setWidth(700);
            win.setHeight(500);
      }
});
```

Add 'Close window' link to the footer

```
var win = Runner.displayPopup( {
      url: "products_add.php",
      header: 'Add new product',
      footer: '<a href="#" onclick="window.win.close();">Close window</a>',
      afterCreate: function(win) {
            window.win = win;
      }
});
```

Use of beforeClose() event

In this function you can return false to prevent window from being closed.

Do not allow to close window if 'Product Name' field is empty.

```
var win = Runner.displayPopup( {
      url: "products_add.php",
      header: 'Add new product',
      footer: '<a href="#" onclick="window.win.close();">Close window</a>',
      afterCreate: function(win) {
            window.win = win;
      },
      beforeClose: function(win) {
            if
($('iframe').contents().find("input#value_ProductName_1").val()=="")
                  return false;
            else
                  return true;
      }
});
```

Show 'View customer' button on each row of Orders List page

Insert a button into Orders List page grid.

Server code

```
$record = $button->getCurrentRecord();
$result["CustomerID"] = $record["CustomerID"];
```

ClientAfter code

```
var win = Runner.displayPopup( {
      url: "customers_view.php?editid1="+result["CustomerID"],
      width: 700,
      height: 500,
      header: 'View customer'
});
```

Show Add page in popup, close popup on 'Save' and refresh the List page

There is added a button to the list page to display Add page in popup and once record is saved we close the popup and refresh the list page to show a new record.

Button's ClientBefore code

```
window.popup = Runner.displayPopup({
    url: Runner.pages.getUrl("carsmake","add"),
    width: 700,
    height: 700,
    header: 'Add Category'
});
```

AfterAdd event

```
$pageObject->setProxyValue('saved', true);
$pageObject->stopPRG = true;
```

Javascript OnLoad event of Add page

```
if ((proxy['saved']) && window.parent && window.parent.popup) {
    window.parent.location.reload();
    window.parent.popup.close();
}
```

If you need to close popup without refreshing the page you need to comment the line: `window.parent.location.reload();`

See also

- Javascript API

How to enable/disable a button

To enable/disable standard application buttons and manually added buttons, you need to complete the following steps. Note that this will not work for buttons inserted directly into the grid.

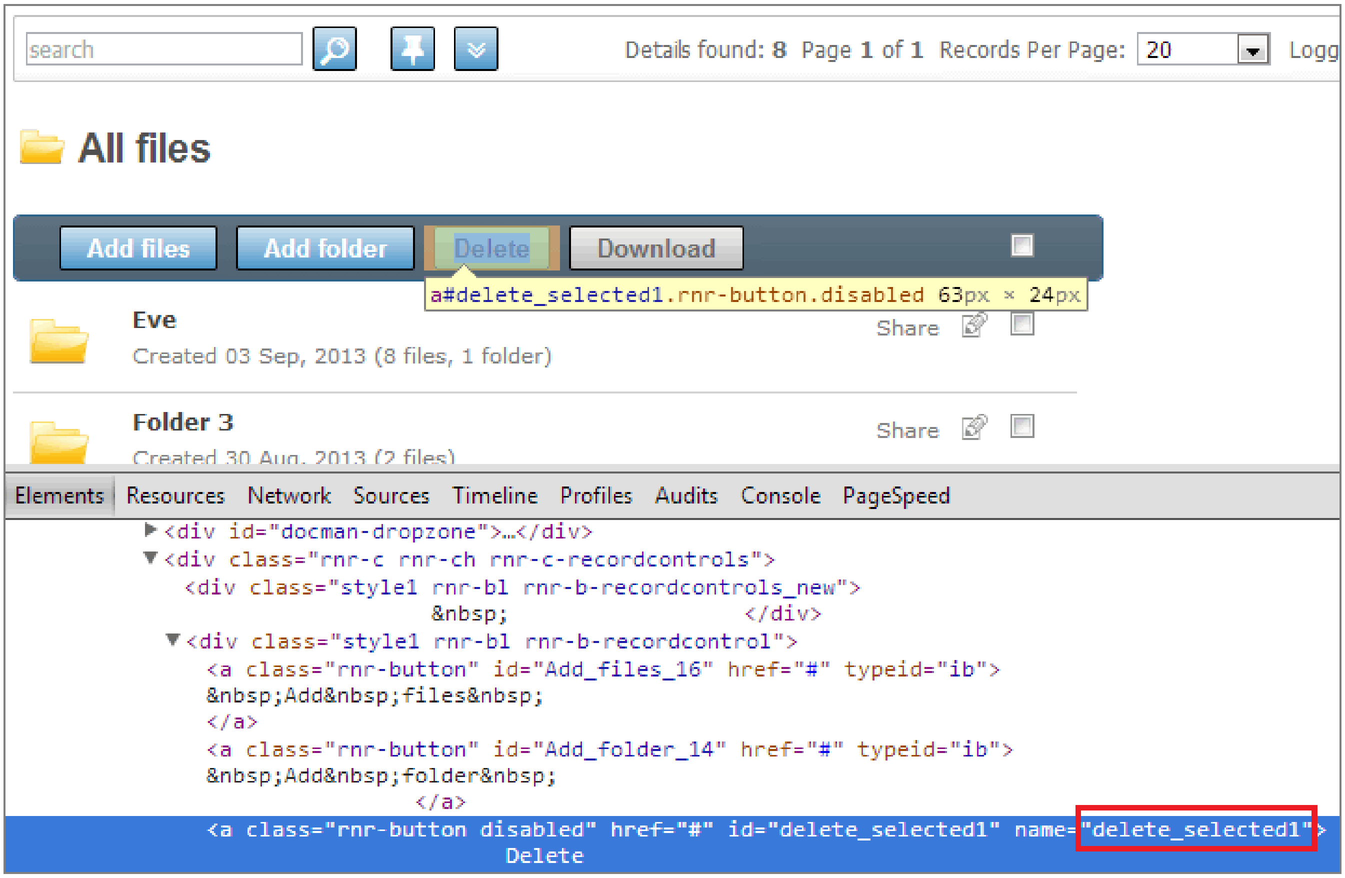
Note: Change the values listed in red to match your specific needs.

1. Get button ID

To find out id of manually added button, select the button in Visual Editor and switch to HTML mode. In the highlighted "button" code look for *id="..."*. In the example below button id is *My_Button*:

```
<A class="rnr-button" id="My_Button" href="#" typeid="ib">
My&nbsp;Button
</A>
```

In the case of standard application buttons you need to do the following to get button id: build application, open it in the browser and view the properties of the button using Chrome Developer Tools or Firebug.


search
Details found: 8 Page 1 of 1 Records Per Page: 20
Logg
All files
Add files
Add folder
Delete
Download
a#delete_selected1.rnr-button.disabled 63px × 24px
Eve
Created 03 Sep, 2013 (8 files, 1 folder)
Share
Folder 3
Created 30 Aug, 2013 (2 files)
Share
Elements
Resources
Network
Sources
Timeline
Profiles
Audits
Console
PageSpeed

```
▶<div id="docman-dropzone">…</div>
▼<div class="rnr-c rnr-ch rnr-c-recordcontrols">
   <div class="style1 rnr-bl rnr-b-recordcontrols_new">
                  &nbsp;                      </div>
 ▼<div class="style1 rnr-bl rnr-b-recordcontrol">
     <a class="rnr-button" id="Add_files_16" href="#" typeid="ib">
     &nbsp;Add&nbsp;files&nbsp;
     </a>
     <a class="rnr-button" id="Add_folder_14" href="#" typeid="ib">
     &nbsp;Add&nbsp;folder&nbsp;
                     </a>
     <a class="rnr-button disabled" href="#" id="delete_selected1" name="delete_selected1">
                            Delete
```

Use only the initial part of the button id (without numbers) in the code, such as *delete_selected*.

Note that button id is case-sensitive.

2. Add code

Add the following code to any Javascript OnLoad event or ClientAfter/ClientBefore event of other buttons.

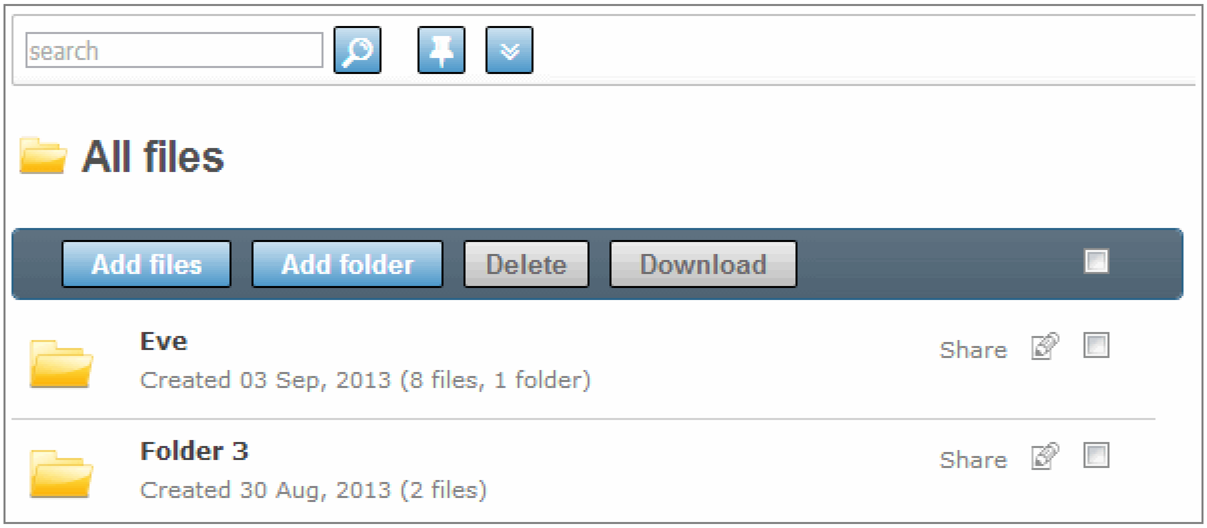
To disable a button:

```
var id = "My_Button";
var button = $("[id^=" + id + "]");
Runner.addDisabledClass(button);
```

To enable a button:

```
var id = "My_Button";
var button = $("[id^=" + id + "]");
Runner.delDisabledClass(button);
```

Example of how disabled buttons are seen in browser:



Eve
Created 03 Sep, 2013 (8 files, 1 folder)

Folder 3
Created 30 Aug, 2013 (2 files)

How to hide 'Edit selected'/'Delete selected' buttons

Sometimes you need to show **Edit selected** and **Delete selected** buttons only when there are selected records on the List page. For this purpose add the following code to List page: JavaScript OnLoad event:

```
var $editSelected = $('#edit_selected' + pageid),
  $deleteSelected = $('#delete_selected' + pageid);
$editSelected.hide();
$deleteSelected.hide();

$('input[type=checkbox][name=^selection], #select_all' + pageid +',
#chooseAll_' + pageid).change( function(e) {
  var $target = $(e.target),
    selBoxes;

  if ( $target.attr('name') == 'selection[]' || $target.attr('id') ==
'chooseAll_' + pageid ) {
   selBoxes = pageObj.getSelBoxes( pageid );
   $editSelected.toggle( selBoxes.length > 0 );
    $deleteSelected.toggle( selBoxes.length > 0 );
 }
});
```

See also

- Javascript API

How to pass values from PHP code to Javascript

To pass values from ASP code to Javascript, use the setProxyValue() method.

How to refresh List page after Edit in popup

By default PHPRunner updates List page automatically with new data. However, if you perform some actions behind the scene like adding a new record in AfterEdit event you might need to refresh List page manually after 'Edit in popup' action.

To do that add the following code to List page: Javascript OnLoad event

```
this.on('afterInlineEdit', function( fieldsData ) {
location.reload();
}):
```

To refresh grid on the List page without page reloading, use the following code in Javascript OnLoad event. Note that option **AJAX search, pagination and sorting** should be enabled (Choose pages -> List page settings).

```
Runner.runnerAJAX(Runner.pages.getUrl(pageObj.tName, pageObj.pageType)+"?
a=return",
     pageObj.ajaxBaseParams,
    function(respObj){
      pageObj.pageReloadHn.call(pageObj, respObj)
     });
```

If you need to use both snippets at the same time, add the following code to the List page Javascript OnLoad event. Note that option **AJAX search, pagination and sorting** should be enabled (Choose pages -> List page settings).

```
window.listPage = pageObj;
```

And add the following code to the Edit page Javascript OnLoad.

```
this.on('afterSave', function() {
var pageObj = window.listPage;
Runner.runnerAJAX(Runner.pages.getUrl(pageObj.tName, pageObj.pageType)+"?
a=return",
pageObj.ajaxBaseParams,
function(respObj){
pageObj.pageReloadHn.call(pageObj, respObj)
});
});
```

See also

- Javascript API

How to show dropdown list of US states

To show dropdown list of US states if US was selected in the country list or hide it otherwise, use the following code in Javascript OnLoad event on Add/Edit pager in popup and inline.

Note: Change the values listed in red to match your specific needs.

```
var ctrlCountry = Runner.getControl(pageid, 'country');
var ctrlState = Runner.getControl(pageid, 'state');

ctrlCountry.on('change', function(e) {
    if (this.getValue() == 'US') {
        ctrlState.show();
    } else {
        ctrlState.hide();
    }
});
```

You may want to hide field label as well. Use the following code to hide or show the whole table row with state edit control based on country field selection. It will work in popup and inline.

```
var ctrlCountry = Runner.getControl(pageid, 'country');

ctrlCountry.on('change', function(e) {
    if (this.getValue() == 'US') {
        pageObj.showField("state");
    } else {
        pageObj.hideField("state");
    }
});
```

See also

- Javascript API

How to work with tabs

Note: The following example works with old, non-Bootstrap layouts. For Bootstrap layouts use Tabs/Sections API

To manipulate a tab, you need to know the tab group name. To find this name proceed to the **Visual Editor** page, select the page with tab, make tab group selected and switch to the HTML mode. Tab group name will be highlighted (e.g. *{$TabGroup New_tab1}*). Here *New_tab1* is the tab group name. Now paste it to the code below.

Besides the tab group name you need to know the tab index. Tab index is zero based: 0 - the first tab, 1 - the second tab etc. Now you can use one of the following code samples.

Note: Change the values listed in red to match your specific needs.

1. To make the tab selected, use the following code in Javascript OnLoad event:

```
// make second tab selected
var tab = "New_tab1",
   myTabs = pageObj.tabs['tabGroup_'+tab];

myTabs.selectChild(1);
```

2. To make the tab disabled, use the following code in Javascript OnLoad event:

```
var tab = "New_tab1",
   prevSelected = 0,
   myTabs = pageObj.tabs["tabGroup_" + tab];

myTabs.item(1).disable();

myTabs.on('click', function(e) {
   if ( this.get("selection").get("disabled") ) {
      myTabs.selectChild( prevSelected );
   }
});

myTabs.on('selectionChange', function(e) {
   prevSelected = e.prevVal.get('index');
});
```

3. To remove the tab, use the following code in Javascript OnLoad event:

```
var tab = "New_tab1",
   myTabs = pageObj.tabs['tabGroup_'+tab],
   tabIndex = 1,   // 0 means first tab, 1 means second tab etc
   savedTab;

if ( myTabs.item(tabIndex) ) {
   savedTab = myTabs.item(tabIndex).getAttrs();
   myTabs.remove(tabIndex);
}
```

4. To hide the tab, use the following code in Javascript OnLoad event:

```
var tab = "New_tab1",
   myTabs = pageObj.tabs['tabGroup_'+tab],
   tabIndex = 1,   // 0 means first tab, 1 means second tab etc
   tabToHide = myTabs.item( tabIndex ),
   siblingTab;

tabToHide.get("boundingBox").hide();
tabToHide.hide();

if ( tabToHide.get('selection') ) {
   siblingTab = tabToHide.next() || tabToHide.previous()
   if ( siblingTab && siblingTab.get('visible') ) {
      myTabs.selectChild( siblingTab.get("index") );
   }  else {
      tabToHide.set('selection', 0);
   }
}
```

5. To show the hidden tab, use the following code in Javascript OnLoad event:

```
var tab = "New_tab1",
   myTabs = pageObj.tabs['tabGroup_' + tab],
   tabIndex = 1,    // 0 means first tab, 1 means second tab etc
   tabToShow = myTabs.item( tabIndex );

tabToShow .get("boundingBox").show();
myTabs.selectChild( tabIndex );
```

See also

- Javascript API

How to control multi-step pages

In event like Javascript **OnLoad of Add** or **Edit pages**, use the following code:

```
// 0 - first tab, 1 - second tab etc
pageObj.setCurrentStep(1);
```

This code will work in Javascript OnLoad event or ClientBefore/ClientAfter event of custom buttons.

How to return other field control in the inline mode

ctrl.getPeer(field);

Returns other field control from the same page of the same row in the inline mode. Works similatr to the Runner.getControl function, but should be used in the field events

Example

Gets the 'price' control in the inline mode and then sets its value to 1000.

```
var ctlPrice = ctrl.getPeer('price');
ctlPrice.setValue( 1000 );
```

3.2.6 Tabs/Sections Javascript API

3.2.6.1 About Tabs/Sections API

This API allows you to manage tabs and sections added to Add, Edit and View pages. This API is only designed to work with Bootstrap layouts. All tab group and tab indexes are zero-based. 0 - first tab, 1 - second tab etc.
These methods will work in the JavaScript OnLoad event on the Add/Edit/View pages.

Methods

Method	Description
getTabs([n])	returns RunnerTabs object, which represents a single tab group
Tabs methods	
count()	returns the number tabs in the Tab control object
activate(n)	activates/shows n-th tab
activeIdx()	gets an index of the currently selected tab.
hide(n)	hides n-th tab (zero-based).
show(n)	shows n-th tab (zero-based)
disable(n)	makes n-th tab disabled
enable(n)	makes n-th tab enabled
headerElement(n)	returns jQuery object of tab header
bodyElement(n)	returns jQuery object of tab content
addTab(headerHtml, paneHtml)	creates a new tab and adds it to the end of tab group
moveTo(n, m)	changes tabs order. n-th tab will be moved to m-th position. m-th tab will be moved to n-th position.
Sections methods	
getSectionCount()	returns number of sections on the page
getSection(n)	returns RunnerSection object representing n-th section on the page (zero-based)
headerElement()	
bodyElement()	
expand()	expands section

collapse()	collapses section

3.2.6.2 Tabs methods

count

Syntax

```
count()
```

Arguments

No arguments.

Return value

Returns the number of tabs in the Tab control object.

Example

Count number of tabs in a tab group.

```
var tabs = pageObj.getTabs();
var count = tabs.count();
```

See also

- RunnerPage.getTabs

activate

Activates/shows n-th tab. Tab index is zero-based.

Syntax

```
activate( n )
```

Arguments

n - zero based index of the tab to be activated

Return value

No return value.

Example

Activate third tab in the first tab group.

```
var tabs = pageObj.getTabs(0);
tabs.activate(2);
```

See also

- RunnerPage.getTabs
- ActiveIdx

activeIdx

Gets an index of the currently selected tab.

Syntax

```
activeIdx()
```

Arguments

No arguments.

Return value

Zero based index of the active tab.

Example

Gets an index of the currently selected tab.

```
var tabs = pageObj.getTabs();
var idx = tabs.activeIdx();
```

See also

- RunnerPage.getTabs
- Activate

hide

Hides n-th tab (zero-based).

Syntax

```
hide( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

Hides the third tab.

```
var tabs = pageObj.getTabs();
tabs.hide(2);
```

See also

- RunnerPage.getTabs
- RunnerTabs.show

show

Shows n-th tab (zero-based).

Syntax

```
show( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

Shows the third tab.

```
var tabs = pageObj.getTabs();
tabs.show(2);
```

See also

- RunnerPage.getTabs
- RunnerTabs.hide

disable

Makes n-th tab disabled. Tab remains visible.

Syntax

```
disable( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

Makes the fourth tab disabled.

```
var tabs = pageObj.getTabs();
tabs.disable(3);
```

See also

- RunnerPage.getTabs
- RunnerTabs.enable

enable

Makes n-th tab enabled.

Syntax

```
enable( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

Makes the fourth tab enabled.

```
var tabs = pageObj.getTabs();
tabs.enable(3);
```

See also

- RunnerPage.getTabs
- RunnerTabs.disable

headerElement

Returns the jQuery object of a tab header.

Syntax

```
headerElement( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

How to change a tab's title to "Active tab" written in bold.

```
var tabs = pageObj.getTabs();
```

```
tabs.headerElement(1).html('<b>Active tab</b>');
```

See also

- RunnerPage.getTabs

bodyElement

Returns jQuery object of tab content.

Syntax

```
bodyElement( n )
```

Arguments

n - zero based index of the tab

Return value

No return value.

Example

Change tab content to read "This is tab content".

```
var tabs = pageObj.getTabs();
tabs.bodyElement(0).html( '<h1>This is tab content</h1>' );
```

See also

- RunnerPage.getTabs

addTab

Creates a new tab and adds it to the end of tab group.

Syntax

```
addTab( headerHtml, paneHtml )
```

Arguments

headerHtml

paneHtml

Return value

Returns index of new tab.

Example

Creates a new tab "New Tab" with content "New tab content" and adds it to the end of tab group.

```
var tabs = pageObj.getTabs();
tabs.addTab('New tab', 'New tab content');
```

See also

- RunnerPage.getTabs

moveTo

Changes tabs order. n-th tab will be moved to m-th position. m-th tab will be moved to n-th position.

If m is greater than number of tabs n-th tab will be moved to the end.

Syntax

```
moveTo( n, m )
```

Arguments

n - zero based index of the tab

m - zero based index of the tab

Return value

Returns new index of n-th tab.

Example

Third tab will be moved to fourth position. Fourth tab will be moved to third position.

```
var tabs = pageObj.getTabs();
```

```
tabs.moveTo(2,3);
```

3.2.6.3 Sections methods

getSectionCount

Syntax

```
getSectionCount()
```

Arguments

No arguments.

Return value

Returns number of sections on the page

Example

Count number of sections.

```
var sections = pageObj.getSectionCount();
```

getSection

Syntax

```
getSection( n )
```

Arguments

n - zero based index of the section.

Return value

Returns RunnerSection object representing n-th section on the page (zero-based).

Example

Returns the first section.

```
var sec = pageObj.getSection(0);
```

headerElement

Syntax

```
headerElement()
```

Arguments

No arguments.

Return value

No return value.

Example

Change first section title to "Active Section" written in bold.

```
var section = pageObj.getSection(0);
section.headerElement().find('a').html('<b>Active Section</b>');
```

bodyElement

Syntax

```
bodyElement()
```

Arguments

No arguments.

Return value

No return value.

Example

Change first section content to read "This is section content".

```
var section = pageObj.getSection(0);
section.bodyElement().html( '<h1>This is section content</h1>' );
```

expand

Expands section.

Syntax

```
expand()
```

Arguments

No arguments.

Return value

No return value.

Example

Expands the first section

```
var sec = pageObj.getSection(0);
sec.expand();
```

See also

- RunnerSection.collapse

collapse

Collapses section.

Syntax

```
collapse()
```

Arguments

No arguments.

Return value

No return value.

Example

Collapses the first section

```
var sec = pageObj.getSection(0);
sec.collapse();
```

See also

- RunnerSection.expand

3.2.7 Labels/Titles API

3.2.7.1 About Labels/Titles API

Labels/Titles API allows you to work with the table/field labels and titles. Table/field names are case-insensitive.
API functions work in events like AfterAppInit, AfterTableInit, BeforeProcess of the appropriate page.

If $language parameter is not specified, current language will be used.

Functions

Function	Description
getFieldLabel setFieldLabel	Gets or sets the field label.

getTableCaption setTableCaption	Gets or sets the table caption.
getProjectLogo setProjectLogo	Gets or sets the project logo.
getPlaceholder setPlaceholder	Gets or sets the field placeholder.
getFieldTooltip setFieldTooltip	Gets or sets the field tooltip.
getPageTitleTempl setPageTitleTempl	Gets or sets the page title.
getBreadcrumbsLabelTempl setBreadcrumbsLabelTempl	Gets or sets the breadcrumbs label template.

3.2.7.2 Methods

getFieldLabel

Gets the field label.

Syntax

```
Labels::getFieldLabel($table, $field, $language)
```

Arguments

$table - table name

$field - field name.

$language - language. If $language parameter is not specified current language will be used.

Return value

Field label. Labels can be set in the Label Editor.

Example

Gets the "carscars" table "descr" field label.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getFieldLabel("carscars", "descr");
```

See also

- setFieldLabel()

setFieldLabel

Sets the field label.

Syntax

```
Labels::setFieldLabel($table, $field, $label, $language)
```

Arguments

$table - table name

$field - field name.

$label - new field label. Labels can be set in the Label Editor.

$language - language. If $language parameter is not specified current language will be used.

Return value

No return value.

Example

Labels can be set in the Label Editor. This example shows how to change the "carscars" table "descr" field label to "New Description".

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setFieldLabel("carscars", "descr", "New Description");
```

See also

- getFieldLabel()

getTableCaption

Gets the table caption.

Syntax

```
Labels::getTableCaption($table, $language)
```

Arguments

$table - table name

$language - language. If $language parameter is not specified current language will be used.

Return value

Table caption. Captions can be set in the Label Editor.

Example

Gets the table "carscars" table caption.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getTableCaption("carscars");
```

See also

- setTableCaption()

setTableCaption

Sets the table caption.

Syntax

```
Labels::setTableCaption($table, $caption, $language)
```

Arguments

$table - table name

$caption - new table caption. Captions can be set in the Label Editor

$language - language. If $language parameter is not specified current language will be used.

Return value

No return value.

Example

Captions can be set in the Label Editor. This example shows how to change the "carscars" table caption to "New Cars".

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setTableCaption("carscars", "New Cars");
```

See also

- getTableCaption()

getProjectLogo

Gets the project logo.

Syntax

```
Labels::getProjectLogo($language)
```

Arguments

$language - language. If $language parameter is not specified current language will be used.

Return value

Project logo. Project logo can be set in the Label Editor.

Example

Gets the Project logo. Project logo can be set in the Label Editor.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getProjectLogo();
```

See also

- setProjectLogo()

setProjectLogo

Sets the project logo.

Syntax

```
Labels::setProjectLogo($label, $language)
```

Arguments

$label - Project logo. $label parameter may contain any text or HTML code

$language - language. If $language parameter is not specified current language will be used.

Return value

No return value.

Example

Sets the Project logo. Project logo can be set in the Label Editor.

AfterAppInit, AfterTableInit, BeforeProcess events

```
 Labels::setProjectLogo("<img
src='http://mywebsite.com/images/logo.png'>");
```

See also

- getProjectLogo()

getFieldTooltip

Gets the field tooltip.

Syntax

```
Labels::getFieldTooltip($table, $field, $language)
```

Arguments

$table - table name

$field - field name.

$language - language. If $language parameter is not specified current language will be used.

Return value

Field tooltip. Tooltips can be set in the Label Editor.

Example

Gets the "carscars" table "descr" field tooltip.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getFieldTooltip("carscars", "descr");
```

See also

- setFieldTooltip()

setFieldTooltip

Sets the field tooltip.

Syntax

```
Labels::setFieldTooltip($table, $field, $tooltip, $language)
```

Arguments

$table - table name

$field - field name.

$tooltip - field tooltip.

$language - language. If $language parameter is not specified current language will be used.

Return value

No return value.

Example

This example shows how to change the "carscars" table "descr" field tooltip to "My new tooltip".

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setFieldTooltip("carscars", "descr", "My new tooltip");
```

See also

- getFieldTooltip()

getPlaceholder

Gets the field placeholder.

Syntax

```
Labels::getPlaceholder($table, $field, $language)
```

Arguments

$table - table name

$field - field name.

$language - language. If $language parameter is not specified current language will be used.

Return value

Field placeholder. Placeholder can be set in the Edit As settings or the Label Editor on the Misc settings.

Example

Gets the "carscars" table "descr" field placeholder.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getPlaceholder("carscars", "descr");
```

See also

- setPlaceholder()

setPlaceholder

Sets the field placeholder.

Syntax

```
Labels::setPlaceholder($table, $field, $placeholder, $language)
```

Arguments

$table - table name

$field - field name.

$placeholder - field placeholder.

$language - language. If $language parameter is not specified current language will be used.

Return value

No return value.

Example

This example shows how to change the "carscars" table "descr" field placeholder to "Cars description".

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setPlaceholder("carscars", "descr", "Cars description");
```

See also

- getPlaceholder()

getPageTitleTempl

This function work with page title templates and not with titles directly. Examples of page title templates can be found in Label Editor

Syntax

```
Labels::getPageTitleTempl($table, $page, $language)
```

Arguments

$table - table name

$page - page name.

$language - language. If $language parameter is not specified current language will be used.

$page parameter accepts one of the following values:

```
PAGE_LIST
PAGE_PRINT
PAGE_ADD
PAGE_EDIT
PAGE_VIEW
PAGE_SEARCH
PAGE_EXPORT
PAGE_IMPORT
PAGE_REPORT
PAGE_RPRINT
PAGE_CHART

PAGE_MASTER_INFO_LIST
PAGE_MASTER_INFO_PRINT
PAGE_MASTER_INFO_REPORT
PAGE_MASTER_INFO_RPRINT

PAGE_MENU
PAGE_LOGIN
PAGE_REGISTER
PAGE_REMIND
PAGE_CHANGEPASS
```

Return value

Page title. Page titles can be set in the Label Editor.

Example

This example shows how to get Edit page title template.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::getPageTitleTempl("carscars", PAGE_EDIT);
```

See also

- setPageTitleTempl()

setPageTitleTempl

This function work with page title templates and not with titles directly. Examples of page title templates can be found in Label Editor

Syntax

```
Labels::setPageTitleTempl($table, $page, $title, $language)
```

Arguments

$table - table name

$page - page name.

$title - page title.

$language - language. If $language parameter is not specified current language will be used.

$page parameter accepts one of the following values:

```
PAGE_LIST
PAGE_PRINT
PAGE_ADD
PAGE_EDIT
PAGE_VIEW
PAGE_SEARCH
PAGE_EXPORT
PAGE_IMPORT
PAGE_REPORT
PAGE_RPRINT
PAGE_CHART

PAGE_MASTER_INFO_LIST
PAGE_MASTER_INFO_PRINT
PAGE_MASTER_INFO_REPORT
PAGE_MASTER_INFO_RPRINT
```

```
PAGE_MENU
PAGE_LOGIN
PAGE_REGISTER
PAGE_REMIND
PAGE_CHANGEPASS
```

Return value

No return value.

Example

This example shows how to change Edit page title template.

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setPageTitleTempl("carscars", PAGE_EDIT, "Edit car [{%id}]");
```

See also

- getPageTitleTempl()

getBreadcrumbsLabelTempl

This function gets the current breadcrumbs template

Syntax

```
Labels::getBreadcrumbsLabelTempl($table, $masterTable, $page, $language)
```

Arguments

$table - table name

$masterTable – master table name

$page - page name.

$language – language. If $language parameter is not specified current language will be used.

$page parameter accepts one of the following values:

```
PAGE_LIST
PAGE_PRINT
PAGE_ADD
PAGE_EDIT
PAGE_VIEW
PAGE_SEARCH
```

```
PAGE_EXPORT
PAGE_IMPORT
PAGE_REPORT
PAGE_RPRINT
PAGE_CHART

PAGE_MASTER_INFO_LIST
PAGE_MASTER_INFO_PRINT
PAGE_MASTER_INFO_REPORT
PAGE_MASTER_INFO_RPRINT

PAGE_MENU
PAGE_LOGIN
PAGE_REGISTER
PAGE_REMIND
PAGE_CHANGEPASS
```

Return value

The current breadcrumbs template

See also

- setBreadcrumbsLabelTempl()

setBreadcrumbsLabelTempl

This function sets the current breadcrumbs template

Syntax

```
Labels::setBreadcrumbsLabelTempl($table, $label, $masterTable, $page,
$language)
```

Arguments

$table - table name

$label - breadcrumb template

$masterTable – master table name (if we are on details table page at the moment).

$page - page name.

$language – language. If $language parameter is not specified current language will be used.

```
$page parameter accepts one of the following values:
PAGE_LIST
PAGE_PRINT
PAGE_ADD
PAGE_EDIT
PAGE_VIEW
PAGE_SEARCH
PAGE_EXPORT
PAGE_IMPORT
PAGE_REPORT
PAGE_RPRINT
PAGE_CHART

PAGE_MASTER_INFO_LIST
PAGE_MASTER_INFO_PRINT
PAGE_MASTER_INFO_REPORT
PAGE_MASTER_INFO_RPRINT

PAGE_MENU
PAGE_LOGIN
PAGE_REGISTER
PAGE_REMIND
PAGE_CHANGEPASS
```

Return value

No return value.

Example

By default on Order Details table we will see the following breadcrumbs menu:
Home / Orders / Order Details [5187]

We want to change it this way, removing "Order details" part:
Home / Orders / 5187

You can use any field from details table here (OrderID) or from master table (master.OrderID)

We can use either

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setBreadcrumbsLabelTempl("order details", "{%OrderID}",
"orders");
```

or

AfterAppInit, AfterTableInit, BeforeProcess events

```
Labels::setBreadcrumbsLabelTempl("order details", "{%master.OrderID}",
"orders");
```

See also

- getBreadcrumbsLabelTempl()

3.2.8 Page class

3.2.8.1 About RunnerPage class

An object of the RunnerPage class represents the current page and is passed to the event as a *$pageObject* parameter.

Methods

Method	Description	Available on
getCurrentRecord()	Gets the current record.	Edit/View pages.
getMasterRecord()	Gets the master record.	All pages.
hideField()	Hides the field and field label.	BeforeDisplay event of the Add/Edit/View/Register pages.
setProxyValue()	Sets the variable to the given value. You may use this method to pass values from PHP to JavaScript.	All pages.
showField()	Shows the previously hidden field and field label.	BeforeDisplay event of the Add/Edit/View/Register pages.
addTab()	Adds a tab to the additional WHERE tabs on the List or Chart page	BeforeProcess event of the List or Chart page.
deleteTab()	Deletes the additional WHERE tab	BeforeProcess event of the List or Chart page.
setTabTitle()	Sets the additional WHERE tab title	BeforeProcess event of the List or Chart page.

setTabWhere()	Sets tab WHERE clause for the additional WHERE tab title	BeforeProcess event of the List or Chart page.

See also

- Using Additional WHERE tabs

3.2.8.2 Methods

getCurrentRecord

Gets the current record. Available on Edit/View pages.

Syntax

```
getCurrentRecord()
```

Arguments

No arguments.

Return value

Returns the array.

Example

Get the current record and display the *Make* and *Model* fields' values:

```
$data = $pageObject->getCurrentRecord();
echo $data["Make"] ." ".$data["Model"];
```

See also

- getMasterRecord()

getMasterRecord

Gets the master record. Available on all pages.

Syntax

```
getMasterRecord()
```

Arguments

No arguments.

Return value

Returns the array.

Example

Get the master record and display the *Make* and *Model* fields' values:

```
$data = $pageObject->getMasterRecord();
echo $data["Make"] ." ".$data["Model"];
```

See also

- getCurrentRecord()

hideField

The *hideField()* method allows you to hide a field and its label. This method is available in the BeforeDisplay event of the List/Add/Edit/View/Register pages.

Syntax

```
hideField($field)
```

Arguments

$field - field name. Example: *"Make"*.

Return value

No return value.

Example

Hide the *Make* field:

```
$pageObject->hideField("Make");
```

Remarks

Fields hidden by the hideField() method can be displayed using JavaScript API: showField() method.

See also

- showField()
- JavaScript API: hideField()
- JavaScript API: showField()

setProxyValue

Sets the variable to the given value. Use the *setProxyValue()* method to pass values from PHP to JavaScript: you assign value to a variable in PHP and then use it in Javascript. Method is available on all pages.

Syntax

```
setProxyValue($name, $value)
```

Arguments

$name - any variable name. Example: *"master"*.

$value - value assigned to the variable. *$value* may be a simple value (e.g. string, number) or an array.

Return value

No return value.

Example

Set the variables *name* and *master* to some values:

```
$pageObject->setProxyValue("name", $value);
$pageObject->setProxyValue("master",$pageObject->getMasterRecord());
```

Use *name* and *master* variables in JavaScript OnLoad event:

```
alert(proxy['name']);
alert(proxy.master['Make']);
```

showField

Shows the previously hidden field and field label. *showField()* method is available in the BeforeDisplay event of the Add/Edit/View/Register pages.

Syntax

```
showField($field)
```

Arguments

$field - field name. Example: *"Make"*.

Return value

No return value.

Example

Show the *Make* field:

```
$pageObject->showField("Make");
```

See also

- hideField()
- JavaScript API: hideField()
- JavaScript API: showField()

addTab

This method adds a tab to the Additional WHERE tabs on the list page. the *addTab()* method is available in the BeforePocess event of List or Chart page.

Syntax

```
$pageObject->addTab($where, $title, $id);
```

Arguments

$where - WHERE clause for the new tab

$title - tab title that will be displayed

$id - id of the new tab

Return value

true - if the tab was added successfully

false - if the tab was not added successfully

Example

An example of adding a tab:

```
$pageObject->addTab("CustomerID='ANTON'", "ANTON orders", "anton");
```

See also

- Using Additional WHERE tabs
- deleteTab()
- setTabTitle()
- setTabWhere()

deleteTab

This method deletes one of the Additional WHERE tabs on the list page. The delete*Tab()* method is available in the BeforePocess event of List or Chart page.

Syntax

```
$pageObject->deleteTab($id);
```

Arguments

$id - id of the tab to be deleted

Return value

no return value

Example

An example of using the *deleteTab()* method to delete a tab:

```
$pageObject->deleteTab("anton");
```

See also

- Using Additional WHERE tabs
- addTab()
- Method: setTabTitle()
- Method: setTabWhere()

setTabTitle

This method sets the title of one of the Additional WHERE tabs on the list page. The *setTabTitle()* method is available in the BeforePocess event of the List or Chart page.

Syntax

```
$pageObject->setTabTitle($id, $title);
```

Arguments

$id - id of the tab

$title - tab title that will be displayed

Return value

true - if the title was set

false - in all other cases

Example

An example of using this method to set the title of a tab:

```
$pageObject->setTabTitle("anton", "ANTON orders");
```

See also

- Using Additional WHERE tabs
- addTab()
- deleteTab()
- setTabWhere()

setTabWhere

This method sets the WHERE clause of one of the Additional WHERE tabs on the list page. The *setTabWhere()* method is available in the BeforePocess event of the List or Chart page.

Syntax

```
$pageObject->setTabWhere($id, $where);
```

Arguments

$id - id of the tab

$where - WHERE clause for the tab

Return value

true - if WHERE clause was set successfully

false - in all other cases

Example

An example of using the *setTabWhere()* method to set the WHERE clause to the tab:

```
$pageObject->setTabWhere("anton", "CustomerID='ANTON'");
```

See also

- Using Additional WHERE tabs
- addTab()
- deleteTab()
- setTabTitle()

3.2.9 Search API

3.2.9.1 About Search API

This API allows you to control search panel, advanced search and 'All fields search' behaviour.

Where to use Search API?

AfterTableInitialized, List page: BeforeProcess events - you can read and write search conditions here. Changes will be applied to SQL Query and also will be shown on search panel.
List page: BeforeDisplay event - you can read and write search conditions here. Changes will not be applied to SQL Query but will be shown on search panel.

See also

- Search Master and Details tables together

Functions

Function	Description
getSearchObject()	Get SearchClause object
$srchObj->getFieldValue()	Returns search control value.
$srchObj->setFieldValue()	Will replace existing value of search field if field already added to search panel. Will add field to search panel if didn’t exist there.
$srchObj->getSecondFieldValue()	Returns second search control value.
$srchObj->setSecondFieldValue()	Sets second search control value if Between option is selected.
$srchObj->getSearchOption()	Returns search option.
$srchObj->setSearchOption()	Sets search option for the field.
$srchObj->setSearchSQL()	Sets SQL expression for search. Replaces current search expression for this field.
$srchObj->getAllFieldsSearchValue()	Gets 'All fields search' value.
$srchObj->setAllFieldsSearchValue()	Sets 'All fields search' value.

3.2.9.2 Methods

getSearchObject

If $table parameter is not specified current table will be used

Syntax

```
SearchClause::getSearchObject($table)
```

Arguments

$table - table name

Return value

SearchClause object

getFieldValue

Returns search control value

Syntax

```
$srchObj->getFieldValue($field);
```

Arguments

$field - field name

Return value

Returns search control value

Example

Get search value of Make field.

```
$srchObj = SearchClause::getSearchObject("Cars");
$srchObj->getFieldValue("Make");
```

See also

setFieldValue()

setFieldValue

Will replace existing value of search field if field already added to search panel. Will add field to search panel if didn't exist there.

Syntax

```
$srchObj->setFieldValue($field, $value);
```

Arguments

$field - field name

$value - value of search field

Return value

No return value

Example

Add the following code to the AfterTableInit event. It will replace existing value of search field if field is already added to search panel and will add field to search panel if didn't exist there.

```
$srchObj = SearchClause::getSearchObject("Employees");

$value = $srchObj->getFieldValue("Department");
   if( $value == null ) {
$srchObj->setFieldValue("Department", 10 );
}
```

See also

getFieldValue()

getSecondFieldValue

Returns second search control value, when Between is selected as a search option

Syntax

```
$srchObj->getSecondFieldValue($field);
```

Arguments

$field - field name

Return value

Returns second search control value, when Between is selected as a search option

See also

setSecondFieldValue()

setSecondFieldValue

Sets second search control value if Between option is selected.

Syntax

```
$srchObj->setSecondFieldValue($field, $value);
```

Arguments

$field - field name

$value - value of search field

Return value

No return value

See also

getSecondFieldValue()

getSearchOption

Returns search option.

Syntax

```
$srchObj->getSearchOption($field);
```

Arguments

$field - field name

Return value

Returns search option, which is one of the following constants:

CONTAINS
EQUALS
STARTS_WITH
MORE_THAN
LESS_THAN
BETWEEN
EMPTY_SEARCH
NOT_CONTAINS
NOT_EQUALS
NOT_STARTS_WITH
NOT_MORE_THAN
NOT_LESS_THAN
NOT_BETWEEN
NOT_EMPTY

See also

setSearchOption()

setSearchOption

Sets search option for the field. Accepts one of the following as a search option:

Syntax

```
$srchObj->setSearchOption($field, $option);
```

Arguments

$field - field name

$option - option to be set. Accepts one of the following as a search option:

CONTAINS
EQUALS
STARTS_WITH
MORE_THAN
LESS_THAN
BETWEEN
EMPTY_SEARCH
NOT_CONTAINS
NOT_EQUALS
NOT_STARTS_WITH
NOT_MORE_THAN
NOT_LESS_THAN
NOT_BETWEEN
NOT_EMPTY

Example

Make sure that for Year field 'Between' option is always selected. This examples shows how to set "BETWEEN" search option for the "Year" field.

```
$srchObj = SearchClause::getSearchObject("Employees");
$value = $srchObj->getSearchOption("Year");
if ($value != BETWEEN) {
    $srchObj->setSearchOption("Year", BETWEEN);
}
```

See also

getSearchOption()

setSearchSQL

Sets SQL expression for search. Replaces current search expression for this field.

Syntax

```
$srchObj->setSearchSQL($field, $sql);
```

Arguments

$field - field name

$sql - sql query that should be used

Return value

No return value

Example 1

How to change a search condition for a field.

```
$srchObj = SearchClause::getSearchObject("carsmodels");

$value = $srchObj->getFieldValue("make");

if( $value != null ) {
$srchObj->setSearchSQL("make", "make='$value' or id>12 and id<15");
}
```

Example 2

Here is how you can **search master and details tables together.** For instance you have Orders and OrderDetails tables and need to find orders that contain a certain product.

1. Modify Orders SQL Query to add a dummy field named 'product'. Make sure this field is searchable.

```
SELECT
        OrderID,
        CustomerID,
        EmployeeID,
        OrderDate,
        ShipAddress,
        ShipCity,
        ShipRegion,
        ShipPostalCode,
        ShipCountry,
        '' as product
FROM orders
```

2. Orders table, AfterTableInit event:

```
$srchObj = SearchClause::getSearchObject("orders");
```

```
$value = $srchObj->getFieldValue("product");

if( $value != null ) {
$srchObj->setSearchSQL("product", "OrderID in (select OrderID from OrderDetails wh
}
```

In this event we do a subquery to find all orders that contain the product in question.

See also

- getSearchObject()
- getFieldValue()

getAllFieldsSearchValue

Gets 'All fields search' value

Syntax

```
$srchObj->getAllFieldsSearchValue()
```

Arguments

No arguments

Return value

Returns 'All fields search' value

```
getFieldValue
```

See also

setAllFieldsSearchValue()

setAllFieldsSearchValue

Sets 'All fields search' value

Syntax

```
$srchObj->setAllFieldsSearchValue($value)
```

Arguments

$value - value to be set

Return value

No return value

See also

getAllFieldsSearchValue()

3.2.10 Security API

3.2.10.1 About Security API

Security API allows you to work with permission level in your application.
Permissions need to be set only once per user session, i.e. in the After Successful Login event.

Functions

Function	Description
getUserGroup()	Returns current user group name. Makes more sense to use with static permissions where each user belongs to single user group.
getUserGroups()	Returns an array with all user groups
getUserName()	Returns current Username
getDisplayName()	Returns current Display Name
setDisplayName()	Set current Display Name
isGuest()	Returns true if user logged in as Guest, returns false otherwise
isAdmin()	Returns true if user is an admin, returns false otherwise
isLoggedIn()	Returns true if user is logged in, returns false otherwise
loginAs()	Logs user in, no redirects
logout()	Just logout, no redirects.

getOwnerId()
setOwnerId()
	These functions get or set OwnerID for specific table when Advanced Security like 'Users can see and edit their own data' is in use.

checkUsernamePassword()	This function checks username and password and returns true if username/password are correct,

	returns false otherwise.
getUserData()	Returns array with user data from login table. Returns false if user not found.
currentUserData()	Returns array with current user data from login table.

getPermissions()
setPermissions()
	Gets/sets permissions for certain table

3.2.10.2 Methods

getUserGroup

Returns current user group name. Makes more sense to use with static permissions where each user belongs to single user group.

Syntax

```
Security::getUserGroup()
```

Arguments

```
No arguments
```

Return value

Returns the current user group name.

getUserGroups

Returns an array with all user groups

Syntax

```
Security::getUserGroups()
```

Arguments

No arguments

Return value

Returns an array with all user groups.

Example

```
$groups = Security::getUserGroups()
   if( $groups["Accounting"] ) {
   ... do something
}
```

getUserName

Returns current Username

Syntax

```
Security::getUserName()
```

Arguments

`No arguments`

Return value

Returns current Username

getDisplayName

Returns current Display Name

Syntax

```
Security::getDisplayName()
```

Arguments

`No arguments`

Return value

Returns current Display Name

setDisplayName

Sets current Display Name

Syntax

```
Security::setDisplayName($str)
```

Arguments

$str - the Display name

Return value

No return value

isGuest

Returns true if user logged in as Guest, returns false otherwise

Syntax

```
Security::isGuest()
```

Arguments

```
No arguments
```

Return value

Returns true if user logged in as Guest, returns false otherwise

isAdmin

Returns true if user is an admin, returns false otherwise

Syntax

```
Security::isAdmin()
```

Arguments

```
No arguments
```

Return value

Returns true if user is an admin, returns false otherwise

isLoggedIn

Returns true if user is logged in and not guest, returns false otherwise

Syntax

```
Security::isLoggedIn()
```

Arguments

```
No arguments
```

Return value

Returns true if user is logged in and not guest, returns false otherwise

loginAs

Logs user in, no redirects.

AftersuccessfulLogin will be called if second parameter is set to true.

This function will not work with Active Directory or Facebook login

Syntax

```
Security::loginAs($username, $callEvent = true)
```

Arguments

$username - name of the user to be logged in.

$callEvent = true - AftersuccessfulLogin will be called if second parameter is set to true.

Return value

No return value.

logout

Just logout, no redirects.

Syntax

```
Security::logout()
```

Arguments

No arguments

Return value

No return value.

getOwnerId

This functions get OwnerID for specific table when Advanced Security like 'Users can see and edit their own data' is in use.

Syntax

```
Security::getOwnerId($table)
```

Arguments

$table - table name

Return value

OwnerID for the specified table

See also

setOwnerId()

setOwnerId

This functions set OwnerID for specific table when Advanced Security like 'Users can see and edit their own data' is in use.

Syntax

```
Security::setOwnerId($table, $ownerid)
```

Arguments

$table - table name

$ownerid - owner id

Return value

OwnerID for the specified table

See also

getOwnerId()

checkUsernamePassword

This function checks username and password and returns true if username/password are correct, returns false otherwise.

If username/password are incorrect and $fireEvents is true AfterUnsuccessfulLogin event will be fired.

This function doesn't actually performs the login, just validates username and password.

Syntax

```
Security::checkUsernamePassword($username, $password, $fireEvents =
false)
```

Arguments

$username – user name

$password – user password

$fireEvents = false – If username/password are incorrect and $fireEvents is true AfterUnsuccessfulLogin event will be fired

Return value

true - if username/password are correct

false - if username/password are not correct

getUserData

Returns array with user data from login table.

Returns false if user not found.

Syntax

```
Security::getUserData($username)
```

Arguments

$username – user name

Return value

Array with user data from login table.

Returns false if user not found.

See also currentUserData()

currentUserData

Returns array with current user data from login table.

Syntax

```
Security::currentUserData()
```

Arguments

`No arguments`

Return value

Returns array with current user data from login table.

Example

This example shows how to output the current user data

```
$userData = Security::currentUserData();
print_r($userData );
```

See also getUserData()

getPermissions

Gets permissions for certain table

Syntax

```
Security::getPermissions($table)
```

Arguments

`$table` `- table name`

Return value

Array with table permissions.

See also

setPermissions()

setPermissions

Sets permissions for certain table for the current user. Permissions need to be set only once per user session, i.e. in the After Successful Login event.

Permissions should be passed in the form of array where keys are specific permission letters:

A - add,

D - delete,

E - edit,

S - search/list,

P - print/export,

I - import,

M - admin permission. When advanced permissions are in effect (users can see/edit their own records only), this permissions grants access to all records.

Syntax

```
Security::setPermissions($table, $rights)
```

Arguments

$table - table name

$rights - array where keys are specific permission letters

Return value

No return value

Example 1

Enable Add and disable Delete functionality on "Cars" table for the current user.

```
$rights = Security::getPermissions("Cars");
$rights["A"] = true;
$rights["D"] = false;
Security::setPermissions("Cars", $rights);
```

Example 2

Check if current user has Add permissions on "Cars" table.

```
$rights = Security::getPermissions("Cars");
if($rights["A"])
    echo "add permission available";
```

See also

getPermissions()

Password hashing

You can hash your password manually using events

For instance, you want to provide admin with direct access to the login table. To do so add the following code to BeforeAdd/BeforeEdit events of login table:

For BCRYPT:

```
$values["password"] = getPasswordHash($values["password"]);
```

For MD5:

```
$values["password"] = md5($values["password"]);
```

See also Registration and passwords.

3.2.11 SQLQuery class

3.2.11.1 About SQLQuery class

SQLQuery class allows you to modify current SQL query to a table stored in the *$query* object. The *$query* object is available only in the After table initialized event. It is immediately usable and does not require initialization.

All the methods described below should be used in the After table initialized event. This ensures that all changes made will be applied to all pages.

Methods

Method	Description
addWhere()	Adds WHERE clause to the current SQL query.
replaceWhere()	Replaces WHERE clause of the current SQL query.
addField()	Adds a field name to the end of SELECT clause of the current SQL query.
deleteField()	Removes a field name from the SELECT clause of the current SQL query.
replaceField()	Replaces a field name in the SELECT clause of the current SQL query with new one.

3.2.11.2 Methods

addField

Adds a field name to the end of SELECT clause of the current SQL query.

Syntax

```
addField($calculatedField, $alias)
```

Arguments

$calculatedField - field name. Example: *"size"*.

$alias - alias of the field name. Example: *"new_size". $alias* field values can be used in the following events:

- Before record processed, After record processed for the List page like *$data[$alias]*;
- Process record values, Before display for the Edit page like *$values[$alias]*;
- Copy page: Onload, Before record added, After record updated, Process record values for the Add page like *$values[$alias]*;
- Process record values, Before display for the View page like *$values[$alias]*.

Return value

No return value.

Example 1.

To add now() as 'current_date' to the end of the SELECT clause:

After table initialized event

```
$query->addField("now()", "current_date");
```

Example 2.

To add 'size+adjust' as 'new_size' to the end of the SELECT clause:

After table initialized event

```

$query->addField("size+adjust", "new_size");
```

addWhere

If the current SQL query does not include WHERE clause, *addWhere()* adds it as *where($condition).* Otherwise, WHERE clause will be added as a new condition as *and($condition)*.

Syntax

```
addWhere($condition)
```

Arguments

$condition - any condition clause. Example: *"id_sizes < 3 or id_sizes > 6"*. *id_sizes* is the field name.

Return value

No return value.

Example

To add WHERE clause:

After table initialized event

```
$query->addWhere("id_sizes < 3 or id_sizes > 6");
$query->addWhere("notes='".$_SESSION["UserID"]."'");
$query->addWhere("test ='passed'");
```

deleteField

Removes a field name from the SELECT clause of the current SQL query.

Syntax

```
deleteField($field)
```

Arguments

$field - field name. Example: *"size"*.

Return value

No return value.

Example

To remove field name from the SELECT clause (data from the field 'id_sizes' are no longer requested):

```
                                                    After table initialized event
$query->deleteField("id_sizes");
```

Remarks

Using *deleteField* function you can delete the alias field created by the addField function.

replaceField

Replaces a field name in the SELECT clause of the current SQL query with new one.

Syntax

```
replaceField($replaceableField, $calculatedField, $alias)
```

Arguments

$replaceableField - field name to be replaced. Example: *"size"*.

$calculatedField - new field name. Example: *"new_size"*.

$alias - alias of the new field name. *$alias* parameter can be omitted. In this case *$replaceableField* value is taken as alias.

Return value

No return value.

Example 1.

To replaces the 'name' with the 'englishName' as 'name':

```
                                                    After table initialized event
$query->replaceField("name", "englishName");
```

Example 2.

To replaces the 'price' with the 'price*1.2' as 'price':

```
                                                    After table initialized event
$query->replaceField("price", "price*1.2");
```

replaceWhere

Replaces WHERE clause of the current SQL query with new one.

Syntax

```
replaceWhere( $condition )
```

Arguments

$condition - any condition clause. Example: *"id_sizes < 3 or id_sizes > 6"*. *id_sizes* is the field name.

Return value

No return value.

Example 1.

To display records from the table where the value of the field test is 'passed':

```
                                                  After table initialized event
$query->replaceWhere( "test = 'passed'" );
```

Example 2.

To display records from a table where the value of the field size is 3 and the value of the field adjust is less than 2:

```
                                                  After table initialized event
$query->replaceWhere( "size = 3 and adjust < 2" );
```

Example 3.

To display all table records:

```
                                                  After table initialized event
$query->replaceWhere( "" );
```

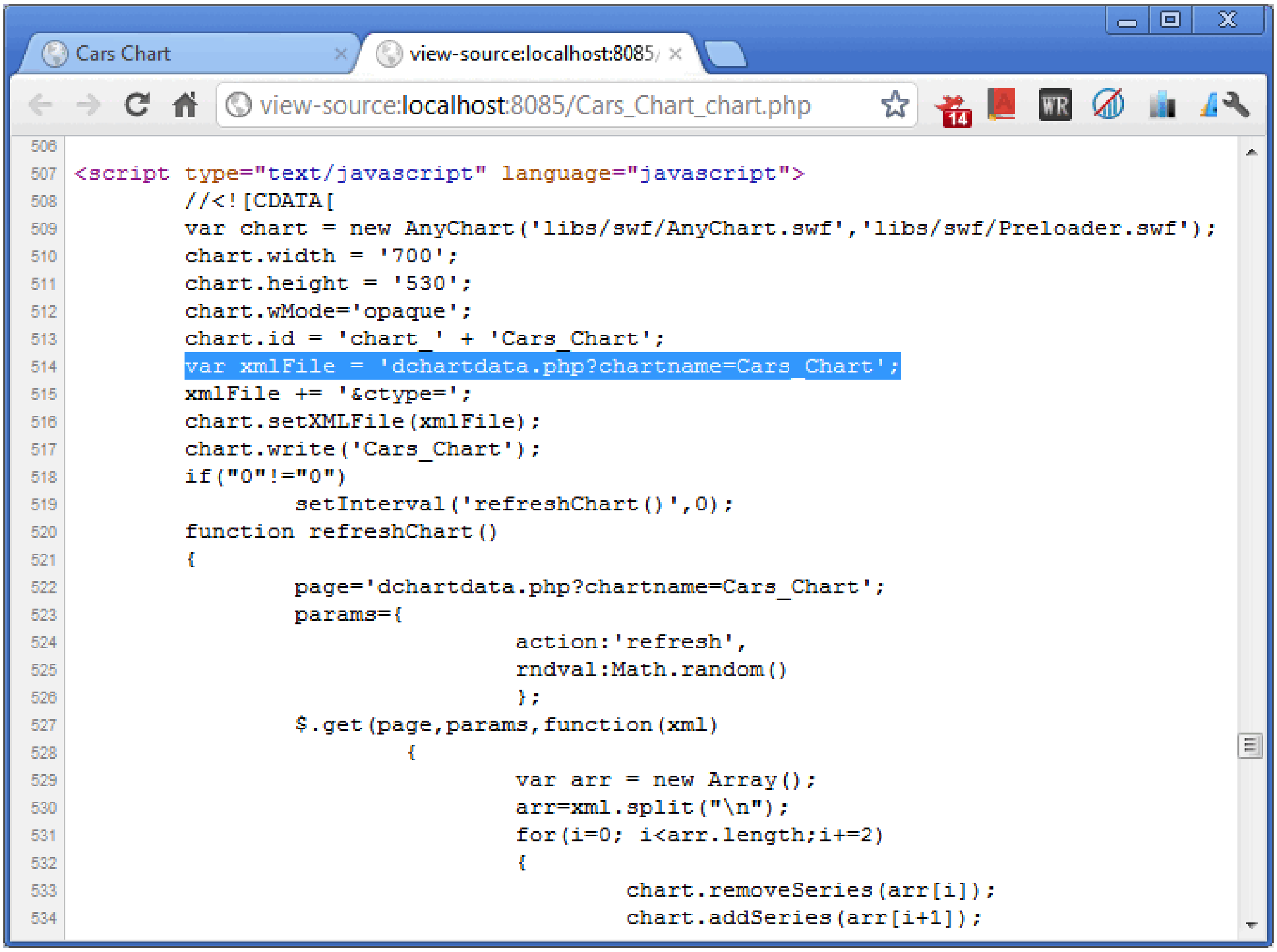
3.2.12 Troubleshooting tips

3.2.12.1 Troubleshooting charts

In this tutorial we will find out how to troubleshoot chart errors in our applications. PHPRunner applications use Flash-based charting component that receives data in XML format. If any error happens while generating XML input you will see the following message instead of chart:

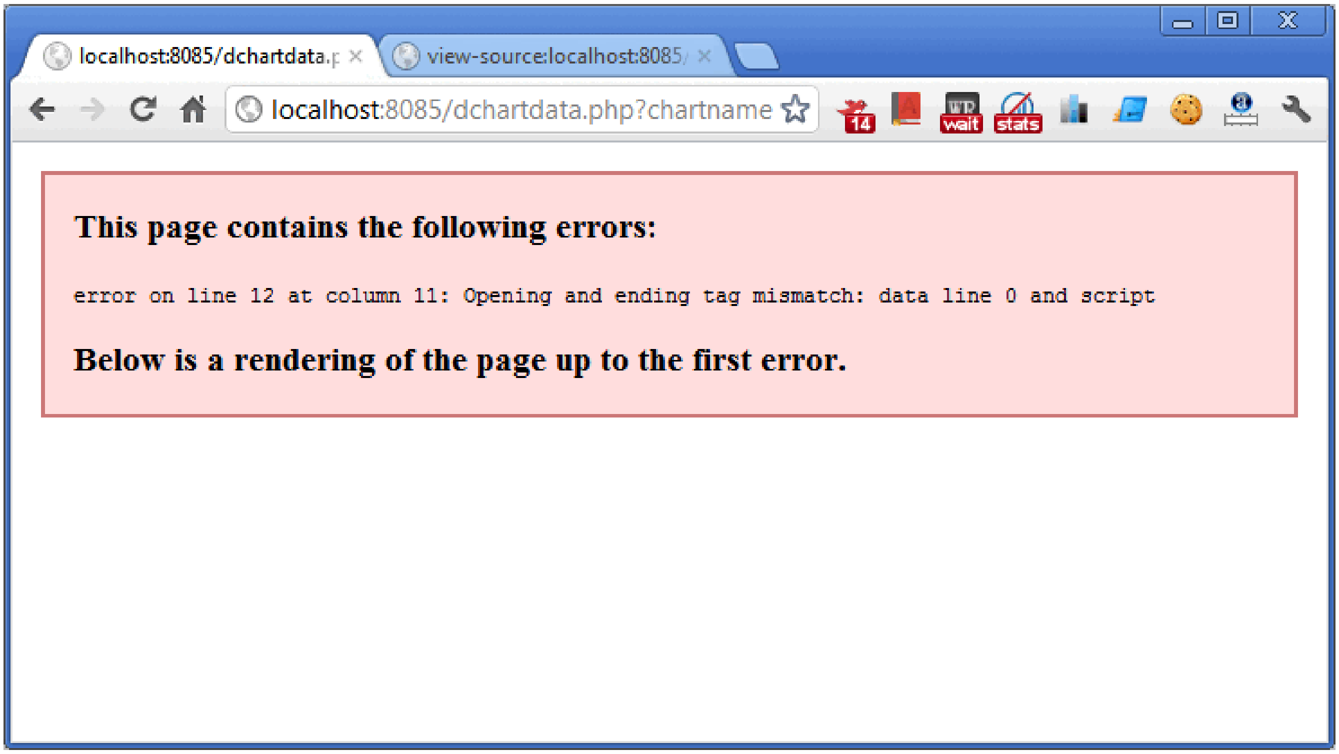
```
XML Parser failure:
The element type must be terminated by the matching end-tag.
```

Lets go a bit deeper. Right click anywhere on the page (except on chart itself) and choose **View source**. In HTML source search for the first occurrence of *dchartdata.php?* (*dchartdata.asp?*) string. You should see something like this:


Cars Chart
view-source:localhost:8085/
view-source:localhost:8085/Cars_Chart_chart.php

```

<script type="text/javascript" language="javascript">
        //<![CDATA[
        var chart = new AnyChart('libs/swf/AnyChart.swf','libs/swf/Preloader.swf');
        chart.width = '700';
        chart.height = '530';
        chart.wMode='opaque';
        chart.id = 'chart_' + 'Cars_Chart';
        var xmlFile = 'dchartdata.php?chartname=Cars_Chart';
        xmlFile += '&ctype=';
        chart.setXMLFile(xmlFile);
        chart.write('Cars_Chart');
        if("0"!="0")
                setInterval('refreshChart()',0);
        function refreshChart()
        {
                page='dchartdata.php?chartname=Cars_Chart';
                params={
                                action:'refresh',
                                rndval:Math.random()
                                };
                $.get(page,params,function(xml)
                        {
                                var arr = new Array();
                                arr=xml.split("\n");
                                for(i=0; i<arr.length;i+=2)
                                {
                                        chart.removeSeries(arr[i]);
                                        chart.addSeries(arr[i+1]);
```

Highlight *dchartdata.php?chartname=Cars_Chart* part, right click on it and choose **Copy**. Now paste it to the browser address line replacing *Cars_Chart_chart.php*. URL is supposed to look like this: *http://yourwebsite.com/dchartdata.php?chartname=Cars_Chart*. Hit **Enter**.

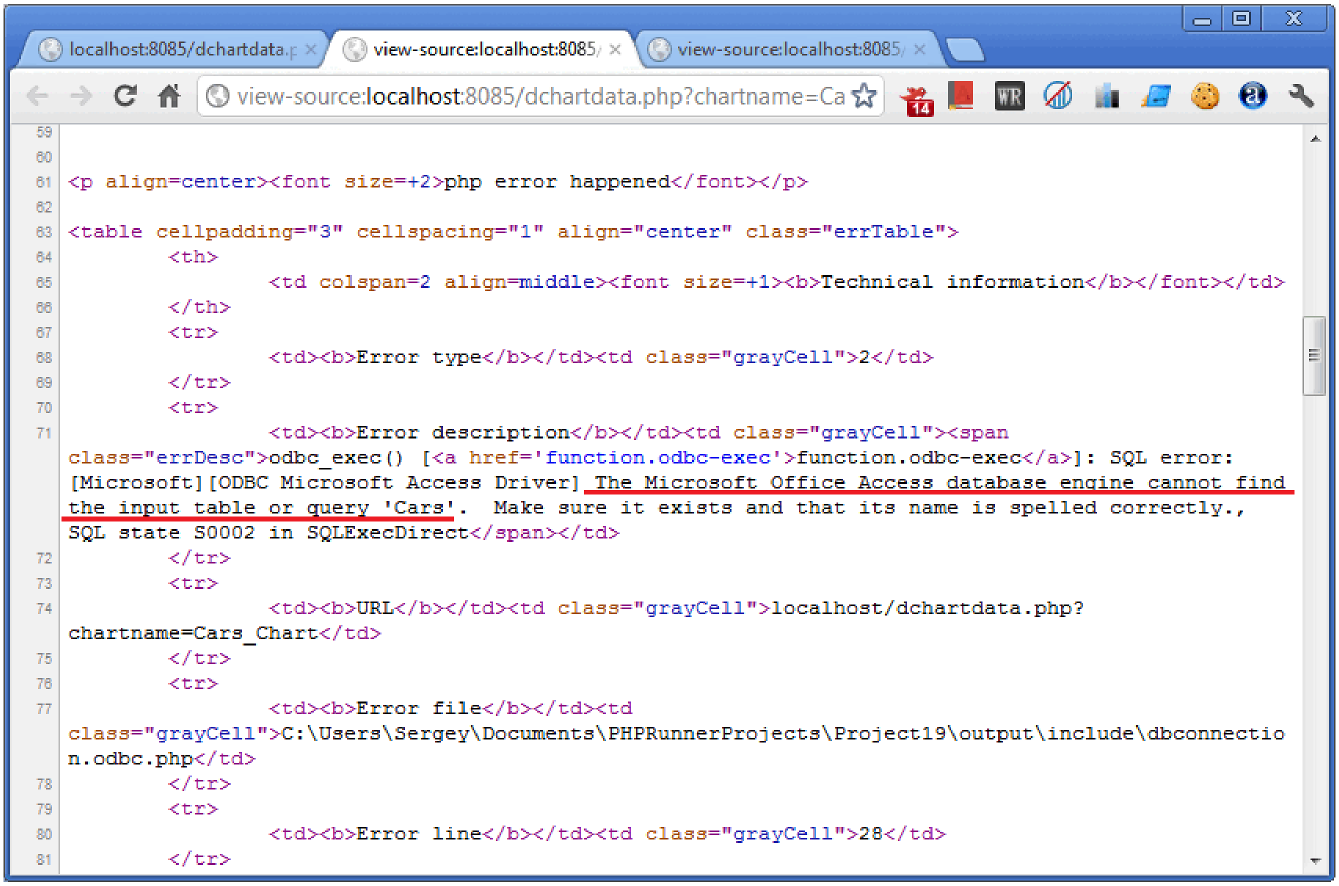

localhost:8085/dchartdata.p
view-source:localhost:8085/
localhost:8085/dchartdata.php?chartname

This page contains the following errors:

```
error on line 12 at column 11: Opening and ending tag mismatch: data line 0 and script
```

Below is a rendering of the page up to the first error.

dchartdata file generates XML. As we can see something went wrong and this file produces the error. View the source of this page again. Search for *php error happened* string ('asp error happened' in ASPRunnerPro).


localhost:8085/dchartdata.p
view-source:localhost:8085/
view-source:localhost:8085/
view-source:localhost:8085/dchartdata.php?chartname=Ca
14

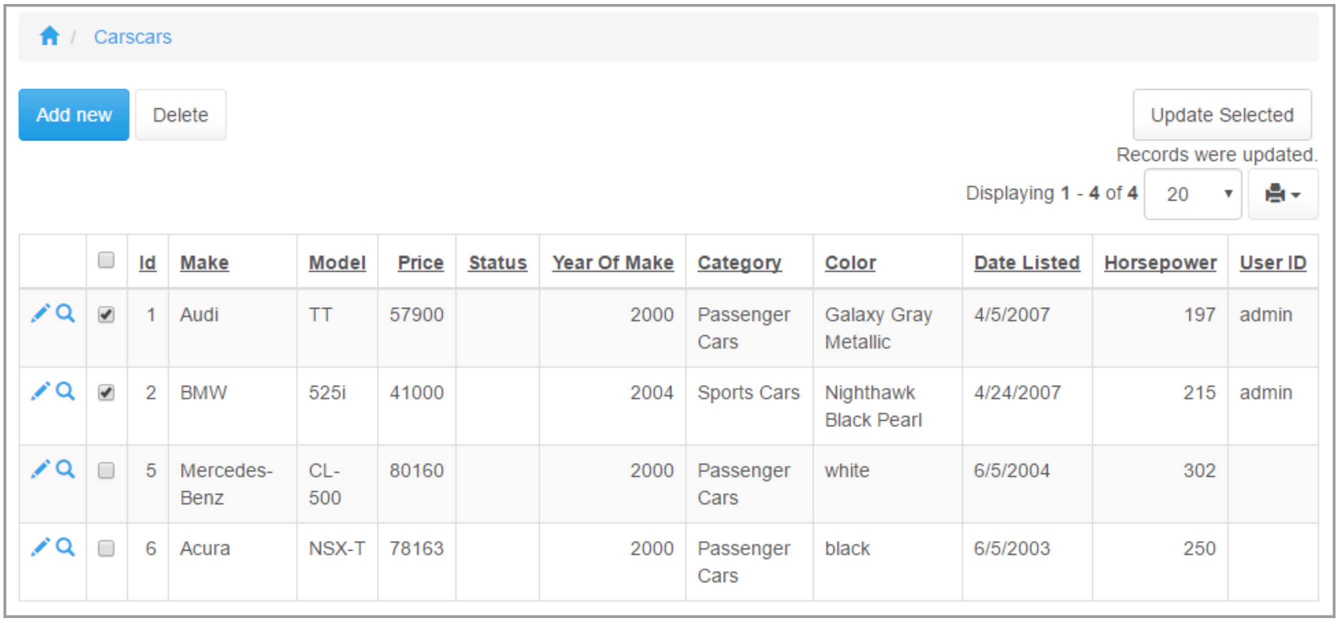
```


<p align=center><font size=+2>php error happened</font></p>

<table cellpadding="3" cellspacing="1" align="center" class="errTable">
        <th>
                <td colspan=2 align=middle><font size=+1><b>Technical information</b></font></td>
        </th>
        <tr>
                <td><b>Error type</b></td><td class="grayCell">2</td>
        </tr>
        <tr>
                <td><b>Error description</b></td><td class="grayCell"><span class="errDesc">odbc_exec() [<a href='function.odbc-exec'>function.odbc-exec</a>]: SQL error: [Microsoft][ODBC Microsoft Access Driver] The Microsoft Office Access database engine cannot find the input table or query 'Cars'.  Make sure it exists and that its name is spelled correctly., SQL state S0002 in SQLExecDirect</span></td>
        </tr>
        <tr>
                <td><b>URL</b></td><td class="grayCell">localhost/dchartdata.php?chartname=Cars_Chart</td>
        </tr>
        <tr>
                <td><b>Error file</b></td><td class="grayCell">C:\Users\Sergey\Documents\PHPRunnerProjects\Project19\output\include\dbconnection.odbc.php</td>
        </tr>
        <tr>
                <td><b>Error line</b></td><td class="grayCell">28</td>
        </tr>
```

Now we can see the actual error message. In this specific case table named 'Cars' missing in the database. Probably table was removed or renamed after project was built. To fix this you either need to change the chart definition in your project or to rename this table back.

3.2.12.2 Troubleshooting custom buttons

Let's say you have a database of cars. You have added a button to the List page that should update selected cars statuses as 'Sold'. Your code looks good and passes syntax check but still doesn't work when you run your application. The worst of all - it doesn't produce any visible errors. In this article I'll show you how to catch errors like this one.

All modern browsers provide developer tools. Our screenshots are taken in Chrome. Open your list page, hit F12 to display developers tools panel and proceed to Network tab. You should see something like this:

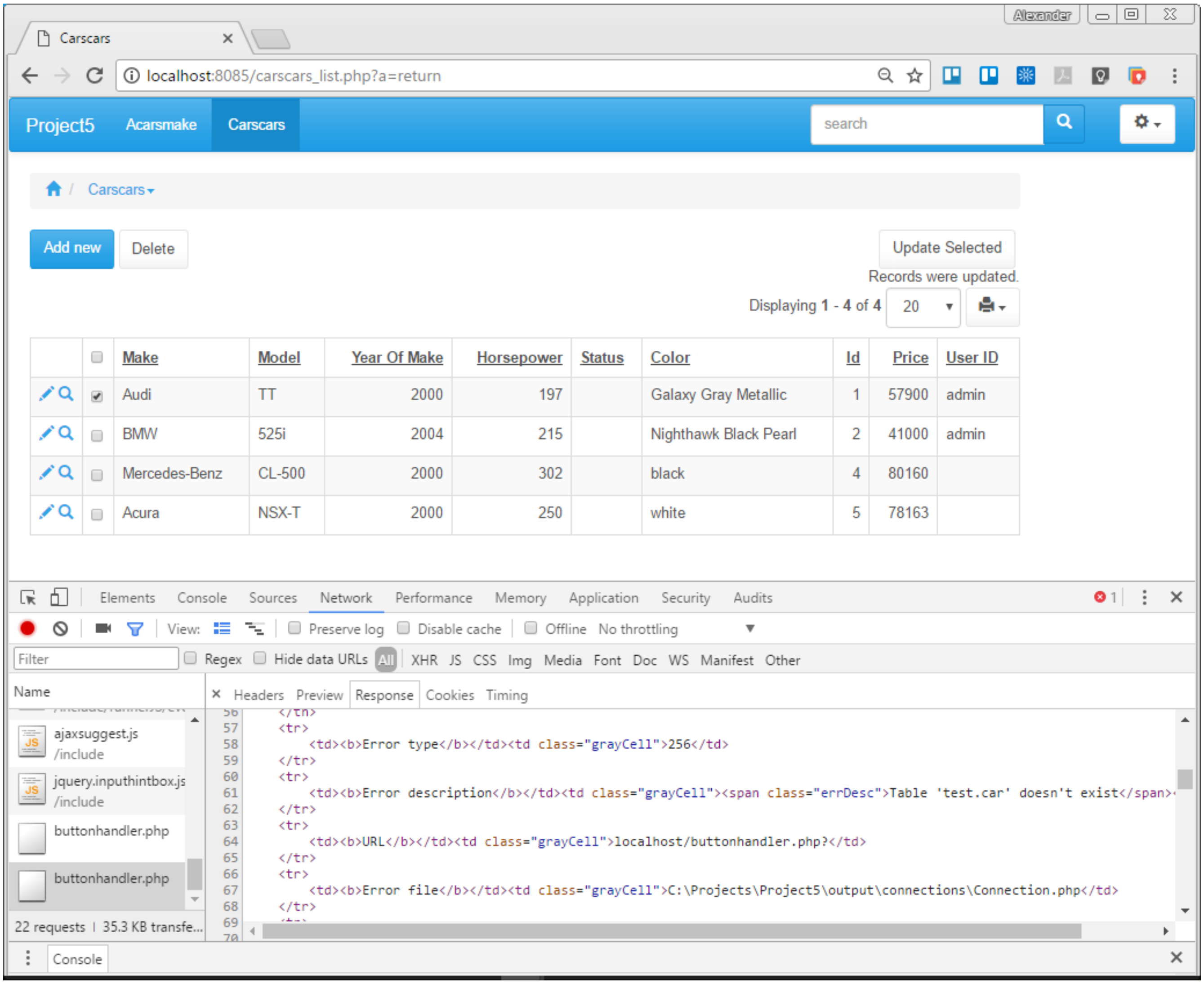


Records were updated.

Displaying **1** - **4** of **4** 20

	□	Id	Make	Model	Price	Status	Year Of Make	Category	Color	Date Listed	Horsepower	User ID
	☑	1	Audi	TT	57900		2000	Passenger Cars	Galaxy Gray Metallic	4/5/2007	197	admin
	☑	2	BMW	525i	41000		2004	Sports Cars	Nighthawk Black Pearl	4/24/2007	215	admin
	□	5	Mercedes-Benz	CL-500	80160		2000	Passenger Cars	white	6/5/2004	302	
	□	6	Acura	NSX-T	78163		2000	Passenger Cars	black	6/5/2003	250	

Now select a few records and click 'Update Selected'. A new entry appears under Network tab - browser executes *buttonhandler.php* file where our server side code is stored. If we expand Response tab we can see our error description if any.


Carscars
localhost:8085/carscars_list.php?a=return
Project5
Acarsmake
Carscars
search
Carscars
Add new
Delete
Update Selected
Elements Console Sources Network Performance Memory Application Security Audits
Preserve log
Disable cache
Offline
No throttling
Filter
Regex
Hide data URLs
All XHR JS CSS Img Media Font Doc WS Manifest Other
Name
ajaxsuggest.js /include
jquery.inputhintbox.js /include
buttonhandler.php
buttonhandler.php
22 requests | 35.3 KB transfe...
Headers Preview Response Cookies Timing
Console

Records were updated.

Displaying 1 - 4 of 4 | 20

	□	Make	Model	Year Of Make	Horsepower	Status	Color	Id	Price	User ID
	☑	Audi	TT	2000	197		Galaxy Gray Metallic	1	57900	admin
	□	BMW	525i	2004	215		Nighthawk Black Pearl	2	41000	admin
	□	Mercedes-Benz	CL-500	2000	302		black	4	80160	
	□	Acura	NSX-T	2000	250		white	5	78163	

```
    </th>
    <tr>
        <td><b>Error type</b></td><td class="grayCell">256</td>
    </tr>
    <tr>
        <td><b>Error description</b></td><td class="grayCell"><span class="errDesc">Table 'test.car' doesn't exist</span>
    </tr>
    <tr>
        <td><b>URL</b></td><td class="grayCell">localhost/buttonhandler.php?</td>
    </tr>
    <tr>
        <td><b>Error file</b></td><td class="grayCell">C:\Projects\Project5\output\connections\Connection.php</td>
    </tr>
```

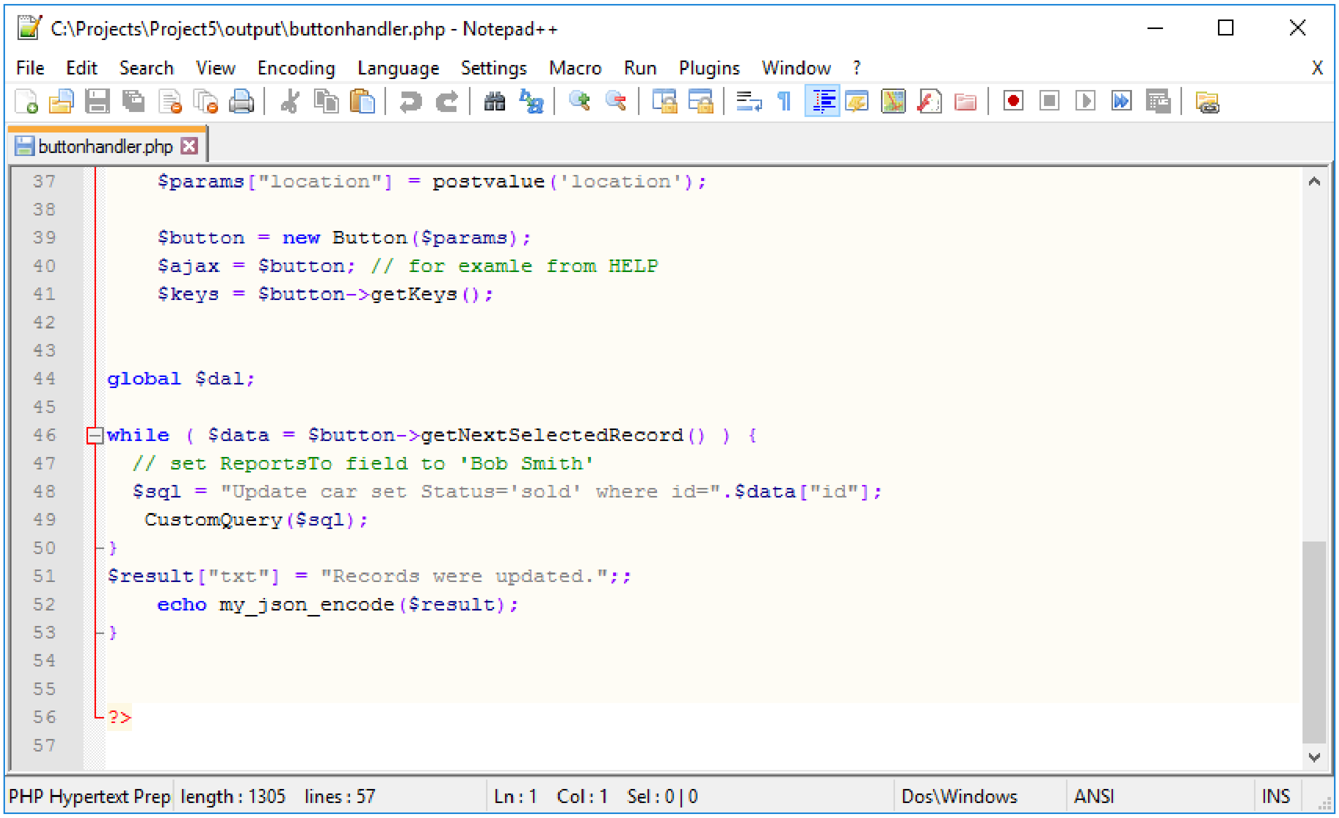
In this specific case it says the following under Error description:

```
Table 'test.car' doesn't exist
```

It looks like we have misspelled the table name in our code. Replacing 'car' with 'carscars' fixes the issue. This was definitely helpful though some events can be tens or hundreds lines of code long and single error message doesn't provide much help. To find the exact line of code that produces the error scroll down the content of Response tab to find the entry that points to *buttonhandler.php* file. Here it is:

```
<td nowrap="nowrap">#4.&nbsp;</td>
<td nowrap="nowrap">buttonhandler.php:49</td>
<td nowrap="nowrap">buttonHandler_Update_Selected</td>
```

Now we can open *buttonhandler.php* file in any text editor (Notepad++ recommended) and find line 40:


C:\Projects\Project5\output\buttonhandler.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
X
buttonhandler.php
PHP Hypertext Prep length : 1305 lines : 57 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows ANSI INS

```
        $params["location"] = postvalue('location');

        $button = new Button($params);
        $ajax = $button; // for examle from HELP
        $keys = $button->getKeys();


global $dal;

while ( $data = $button->getNextSelectedRecord() ) {
  // set ReportsTo field to 'Bob Smith'
  $sql = "Update car set Status='sold' where id=".$data["id"];
   CustomQuery($sql);
}
$result["txt"] = "Records were updated.";;
    echo my_json_encode($result);
}


?>
```

This also points to the fact that something is not right with our SQL query.

Additional troubleshooting tips

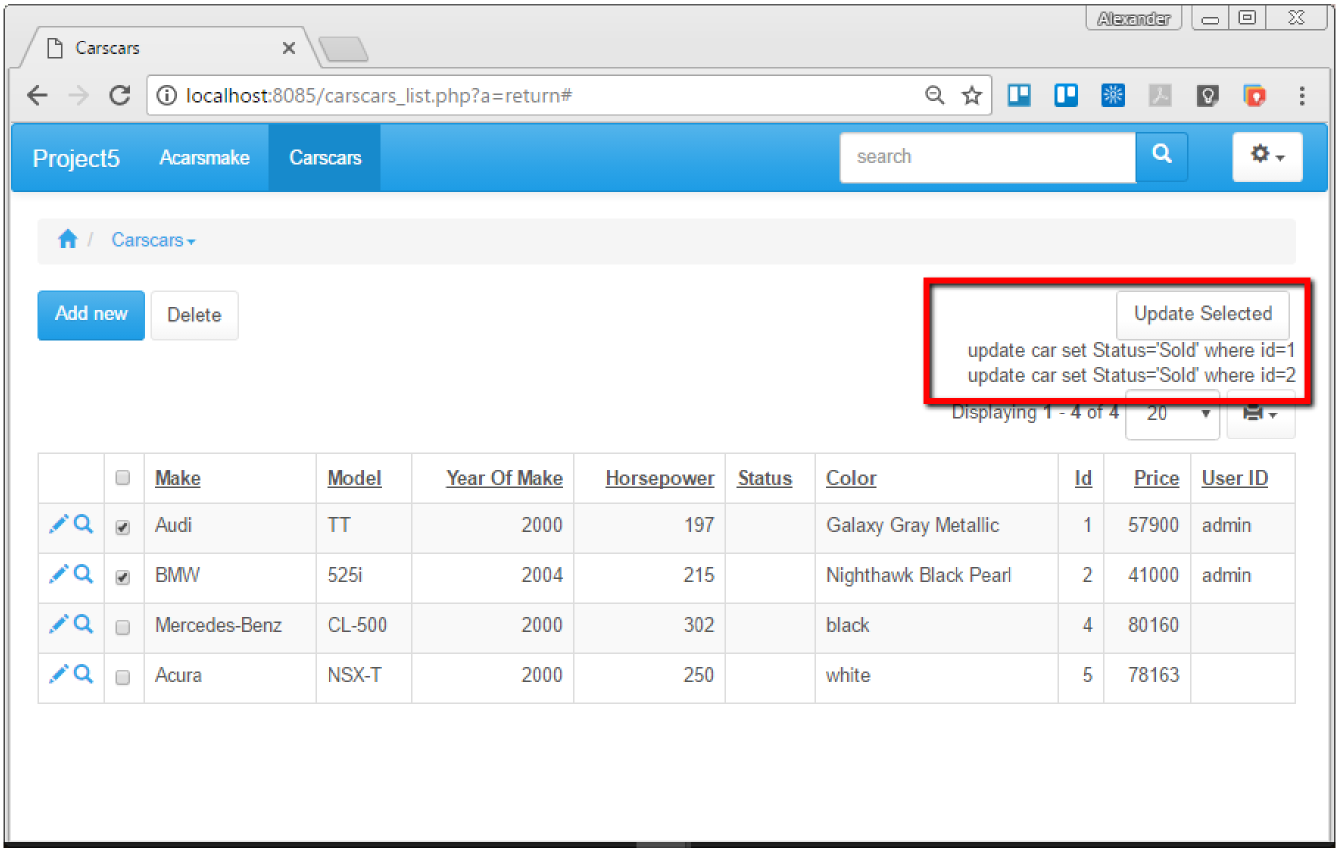
You may want save a few troubleshooting steps printing your SQL Queries on the web page instead of executing them. This way you can see what exactly is happening on the server and catch syntax errors faster. To do so assign SQL queries to *result["txt"]* variable and print it on the page using the following code in ClientAfter event.

```
ctrl.setMessage(result["txt"]);
```

Sample server PHP code:

```
$result["txt"]="";
foreach($keys as $idx=>$val)
{
$sql = "update car set Status='Sold' where id=".$val["ID"];
$result["txt"].=$sql."<br>";
//CustomQuery($sql);
}
```

And here is how this looks in action:



update car set Status='Sold' where id=1
update car set Status='Sold' where id=2

Displaying **1 - 4 of 4** 20

	□	Make	Model	Year Of Make	Horsepower	Status	Color	Id	Price	User ID
	☑	Audi	TT	2000	197		Galaxy Gray Metallic	1	57900	admin
	☑	BMW	525i	2004	215		Nighthawk Black Pearl	2	41000	admin
	□	Mercedes-Benz	CL-500	2000	302		black	4	80160	
	□	Acura	NSX-T	2000	250		white	5	78163	

3.2.12.3 Troubleshooting Javascript errors

In this article we will be focusing on debugging and troubleshooting the JavaScript errors. We will look at few examples to illustrate the methods used.

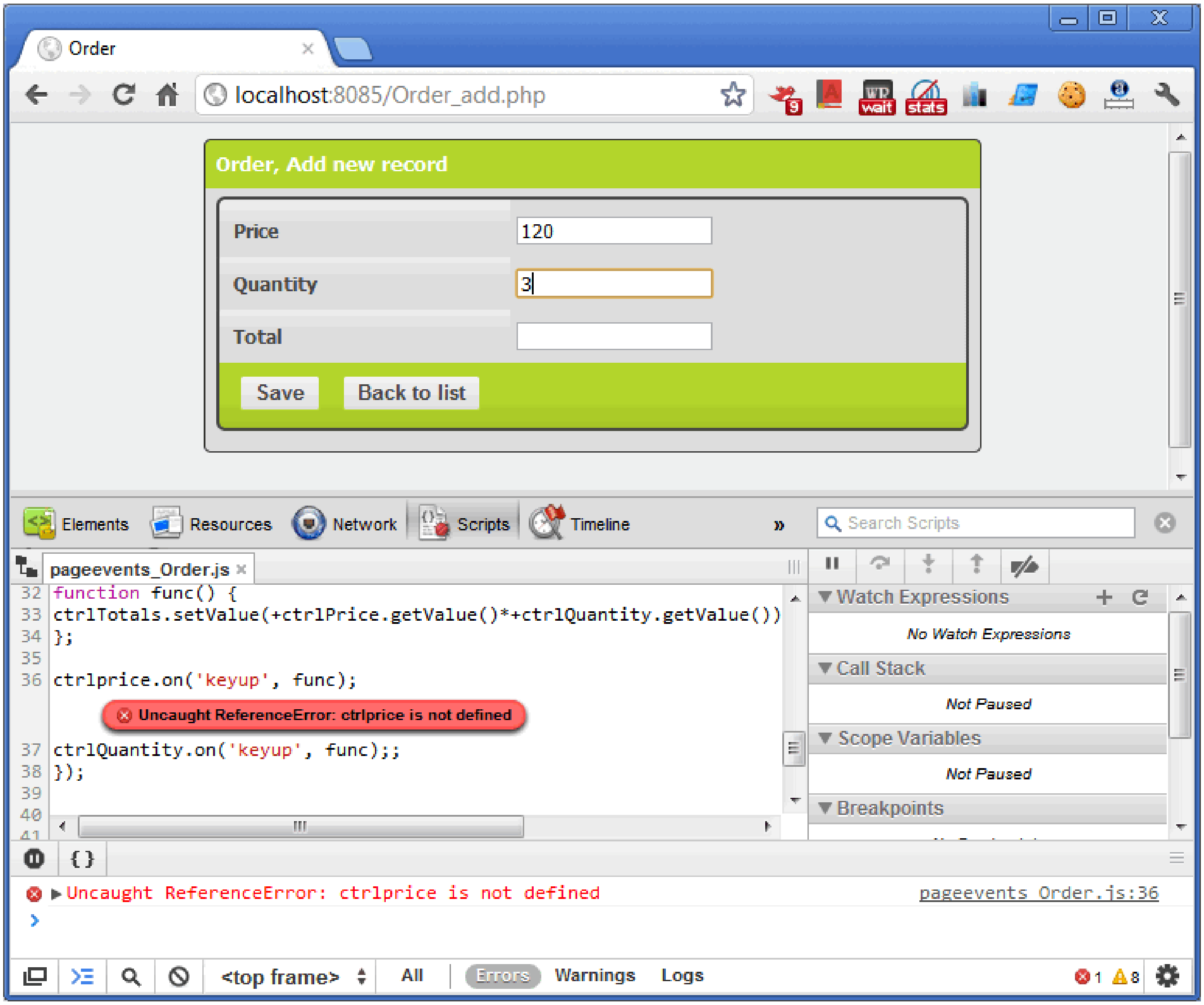
For this purpose we will be using developers tools that come with Chrome browser. If you use Firefox - download and install the Firebug. Firebug is an extension to Mozilla Firefox web browser which allows us to monitor and debug the JavaScript in any web page.

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotals = Runner.getControl(pageid, 'Total');

function func() {
ctrlTotals.setValue(+ctrlPrice.getValue()*+ctrlQuantity.getValue());
};

ctrlprice.on('keyup', func);
ctrlQuantity.on('keyup', func);
```

Let take a look at our first example where we intentionally misspelled the name of the field used in defining the *ctrlPrice* variable on the JavaScript OnLoad event (*ctrlPrice* vs *ctrlprice*, case sensitive). Since this is not a syntax error the Syntax Check won't pick up on it but browser will.

One thing to note here is when you are troubleshooting and you build your project, you might want to uncheck the **Compress javascript files** option on the Output step. This will organize the code in the way that is much easier to follow.

So, when the page is loaded we click F12 to display developers tools panel. We can see our error message there and by clicking error message we can get right to the line of code that causes the trouble. Error message in popup says: 'ctrlprice is not defined'. Replacing *ctrlprice* with correct *ctrlPrice* fixes the issue.



localhost:8085/Order_add.php

Order, Add new record

Price	120
Quantity	3
Total	

Save Back to list

pageevents_Order.js

```
function func() {
ctrlTotals.setValue(+ctrlPrice.getValue()*+ctrlQuantity.getValue())
};

ctrlprice.on('keyup', func);
    Uncaught ReferenceError: ctrlprice is not defined
ctrlQuantity.on('keyup', func);;
});
```

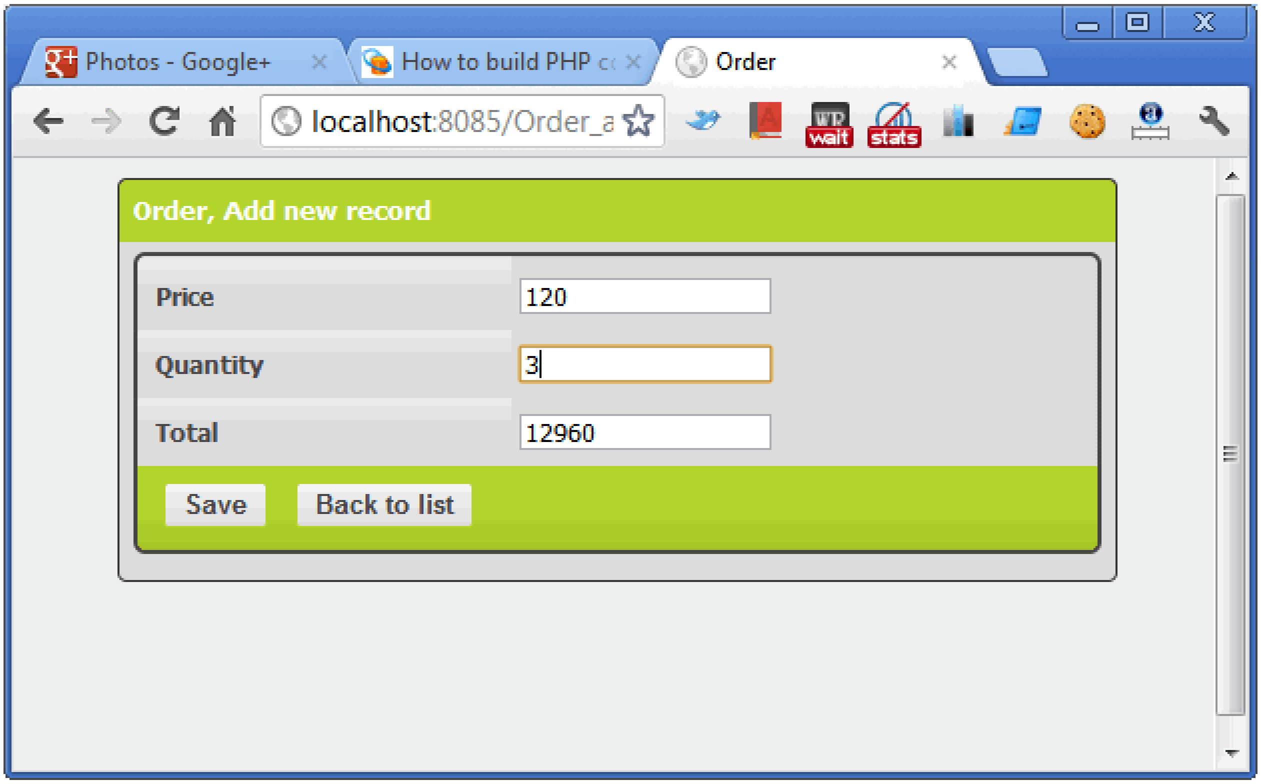
```
Uncaught ReferenceError: ctrlprice is not defined        pageevents_Order.js:36
```

Let's take a look at another example where we are calculating amount on the fly and are not getting the anticipated result. We are calculating the order by multiplying the number of units in the order by the price of the unit and adding a tax to the equation.

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotals = Runner.getControl(pageid, 'Total');
```

```
function func() {
var total=ctrlPrice.getValue()*+ctrlQuantity.getValue();
ctrlTotals.setValue(total * total*0.1 );
};

ctrlprice.on('keyup', func);
ctrlQuantity.on('keyup', func);
```

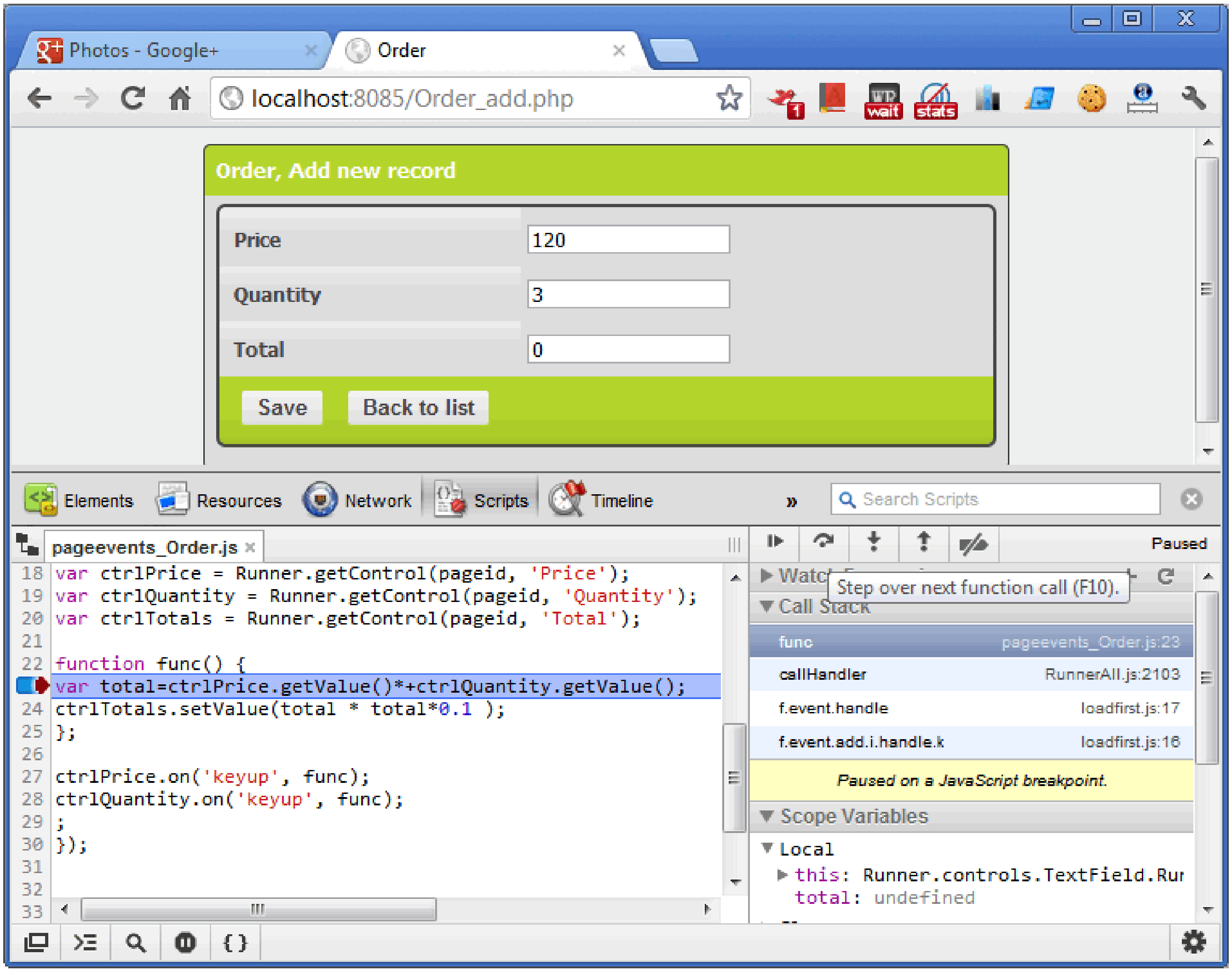
And here is the result it produces:



Order, Add new record

Price	120
Quantity	3
Total	12960

So, in the order where we have 3 units at $120 each with 10% sales tax our total should be $396 dollars. However, the total we are getting is $12960. To troubleshoot the issue we will insert a breakpoint in our Javascript code and watch the values of each part of the equation separately to identify the error. To add a breakpoint we need to proceed to **Scripts** tab in Developers tools, find our file (*pageevents_Order.js*) and click the line number we we want to set our breakpoint.



Order, Add new record

Price	120
Quantity	3
Total	0

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotals = Runner.getControl(pageid, 'Total');

function func() {
var total=ctrlPrice.getValue()*+ctrlQuantity.getValue();
ctrlTotals.setValue(total * total*0.1 );
};

ctrlPrice.on('keyup', func);
ctrlQuantity.on('keyup', func);
;
});
```

Step over next function call (F10).

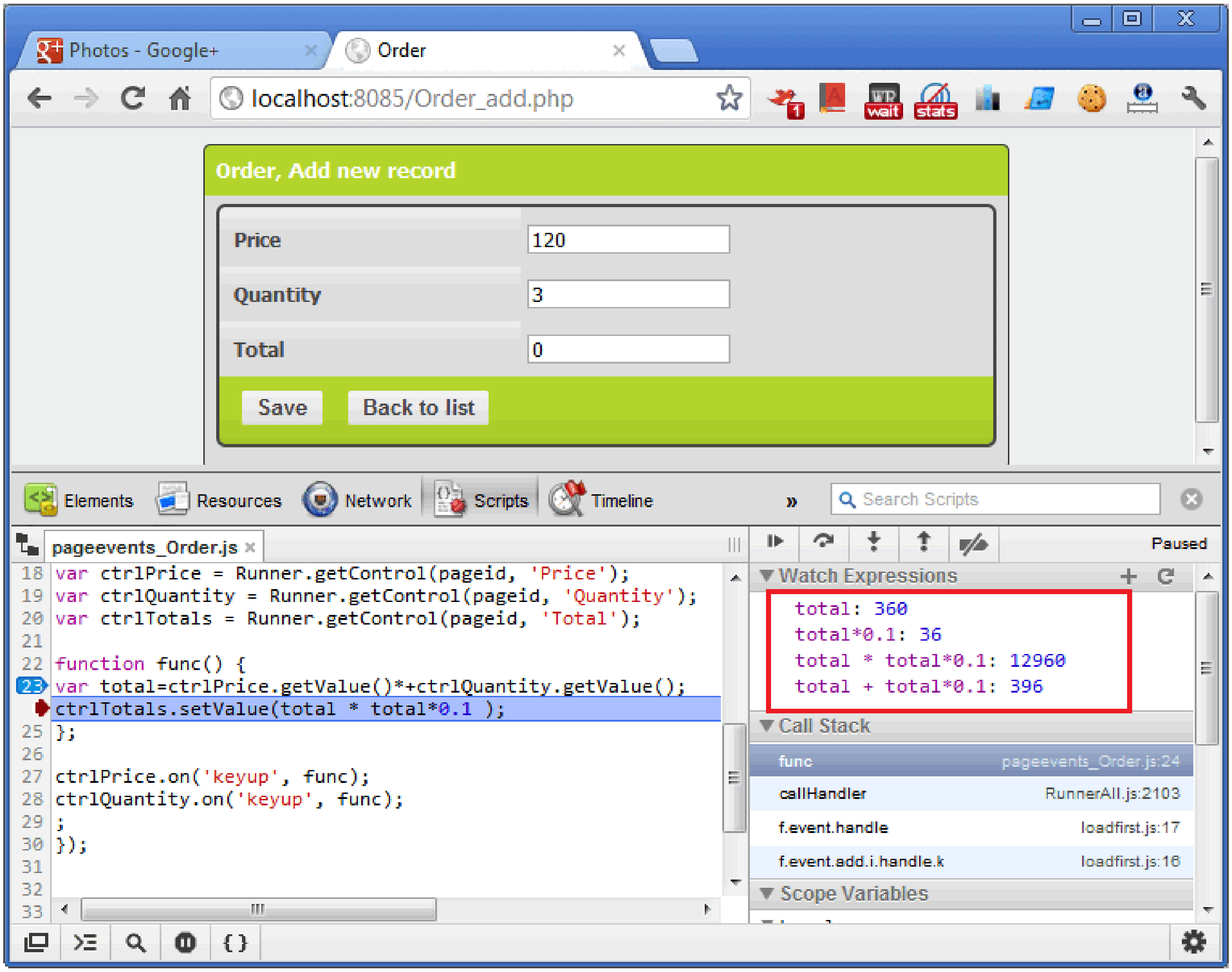
func	pageevents_Order.js:23
callHandler	RunnerAll.js:2103
f.event.handle	loadfirst.js:17
f.event.add.i.handle.k	loadfirst.js:16

Paused on a JavaScript breakpoint.

```
this: Runner.controls.TextField.Rur
total: undefined
```

Once breakpoint is inserted put cursor to *Quantity* field and hit any arrow key on the keyboard. Program execution stops and we can see what's going on. We can use F10 key to move to the next line of code (Step over).

Now lets add a few watch expressions on the right side. We can see that total and tax are calculated properly however we multiplying them instead of adding up which causes the error. A simple correction in the equation logic solves the issue.



Price	120
Quantity	3
Total	0

```
var ctrlPrice = Runner.getControl(pageid, 'Price');
var ctrlQuantity = Runner.getControl(pageid, 'Quantity');
var ctrlTotals = Runner.getControl(pageid, 'Total');

function func() {
var total=ctrlPrice.getValue()*+ctrlQuantity.getValue();
ctrlTotals.setValue(total * total*0.1 );
};

ctrlPrice.on('keyup', func);
ctrlQuantity.on('keyup', func);
;
});
```

```
total: 360
total*0.1: 36
total * total*0.1: 12960
total + total*0.1: 396
```

func	pageevents_Order.js:24
callHandler	RunnerAll.js:2103
f.event.handle	loadfirst.js:17
f.event.add.i.handle.k	loadfirst.js:16

Real life examples are more complicated though basic troubleshooting techniques are the same. As a first step make sure there are no runtime Javascript errors then set a breakpoint and step through the code to find logic flaws.

You can also watch a video version of this tutorial.

3.2.12.4 Troubleshooting tips

When your projects do not work as expected, especially when you use a load of events and custom code, you can use the following debugging techniques to resolve the issue.

1. Print all executed SQL statements on the Web page

For this purpose set **$dDebug** variable in **include/appsettings.php** file to **true**. Especially useful when you need to resolve master-details or advanced security issues.

```
$dDebug = true;
```

```
$dSQL = "";

$bUseMobileStyleOnly = false;
```

You can copy SQL query and run it against your database manually to see if it returns correct results. To run SQL query manually use tools that comes with your database (Query designer in MS Access, Enterprise Manager in SQL Server, phpMyAdmin in MySQL).

Another way is to add the following code to the AfterAppInit event:

```
$dDebug = true;
```

2. Print out variables using echo statement

The echo command is used to write output to a browser. The following example sends the text "Hello World" to the browser:

```
echo "Hello World";
```

Here are several examples of how you can use echo command:

- Print variable value;

```
echo "Variable name: " . $variable;
```

- In BeforeAdd/BeforeEdit events you can print form field value;

```
echo "field_name: " . $values["field_name"];
```

- Displaying Session variable that stores logged in user id;

```
echo "UserID: " . $_SESSION["UserID"];
```

- Displaying IP address of logged in user;

```
echo "Your IP address: " . $_SERVER['REMOTE_ADDR'];
```

- Print all values in the array.

```
echo "all entered values: ";
print_r($values);
```

3. Troubleshooting charts

4. Troubleshooting custom buttons

5. Troubleshooting Javascript errors

3.2.13 Data formatting

As you know, data in a database is stored in raw format. For example, data of *Date* type is stored as *yyyy-mm-dd*. For the pages like List, View, etc. you can customize the data appearance by defining View as settings in Visual Editor.

Using the formatted data in events. ViewControl::Format function

If there is a need to use the formatted data in events, e.g. when sending emails to users, you can use the *ViewControl::Format* function.

Syntax:

```
ViewControl::Format($data, $fieldName, $TableName);
```

Arguments:

- *$data* - raw data to be formatted;
- *$fieldName* - field that contains row data to be formatted;
- *$TableName* - table that contains row data to be formatted.

Example 1. Using *ViewControl::Format* function in *AfterAdd* event

```
echo $values['Date Listed'] ."<br>";
echo ViewControl::Format($values, 'Date Listed', 'Cars' );
```

The first line of code sample will display raw data (e.g. *2014-10-15*), the second one - formatted data (US date format, e.g. *10/15/2014*).

Example 2. Data formatting for a button on View page or button inserted into the grid

```
$data = $button->getCurrentRecord();
$var = ViewControl::Format($data, 'Date Listed', 'Cars');
```

This function can be also used for the field that is configured as a Lookup wizard. In this case raw data is stored in the *Link* field and *ViewControl::Format* function will display the values of *Display* field.

3.2.14 Email temlpates

After enabling Send email to user/admin you will see that **Email templates** button will become active.

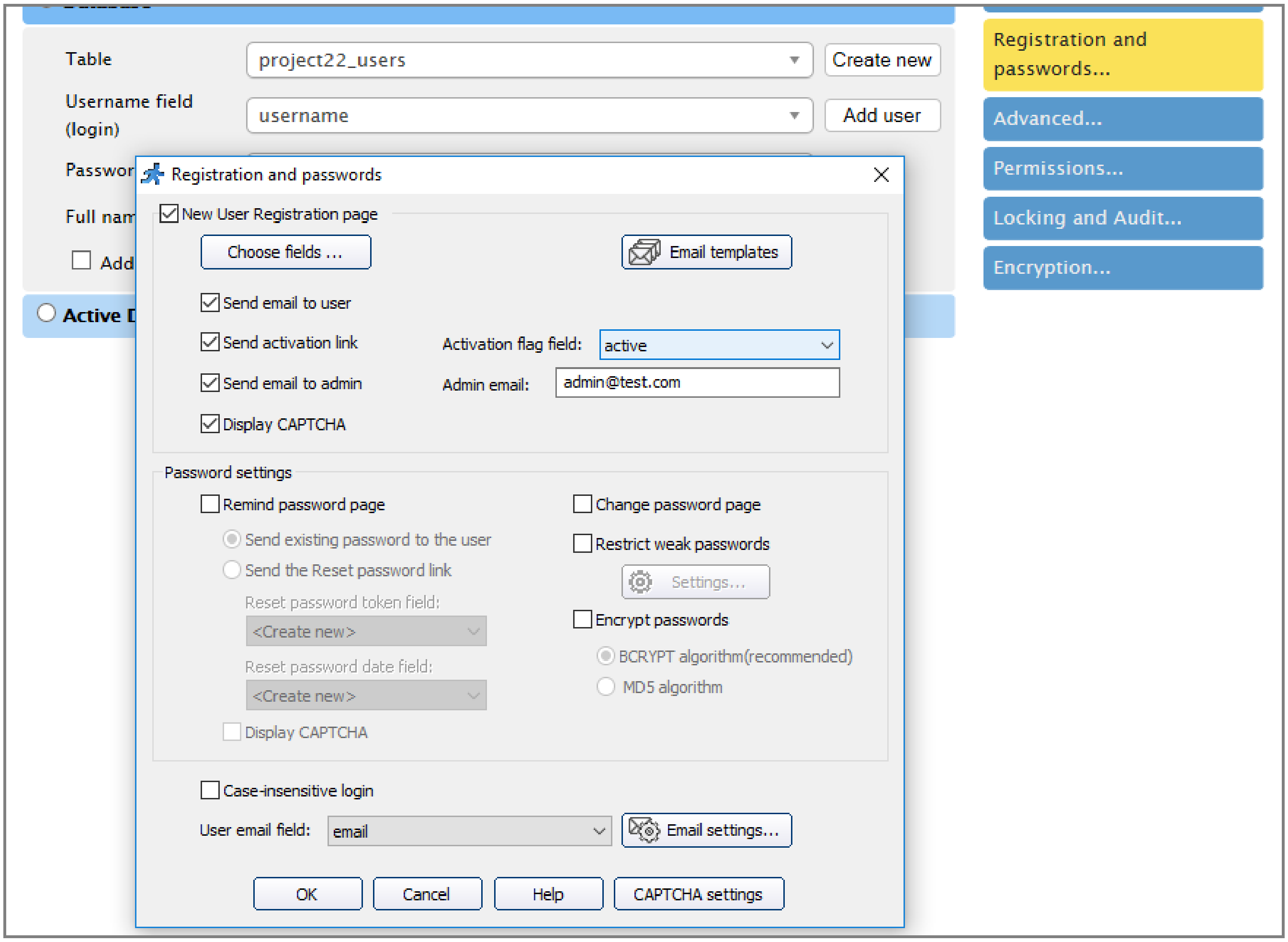


Table: project22_users
Create new

Username field (login): username
Add user

Passwor

Full nam

□ Add

○ Active D

Registration and passwords...

Advanced...

Permissions...

Locking and Audit...

Encryption...

Registration and passwords

☑ New User Registration page

Choose fields ...

Email templates

☑ Send email to user

☑ Send activation link — Activation flag field: active

☑ Send email to admin — Admin email: admin@test.com

☑ Display CAPTCHA

Password settings

□ Remind password page

○ Send existing password to the user

○ Send the Reset password link

Reset password token field: <Create new>

Reset password date field: <Create new>

□ Display CAPTCHA

□ Change password page

□ Restrict weak passwords

Settings...

□ Encrypt passwords

○ BCRYPT algorithm(recommended)

○ MD5 algorithm

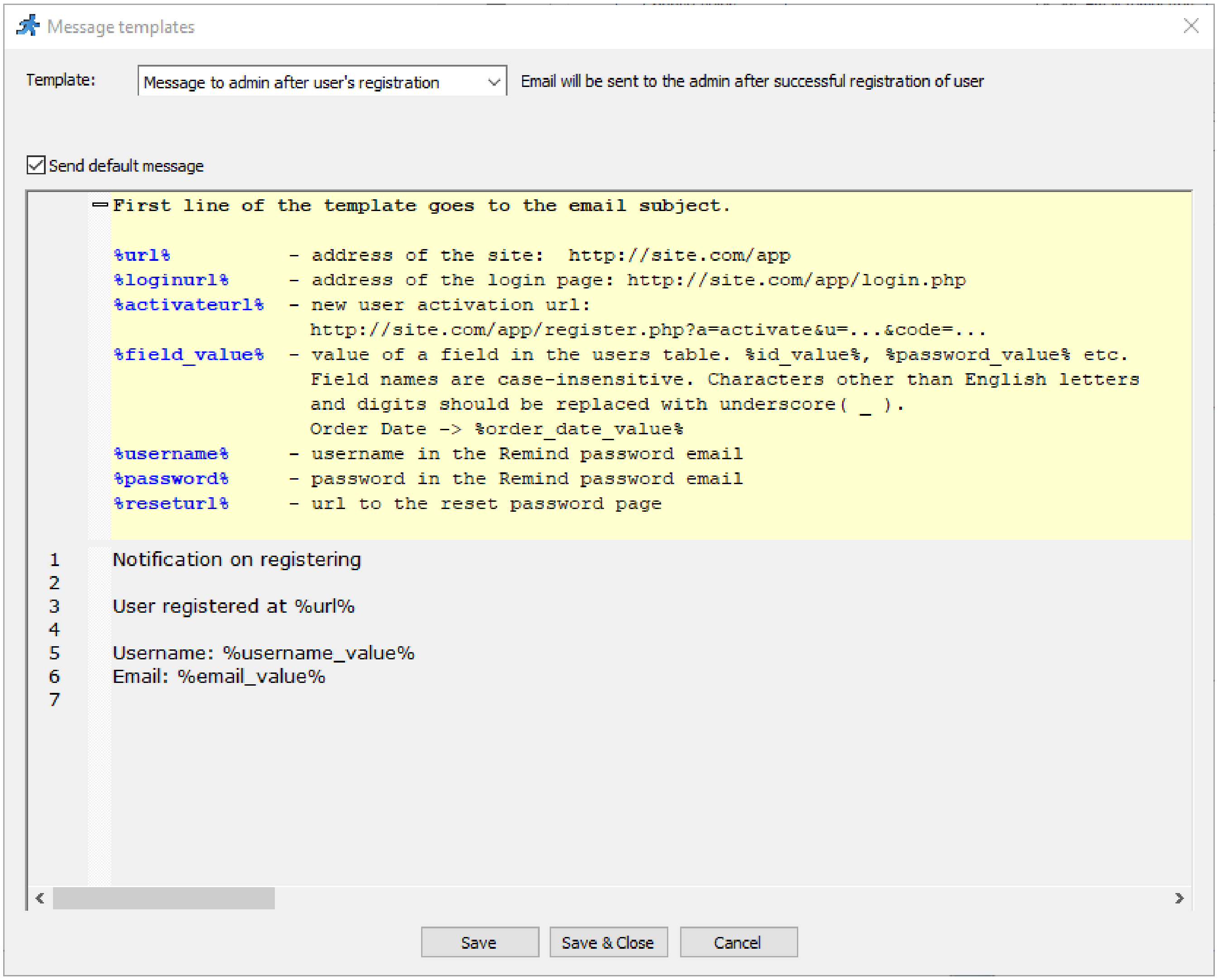
□ Case-insensitive login

User email field: email

Email settings...

OK | Cancel | Help | CAPTCHA settings

Click the button to open the dialog. Here you can customize email templates that will be send to the users.


Message templates
Template:
Message to admin after user's registration
Email will be sent to the admin after successful registration of user
☑ Send default message
Save
Save & Close
Cancel

```
First line of the template goes to the email subject.

%url%           - address of the site:  http://site.com/app
%loginurl%      - address of the login page: http://site.com/app/login.php
%activateurl%   - new user activation url:
                  http://site.com/app/register.php?a=activate&u=...&code=...
%field_value%   - value of a field in the users table. %id_value%, %password_value% etc.
                  Field names are case-insensitive. Characters other than English letters
                  and digits should be replaced with underscore( _ ).
                  Order Date -> %order_date_value%
%username%      - username in the Remind password email
%password%      - password in the Remind password email
%reseturl%      - url to the reset password page
```

```
Notification on registering

User registered at %url%

Username: %username_value%
Email: %email_value%
```

3.2.15 How to access fields in the field events

How to access fields in the field events

You can access fields from the ClientBefore, ClientAfter events:

Variables available in the field events:

ctrl - current field, Control object

pageObj - RunnerPage object

```
ctrl.setValue(100);
```

ctrl.getPeer(field)

Returns other field control from the same page of the same row in the inline mode:

```
var ctlPrice = ctrl.getPeer('price');
ctlPrice.setValue( 1000 );
```

pageObj.getControl(fieldName)

Returns field control. Works in any events where pageObj variable is available

```
var ctlPrice = pageObj.getControl('price');
ctlPrice.setValue( 1000 );
```

You can also access fields from the Server event:

$result - array of values to return from the server

getCurrentRecord() - returns an associative array with field values (field name => value)

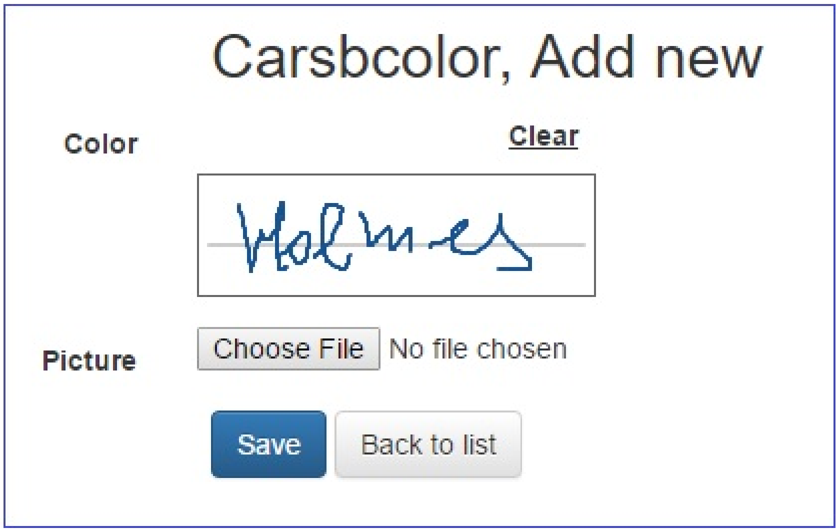
```
$data = $ajax->getCurrentRecord();
$result["record"] = $data;
$result["email"] = $data["email"];
```

3.2.16 How to create your own Edit control plugin

PHPRunner adds an exciting new feature - custom edit controls. You are no longer limited by stock Edit controls that come with the software. And the best of all, creating new Edit controls is not complicated and we'll show how this can be done.

We will show you how to create ColorPicker and SignaturePad plugins. Before we proceed check the following live demo that showcases both edit controls plugins. SignaturePad control works on mobile devices as well.

ColorPicker and SignaturePad edit controls as seen in browser:



ColorPicker control

Lets say we want to add a color picker control that allows users select color the same way they do in Adobe Photoshop. Since our software comes with jQuery bundled we'll be searching web for "jQuery colorpicker". This search returns a number of results and the one we'll be using is miniColors. It looks nice and easy to integrate. Sources can be find at Github:

- create a new folder under *<Documents>\PHPRunnerPlugins\edit* named *ColorPicker;*
- copy files *EditMyField.js* and *EditMyField.php* from *MyField* folder to *ColorPicker* folder;
- rename those files to *EditColorPicker.js* and *EditColorPicker.php* respectively;
- if your plugin needs extra files copy them to *ColorPicker* folder keeping folders structure;
- open *EditColorPicker.js* in any text editor and replace all occurrences of *EditMyField* with *EditColorPicker*. Do the same with *EditColorPicker.php* file.
- use *addJSFiles()* function in *ColorPicker.php* to add references to external Javascript files:

```
$this->pageObject->AddJSFile("jquery.miniColors.min.js");
```

  You may need to specify the load order of Javascript files. In this example file *second.js* will be loaded after *first.js*:

```
$this->pageObject->AddJSFile("second.js", "first.js");
```

- use *addCSSFiles()* function to add CSS files:

```
$this->pageObject->AddCSSFile("jquery.miniColors.css");
```

In *EditColorPicker.php* file find *buildUserControl()* function. This is were you build HTML code for your control. If you leave predefined code as is it will display a simple text edit box. We only going add a minor change telling colorpicker control to use black color theme:

```
class="black"
```

Lets see how we can turn this edit box into a colorpicker control. According to colorpicker instructions we need to call miniColors Javascript referencing edit box. We'll do so adding the following code to constructor function:

```
$("#"+this.valContId).miniColors({
letterCase: 'uppercase'
});
```

letterCase option tells control to convert color values entered manually to upper case. *this.valContId* is the id of the control.

This is it, our control is ready. You can now launch PHPRunner, select ColorPicker as *Edit as* type for any text field and enjoy your color picker on Add/Edit pages.

You may consider some additional enhancements:

1. It would be nice if instead of hex color value we can show some visual representation of selected color on List/View pages. We'll do so choosing 'View as' type 'Custom' and putting the following code there:

```
$value="<span style='padding: 5px; background:
".$value."'>".$value."</span>";
```

2. By default PHPRunner sets focus to the first edit control when Add or Edit page is loaded. This is not a desired behaviour for colorpciker control as we do not want to see colorpicker popup window to open every time page is loaded. To prevent this from happening implement setFocus function - simply return false every time.

If you need to change control behaviour check all functions and events in *source\include\common\runnerJS\Control.js* file.

SignaturePad

The SignaturePad plugin allows to a add signature pad to your forms. SignaturePad works with both mouse and touch devices. We will be using SignaturePad jQuery plugin that comes with excellent documentation and examples.

The basic setup is the same: create new folder for SignaturePad plugin, copy and rename files, add files that plugin needs to plugin directory. This plugin is a bit more complicated and takes a few extra steps to integrate.

1. Build control's HTML code

Here is how we do this in *buildUserControl()* function:

```
echo '
<div class="sigPad" style="width: '.($this->width+2).'px;">
<ul class="sigNav">
<li class="clearButton"><a href="#clear">Clear</a></li>
</ul>
```

```
<div class="sig sigWrapper">
<div class="typed"></div>

<canvas class="pad" width="'.$this->width.'" height="'.$this-
>height.'"></canvas>
<input id="'.$this->cfield.'" type="hidden" '
.'name="'.$this->cfield.'" class="output">
</div>
</div>

';
```

2. Convert signature data to image

Signature data is recorded in JSON format as a sequence of lines and passed to the web server this way. Now we need to convert JSON data to PNG file and save it in 'files' folder on the web server. Luckily all the hard work was already done and all we need to do is to add a few lines of code to *readWebValue()* function:

```
if ($this->webValue) {
   // save signature to file
   require_once 'signature-to-image.php';
   $img = sigJsonToImage($this->webValue, array(
            'imageSize' => array($this->width, $this->height)
            ,'bgColour' => $this->bgcolor
            ));
   $filename= $this->folder."/".generatePassword(15).".png";
   imagepng($img, $filename);
   $filesize = filesize($filename);

   // prepare image info to be saved in db
   $result[] = array("name" => $filename,
   "usrName" => 'signature.png', "size" => $filesize, "type" =>
"image/png",
   "searchStr" => 'signature.png'.":sStrEnd");
   $this->webValue = my_json_encode($result);
   }
```

You can read more info JSON to image conversion here.

Note the way how file info is stored in the database. Since new version offers multiple files upload we need to be able to store more info there than just file name. Besides the file name itself we also save file size, file type and path to the file in JSON format. Here is how typical file upload field looks in the database:

```
[{"name":"files\/h8hsoz5hd23b0ik.jpg", "usrName":"Chrysanthemum.jpg",
"size":879394,"type":"image\/jpeg",
"searchStr":"Chrysanthemum.jpg:sStrEnd"},
{"name":"files\/2p85jz854o6fbv8.jpg",
"usrName":"Desert.jpg","size":845941, "type":"image\/jpeg",
"searchStr":"Desert.jpg:sStrEnd"},
{"name":"files\/pm4fu8uv2u6xc1w.jpg", "usrName":"Hydrangeas.jpg",
"size":595284,"type":"image\/jpeg", "searchStr":"Hydrangeas.jpg:sStrEnd"}]
```

3. Customization

Now it's the time to customize our plugin. Users may need to change the appearance and behaviour of signature pad i.e.:

- change width and height of signature pad
- change background color
- change folder where image files are stored
- make signature pad field required

As a first step we need to learn to pass settings from PHPRunner wizard to plugin. Proceed to "Edit as" dialog in PHRRunner and click **Add initialization script** button. Here is the sample set of settings of SignaturePad control:

```
// signature field height
$this->settings["height"] = 100;
// signature field width
$this->settings["width"] = 300;
// signature background color
$this->settings["bgcolor"] = array(0xff, 0xff, 0xff);
// set it to true to make signature field required
$this->settings["required"]=false;
// folder to store signature files
$this->settings["folder"]="files";
```

This code is self-descriptive, you can pass any number of settings there. If you create your own edit control plugin place sample initialization script to *sample.php* file that needs to be located in the plugin folder.

Now we can access those settings in plugin's *initUserControl()* function. We can also pass settings to Javascript part of the plugin.

```
// setting default value
$this->required = false;

// saving settings to the local variable
if ($this->settings["required"])
   $this->required=$this->settings["required"];

// passing settings to Javascript
$this->addJSSetting("required", $this->required);
```

Now let's actually add some new functionality.

Making signature field required

In Javascript *init()* function add the following code:

```
if (this.required)
this.addValidation("IsRequired");
```

Signature is not required by default. To make it required add the following line of code to initialization script under 'Edit as' properties:

```
$this->settings["required"]=false;
```

Setting width and height of signature pad

Passing width and height from 'Edit as' settings in PHPRunner:

```
$this->settings["height"] = 100;
$this->settings["width"] = 300;
```

Then we can use *$this->width* and *$this->height* in *buildUserControl()* function to specify width and height of our control:

```
<canvas class="pad" width="'.$this->width.'" height="'.$this-
>height.'"></canvas>
```

Changing folder where signature images stored

Passing folder name from 'Edit as' settings in PHPRunner:

```
$this->settings["folder"]="files";
```

Using this variable in *readWebValue()* function:

```
$filename= $this->folder."/".generatePassword(15).".png";
```

Changing signature pad background color

Passing background color from 'Edit as' settings in PHPRunner:

```
$this->settings["bgcolor"] = array(0xff, 0xff, 0xff);
```

bgcolor array contains color value in RGB format (Red, Green, Blue, each color ranges from 0 to 255, 0xff is a hexadecimal representation of 255. array(0xff, 0xff, 0xff) means white color.

Now in Javascript control constructor function we can use *this.bgColor* to pass background color to SignaturePad control:

```
$('.sigPad').signaturePad({drawOnly:true, bgColour: this.bgColor});
```

We also need to pass backgound color to *sigJsonToImage()* function that converts JSON signature data to image. We use *$this->bgcolor* variable here in *readWebValue()* PHP function.

```
$img = sigJsonToImage($this->webValue, array(
      'imageSize' => array($this->width, $this->height)
```

```
        ,'bgColour' => $this->bgcolor
        ));
```

This is it. As you can creating your own edit control plugins is not a rocket science and you can build something useful in just a few lines of code.

If you develop a new edit control plugin on your own and want to share it with other users feel free to contact our support team. We'll be launching our marketplace soon where you can earn a buck or two selling plugins you have developed.

3.2.17 How to display message or tooltip

How to display message or tooltip

ctrl - current field, Control object

These functions are available in Bootstrap layouts only.
They don't work in Inline Add/Edit mode.

ctrl.getTooltip()

Returns current tooltip text including HTML formatting

ctrl.setTooltip(message)

Set tooltip message. HTML formatting is supported.

ctrl.addTooltip(message)

Append message to the tooltip. HTML formatting is supported.

```
ctrl.setTooltip( '<b>' + ctrl.getValue() + '</b> is a great field value!');
```

3.2.18 How to mark field invalid

How to mark field invalid

ctrl - current field, Control object

ctrl.setInvalid(message)

ctrl.setValid()

Mark control invalid, mark it valid again

The page can not be saved when there are invalid fields on it.

setValid() only removes status set by setInvalid(), it has no effect on other validations.

```
if( ctrl.getValue() > 100 )
  ctrl.setInvalid( 'Price can not exceed 100' );
else
  ctrl.setValid();
```

3.2.19 How to display data returned by stored procedure

How to display data returned by stored procedure

The following stored procedure example returns data from the "test" table. It should be created in the in the SQL Server.

```
CREATE PROCEDURE [dbo].[test_proc]
AS
BEGIN
SET NOCOUNT ON;
SELECT * from test
END
```

1. Add 'test' table or any other table to the project

2. Add the following code to List page: CustomQuery event:

```
return DB::Query("EXEC test_proc");
```

3. Add the following code to List page: FetchRecords event:

```
return $rs->fetchAssoc();
```

4. Add the following code to ListGetRowCount event

```
return DBLookup("select count(*) from test");
```

See also

- DB_Query
- ListFetchArray
- fetchAssoc
- How to execute stored procedures

3.2.20 How to execute SQL Server/MySQL/ORACLE stored procedures

Executing SQL Server stored procedure from the event

Calling procedure without parameters:

```
CustomQuery("EXEC StoredProcNameHere");
```

Passing one of field values as a parameter:

```
CustomQuery("EXEC StoredProcNameHere '" . $values["FieldName"] . "'");
```

Executing MySQL stored procedure from the event

Calling procedure without parameters:

```
CustomQuery("CALL StoredProcNameHere");
```

Passing one of field values as a parameter:

```
CustomQuery("CALL StoredProcNameHere ('" . $values["FieldName"] . "')");
```

Executing ORACLE stored procedure from the event

Calling procedure without parameters:

```
CustomQuery("BEGIN STOREDPROCNAME END;");
```

Passing one of field values as a parameter:

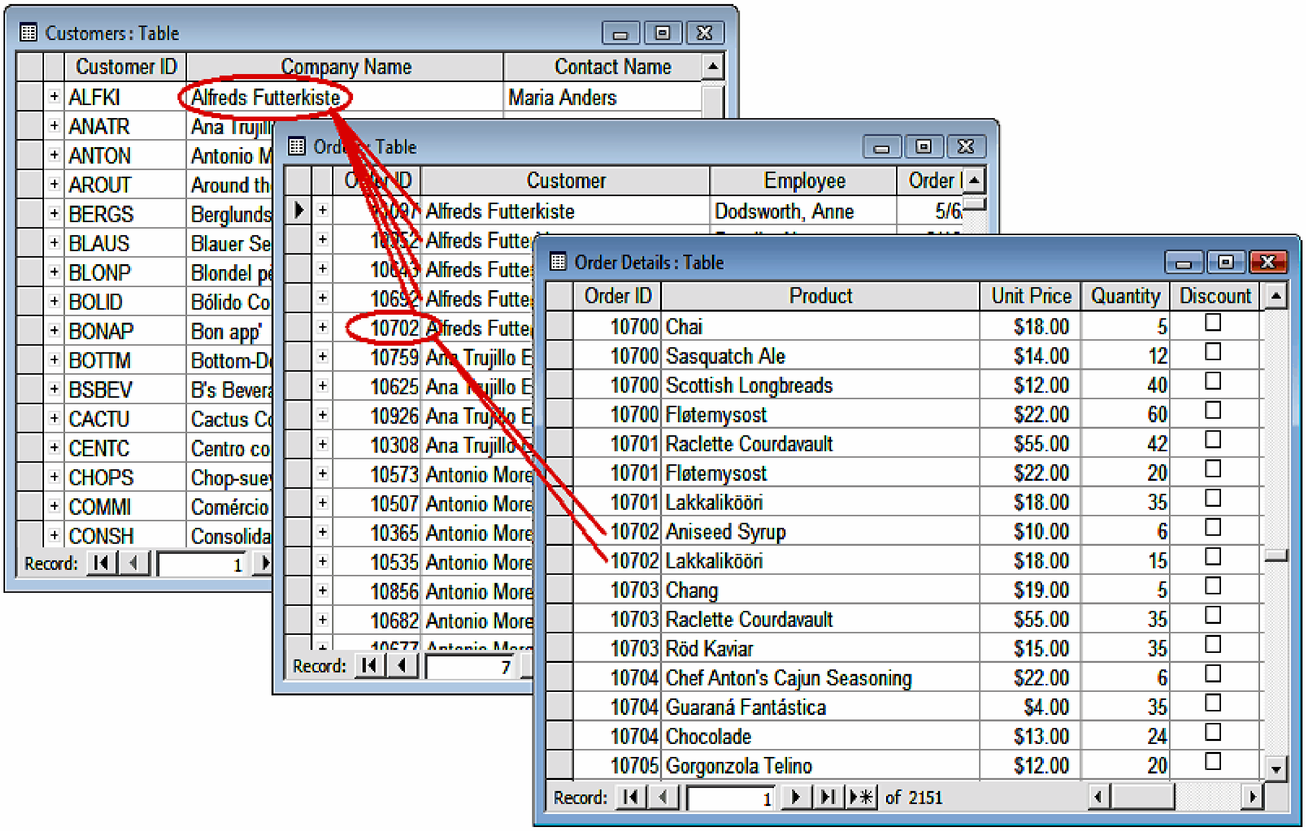
```
CustomQuery("BEGIN STOREDPROCNAME('" . $values["FieldName"] . "'); END;");
```

3.2.21 Master-details relationships

A one-to-many relationship, often referred to as a "master-details" or "parent-child" relationship, is the most usual relationship between two tables in a database.

Common scenarios include customer/purchase data, patient/medical-record data, and student/course-result data. For example, each customer is associated with at least one order record. Valued customers have many order records involving significant sums and often a user needs to view one in connection with the other. In a one-to-many relationship, a record in Table A can have (none or one or) more than one matching record in Table B, but for every record in Table B there is exactly one record in Table A.

For example, say you have **Orders** table and **Order Details** table , where **order number** is a common field in each. You can create a master-details relationship that will enable you to navigate through the Orders and jump to Order Details that belong to current order only.



Customers : Table

	Customer ID	Company Name	Contact Name
+	ALFKI	Alfreds Futterkiste	Maria Anders
+	ANATR	Ana Trujill	
+	ANTON	Antonio M	
+	AROUT	Around th	
+	BERGS	Berglunds	
+	BLAUS	Blauer Se	
+	BLONP	Blondel pe	
+	BOLID	Bólido Co	
+	BONAP	Bon app'	
+	BOTTM	Bottom-D	
+	BSBEV	B's Bevera	
+	CACTU	Cactus Co	
+	CENTC	Centro co	
+	CHOPS	Chop-suey	
+	COMMI	Comércio	
+	CONSH	Consolida	

Record: 1

Orders : Table

	Order ID	Customer	Employee	Order I
+	[illegible]	Alfreds Futterkiste	Dodsworth, Anne	5/6
+	10952	Alfreds Futte		
+	10643	Alfreds Futte		
+	10692	Alfreds Futte		
+	10702	Alfreds Futte		
+	10759	Ana Trujillo E		
+	10625	Ana Trujillo E		
+	10926	Ana Trujillo E		
+	10308	Ana Trujillo E		
+	10573	Antonio More		
+	10507	Antonio More		
+	10365	Antonio More		
+	10535	Antonio More		
+	10856	Antonio More		
+	10682	Antonio More		
+	10677	Antonio More		

Record: 7

Order Details : Table

Order ID	Product	Unit Price	Quantity	Discount
10700	Chai	$18.00	5	☐
10700	Sasquatch Ale	$14.00	12	☐
10700	Scottish Longbreads	$12.00	40	☐
10700	Fløtemysost	$22.00	60	☐
10701	Raclette Courdavault	$55.00	42	☐
10701	Fløtemysost	$22.00	20	☐
10701	Lakkalikööri	$18.00	35	☐
10702	Aniseed Syrup	$10.00	6	☐
10702	Lakkalikööri	$18.00	15	☐
10703	Chang	$19.00	5	☐
10703	Raclette Courdavault	$55.00	35	☐
10703	Röd Kaviar	$15.00	35	☐
10704	Chef Anton's Cajun Seasoning	$22.00	6	☐
10704	Guaraná Fantástica	$4.00	35	☐
10704	Chocolade	$13.00	24	☐
10705	Gorgonzola Telino	$12.00	20	☐

Record: 1 of 2151

3.2.22 PHPRunner session variables

Starting with version 9.7 previously recommended session variables were replaced with functionality available via Security API and Search API.

You can still use the session variable $_SESSION["language"] that stores selected language, e.g. "English" or "Spanish".

Examples

1. To get the current value of this variable use the following function that returns current selected language.

```
mlang_getcurrentlang()
```

2. If you need to assign the value to the variable, use the variable directly.

```
$_SESSION["language"]="English";
```

3.2.23 PHPRunner templates

PHPRunner uses built-in template language. PHPRunner templates cleanly separates your presentation layer (HTML, CSS, etc.) from your application code. The main idea is to simply visual templates moving all logic and javascript to PHP files. This makes Visual Editor stable and eliminates the need to reset pages.

Template language reference

- all templates tags are enclosed within delimiters { and }. All content outside delimiters is displayed as static HTML.

- template variables start with dollar ($) sign. They may contain numbers, letters and underscores.

Example: `{$variable_name}`

`{BEGIN block_name} {END block_name}` is used to define block section (loops, condition statements).

Functions

- `assign($name, $val)` is used to assign value $val to the variable $name.
- `assignbyref($name, &$val)` is similar to assign() except for value $val is passed by reference. Use assignbyref() when $val variable is array. See {block body} example below.
- `assign_section($name, $begin, $end)` is used to define block section.
- `assign_loopsection($name, &$data)` is used to define loop section.
- `display($template)` renders and displays the template file.

How templates work

Template files (*.htm) can be found in **templates** directory under **output** directory.

Here is the basic example of how templates work:

list.php

```
include('libs/xtempl.php');
// create object
$xt = new Xtempl();
// assign some content. This would typically come from
// a database or other source, but we'll use static
// values for the purpose of this example.
$xt->assign('name', 'george smith');
$xt->assign('address', '45th & Harris');
// display it
$xt->display('list.htm');
```

The template file then contains the output interspersed with tags that PHPRunner replaces with assigned content.

list.htm

```
<html>
<head>
<title>User Info</title>
</head>
<body>
User Information:<p>
Name: {$name}<br>
Address: {$address}<br>
</body>
</html>
```

output

```
<html>
<head>
<title>User Info</title>
</head>
<body>
User Information:<p>
Name: george smith<br>
Address: 45th & Harris<br>
</body>
</html>
```

{BEGIN} ... {END} blocks

Template file contains a set of code sections wrapped by {BEGIN ...} and {END ...}.

view.htm

```
{BEGIN Model_fieldblock}
  <tr><td class=shade width=150>Model</td><td width=250>
  {$Model_value}&nbsp;
  </td></tr>
{END Model_fieldblock}
```

In view.php file use the following:

`$xt->assign("Model_fieldblock",true);` - code snippet between {BEGIN ...} and {END ...} app

`$xt->assign("Model_fieldblock",true);` - code snippet goes away.

How to deal with Javascript

Earlier we mentioned that javascript and some HTML tags like <form> and <input type=hidden ...> need to be moved to PHP code in order to keep templates clean.

On the server side javascript code needs to be assigned to **body** or **end** variables in block array.

list.php

```
// create an array
$body = array();
// code assigned to "begin" variable replaces {BEGIN body} tag
$body["begin"]='<form name="frmSearch" method="GET"
  action="carsadmin_cars_list.php">
<input type="Hidden" name="a" value="search">
<input type="Hidden" name="value" value="1">
</form>';
// code assigned to "end" variable replaces {END body} tag
$body["end"]="<script>if(document.getElementById('SearchFor'))
document.getElementById('ctlSearchFor').focus();</script>";
// using assignbyref() function as body is an array
$xt->assignbyref("body",$body);
```

Template file and generated output

list.htm

```
{BEGIN body}
...
{END body}
```

output

```
<form name="frmSearch" method="GET"
action="carsadmin_cars_list.php">
<input type="Hidden" name="a" value="search">
<input type="Hidden" name="value" value="1">
</form>
...
<script>
if(document.getElementById('SearchFor'))
document.getElementById('ctlSearchFor').focus();
</script>
```

3.2.24 runner_mail function

Description

Function **runner_mail** is PHPRunner wrapper for *mail()* function.
To use it you need to setup email parameters like From, SMTP server etc in Email settings.

Syntax

```
runner_mail($params)
```

Arguments

$params - array with input parameters. The following parameters are supported:

- *from* - sender email address. If none specified an email address from the wizard will be used.
- *fromName* - sender name.
- *to* - receiver email address.
- *cc* - email addresses of secondary recipients.
- *bcc* - email addresses of recipients whose addresses are not to be revealed to other recipients of the message.
- *replyTo* - reply email address.
- *priority* - message priority (use '1' for urgent, '2' - high, '3' - normal).
- *body* - plain text message body.
- *htmlbody* - html message body (do not use 'body' parameter in this case).
- *charset* - html message charset. If none specified the default website charset will be used.
- *attachments* - attachments description. The *'path'* (a path to the attachment) is required. Other parameters are optional: *'name'* overrides the attachment name, *'encoding'* sets a file encoding, *'type'* sets a MIME type. Attachments will only work when PHP mail is not in use. Email settings.

Return value

Array with the following keys:

- mailed: (true or false) indicates whether email was sent or not.
- message: error message in case the email was not sent

Examples

Send simple email

```
$email="test@test.com";
$message="Hello there\nBest regards";
$subject="Sample subject";
runner_mail(array('to' => $email, 'subject' => $subject,
  'body' => $message));
```

Send HTML email

```
$email="test@test.com";
$message="Hello there\n<b>Best regards</b>";
$subject="Sample subject";
runner_mail(array('to' => $email, 'subject' => $subject,
  'htmlbody' => $message, 'charset' => 'UTF-8'));
```

Example with error handling

```
$email="test@test.com";
```

```
$subject="Sample subject";
$body="test";
$arr = runner_mail(array('to' => $email, 'subject' => $subject,
  'body' => $body));
// if error happened print a message on the web page
if (!$arr["mailed"])
{
  echo "Error happened: <br>";
  echo $arr["message"];
}
```

Send email with BCC and CC fields

```
$email="test@test.com";
$message="Hello there\nBest regards";
$subject="Sample subject";
runner_mail(array('to'  =>  $email, 'cc' => 'test2@test.com',
'bcc' => 'test3@test.com', 'subject' => $subject, 'body' => $message));
```

Send email with From and FromName fields

```
$email="test@test.com";
$message="Hello there\nBest regards";
$subject="Sample subject";
$from="bgates@microsoft.com";
$fromName="Bill Gates";
runner_mail(array('to'  =>  $email, 'subject' => $subject,   'body' =>
$message, 'fromName'=> $fromName, 'from' => $from));
```

Send email to multiple recipients

```
$email="test@test.com,test2@test.com,test3@test.com";
$message="Hello there\nBest regards";
$subject="Sample subject";
runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$message));
```

Send email with attachments

```
$from = "myemail@test.com";
$to = "myclient@test.com";
$msg = "Find some documents (Invoice.pdf, Photo.jpg, signature.jpg)
attached.";
$subject="Documents";

$attachments = array();
```

```
// Attachments description. The 'path'(a path to the attachment) is
required. Others parameters are optional:
//'name' overrides the attachment name, 'encoding' sets a file encoding,
'type' sets a MIME type
$attachments =  array(
        array('path' => getabspath('files/Invoice.pdf')),
        array('path' => getabspath('files/Photo12.jpg'),
                    'name' => 'Photo.jpg',
                    'encoding' => 'base64',
                    'type' => 'application/octet-stream'),
        array('path' => getabspath('files/signature.jpg'))
) ;

$ret = runner_mail(array('from' => $from, 'to' => $to, 'subject' =>
$subject, 'body' => $msg, 'attachments' => $attachments));

if(!$ret["mailed"]){
    echo $ret["message"];
}
```

Send email with new record data. Use this code in the After Add event.

```

$from = "myemail@test.com";
$to = "myclient@test.com";
$msg = "The new record was added: ";
$subject="New record";

$msg.= "Name: ".$values["name"]."\r\n";
$msg.= "Email: ".$values["email"]."\r\n";
$msg.= "Age: ".$values["age"]."\r\n";

$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' =>
$msg, 'from'=>$from));
if(!$ret["mailed"]){
    echo $ret["message"];
}
```

3.2.25 runner_sms function

Description

Sends text message to specified phone number.

Syntax

```
runner_sms($number, $message)
```

Arguments

$number - phone number
$message - text message

Return value

Array with the following keys:

- success: (true or false) indicates whether sms message was sent or not.
- error: error message in case the sms was not sent

Example

Send sms to number. This function will only work if you have Twilio settings entered under Security -> Two factor authentication

```
$number="+12345678901";
$message="You verification code";

runner_sms($number, $message);
```

3.2.26 Template files processing rules (Files.txt)

The 'Files.txt' file (located in 'source' directory) contains the list of template processing rules.

<source file> <destination file> [table name]

'Table name' is an optional parameter that must be provided for templates that are table specific such as 'list.php', 'edit.php' etc. If the 'table name' parameter is a specified file, it will be processed for each table in the project. Otherwise the template file will be processed just once.

Files.txt supports a conditional compilation. See Template language reference for more info on syntax.

Example:

```
##if @t.bAdd || @t.bInlineAdd##
add.php ##@t.strShortTableName##_add.php ##@t.strDataSourceTable##
```

```
##endif##
```

This code snippet tells the wizard to create Add page (tablename_add.php file) if the '**Add**' or '**Inline Add**' options have been selected for this table.

Each business template (Cars, Vacations, Paypal etc) has its own 'files.txt' which overrides the rules defined in the main file.

If you add your own files to the template, it's necessary for you to add a new line to the 'files.txt' file that defines how new file should be processed.

3.2.27 Template language

Quick jump
Expressions
Operators
Statements
Output modifiers
Macros and constants
Additional language elements

Template language is the framework that powers visual and code templates in PHPRunner. Most template language expressions are references to the project file. Template language elements are wrapped by ## characters:

```
##if @field.m_bAddPage##
```

Expressions

1. Strings

Example:

- "string1" - string1;
- "this is a \"string\"" - this is a "string";
- "\"\\\" is a backslash" - "\" is a backslash.

2. Numbers

Examples:

- 2
- 3.3
- -2

3. Variables

Variables start with @ character.

Examples:

- @BUILDER - root element of project file.
- @TABLE - pseudo-variable that points to the current selected table.
- @a, @field - regular variables.
- @TABLE.arrFieldObj - array of fields that belong to the current table.
- @field.EditFormatObj.m_strDefaultValue - default field value.

Variables belong to one of the following data types: strings, numbers, objects and arrays.

4. Boolean expressions

0,"" - false, anything else - true.

Operators

1. Comparison operators

- == - equals.
- = or <> - does not equal.
- < - less than.
- <= - less than or equal to.
- > - greater than.
- >= - greater than or equal to.

You can only compare numbers and strings. Comparison result is either 0 or 1.

2. Boolean

- or or ||
- and or &&
- not or !

Result is either 0 or 1.

3. Parenthesis

Example:

```
(@field.m_bAddPage or @field.m_strLabel=="ID") and not
@Table.m_strCaption==""
```

4. Dot operator

To access structure members, the operator used is the dot operator denoted by (.).

Example:

```
@field.m_ListFormatObj.m_nColWidth
```

5. The Array length property is 'len'

Example:

```
@TABLE.m_arrFieldObj.len
```

6. Priority order

There is an established order with the priority of each operator. From greatest to lowest priority, the priority order is as follows:

1. .
2. .len
3. parenthesis
4. comparison operators
5. not
6. and
7. or

Statements

1. Display a value of a variable or an expression

Examples:

- ##3## - displays 3.
- ##@field.m_bAddPage## - displays 0 or 1.
- ##@field.m_strLabel## - displays 'Year of Make'.

2. Conditional statement

if <Boolean expression>, elseif <Boolean expression>, else, endif

Examples:

```
##if @field.m_bAddPage##
 ...
##elseif @field.m_strLabel=="ID"##
```

```
 ...
##else##
...
##endif##
```

3. Loop statements

Foreach <array> as <variable> , endfor

The variable is created when loop starts and destroyed with the 'endfor'.

Example:

```
##foreach @TABLE.m_arrFieldObj as @field##
if strField="##@field.m_strName##" then
  Label = "##@field.m_strLabel##"
##endfor##
```

Repeat <number> [variable], endrepeat

Repeat loop body N times. Variable, if specified, ranges from 1 to <number>.

Filter

Allows nodes filtering.

Example. Get a list of fields that require validation:

```
##foreach @TABLE.m_arrFieldObj as @field filter @field.m_strValidateAs order
@field. m_nEditPageOrder##
  ##if @first##
    include("include/validate.php");
  ##endif##
##endfor##
```

Nested loops

Repeat and **Foreach** loops can be nested.

Example:

```
##foreach @BUILDER.Tables as @t##
  ##foreach
@t.arrMasterTables[strMasterTable==@TABLE.strDataSourceTable].arrMasterKeys
as @dk##
  $masterquery.="&masterkey##@index##=".rawurlencode($data["##@dk s##"]);
  ##endfor##
  $showDetailKeys["##@t.strShortTableName##"]=$masterquery;
##endfor##
```

Loop variable @index

The loop variable @index will take on the values 1, 2, ..., N through each of the N iterations of the loop's body.

Pseudo-variables @first and @last

@first takes a value of:

- 1 - during the first loop pass;
- 0 - otherwise.

It is useful when you need to perform some action only once i.e. skip a comma in front of table name:

```
$tables = Array("Table1","Table2","Table3");
##if !@first## , ##endif##
```

In loops, @first terminates execution of the nearest enclosing foreach or repeat statement. Control then passes to the statement that follows the terminated statement, if any:

```
##foreach @TABLE.m_arrFieldObj as @field##
##if @field.m_EditFormatObj. m_strValidateAs && @first##
  include("include/validate.php");
##endif##
##endfor##
```

Order - sort order other than default.

Example. Get a list of fields ordered by nEditPageOrder (field order on the edit page):

```
##foreach @TABLE.m_arrFieldObj as @field order @field. nEditPageOrder##
##if @field.m_EditFormatObj. m_strValidateAs && @first ##
  include("include/validate.php");
##endif##
##endfor##
```

Output modifiers

Modifiers are required to encode quotes, slashes and other "bad" characters that can break template language elements. You can combine several modifiers. Modifiers order is important.

Example:

```
##@field.m_strLabel hs##
```

List of modifiers abbreviations:

- hs - shapehtml will be applied first, shapescript will be applied after that.
- s - replaces " with \" for PHP.

- q - PHP strings wrapped by single quotes.
- h - HTML-encodes string.
- j - replaces ' with \'.
- n - replaces spaces with &nbsp.
- u - URL-encodes string.
- w - adds wrappers around the field name ([field name] or `field name`).
- t - adds wrappers around the table name ([dbo].[table name] or `table name`).
- g - replaces all non alphanumeric characters with underscores.
- p - builds parameter name for UPDATE, INSERT, DELETE (.NET specific).
- c - removes spaces.
- 8 - converts UTF8 string to national language charset.
- o - removes owner/schema from table name.
- d - converts name to CamelCase. Example: "Order details" becomes OrderDetails.
- l - replaces line feeds and carriage returns with spaces.
- a - builds valid PHP variable name from table name. Used in Data Access Layer.
- f - builds valid PHP variable name from field name. Used in Data Access Layer.
- x - builds valid property name (.NET Data Access Layer specific).
- y - builds valid class name (.NET Data Access Layer specific).

Macros and constants

Macros and constants are processed and replaced with the actual code. Macros and constants are defined in the *macros.txt* file.

Constant definition example:

```
##define <name> <value>##
##define FORMAT_DATABASE_IMAGE "Database image"##
##define EDIT_DATE_SIMPLE 0##
```

Macro definition example:

```
##define UseRTE(@field)
(@field.strEditFormat==EDIT_FORMAT_TEXT_AREA &&
@field.m_EditFormatObj.m_bUseRTE)
##
```

Macros and constants are processed in the same way. Therefore, we suggest to follow this naming convention: constant names are written in upper case (e.g. FORMAT_DATABASE_IMAGE), macro names use CamelCase convention (e.g. UseCalendar). Spaces are not allowed in macro or constant names.

Example:

```
##if @field.strViewFormat==FORMAT_DATABASE_IMAGE##
##if UseRTE(@field)##
##foreach Fields as @f##
##Master.strCaption##
```

Additional language elements

1. Specific array element

To access specific array element use *<array>[<condition>]*.

Example:

```
##@TABLE.arrFieldObj[strName==@TABLE.strKeyField].strLabel##
```

This example shows how to access the Label property of a key column field or any other field.

3.2.28 Useful links

PHP documentation	http://www.php.net/manual/en/
PHP string functions	http://www.php.net/manual/en/ref.strings.php
PHP date and time functions	http://us.php.net/manual/en/ref.datetime.php
General SQL functions and tutorial	http://www.webcheatsheet.com/sql/interactive_sql_tutorial/
MySQL documentation	http://dev.mysql.com/doc/refman/5.0/en/index.html
MySQL date and time functions	http://dev.mysql.com/doc/refman/5.0/en/date-and-time-functions.html
MySQL string functions	http://dev.mysql.com/doc/refman/5.0/en/string-functions.html
MS Access documentation	http://www.webcheatsheet.com/sql/access_functions/
Oracle documentation	http://www.oracle.com/pls/db111/homepage
SQL Server documentation	http://technet.microsoft.com/en-us/library/ms130214.aspx
HTML reference	http://htmlhelp.com/reference/html40/

CSS reference	http://htmlhelp.com/reference/css/

3.2.29 Using JOIN SQL queries

Lets say you need to pull data from two or more joined tables to appear on List/View/Edit and other pages. Data should be searchable and editable (except for joined fields that cannot be updated).

In this example we use two tables:

Cars

ID	Make	Model	UserID
1	Acura	NSX-T	1
2	Ford	Crown Victoria	2
3	Volkswagen	Passat	1
4	Toyota	Avalon	2
5	Audi	TT	3

Users

ID	Username
1	Bob
2	Admin
3	Bill
4	Tina

Here is the default SQL query PHPRunner builds for table Cars:

```
select [ID],
[Make],
[Model],
[UserID],
From [Cars]
```

Modify it the following way:

```
select Cars.[ID],
[Make],
[Model],
[UserID],
UserName
From [Cars]
inner join users on cars.userid=users.id
```

If you don't specify what table ID field belongs you see the error message similar to this one:

```
Error message:
```

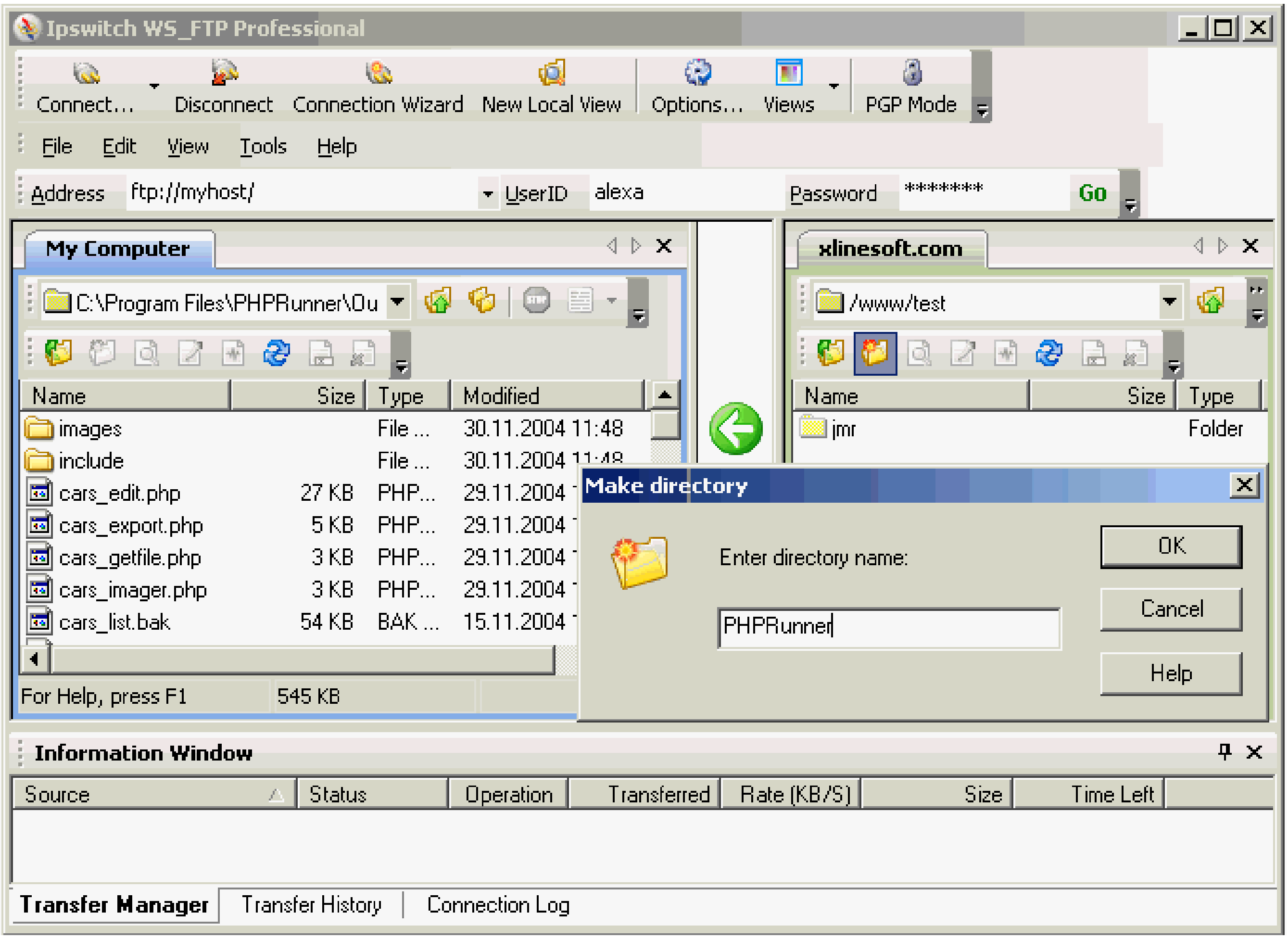
```
[Microsoft][ODBC Microsoft Access Driver] The specified field '[ID]' could
refer to more than one table listed in the FROM clause of your SQL
statement.
```

3.3 Publishing PHP application to the remote Web server

3.3.1 Using FTP client to publish PHP pages to the remote Web server

When you have created a set of PHP pages with PHPRunner, you may use any FTP client to upload them on a remote Web server. We will show you how to do it using popular FTP client WS_FTP Pro as an example.

First of all, create your FTP server connection using WS_FTP Pro Connection Wizard. Follow the directions in the Connection Wizard - type in server name or IP address, username and password. Set connection type to FTP and connect to remote FTP server.



Address ftp://myhost/ UserID alexa Password ********

C:\Program Files\PHPRunner\Ou

Name	Size	Type	Modified
images		File ...	30.11.2004 11:48
include		File ...	30.11.2004 11:48
cars_edit.php	27 KB	PHP...	29.11.2004
cars_export.php	5 KB	PHP...	29.11.2004
cars_getfile.php	3 KB	PHP...	29.11.2004
cars_imager.php	3 KB	PHP...	29.11.2004
cars_list.bak	54 KB	BAK ...	15.11.2004

545 KB

/www/test

Name	Size	Type
jmr		Folder

Enter directory name: PHPRunner

Source	Status	Operation	Transferred	Rate (KB/S)	Size	Time Left

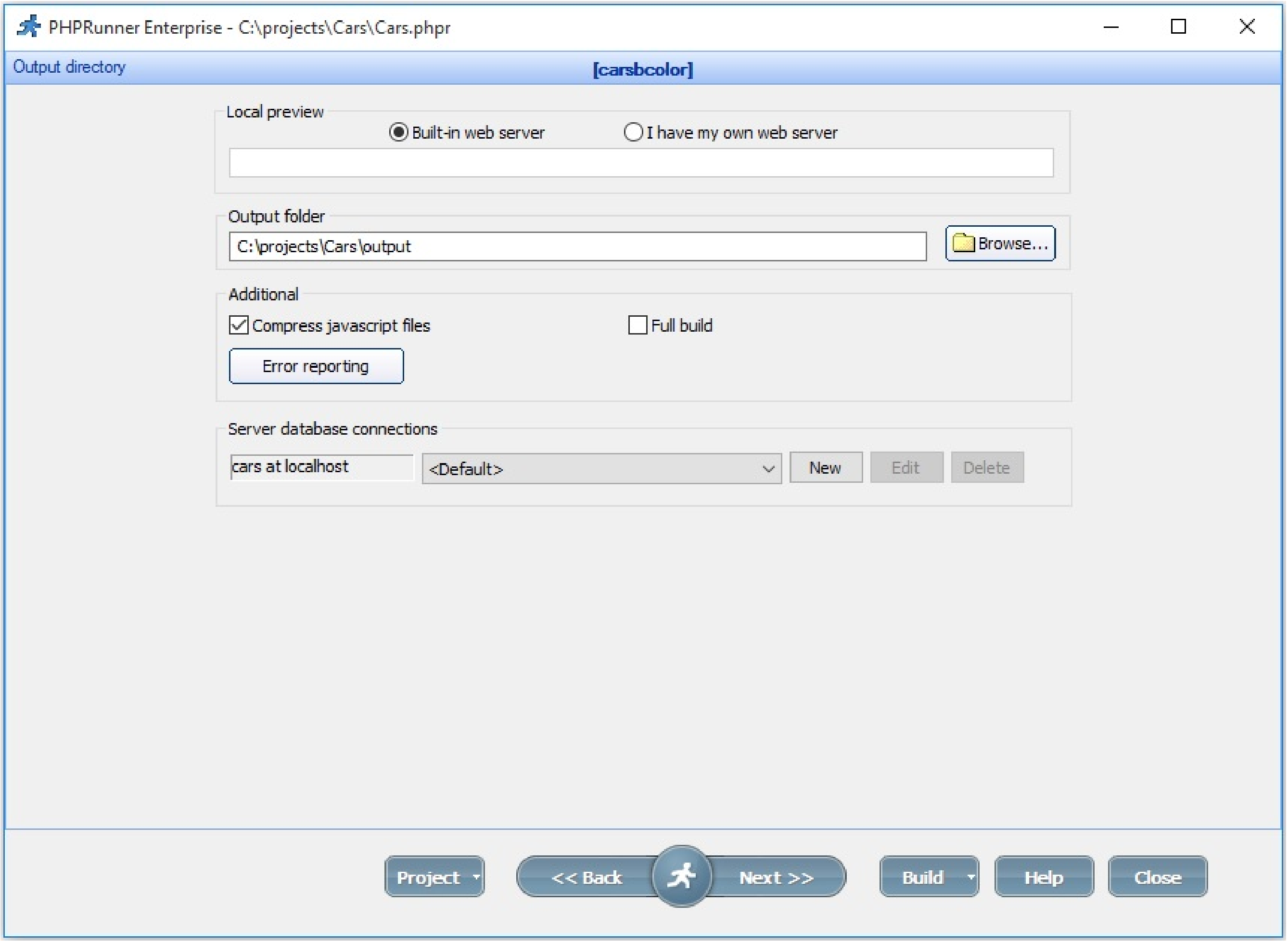
On the left panel (your local machine) proceed to the directory you have chosen as output directory in PHPRunner. On the right panel you may see folders of your remote Web server. Create a new folder on the remote server, name it *PHPrunner* and open it. Select all files on the left panel (menu Edit - Select All) and click the Upload button.

When all necessary files have been copied to the Web server, start browser, type the name of your Web site in the address line and add "/the name of the folder you just made" after it. In our example, it will be: **http://www.xlinesoft.com/test/PHPrunner/**

3.3.2 Using FrontPage to publish PHP pages to the remote Web server

Here is how files generated by PHPRunner can be published on the Web using Frontpage:

- Run PHPRunner. Point it to your database and create your PHP pages. Lets assume that your output folder is *D:\Projects\Project1\output.*



PHPRunner Enterprise - C:\projects\Cars\Cars.phpr

Output directory

[carsbcolor]

Local preview

● Built-in web server ○ I have my own web server

Output folder

C:\projects\Cars\output

Additional

☑ Compress javascript files □ Full build

Server database connections

cars at localhost <Default>

- Start FrontPage. Select **File** -> **New** -> **Web**.

- Select **Import Web Wizard**. Enter your Web site address in the address box (i.e. *http://www.yourservername.com/PHPRunner*) and click **Ok**.

- On the **Import Web Wizard** dialog, select PHPRunner output directory (*D:\Projects\Project1\output*) as a source. Include subfolders and click **Next**.

- Exclude files that you don't need and click **Next**.

- Click **Finish**. Now your Web site is up and running at *http://www.yourservername.com/PHPrunner.*

3.4 Demo Account

3.4.1 What is the Demo Account?

Demo Account is a free service provided by Xlinesoft.com to PHPRunner customers. By using Demo Account you agree to the following Terms and Conditions.

Demo Account allows you quickly put PHP application generated by PHPRunner on our demo web server for testing purposes.

You may find this feature useful if:

- you don't have a web server on your local machine to test PHP pages

- you don't have a webhosting account yet

- you need to show generated application to your boss, friends or to support staff

To open and use Demo Account proceed to the **Finished** tab in PHPRunner and click **Demo Account** button. To create an account enter your email and password. You will need this info to manage your account online. After account is created use Upload button to transfer PHP application to the web server. After successfull upload this application will open in browser.

Demo Account transfers generated pages and your database to the server. Currently supported databases are MS Access, SQL Server, MySQL.

To manage your account online proceed to http://demo.asprunner.net/Account/AccountView.aspx. You can browse uploaded projects, delete projects etc.

Demo account limitations:

- Since this account is designed for demo purposes only first 1000 data records in each tables of SQL Server/MySQL databases will be transferred to the server. MS Access databases are transferred to the server wholly.

- Maximum size of the archived project is 50Mb.

If you are interested in opening a full featured webhosting account that supports one click publishing from PHPRunner, visit http://inspirunner.com for more details.

3.4.2 Terms and Conditions

Xlinesoft.com Demo Account Terms and Conditions

Agreement between User and Xlinesoft.com

A violation of any of the below Terms and Conditions will result in the termination of your account, with or without warning.

We provides free demo account for our users. Xlinesoft.com reserves the right to cancel any account for any reason or no reason at all. Xlinesoft.com provides this service to any user that abides to our terms and conditions. Failure to abide to these rules will result in account termination. Furthermore, any one who conducts illegal activities may be prosecuted and personal information disclosed to the appropriate authority. Xlinesoft.com reserves the right to change the terms and conditions at any time and it is the member’s responsibility to check for any updates of these terms. You are entirely liable for all activities conducted through your account.

Your use of the Xlinesoft.com Demo Account is conditional upon your acceptance without modification of the terms, conditions and notices contained herein. Your use of our Services constitutes your agreement to all such terms, conditions and notices.

The following rules apply while using Xlinesoft.com Demo Account:

1. Sign Up - Basic Terms

1.1 Upon after creating a Demo Account through Xlinesoft.com software you will become a user of Xlinesoft.com Demo Account.

1.2 Xlinesoft.com does not issue credits for outages incurred through service disablement resulting from Terms and Conditions violations.

2. Modification of Terms of Service

2.1 We reserve the right to change the terms and conditions of this agreement at any time without prior notice. Such changes will be posted on our web site at www.xlinesoft.com. You agree to review any changes to this agreement and, if such changes are not acceptable to you, immediately terminate your use of the Xlinesoft.com Demo Account. If you continue to use the Xlinesoft.com service after the effective date of such changes, such use will constitute acceptance of the changes.

3. Modifications to Xlinesoft.com Demo Account

3.1 By accepting this Agreement, you hereby acknowledge and agree that, in our sole discretion, we may modify or discontinue, temporarily or permanently, any aspect of the Xlinesoft.com Demo Account at any time with or without prior notice, including, without limitation, modification or discontinuance of advertising, content and applications appearing as part of the Xlinesoft.com Demo Account. You agree that we will not be liable to you or to any third party for any modification, suspension or discontinuance of the Xlinesoft.com Demo Account.

4. Privacy

4.1 Xlinesoft.com adheres to a Privacy Statement and will not release any confidential information about you unless required by law or regulatory authority or while assisting an investigation concerning fraud.

5. No Unlawful or Prohibited Use, Spam and Termination

5.1 Xlinesoft.com Demo Account may be used for lawful purposes only. As a condition of your use of the Xlinesoft.com Demo Account, you agree that you will not use our Website for any purpose that is unlawful, illegal or prohibited by these Terms of Services, or our other terms, conditions or notices. You agree that you will not modify, copy, distribute, transmit, reproduce, publish, license, transfer, store, sell or create derivative works from, any data, software, Services or products obtained through our Website. This includes, but is not limited to: copyrighted material; trademarks; trade secrets or other intellectual property rights used without proper authorization; material that is obscene, defamatory, constitutes an illegal threat, or violates export control laws. Examples of unacceptable content or links include: pirated software, hacker programs or archives, Warez sites, MP3, and IRC bots. Such misuse can result in the cancellation of your entire account and the blacklisting of your domains without notice. You or Xlinesoft.com may terminate your Demo Account at any time for any reason.

3.5 Web reports

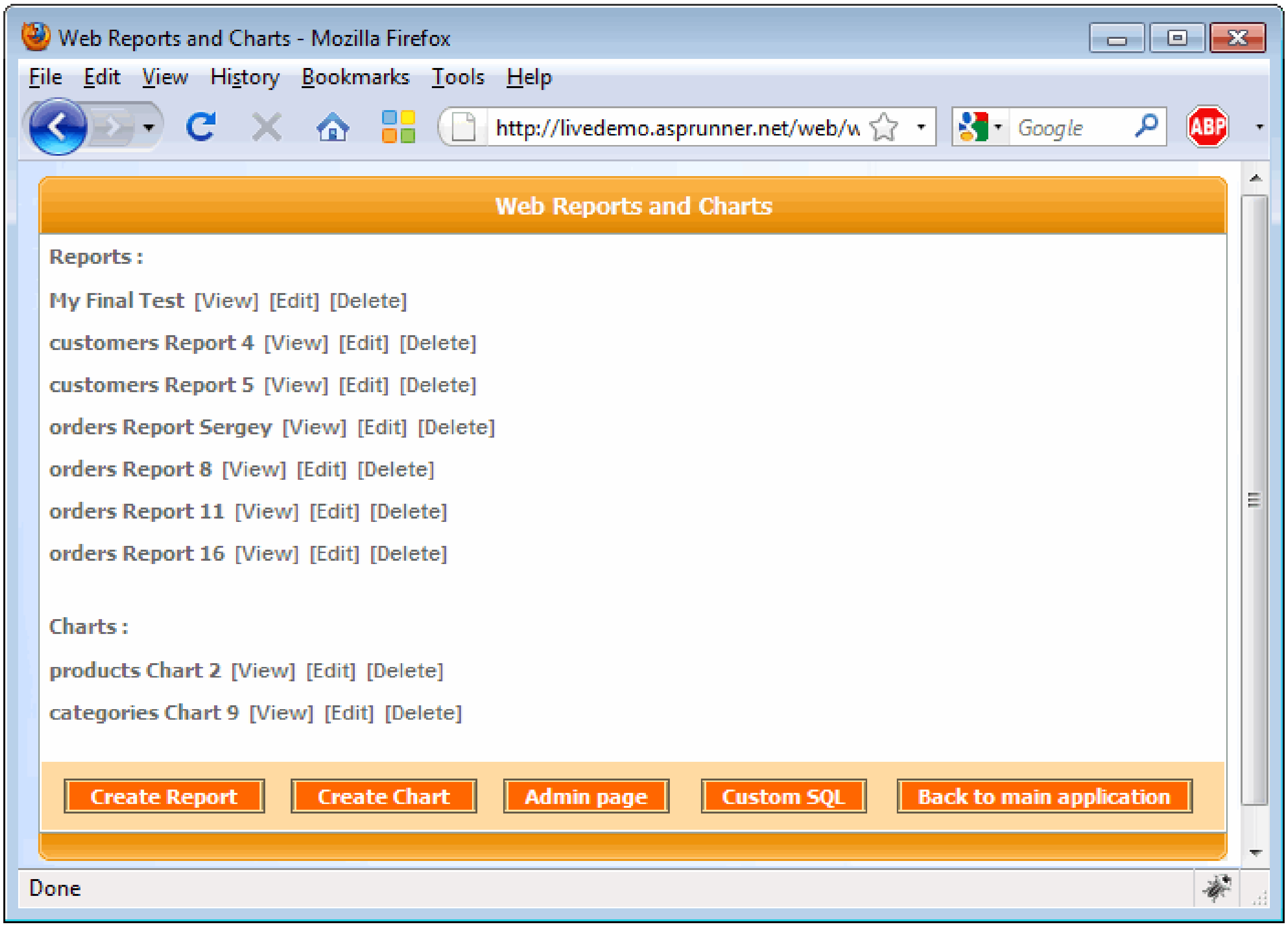
3.5.1 Online report/chart builder

Enterprise Edition of PHPRunner includes the online report/chart builder that allows you to view, create, edit and delete reports and charts in the web browser.

To enable the online report/chart builder, select the **Web Reports** check box on the **Miscellaneous** page and build the project. Then launch online report/chart builder by running *webreport.php* in the browser (e.g. **http://www.yourwebsite.com/project1/webreport.php**). Online report/chart builder uses PHPRunner project security model. So if you use a login page, you need to logon to your application first.

Working with online report/chart builder

Start page presents you with the list of charts and reports. If your project uses permissions, charts and reports are divided into shared and private ones.



Web Reports and Charts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/w
Google
ABP
Create Report
Create Chart
Admin page
Custom SQL
Back to main application
Done

Web Reports and Charts

Reports :

My Final Test [View] [Edit] [Delete]

customers Report 4 [View] [Edit] [Delete]

customers Report 5 [View] [Edit] [Delete]

orders Report Sergey [View] [Edit] [Delete]

orders Report 8 [View] [Edit] [Delete]

orders Report 11 [View] [Edit] [Delete]

orders Report 16 [View] [Edit] [Delete]

Charts :

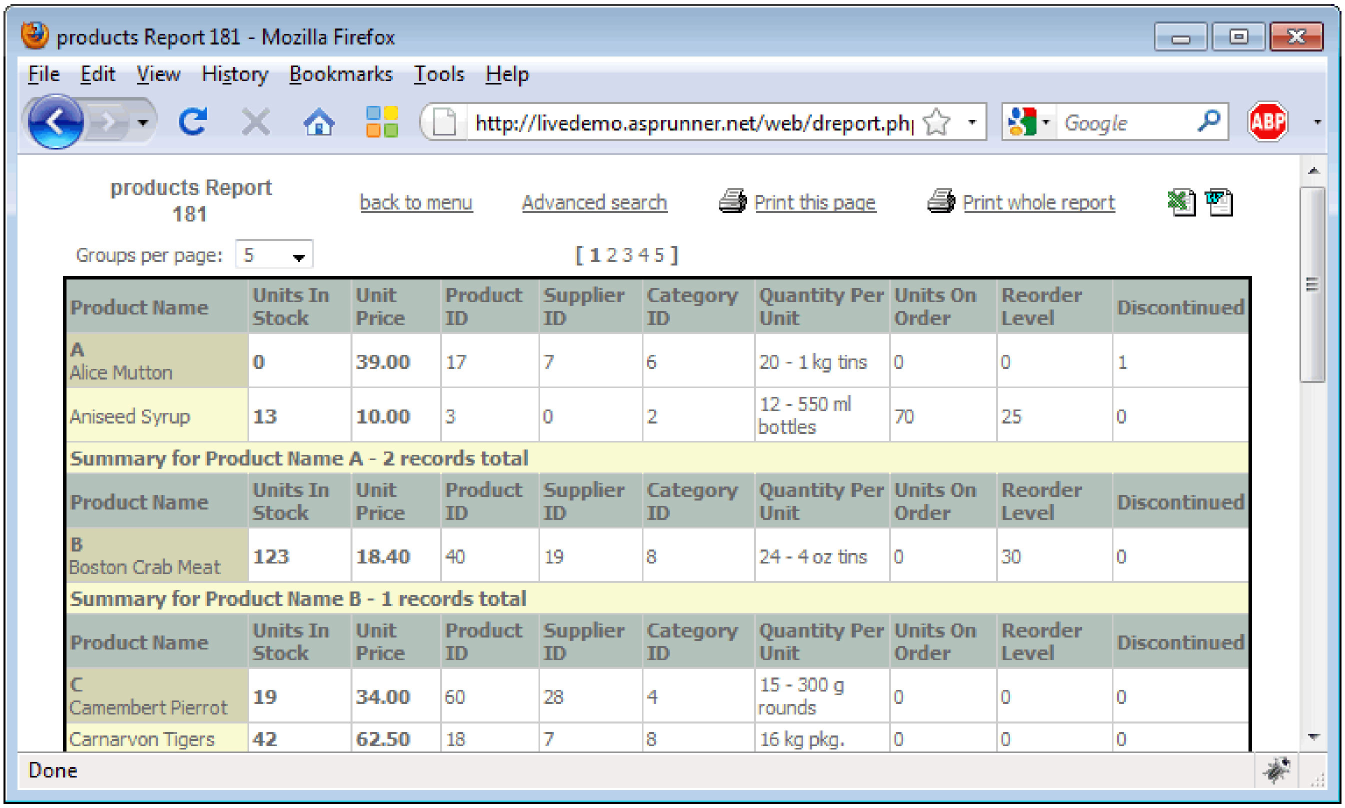
products Chart 2 [View] [Edit] [Delete]

categories Chart 9 [View] [Edit] [Delete]

You can view, edit or delete charts and reports if the permissions allow you to do so. Also you can create new chart or new report.

While viewing a report or chart you can narrow the finding using **Advanced search**. While viewing a report you can also print the current page or the whole report, open a report as Microsoft Word or Microsoft Excel document.

Example of Web report:


products Report 181 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/dreport.ph
Google
back to menu
Advanced search
Print this page
Print whole report
Done

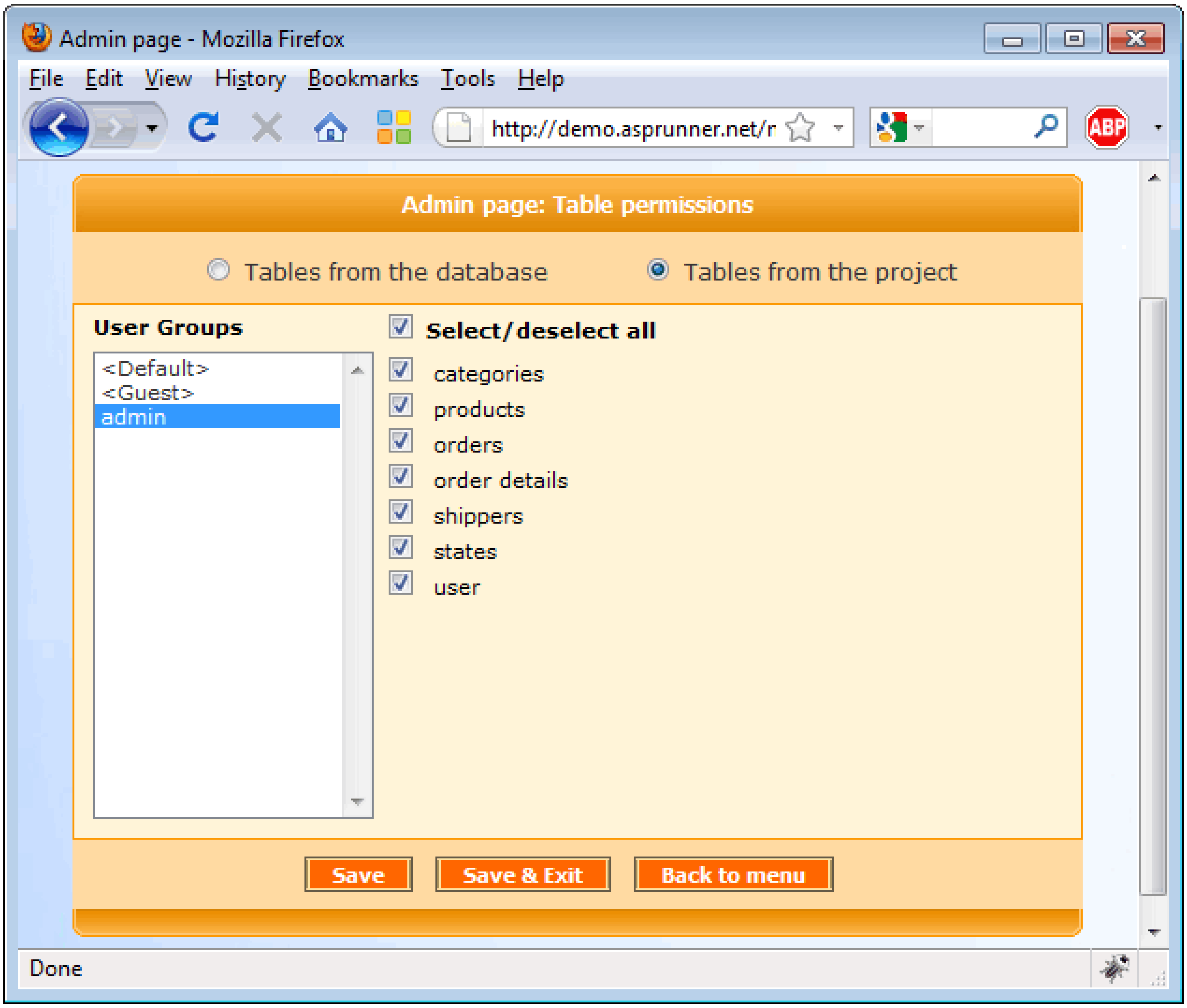
products Report 181

Groups per page: 5

[1 2 3 4 5]

Product Name	Units In Stock	Unit Price	Product ID	Supplier ID	Category ID	Quantity Per Unit	Units On Order	Reorder Level	Discontinued
A Alice Mutton	**0**	**39.00**	17	7	6	20 - 1 kg tins	0	0	1
Aniseed Syrup	**13**	**10.00**	3	0	2	12 - 550 ml bottles	70	25	0
Summary for Product Name A - 2 records total									
Product Name	**Units In Stock**	**Unit Price**	**Product ID**	**Supplier ID**	**Category ID**	**Quantity Per Unit**	**Units On Order**	**Reorder Level**	**Discontinued**
B Boston Crab Meat	**123**	**18.40**	40	19	8	24 - 4 oz tins	0	30	0
Summary for Product Name B - 1 records total									
Product Name	**Units In Stock**	**Unit Price**	**Product ID**	**Supplier ID**	**Category ID**	**Quantity Per Unit**	**Units On Order**	**Reorder Level**	**Discontinued**
C Camembert Pierrot	**19**	**34.00**	60	28	4	15 - 300 g rounds	0	0	0
Carnarvon Tigers	**42**	**62.50**	18	7	8	16 kg pkg.	0	0	0

On the **Admin page** you can select tables to be available for creating web reports and charts.



Admin page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/r
Save
Save & Exit
Back to menu
Done

Admin page: Table permissions

○ Tables from the database ● Tables from the project

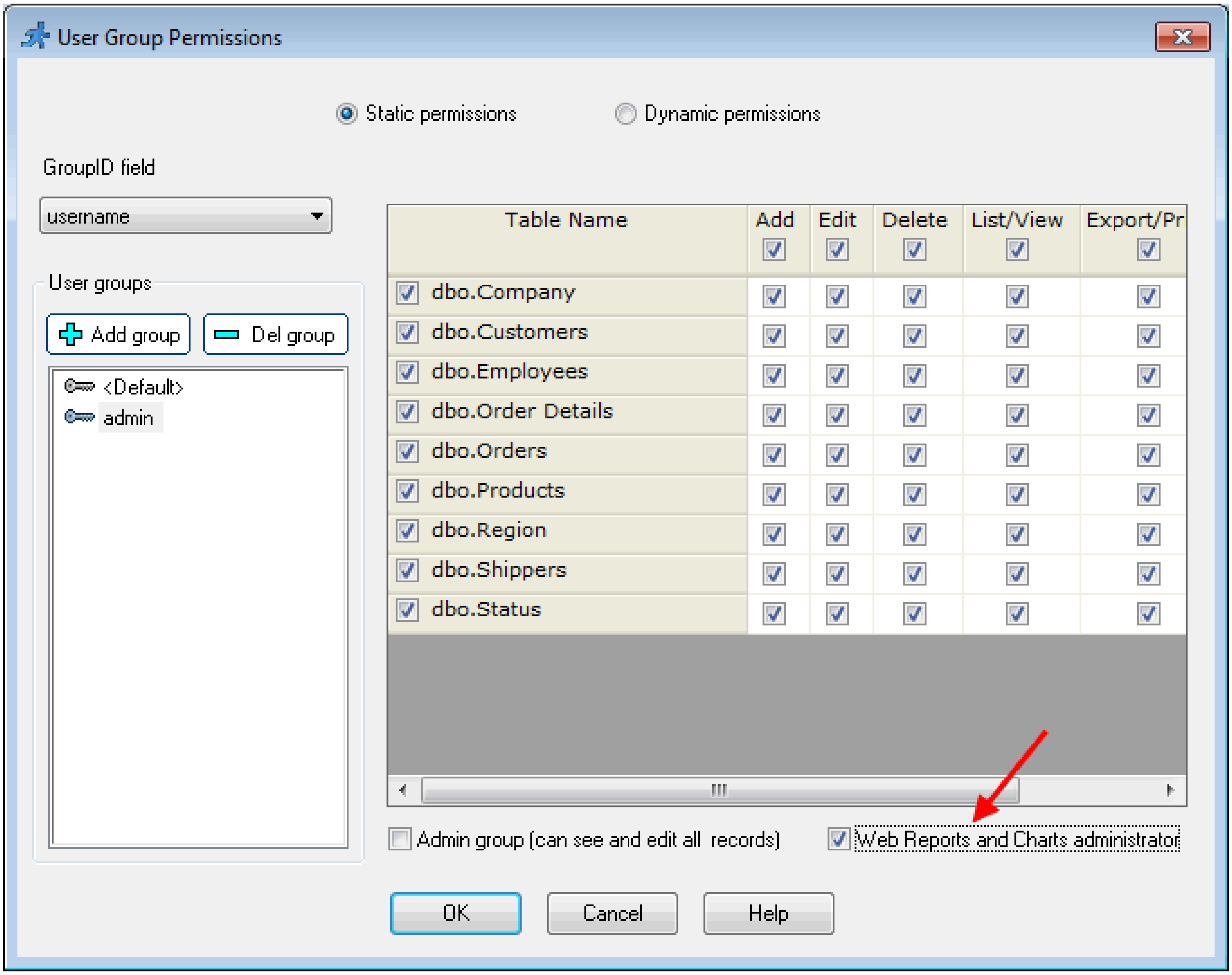
User Groups

- <Default>
- <Guest>
- admin

☑ **Select/deselect all**

- ☑ categories
- ☑ products
- ☑ orders
- ☑ order details
- ☑ shippers
- ☑ states
- ☑ user

The **Admin page** is available only to the web reports and charts administrator that can be assigned on the **Security** -> **Permissions** page. To specify the password to access the Web Reports and Charts admin area, click the **Administrator** button on the **Miscellaneous** page.



User Group Permissions

● Static permissions ○ Dynamic permissions

GroupID field: username

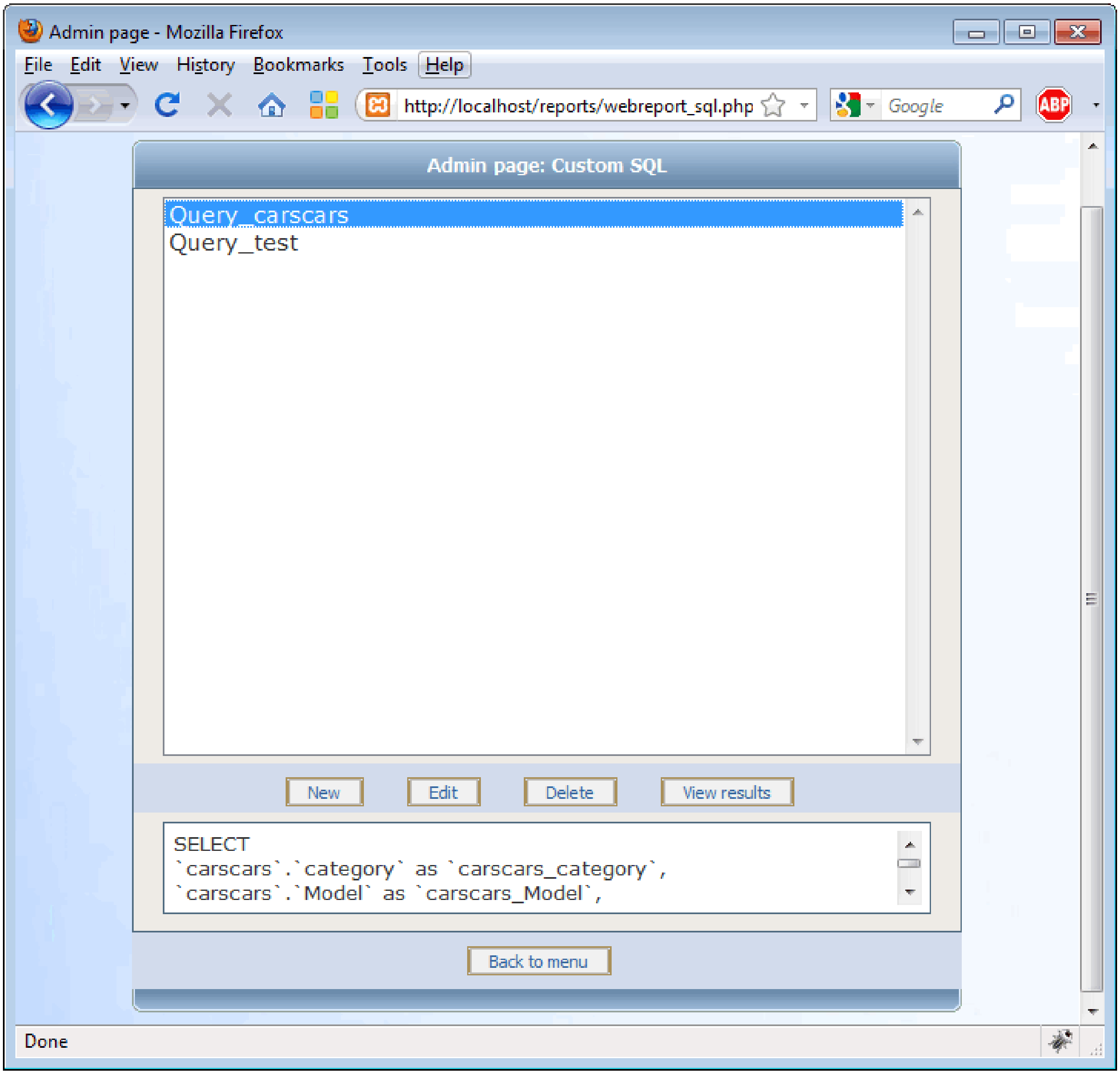
User groups

- <Default>
- admin

Table Name	Add ☑	Edit ☑	Delete ☑	List/View ☑	Export/Pr ☑
☑ dbo.Company	☑	☑	☑	☑	☑
☑ dbo.Customers	☑	☑	☑	☑	☑
☑ dbo.Employees	☑	☑	☑	☑	☑
☑ dbo.Order Details	☑	☑	☑	☑	☑
☑ dbo.Orders	☑	☑	☑	☑	☑
☑ dbo.Products	☑	☑	☑	☑	☑
☑ dbo.Region	☑	☑	☑	☑	☑
☑ dbo.Shippers	☑	☑	☑	☑	☑
☑ dbo.Status	☑	☑	☑	☑	☑

□ Admin group (can see and edit all records) ☑ Web Reports and Charts administrator

A user with admin permissions can create/edit/delete custom SQL queries using the **Custom SQL** button. The custom SQL queries later may be used by other users as a datasource for reports and charts. For more information, see Custom SQL.


Admin page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/reports/webreport_sql.php
Google
Admin page: Custom SQL
Query_carscars
Query_test
New
Edit
Delete
View results
Back to menu
Done

```
SELECT
`carscars`.`category` as `carscars_category`,
`carscars`.`Model` as `carscars_Model`,
```

3.5.2 Creating web report

To create new report, click the **Create Report** button on the start page and follow the steps to define report settings.

The **Back** and **Next** buttons allow you to jump to previous and next page correspondingly. Use **Jump to** button to jump to any other page. The **Save** button saves the report and moves you to the web reports menu (start page). Use the **SQL Query** button to view resulting SQL query and query results. The **Preview** button allow you to see how you report will look.

Below you can find the description of the report creation steps:

Tables

On this page you can choose a table, view or SQL query as a datasource for your report. Note that tables that are not added to the project from a database are not available for selection.

Tables from the database

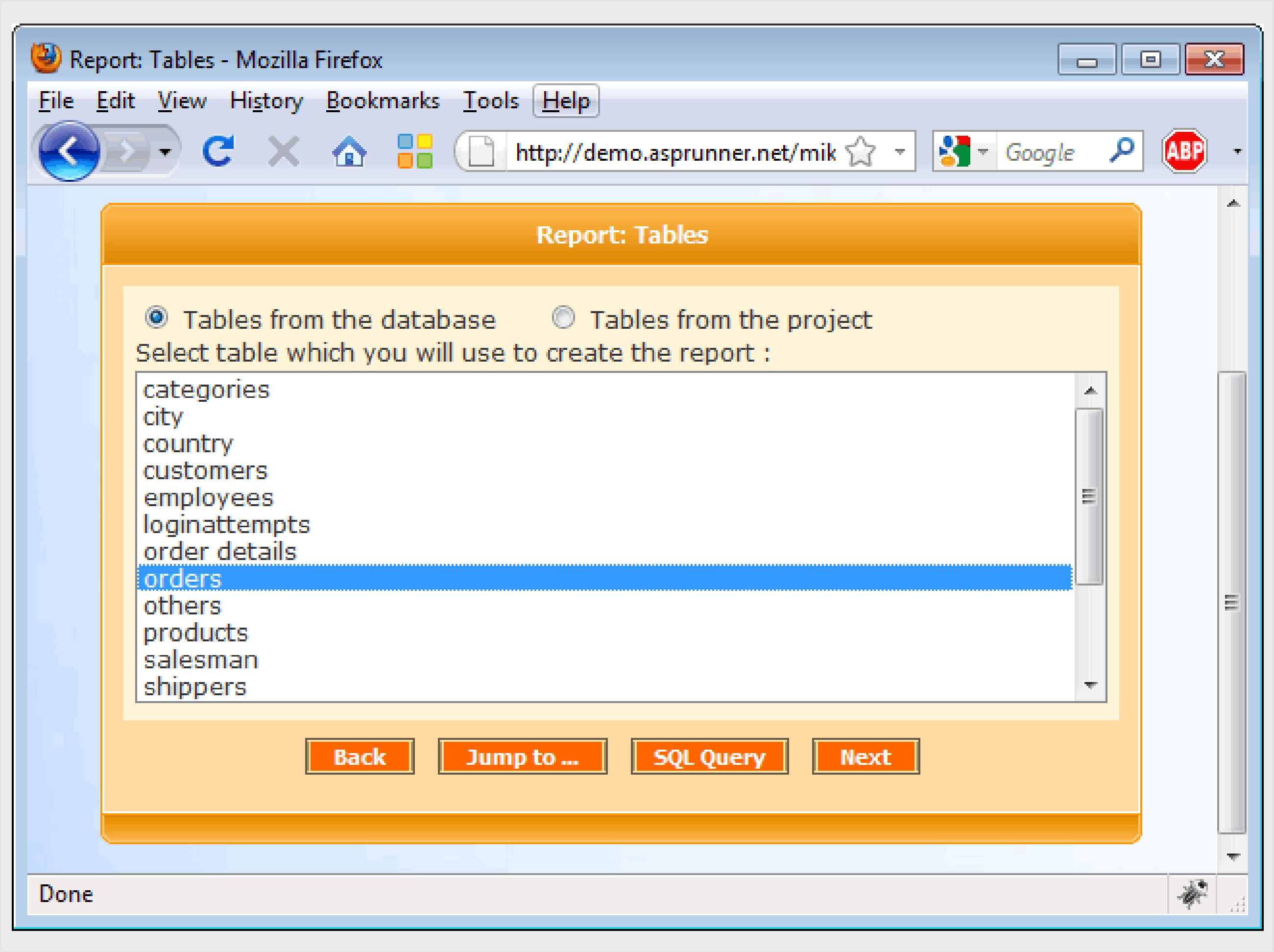
- All tables added to the project are available for selection. User tables (custom views) are not available.
- When viewing a report, all user permissions (static and dynamic permissions, advanced security option) and "view/edit" field settings do not work.
- While creating a report you will be able to create table relations (SQL joins) to query data from two or more tables and add additional search conditions using WHERE clause.

Tables from the project

- All tables added to the project and custom views created in PHPRunner are available for selection.
- When viewing a report, user permissions and "view/edit" field settings work as usual.
- While creating a report you will not be able to create table relations (SQL joins) to query data from two or more tables and add additional search conditions using WHERE clause. The SQL queries defined in PHPRunner will be used to query data for report.
- Tables for selection are displayed as *Caption (table title)*, e.g. Cars (carscars).

SQL queries

On the **SQL queries** tab a user with admin permissions can create new custom SQL query using the **New query** button and edit the existing SQL queries using the **SQL Query** button. For more information about custom SQL queries, see Custom SQL.


Report: Tables - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mik
Google
Done

Report: Tables

● Tables from the database ○ Tables from the project

Select table which you will use to create the report :

categories
city
country
customers
employees
loginattempts
order details
orders
others
products
salesman
shippers

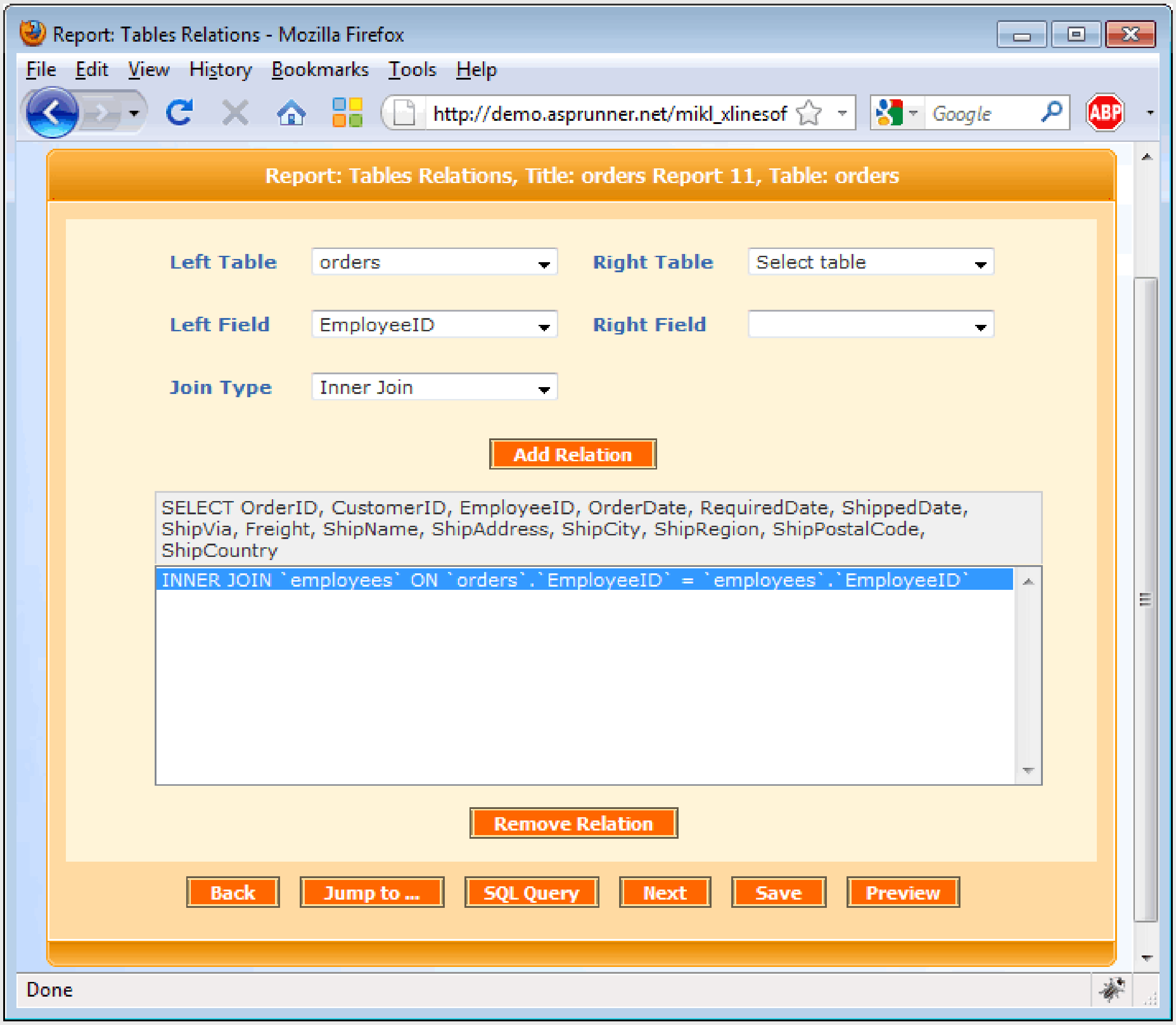
Back
Jump to ...
SQL Query
Next

Table Relations

Note: this page is available if you selected a table from the database as report data source.

On this page you can create table relations (SQL joins) to query data from two or more tables, based on a relationship between certain fields in these tables. You can add Inner Join, Left Join, Right Join and Full Outer Join.

To add an SQL join, choose tables and fields to be joined and click **Add Relation**. The JOIN clause will be added below the SELECT clause. Note that you can add several table relations.

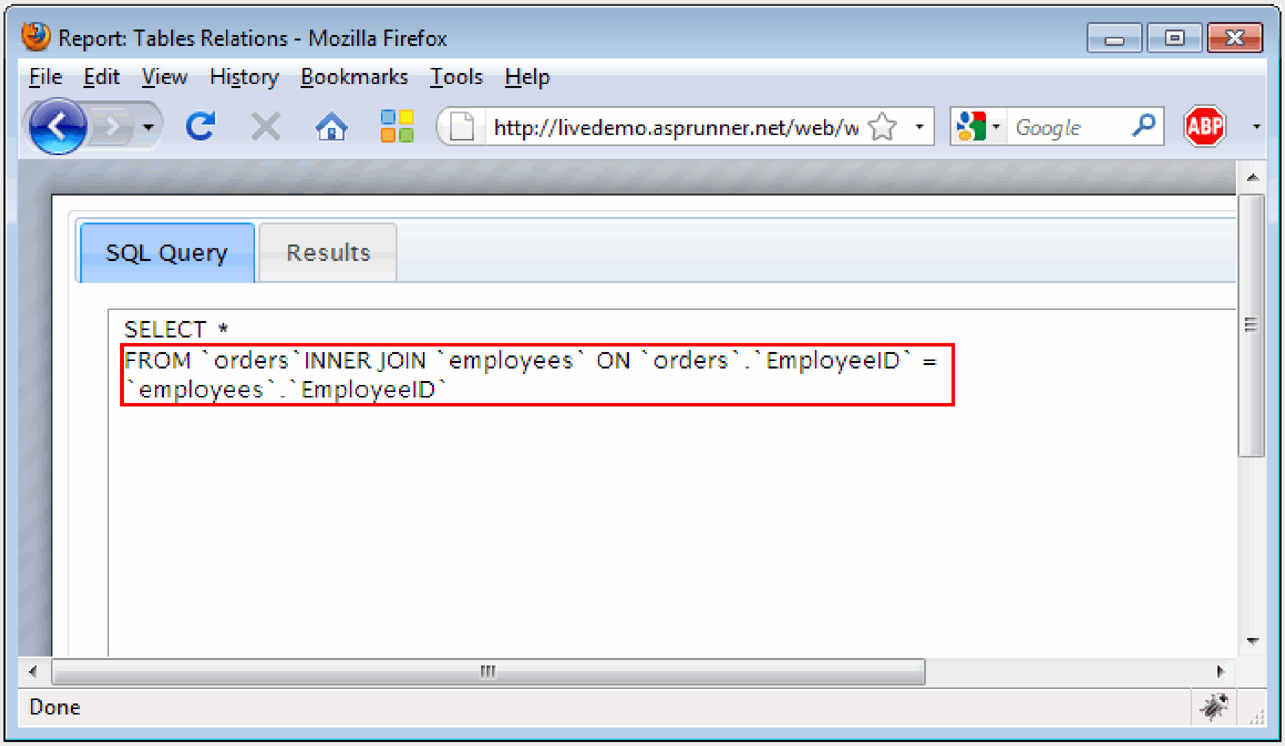

Report: Tables Relations - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesof
Google
Report: Tables Relations, Title: orders Report 11, Table: orders
Add Relation
Remove Relation
Back
Jump to ...
SQL Query
Next
Save
Preview
Done

Left Table: orders
Right Table: Select table
Left Field: EmployeeID
Right Field:
Join Type: Inner Join

```
SELECT OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate,
ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode,
ShipCountry
```

```
INNER JOIN `employees` ON `orders`.`EmployeeID` = `employees`.`EmployeeID`
```

Use **SQL Query** button to view resulting SQL query and query results.


Report: Tables Relations - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/w
Google
SQL Query
Results
Done

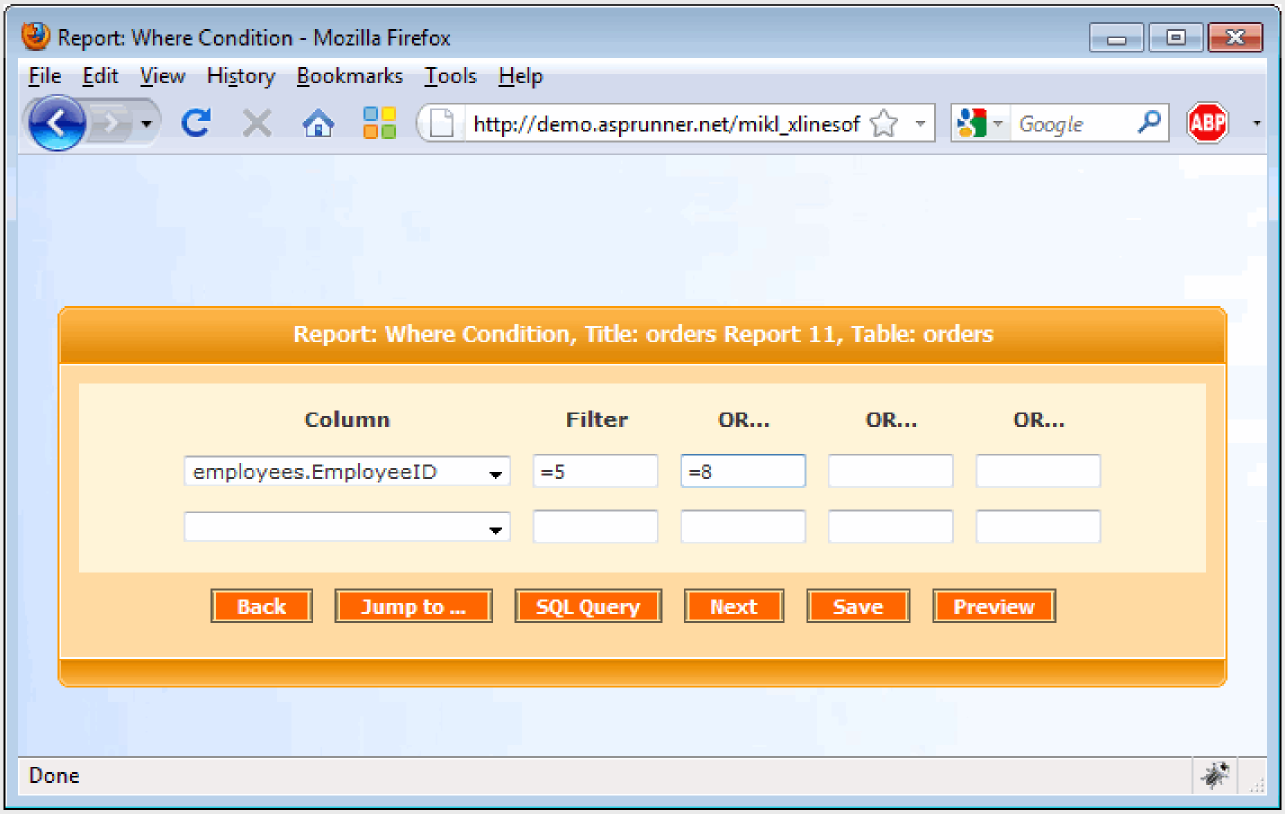
```
SELECT *
FROM `orders`INNER JOIN `employees` ON `orders`.`EmployeeID` =
`employees`.`EmployeeID`
```

Use the **Remove Relation** button to delete the selected table relation.

Where Condition

Note: this page is available if you selected a table from the database as report data source.

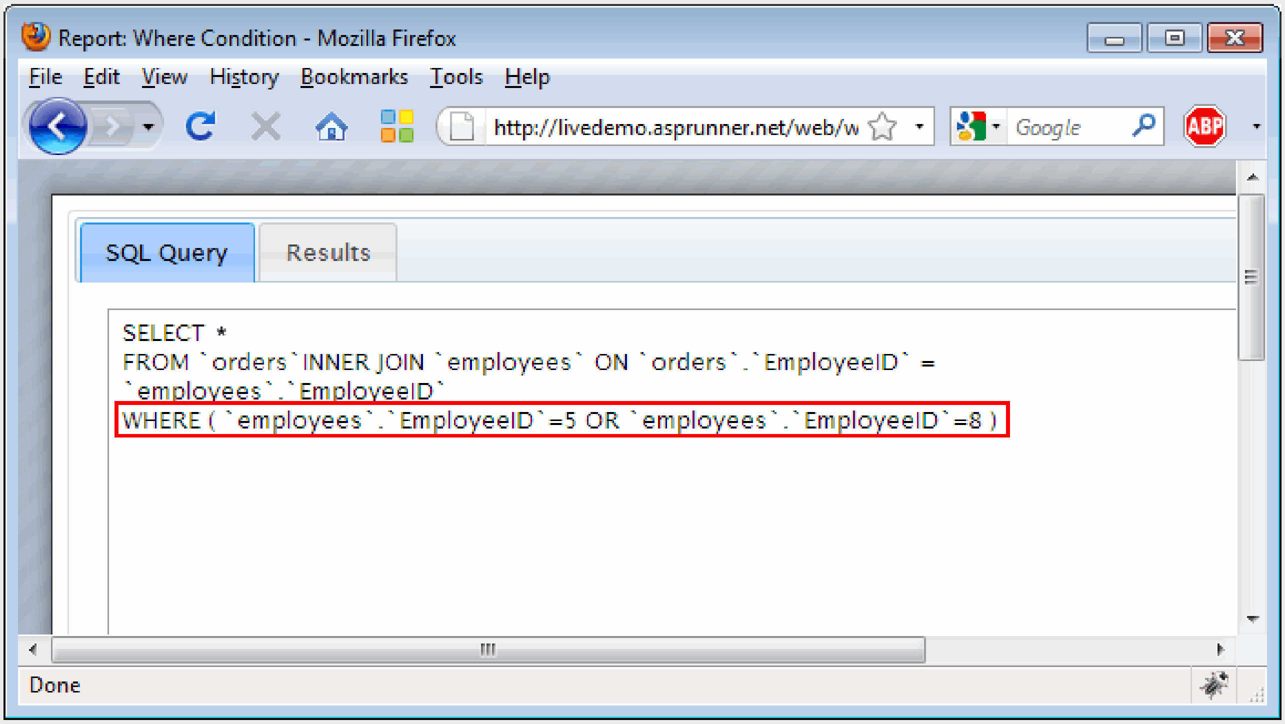
On this page you can add additional search conditions using WHERE clause. Select a field and type search criteria in the text boxes on the right. Search criterion should be added as **<operator><value>**. E.g. ='USA'; =2009; <>'red'; >10.


Report: Where Condition - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesof
Google
Back
Jump to ...
SQL Query
Next
Save
Preview
Done

Report: Where Condition, Title: orders Report 11, Table: orders

Column	Filter	OR...	OR...	OR...
employees.EmployeeID	=5	=8		

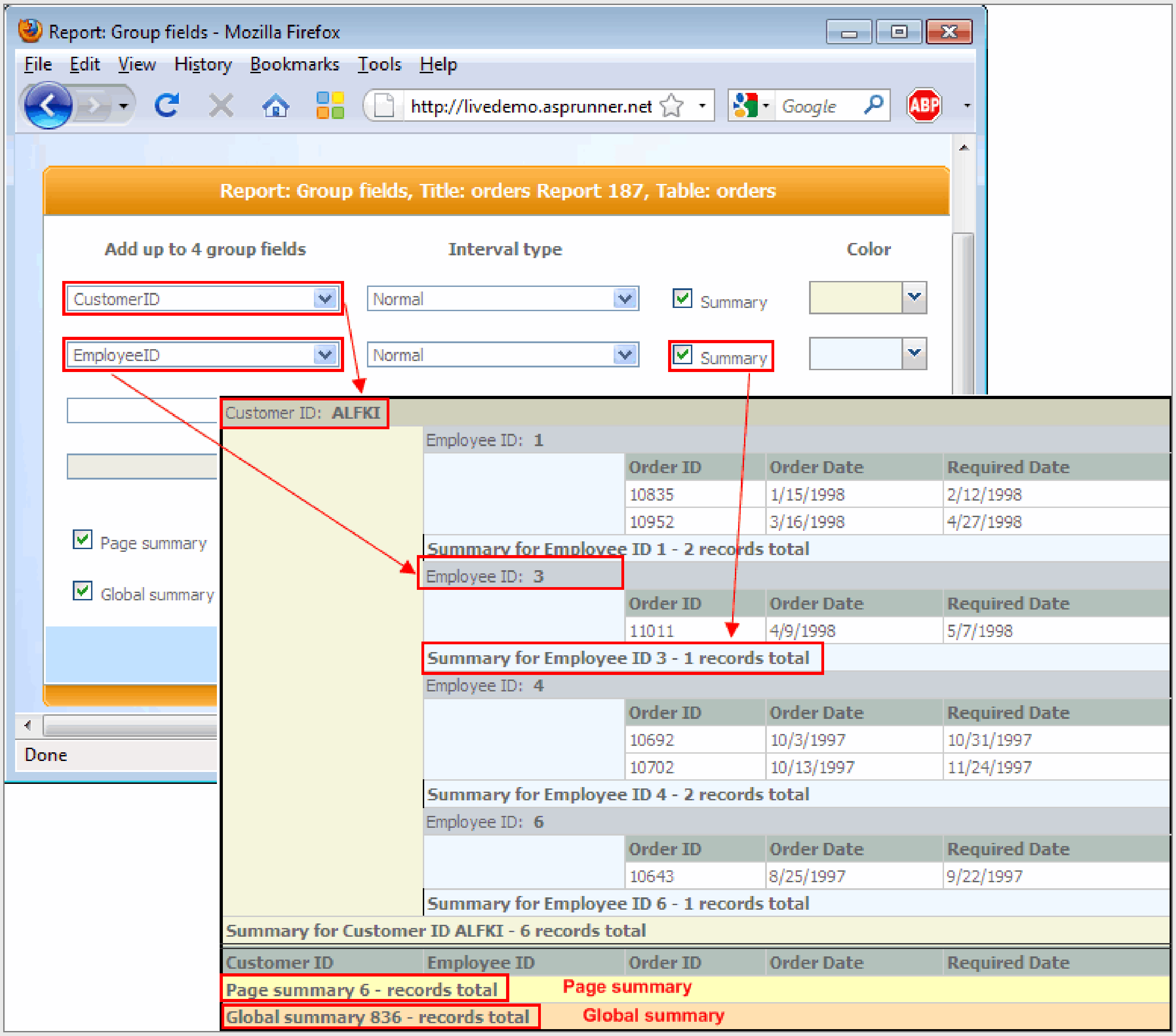
Use the **SQL Query** button to view resulting SQL query and query results.


Report: Where Condition - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/w
Google
SQL Query
Results
Done

```
SELECT *
FROM `orders`INNER JOIN `employees` ON `orders`.`EmployeeID` =
`employees`.`EmployeeID`
WHERE ( `employees`.`EmployeeID`=5 OR `employees`.`EmployeeID`=8 )
```

Group fields

On this page you can add group fields to group the results by one or more columns. The following picture explains how each option works.



Report: Group fields - Mozilla Firefox

File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net
Google
Done

Report: Group fields, Title: orders Report 187, Table: orders

Add up to 4 group fields	Interval type		Color
CustomerID	Normal	☑ Summary	
EmployeeID	Normal	☑ Summary	

☑ Page summary

☑ Global summary

Customer ID: ALFKI

Employee ID: 1

Order ID	Order Date	Required Date
10835	1/15/1998	2/12/1998
10952	3/16/1998	4/27/1998

Summary for Employee ID 1 - 2 records total

Employee ID: 3

Order ID	Order Date	Required Date
11011	4/9/1998	5/7/1998

Summary for Employee ID 3 - 1 records total

Employee ID: 4

Order ID	Order Date	Required Date
10692	10/3/1997	10/31/1997
10702	10/13/1997	11/24/1997

Summary for Employee ID 4 - 2 records total

Employee ID: 6

Order ID	Order Date	Required Date
10643	8/25/1997	9/22/1997

Summary for Employee ID 6 - 1 records total

Summary for Customer ID ALFKI - 6 records total

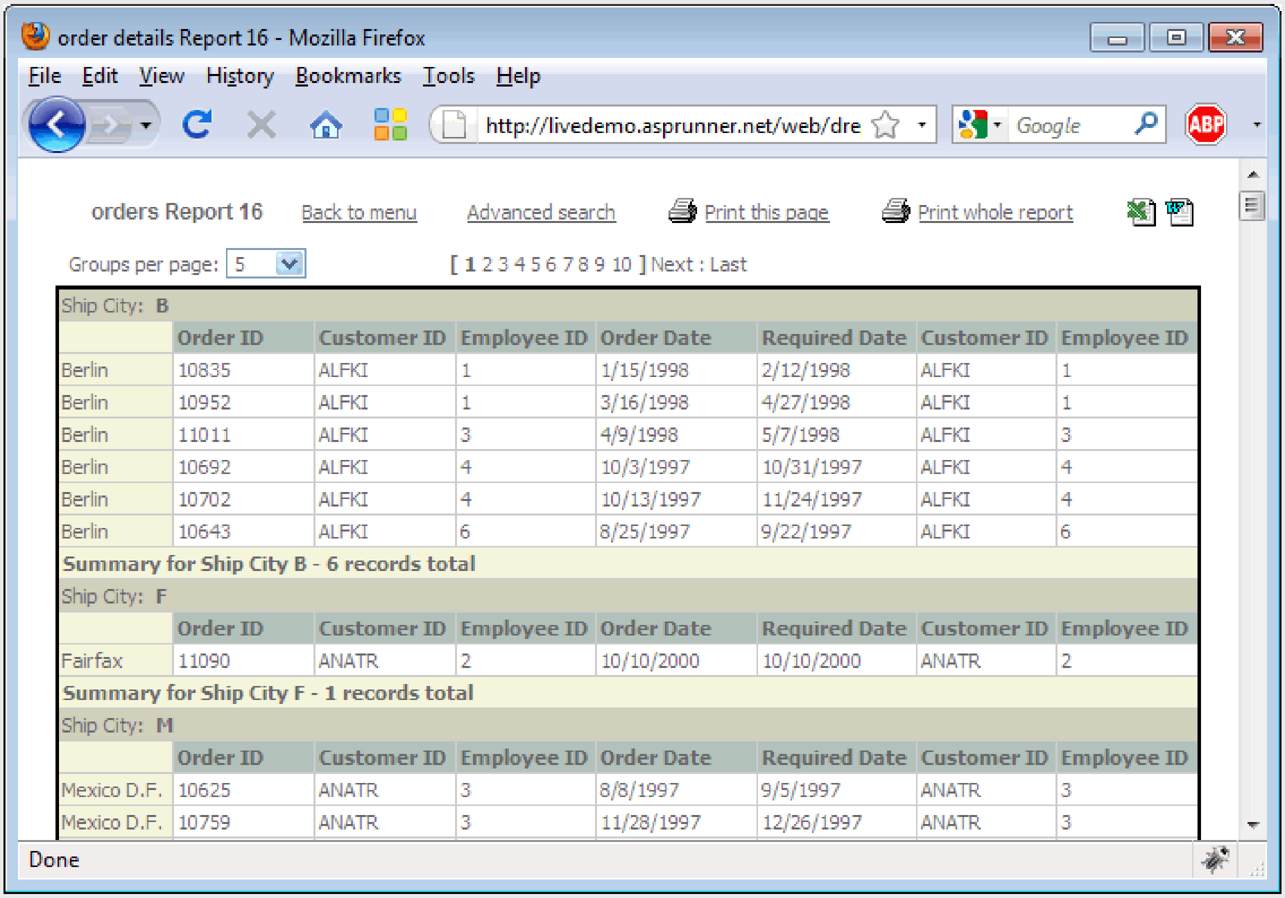
Customer ID	Employee ID	Order ID	Order Date	Required Date

Page summary 6 - records total

Global summary 836 - records total

If you clear the **Details and summary** check box, only summary will be shown in the report.

Besides standard intervals (new group starts when group field value changes) you can use other interval types. Available interval types are different for each data type. Here is the example of text group field using first letter as an interval.

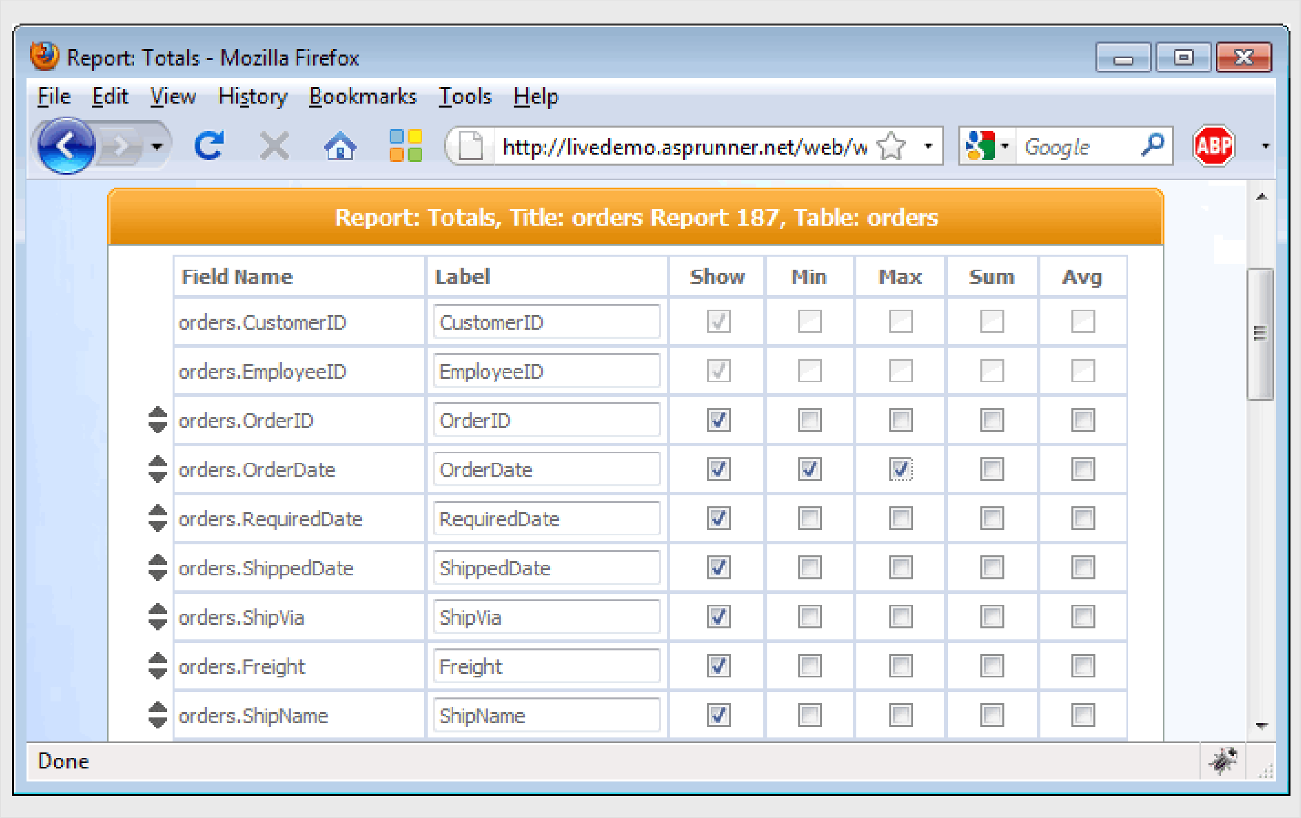

order details Report 16 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/dre
Google
orders Report 16
Back to menu
Advanced search
Print this page
Print whole report
Groups per page: 5
[1 2 3 4 5 6 7 8 9 10] Next : Last
Done

Ship City: B							
	Order ID	Customer ID	Employee ID	Order Date	Required Date	Customer ID	Employee ID
Berlin	10835	ALFKI	1	1/15/1998	2/12/1998	ALFKI	1
Berlin	10952	ALFKI	1	3/16/1998	4/27/1998	ALFKI	1
Berlin	11011	ALFKI	3	4/9/1998	5/7/1998	ALFKI	3
Berlin	10692	ALFKI	4	10/3/1997	10/31/1997	ALFKI	4
Berlin	10702	ALFKI	4	10/13/1997	11/24/1997	ALFKI	4
Berlin	10643	ALFKI	6	8/25/1997	9/22/1997	ALFKI	6
Summary for Ship City B - 6 records total							
Ship City: F							
	Order ID	Customer ID	Employee ID	Order Date	Required Date	Customer ID	Employee ID
Fairfax	11090	ANATR	2	10/10/2000	10/10/2000	ANATR	2
Summary for Ship City F - 1 records total							
Ship City: M							
	Order ID	Customer ID	Employee ID	Order Date	Required Date	Customer ID	Employee ID
Mexico D.F.	10625	ANATR	3	8/8/1997	9/5/1997	ANATR	3
Mexico D.F.	10759	ANATR	3	11/28/1997	12/26/1997	ANATR	3

Use the **SQL Query** button to view resulting SQL query and query results.

Totals

On this page you can choose what fields to display in the report and specify field labels. Also you can apply aggregate functions like MIN, MAX, SUM and AVERAGE. The results of these calculations will be displayed after each group and at the end of page/report. Note than you can't modify settings for group fields. Use arrows on the left of the field names to change the fields order.


Report: Totals - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/w
Google
ABP
Done

Report: Totals, Title: orders Report 187, Table: orders

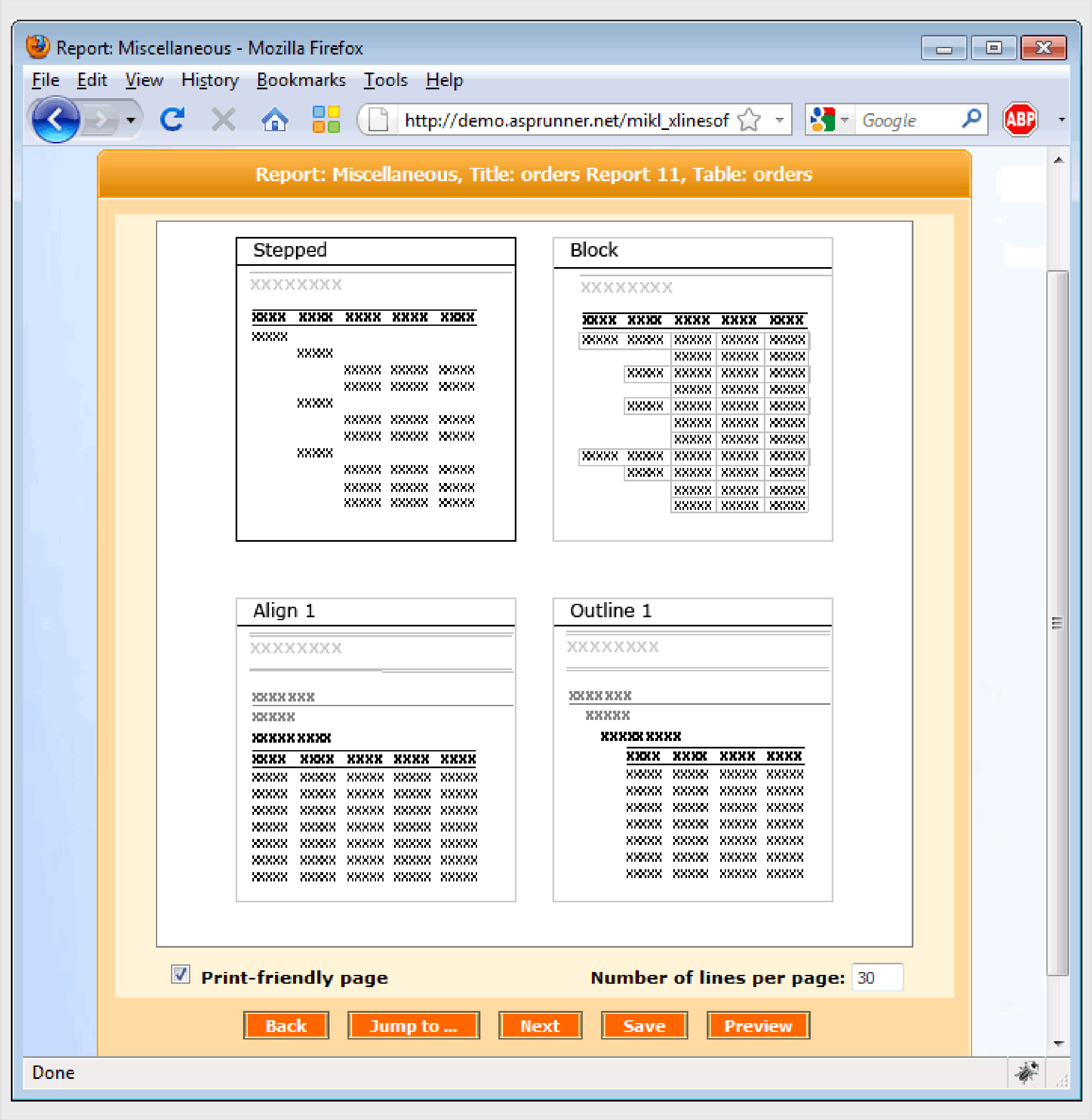
Field Name	Label	Show	Min	Max	Sum	Avg
orders.CustomerID	CustomerID	☑	□	□	□	□
orders.EmployeeID	EmployeeID	☑	□	□	□	□
orders.OrderID	OrderID	☑	□	□	□	□
orders.OrderDate	OrderDate	☑	☑	☑	□	□
orders.RequiredDate	RequiredDate	☑	□	□	□	□
orders.ShippedDate	ShippedDate	☑	□	□	□	□
orders.ShipVia	ShipVia	☑	□	□	□	□
orders.Freight	Freight	☑	□	□	□	□
orders.ShipName	ShipName	☑	□	□	□	□

Miscellaneous

On this page you can choose report layout. If you use grouping you can choose between Stepped, Block, Outline and Align layouts. If you don't use grouping you can use only Tabular layout.

Print-friendly page option enables/disables the following features: printing the whole report or its page in print-friendly version, opening a report as Microsoft Word or Microsoft Excel document.

Use **Number of lines per page** option to determine where to insert the page break, when you print the whole report.



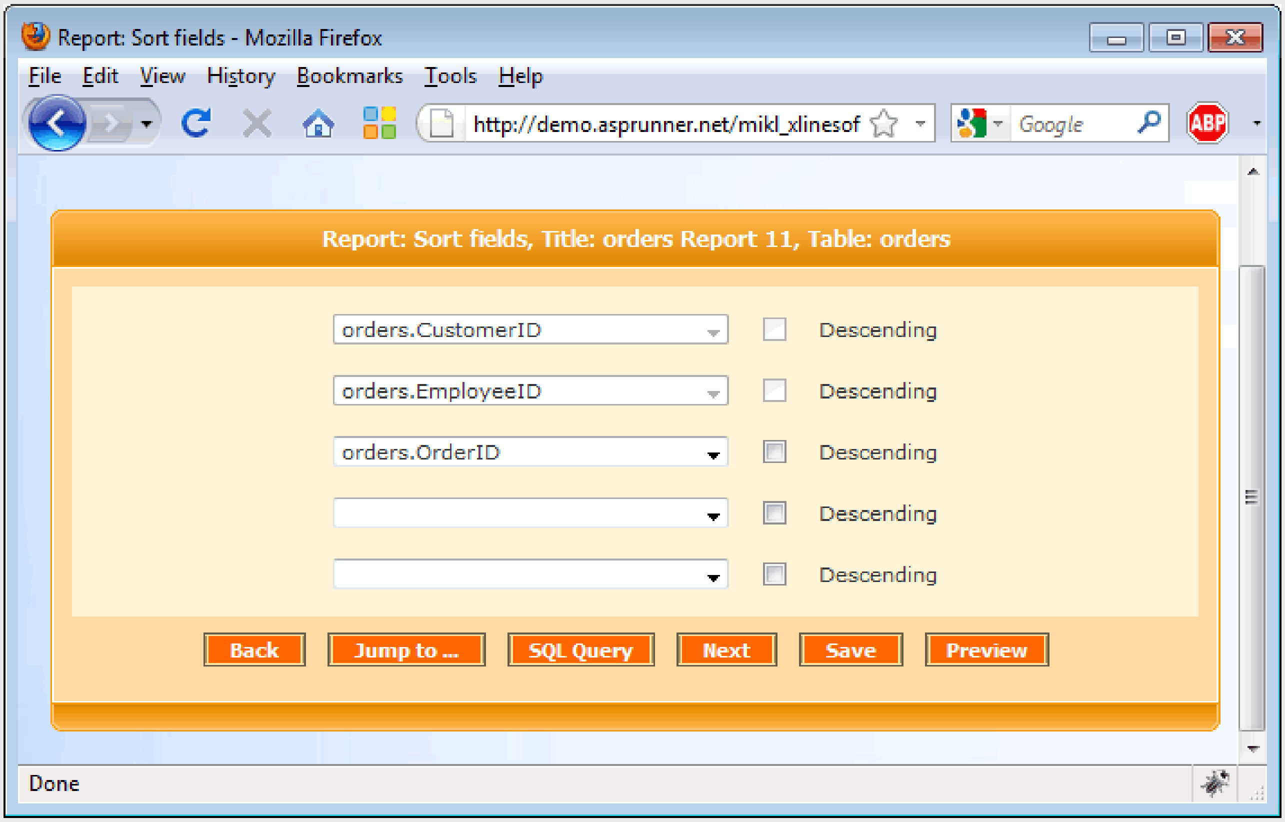
Report: Miscellaneous - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesof
Google
Report: Miscellaneous, Title: orders Report 11, Table: orders
Stepped
Block
Align 1
Outline 1
Back
Jump to ...
Next
Save
Preview
Done

☑ Print-friendly page

Number of lines per page: 30

Sort fields

On this page you can define sort fields to sort the records in the report. Note than you can't modify settings for group fields.


Report: Sort fields - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesof
Google
Done

Report: Sort fields, Title: orders Report 11, Table: orders

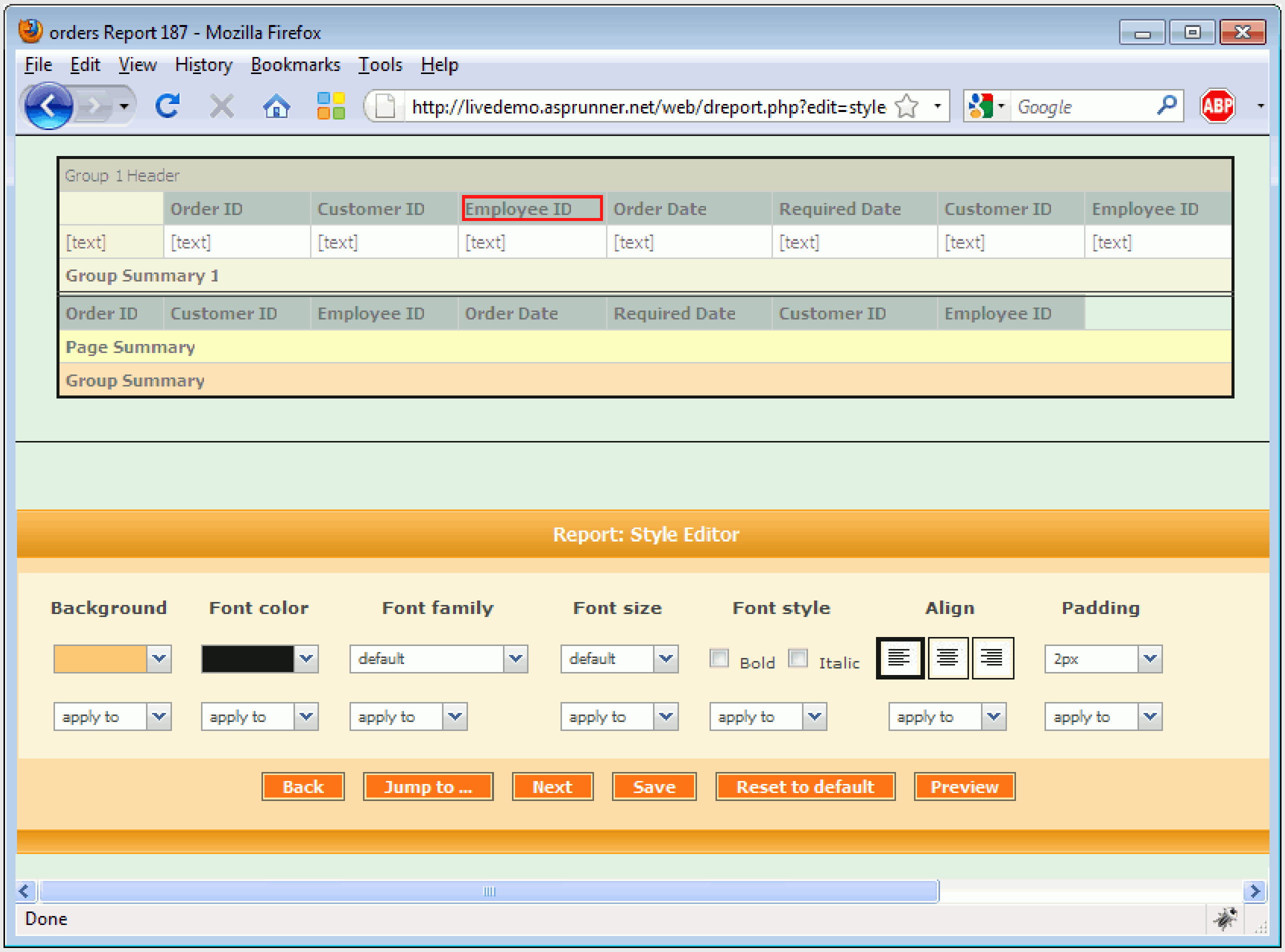
orders.CustomerID	□	Descending
orders.EmployeeID	□	Descending
orders.OrderID	□	Descending
	□	Descending
	□	Descending

Back
Jump to ...
SQL Query
Next
Save
Preview

Use the **SQL Query** button to view resulting SQL query and query results.

Style Editor

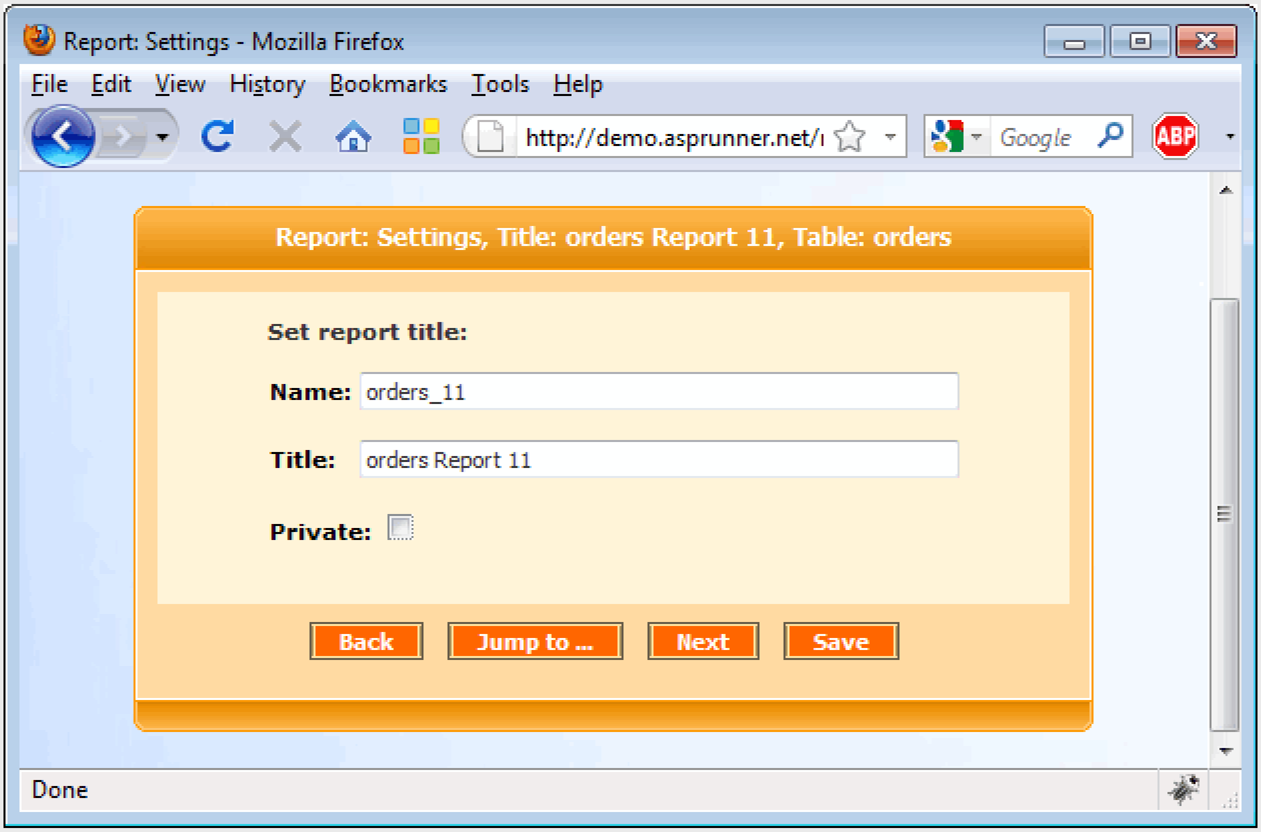
On this page you can define font settings and background color to be used in the report. Firstly, select report element and then define style settings for it. Using **apply to** dropdown list box you can apply changes made to a group (row), field (column) or all text in the report. Use **Reset to default** button to return to default settings.



Font family: default
Font size: default
Padding: 2px

Settings

On this page you can define the report name and title. If your project uses security, you have also an option to make a report private. Private reports are not accessible by anyone but owner. Non-private (public) ones will appear under "shared" section on the start page.


Report: Settings - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/
Google
Done

Report: Settings, Title: orders Report 11, Table: orders

Set report title:

Name: orders_11

Title: orders Report 11

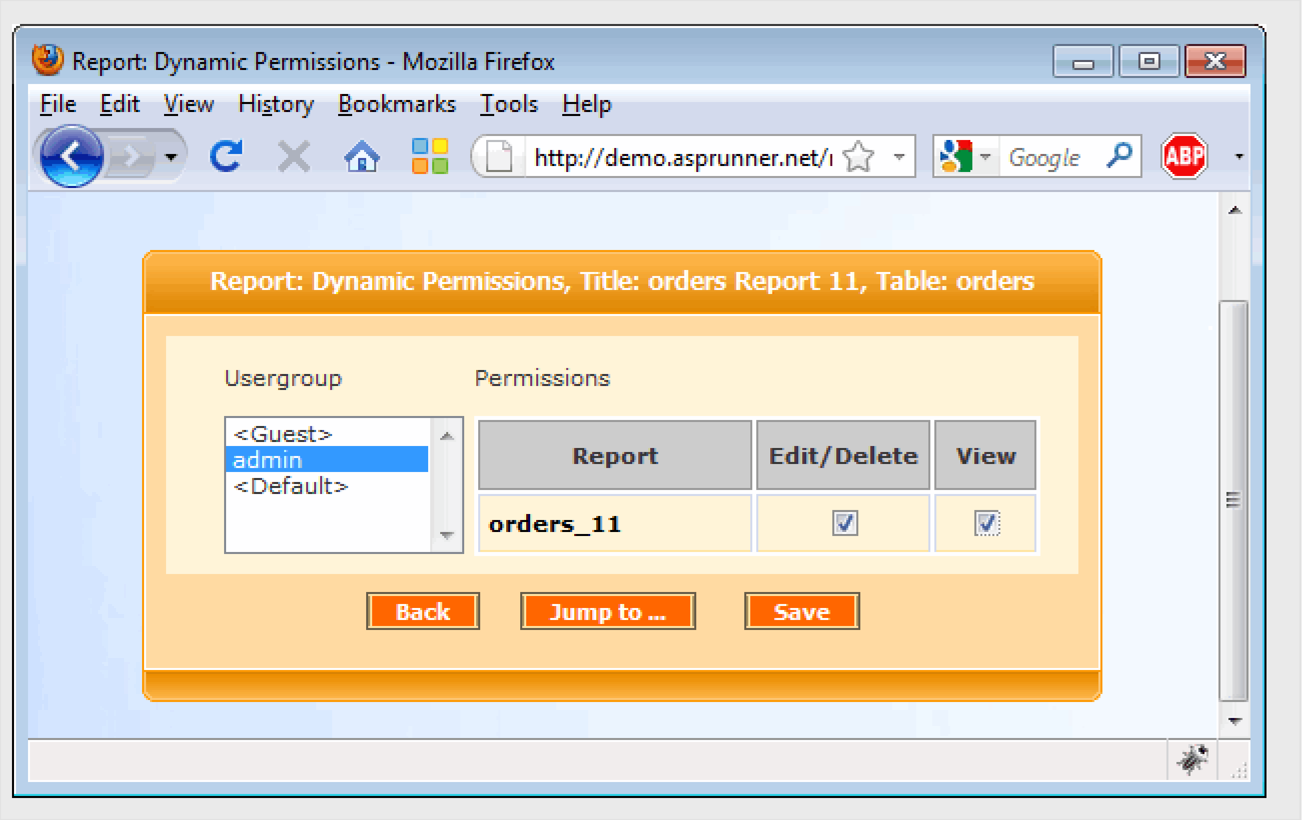
Private: □

Back
Jump to ...
Next
Save

Dynamic Permissions

Note: this page is available if you enabled dynamic permissions in PHPRunner and a report is not marked as private on the previous step.

On this page you can assign user group permissions to view/edit/delete a report.


Report: Dynamic Permissions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/
Google
ABP

Report: Dynamic Permissions, Title: orders Report 11, Table: orders

Usergroup

- <Guest>
- admin
- <Default>

Permissions

Report	Edit/Delete	View
orders_11	☑	☑

3.5.3 Creating web chart

To create new chart, click the **Create Chart** button on the start page and follow the steps to define chart settings.

The **Back** and **Next** buttons allow you to jump to previous and next page correspondingly. Use **Jump to** button to jump to any other page. The **Save** button saves the chart and moves you to the web reports menu (start page). Use the **SQL Query** button to view resulting SQL query and query results. The **Preview** button allow you to see how you chart will look.

Below you can find the description of the chart creation steps:

Tables

On this page you can choose a table, view or SQL query as a data source for your chart. Note that tables that are not added to the project from a database are not available for selection.

Tables from the database

- All tables added to the project are available for selection. User tables (custom views) are not available.

- When viewing a chart, all user permissions (static and dynamic permissions, advanced security option) and "view/edit" field settings do not work.
- While creating a chart you will be able to create table relations (SQL joins) to query data from two or more tables and add additional search conditions using WHERE clause.

Tables from the project

- All tables added to the project and custom views created in PHPRunner are available for selection.
- When viewing a chart, user permissions and "view/edit" field settings work as usual.
- While creating a chart you will not be able to create table relations (SQL joins) to query data from two or more tables and add additional search conditions using WHERE clause. The SQL queries defined in PHPRunner will be used to query data for the chart.
- Tables for selection are displayed as *Caption (table title)*, e.g. Cars (carscars).

SQL queries

On the **SQL queries** tab a user with admin permissions can create new custom SQL query using the **New query** button and edit the existing SQL queries using the **SQL Query** button. For more information about custom SQL queries, see Custom SQL.

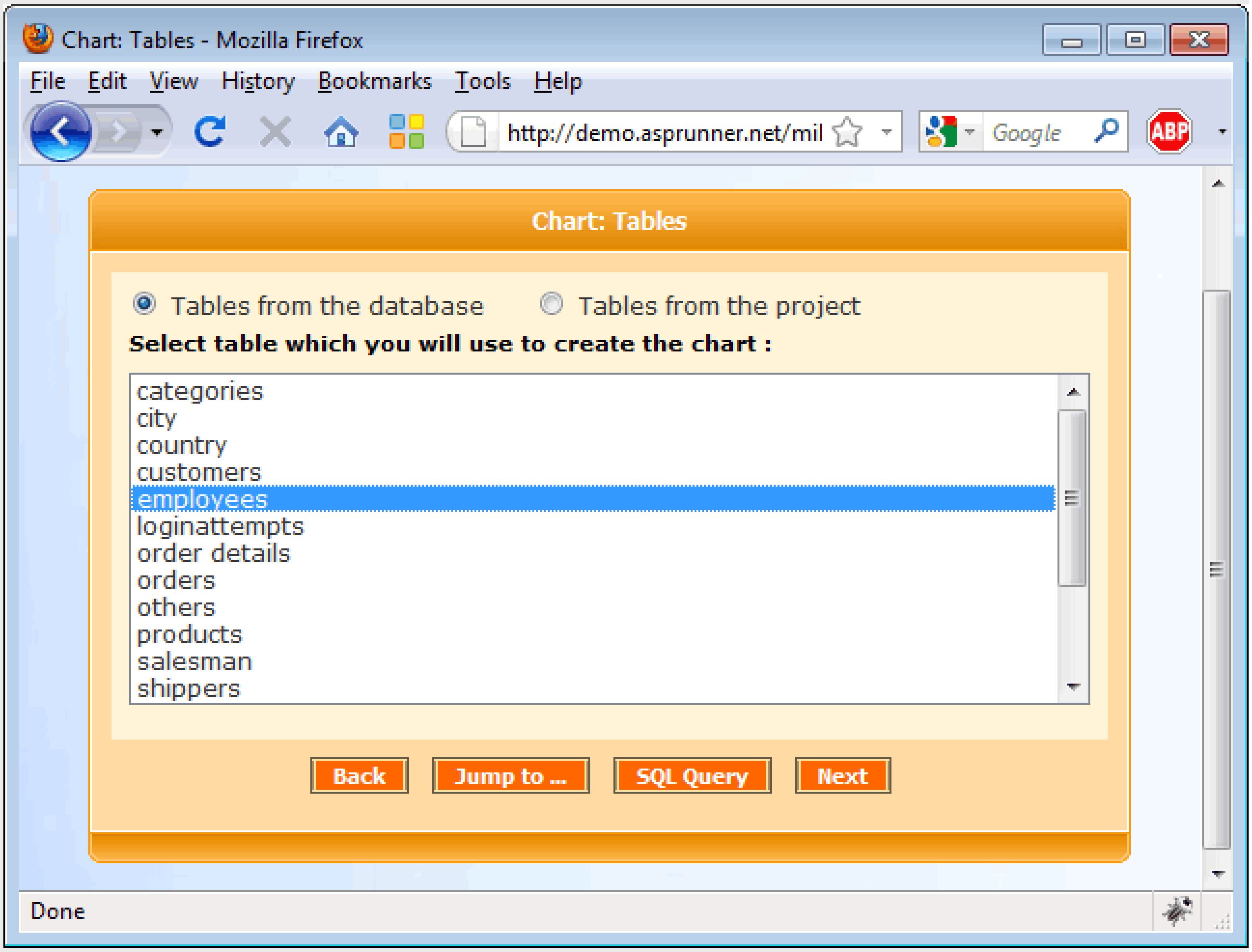


Chart: Tables - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mil
Google
ABP
Back
Jump to ...
SQL Query
Next
Done

Chart: Tables

● Tables from the database ○ Tables from the project

Select table which you will use to create the chart :

categories
city
country
customers
employees
loginattempts
order details
orders
others
products
salesman
shippers

Table Relations

Note: this page is available if you selected a table from the database as chart data source.

On this page you can create table relations (SQL joins) to query data from two or more tables, based on a relationship between certain fields in these tables. You can add Inner Join, Left Join, Right Join and Full Outer Join.

To add an SQL join, choose tables and fields to be joined and click **Add Relation**. The JOIN clause will be added below the SELECT clause. You can add several table relations. Use the **Remove Relation** button to delete the selected table relation.

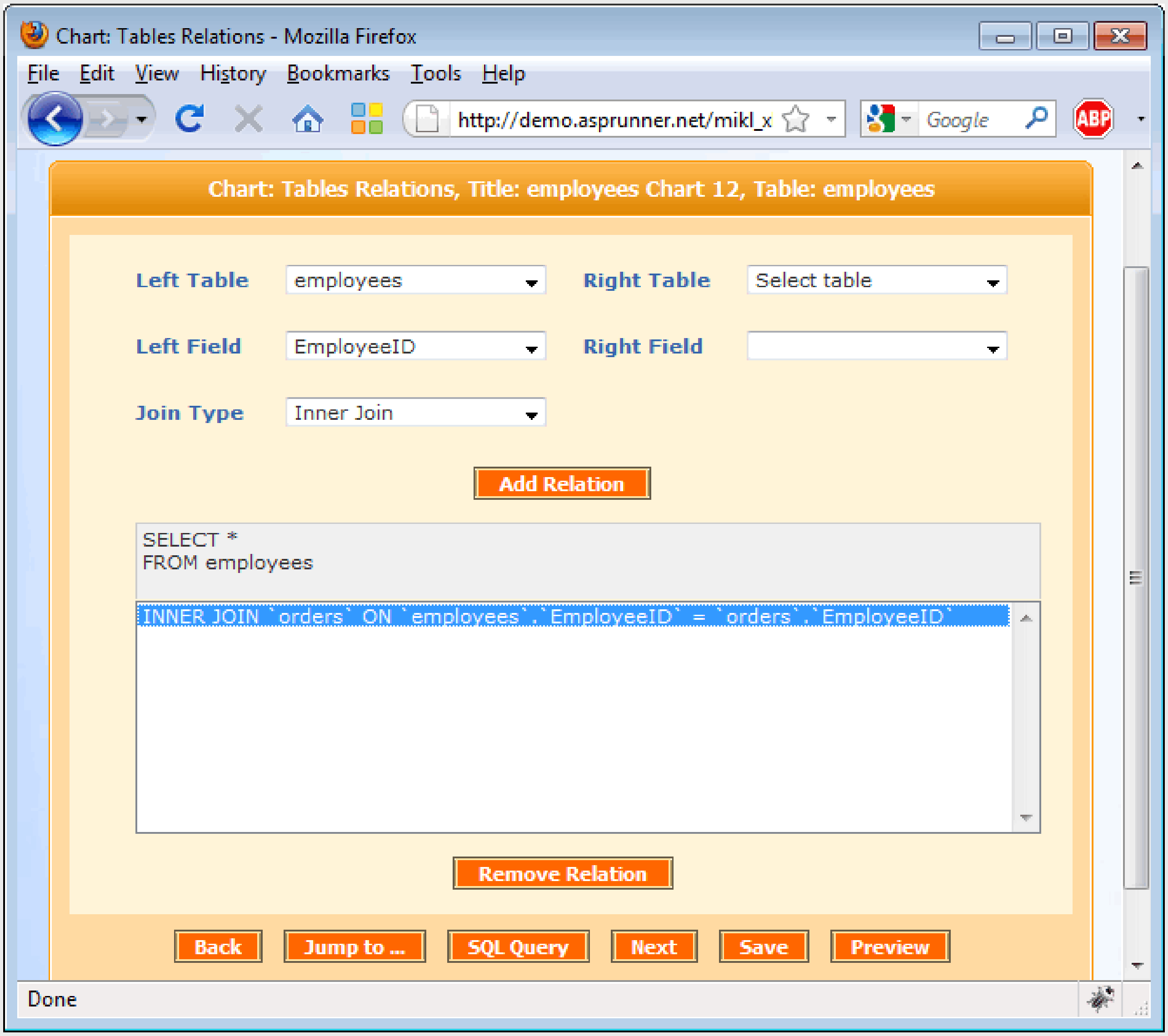


Chart: Tables Relations - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_x
Google
Done

Chart: Tables Relations, Title: employees Chart 12, Table: employees

Left Table: employees
Right Table: Select table
Left Field: EmployeeID
Right Field:
Join Type: Inner Join

```
SELECT *
FROM employees
```

```
INNER JOIN `orders` ON `employees`.`EmployeeID` = `orders`.`EmployeeID`
```

Remove Relation
Back
Jump to ...
SQL Query
Next
Save
Preview

Use **SQL Query** button to view resulting SQL query and query results.

Group By

Note: this page is available if you selected a table from the database as chart data source.

On this page you can add additional search conditions using WHERE clause. To do this select a field in the first column and type search criteria in the **Filter** and **OR**... text boxes on the right. Search criterion should be added as **<operator><value>**. E.g. *='USA'* or *=2009* or *<>'red'* or *>10*.

Also you can define sort fields to sort the records in the chart. To do this select a field in the first column and choose sort type and sort order.

In addition, you can add group fields to group the results by one or more columns and apply aggregate functions like MIN, MAX, SUM, AVERAGE and COUNT. To do this select a field in the first column, select the **Group By** check box and choose one of values in the dropdown list box under this check box. You can filter the records that a GROUP BY clause returns using HAVING clause. To do this select a field in the first column, select one of the aggregate functions and type the condition as **<operator><value>** in the textbox under **Having**. E.g. *>10* or *= 500*.

Example. Number of employees per city

Let's build a chart showing the number of employees per city except 'London'. We need to use WHERE clause to select employees that are not working in London, GROUP BY clause to group records by city and COUNT function to calculate the number of employees in each city.

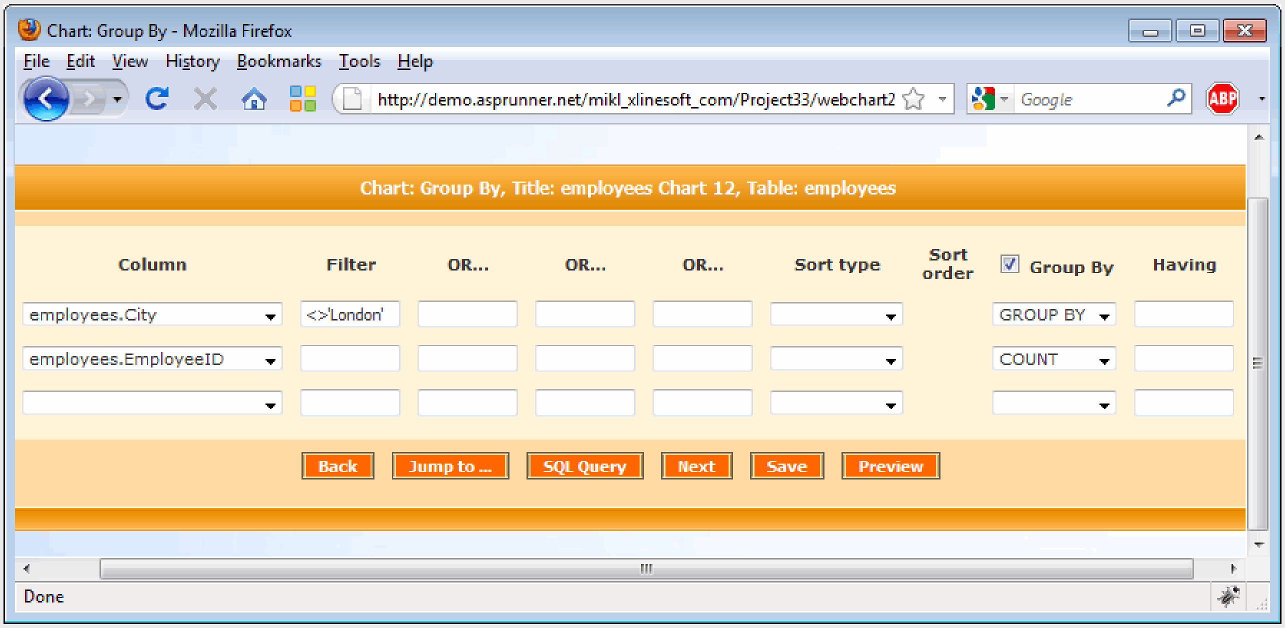


Chart: Group By - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesoft_com/Project33/webchart2
Google
Back
Jump to ...
SQL Query
Next
Save
Preview
Done

Chart: Group By, Title: employees Chart 12, Table: employees

Column	Filter	OR...	OR...	OR...	Sort type	Sort order	☑ Group By	Having
employees.City	<>'London'						GROUP BY	
employees.EmployeeID							COUNT	

Use **SQL Query** button to view resulting SQL query and query results.

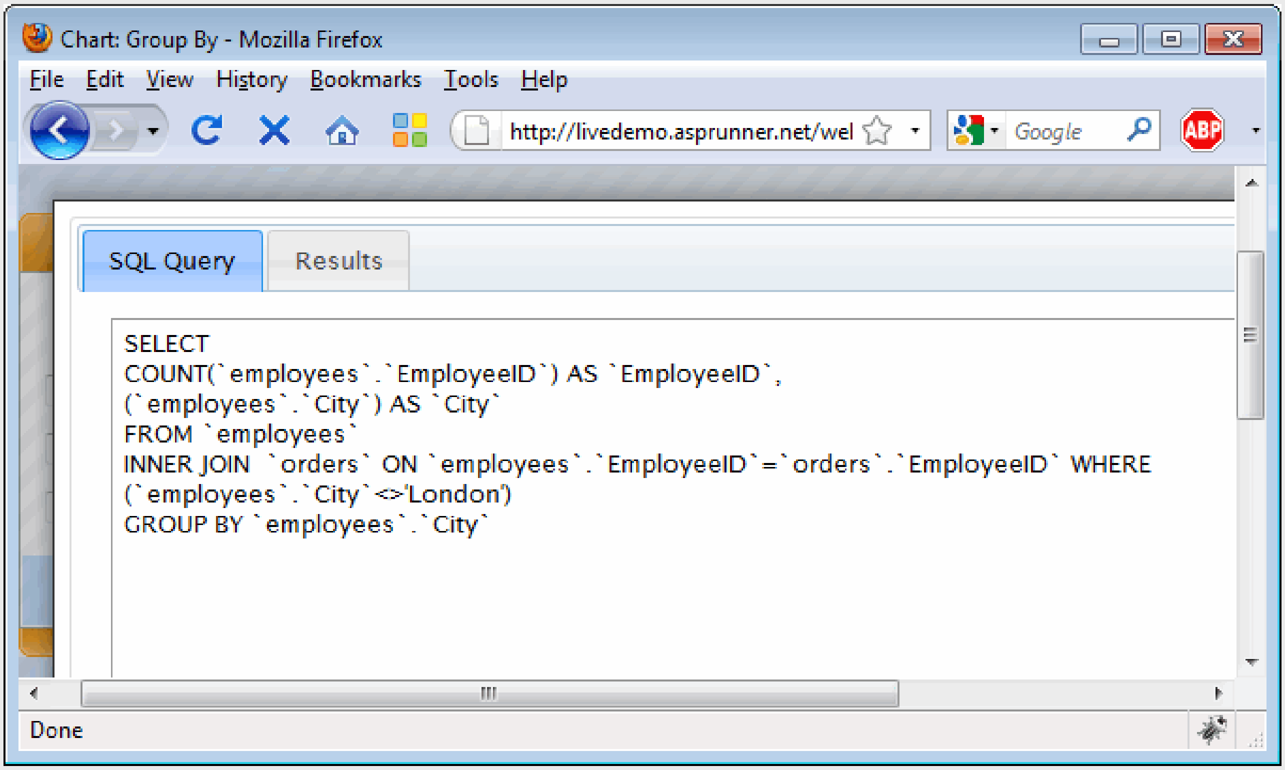


Chart: Group By - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/wel
Google
SQL Query
Results
Done

```
SELECT
COUNT(`employees`.`EmployeeID`) AS `EmployeeID`,
(`employees`.`City`) AS `City`
FROM `employees`
INNER JOIN `orders` ON `employees`.`EmployeeID`=`orders`.`EmployeeID` WHERE
(`employees`.`City`<>'London')
GROUP BY `employees`.`City`
```

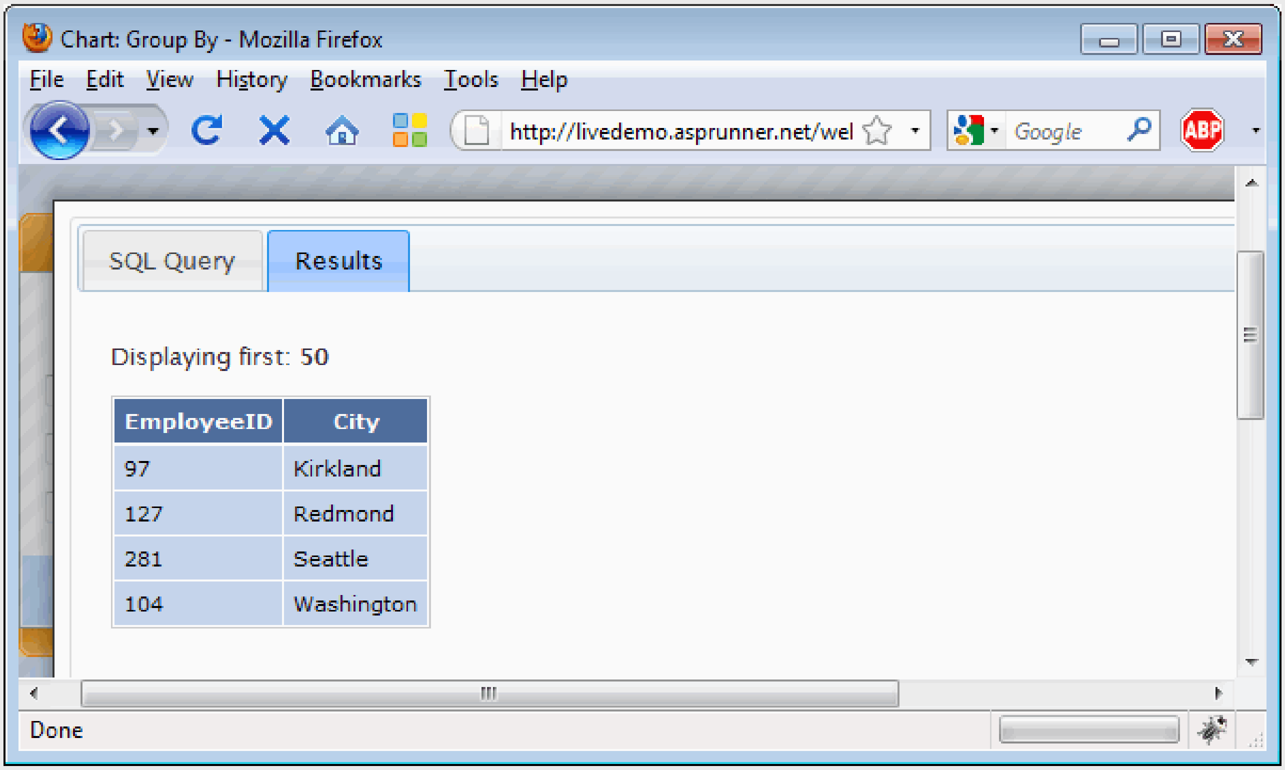


Chart: Group By - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/wel
Google
SQL Query
Results
Done

Displaying first: 50

EmployeeID	City
97	Kirkland
127	Redmond
281	Seattle
104	Washington

Type

On this page you can select a chart type. For more information, see Chart types.

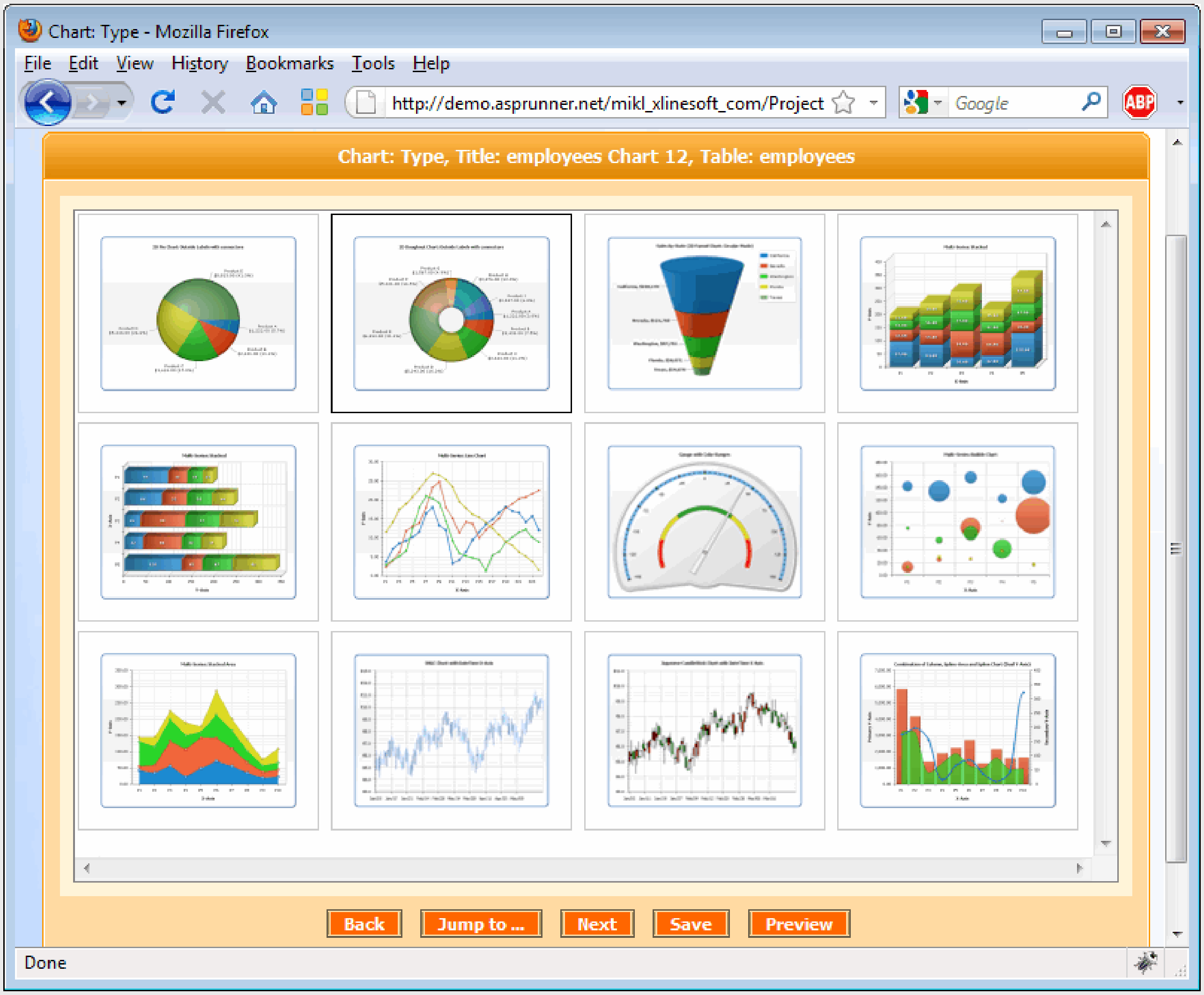


Chart: Type - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesoft_com/Project
Google
Chart: Type, Title: employees Chart 12, Table: employees
Back
Jump to ...
Next
Save
Preview
Done

Parameters

On this page you can choose Data Series fields (fields with data) and label field (field with data labels).

You can add unlimited number of data series. Additional Data series dropdown list boxes are added automatically once you used available ones.

For more information about choosing data series for certain chart type, see Chart types.

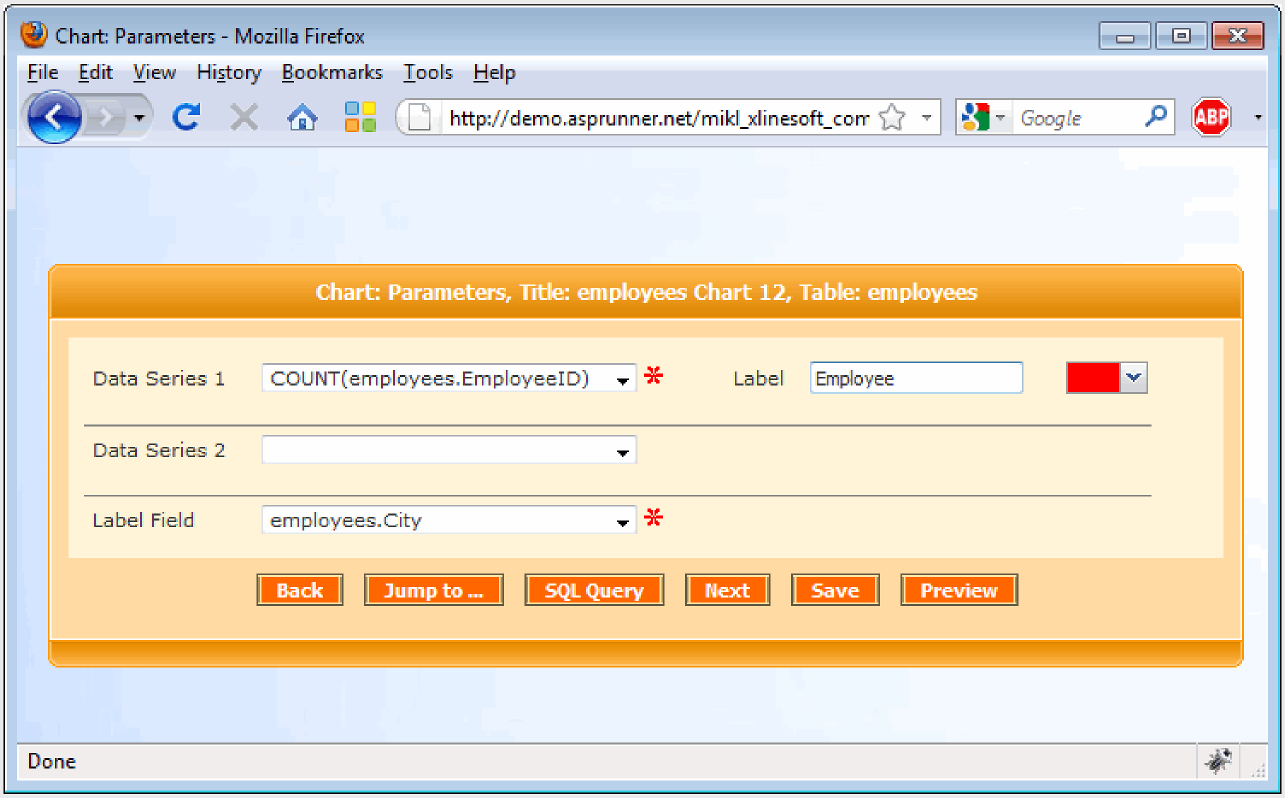


Chart: Parameters - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mikl_xlinesoft_com
Google

Chart: Parameters, Title: employees Chart 12, Table: employees

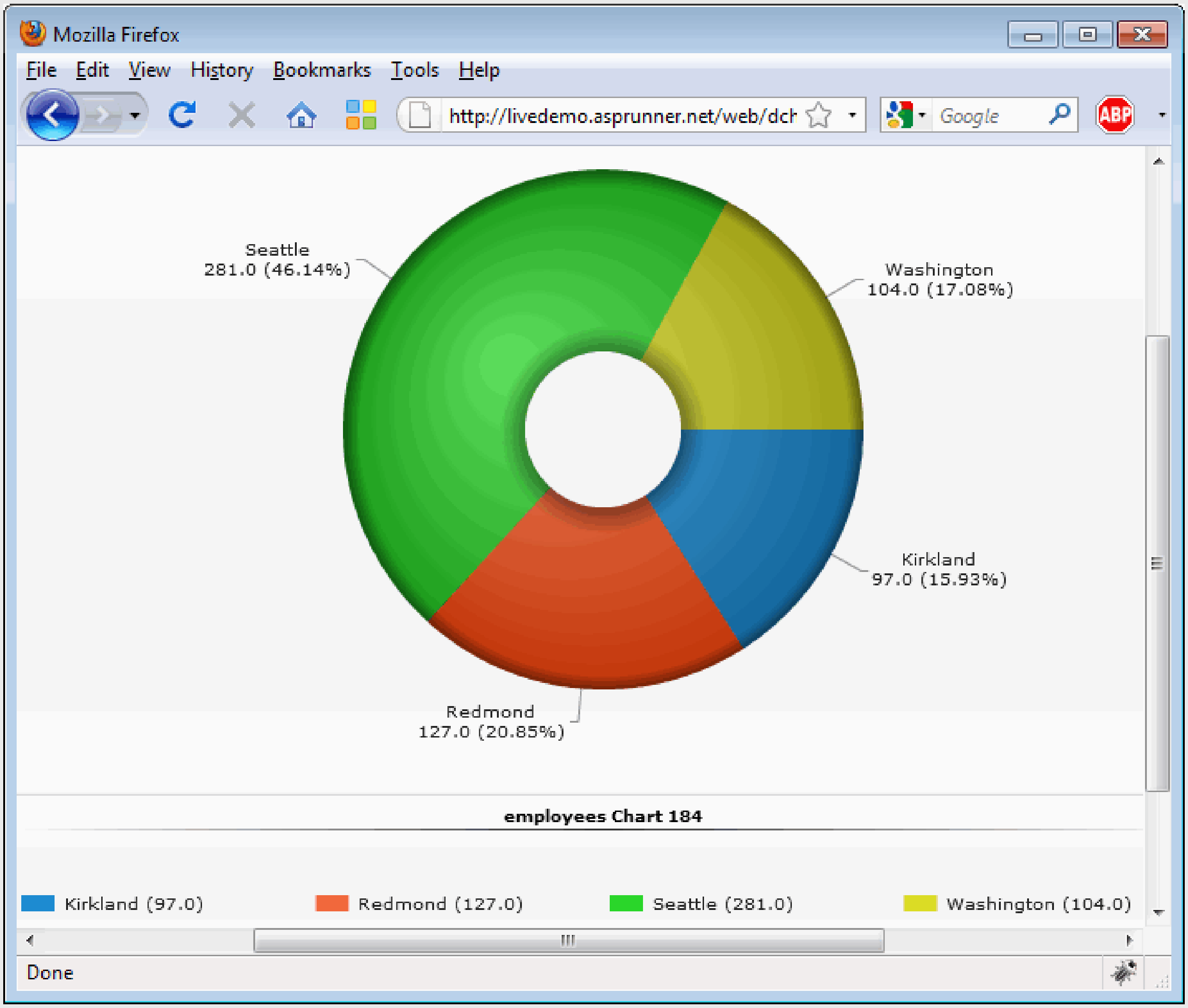
Data Series 1	COUNT(employees.EmployeeID) *	Label	Employee 🟥
Data Series 2			
Label Field	employees.City *		

Back
Jump to ...
SQL Query
Next
Save
Preview
Done

Note: only numeric fields can be chosen as a Data Series. Therefore only numeric fields are available for selection in Data Series dropdown list box.

The color options define the colors of the data series in the Line charts.

If we select Data Series and Label fields as shown on the image above, we receive the following chart:



Appearance

On this page you can define how your chart will be displayed on the web page. The following two pictures explain how each option works (on the first picture we numbered the options; on the second one we showed how these options effect on the chart appearance).

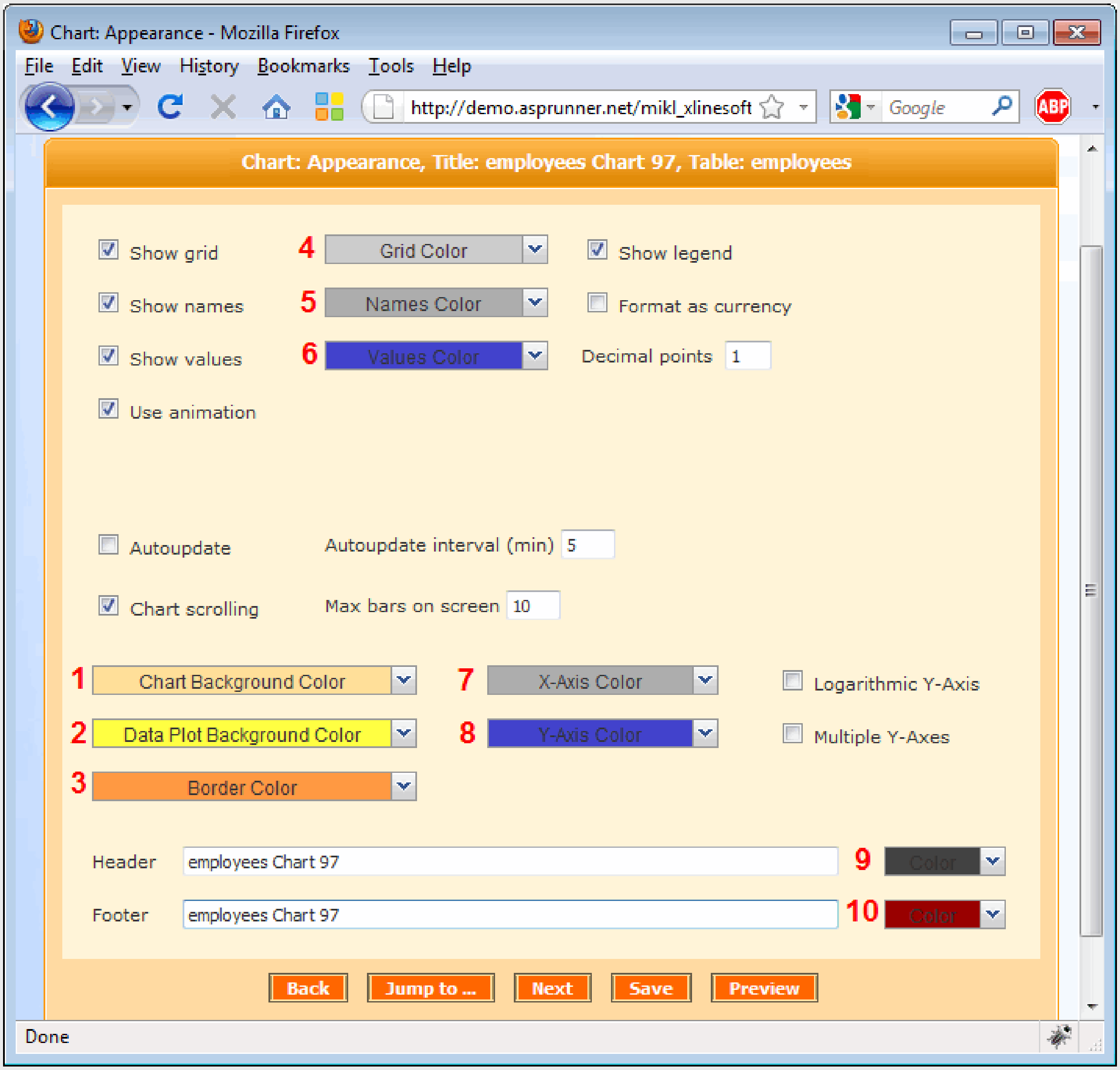


Chart: Appearance - Mozilla Firefox

http://demo.asprunner.net/mikl_xlinesoft

Chart: Appearance, Title: employees Chart 97, Table: employees

☑ Show grid

☑ Show names

☑ Show values

☑ Use animation

☑ Show legend

☐ Format as currency

Decimal points 1

☐ Autoupdate

Autoupdate interval (min) 5

☑ Chart scrolling

Max bars on screen 10

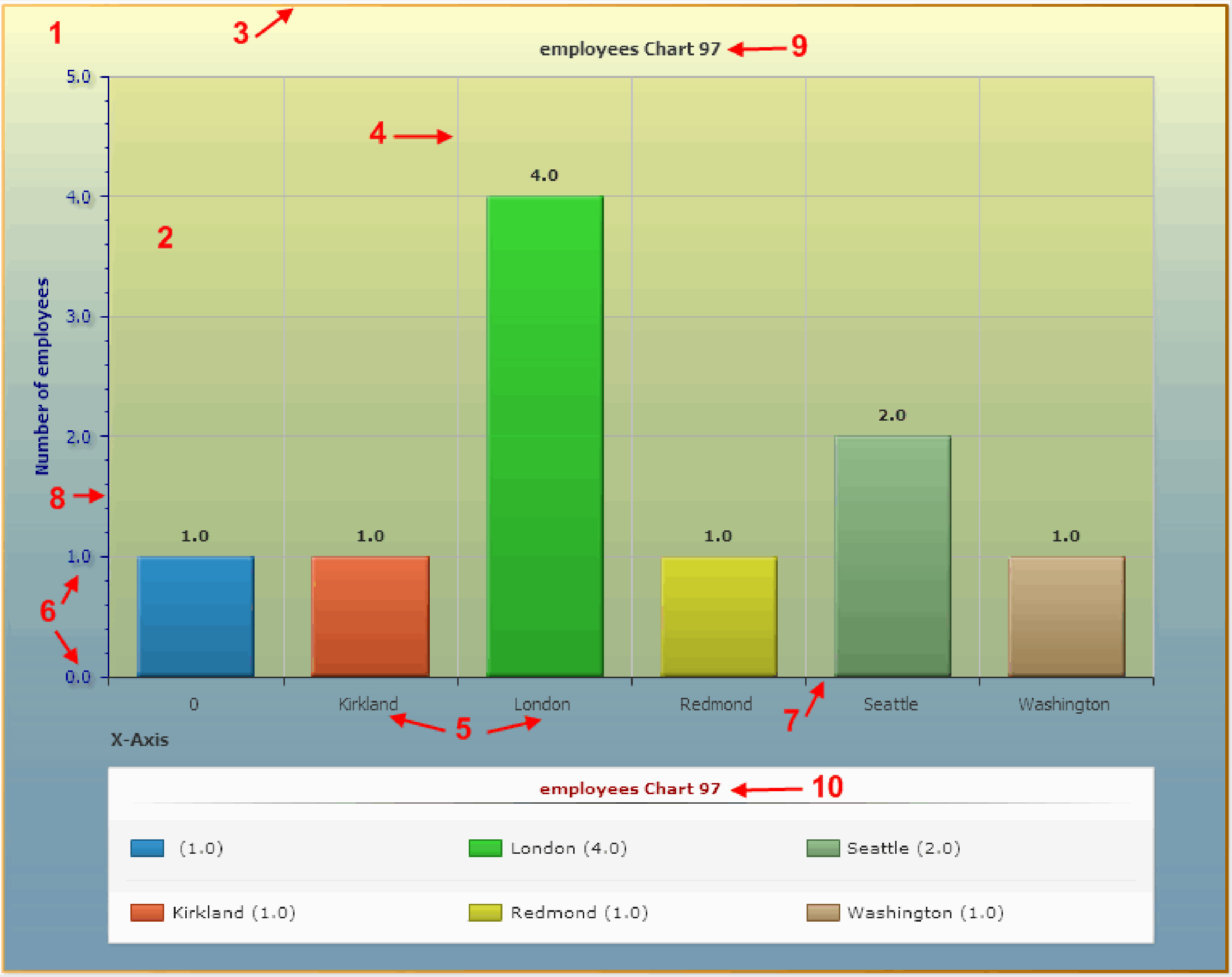
☐ Logarithmic Y-Axis

☐ Multiple Y-Axes

Header employees Chart 97

Footer employees Chart 97

Done



Use the **Autoupdate** check box to enable chart auto-refresh by specified time interval. The **Use animation** check box enables the chart animation while opening a chart.

The **Chart scrolling** option allows you display a scrollable chart. Don't forget to define the number of bars to show on the chart screen.

Use the **Logarithmic Y-Axis** option to convert a linear value axis to a logarithmic value axis. If you have several data series on the chart, you can use the **Multiple Y-Axes** option to position each data series relative to its own Y axis.

For more information about appearance settings for certain chart type, see Chart types.

Settings

On this page you can define the chart name and title. If your project uses security, you have also an option to make a chart private. Private charts are not accessible by anyone but owner.

Non-private (public) ones will appear under "shared" section on the start page.

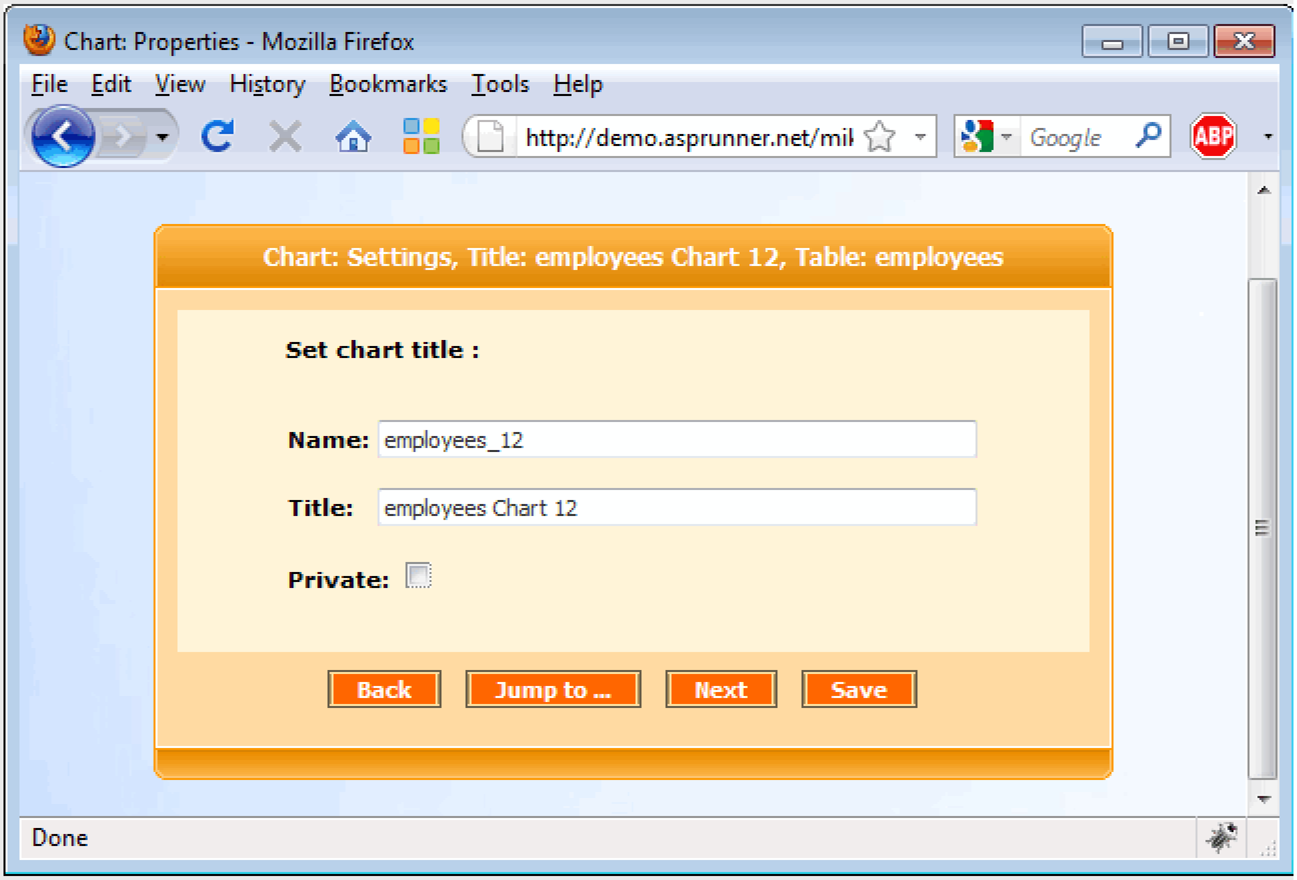


Chart: Properties - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mil
Google
Done

Chart: Settings, Title: employees Chart 12, Table: employees

Set chart title :

Name: employees_12

Title: employees Chart 12

Private: □

Back
Jump to ...
Next
Save

Dynamic Permissions

Note: this page is available if you enabled dynamic permissions in PHPRunner and a chart is not marked as private on the previous step.

On this page you can assign user group permissions to view/edit/delete a chart.

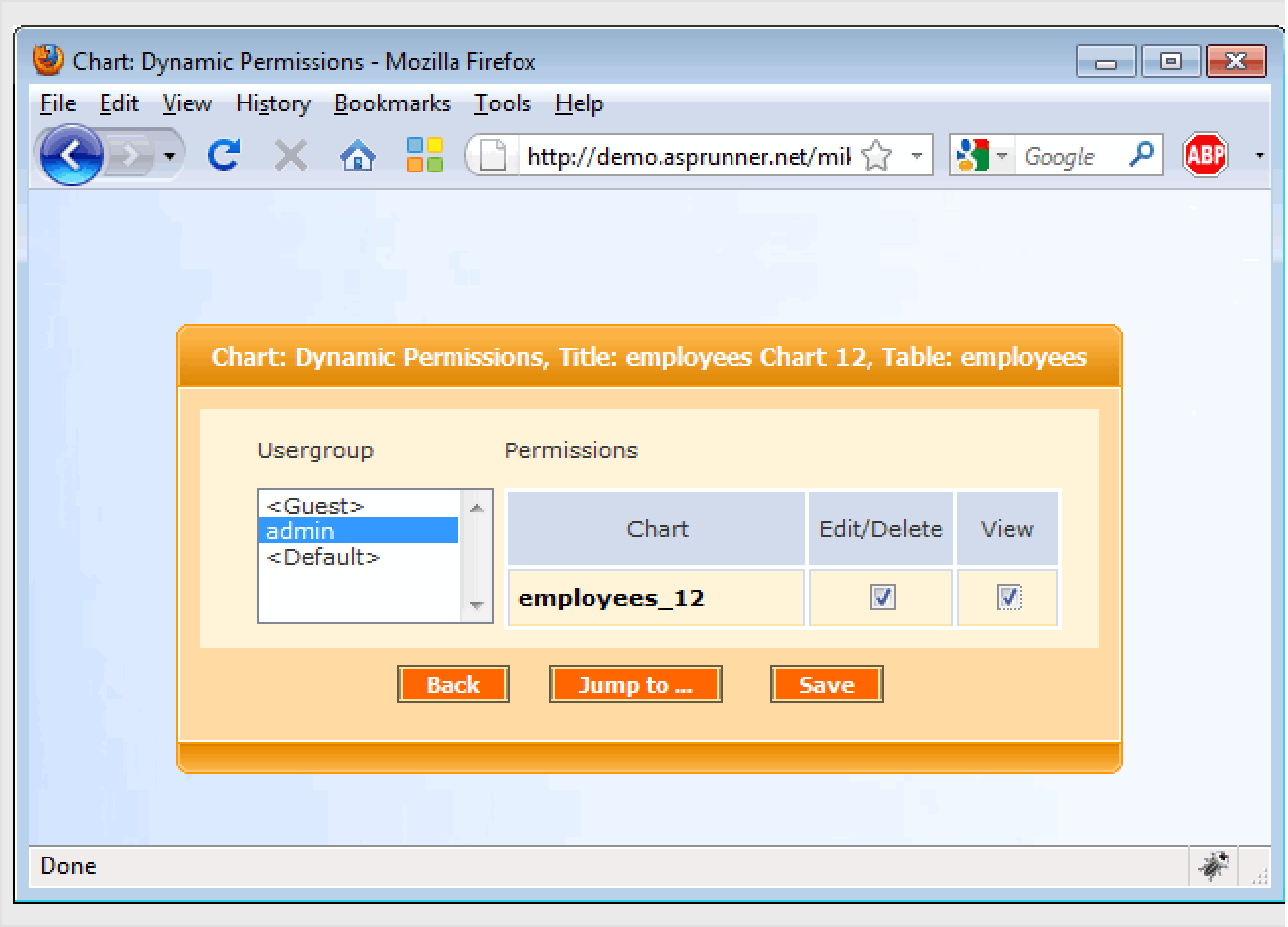


Chart: Dynamic Permissions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://demo.asprunner.net/mil
Google

Chart: Dynamic Permissions, Title: employees Chart 12, Table: employees

Usergroup

<Guest>
admin
<Default>

Permissions

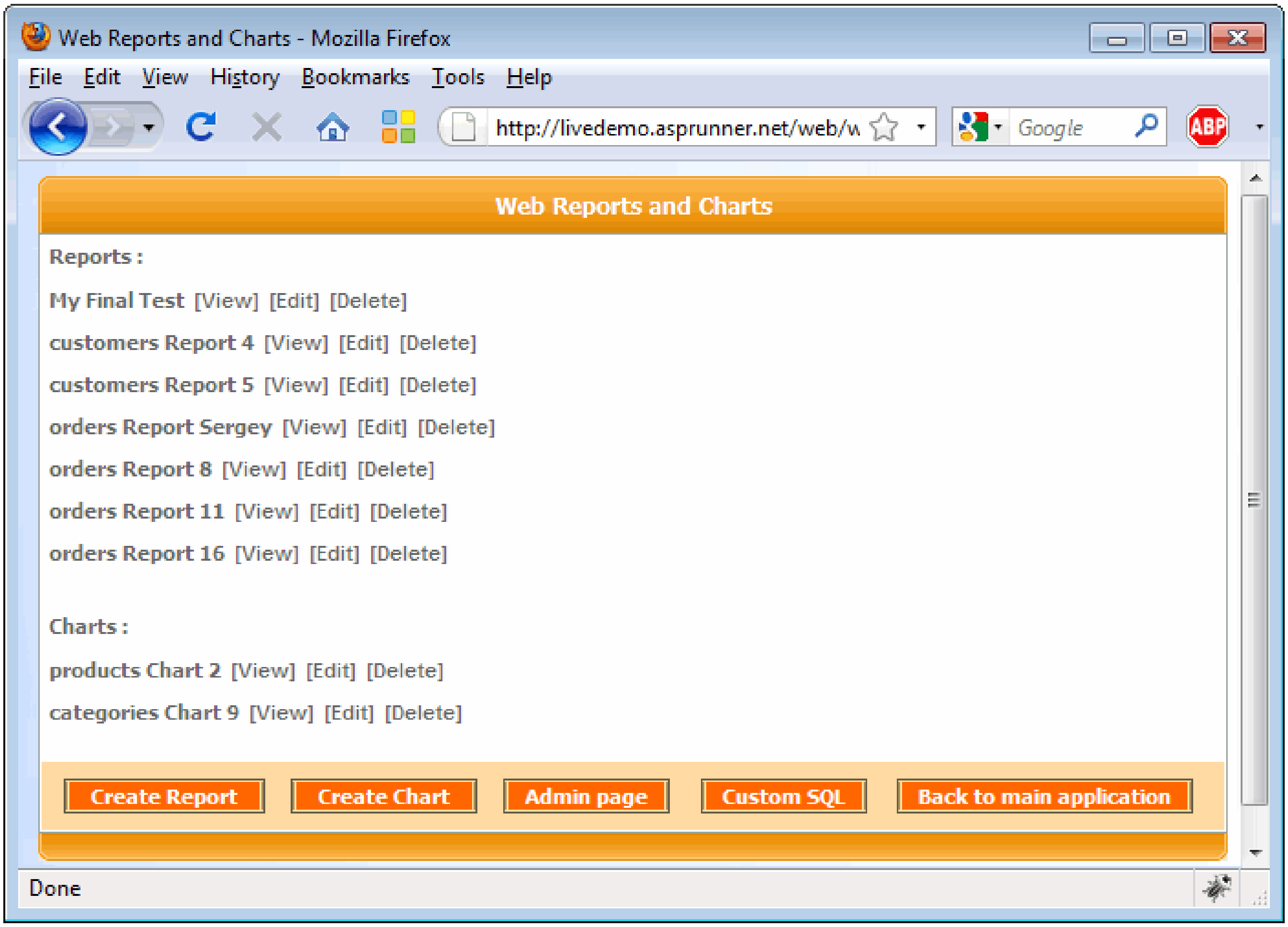
Chart	Edit/Delete	View
employees_12	☑	☑

Back
Jump to ...
Save
Done

3.5.4 Custom SQL

A user with admin permissions (admin) can create custom SQL queries that later may be used by other users as a datasource for reports and charts. Also admin can edit and delete custom SQL queries.

To view/create/edit/delete custom SQL queries use the **Custom SQL** button on the start window.



Web Reports and Charts - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://livedemo.asprunner.net/web/w
Google
Web Reports and Charts

Reports :

My Final Test [View] [Edit] [Delete]

customers Report 4 [View] [Edit] [Delete]

customers Report 5 [View] [Edit] [Delete]

orders Report Sergey [View] [Edit] [Delete]

orders Report 8 [View] [Edit] [Delete]

orders Report 11 [View] [Edit] [Delete]

orders Report 16 [View] [Edit] [Delete]

Charts :

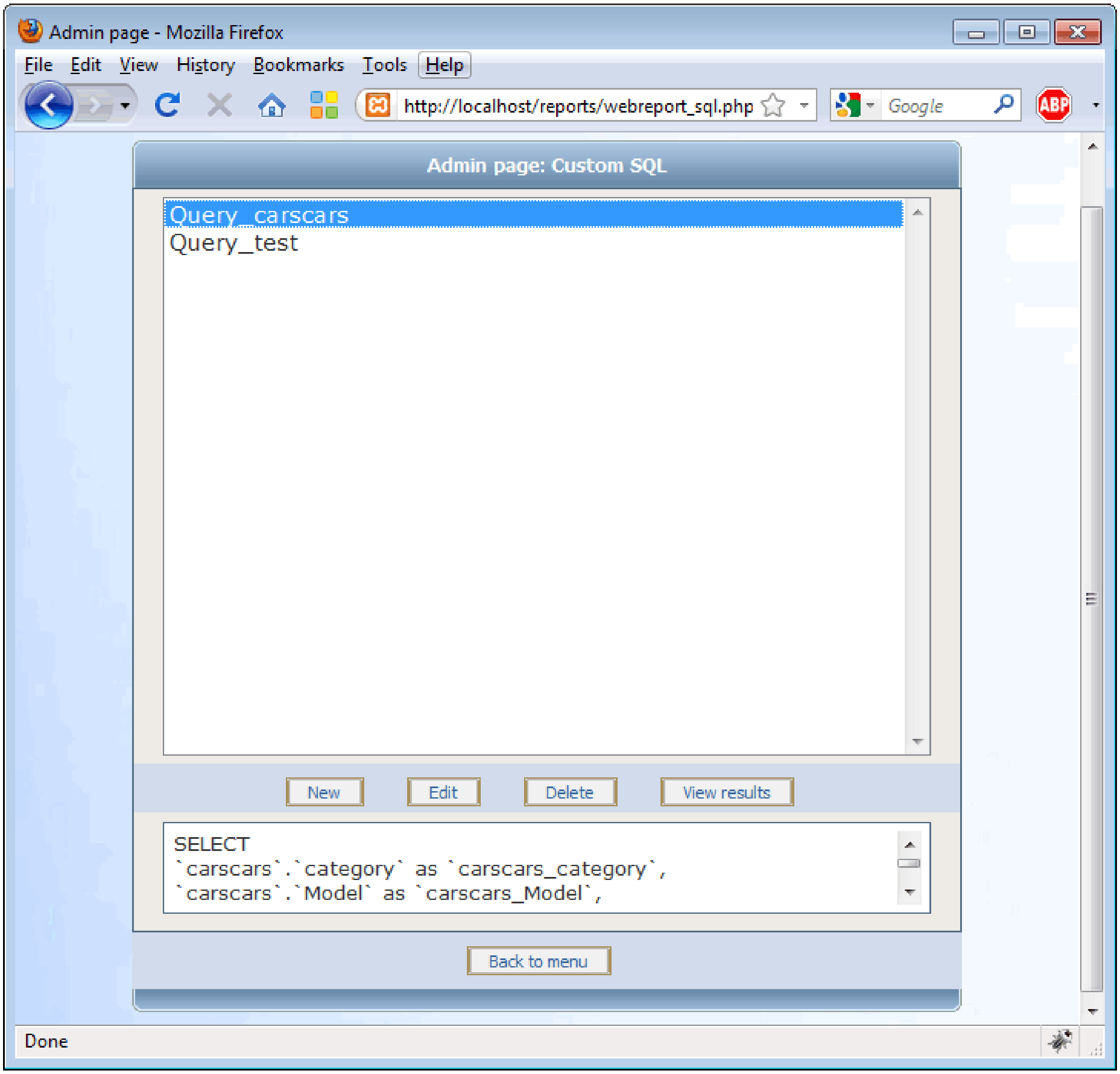
products Chart 2 [View] [Edit] [Delete]

categories Chart 9 [View] [Edit] [Delete]

Create Report
Create Chart
Admin page
Custom SQL
Back to main application
Done

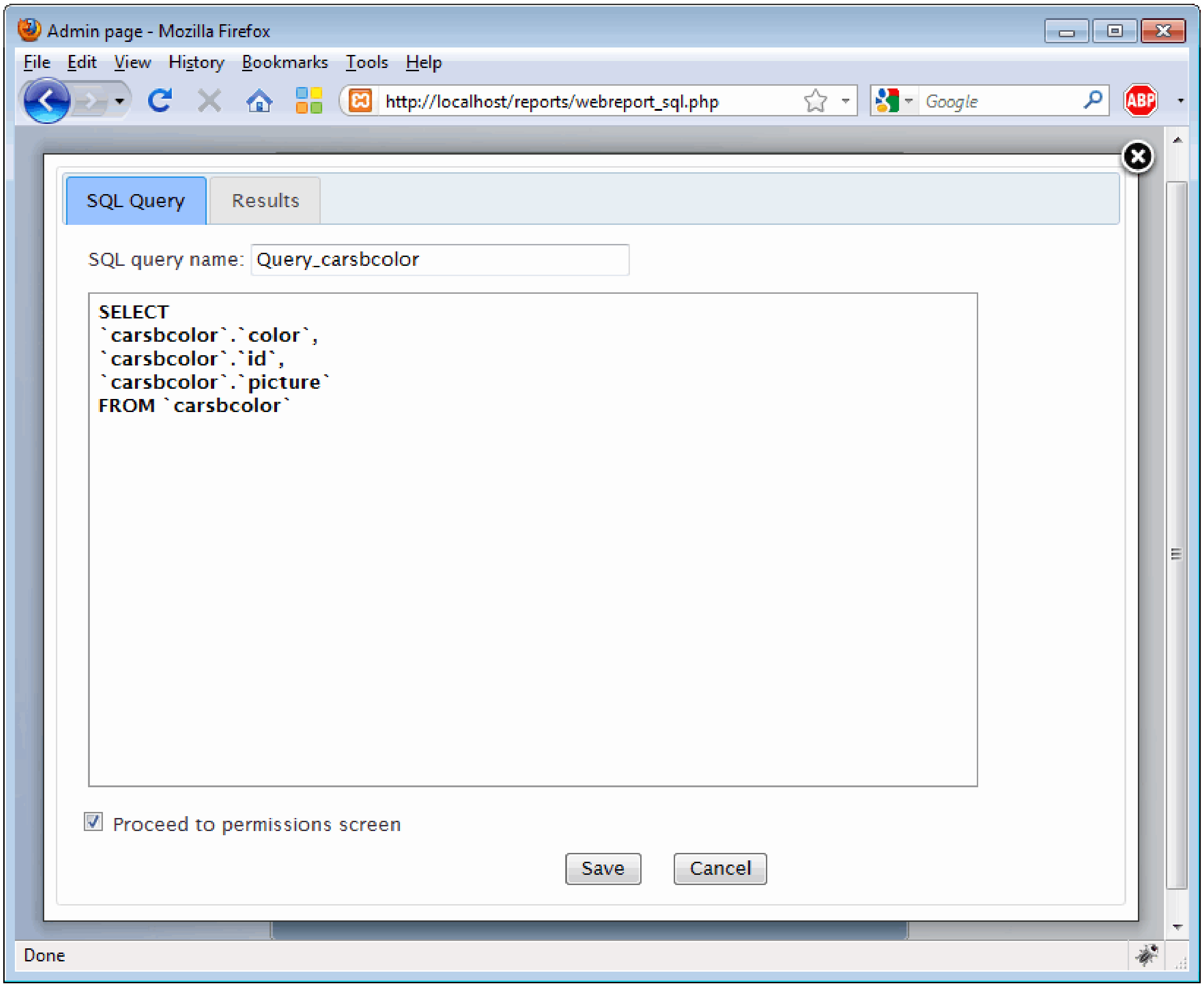
To create new custom SQL query:

1. Click the **Custom SQL** button on the start window.
2. Click the **New** button.


Admin page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/reports/webreport_sql.php
Google
Admin page: Custom SQL
Query_carscars
Query_test
New
Edit
Delete
View results
Back to menu
Done

```
SELECT
`carscars`.`category` as `carscars_category`,
`carscars`.`Model` as `carscars_Model`,
```

3. Enter SQL query name and the query code. If you wish to set permissions for the created query right after its saving, select the **Proceed to permissions screen** check box. Note that you can set permissions later on the Admin page (use the **Admin page** button on the start window).



Admin page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/reports/webreport_sql.php
Google
SQL Query
Results
Save
Cancel
Done

SQL query name: Query_carsbcolor

```
SELECT
`carsbcolor`.`color`,
`carsbcolor`.`id`,
`carsbcolor`.`picture`
FROM `carsbcolor`
```

☑ Proceed to permissions screen

Note that you can use stored procedures in the SQL query for those databases where they are supported (MS SQL Server, MySQL, Oracle, etc.).

Example of calling the stored procedure without parameters in MS SQL Server:

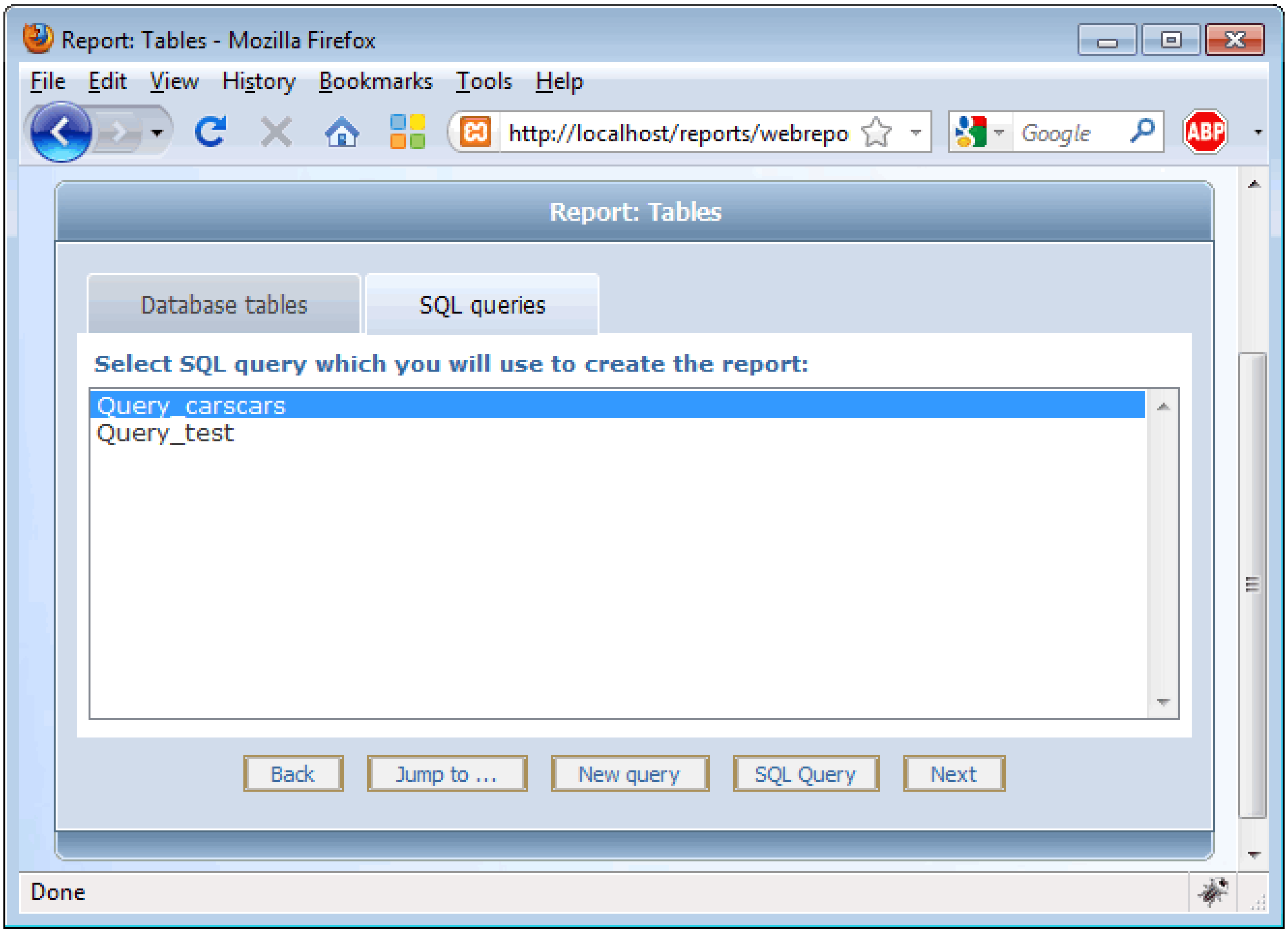
```
EXEC [Ten Most Expensive Products]
```

Example of calling the stored procedure with parameters in MS SQL Server:

```
EXEC [Employee Sales by Country] '10/10/2007','10/10/2009'
```

4. Click **Save**.

When creating a report or chart a user can choose custom SQL query as a datasource (go to the **SQL queries** tab on the **Tables** window). Admin can also create new custom SQL query using the **New query** button and edit the existing SQL queries using the **SQL Query** button.


Report: Tables - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/reports/webrepo
Google
Report: Tables
Database tables
SQL queries
Back
Jump to ...
New query
SQL Query
Next
Done

Select SQL query which you will use to create the report:

Query_carscars
Query_test

3.6 Domain host instructions

3.6.1 Yahoo!

3.6.1.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your Yahoo! account, follow the instructions below.

Step 1. Create a database administrator account

1. Log in to your account at webhosting.yahoo.com.

2. Click the **Web Hosting Control Panel** link.

3. Select **Create & Update** tab.

4. In the **Other Site Building and Editing Tools** section click the **MySQL Database** link.

5. Click the **Database administrator** link.

6. Enter a database administrator user name and password. **Note**: Choose a user name carefully, as you will not be able to change it later.

7. Click **Submit**.

Step 2. Install phpMyAdmin - database administration tool

1. Click the **Install admin tool** link.

2. Read General Public License Agreement and click **I agree to These Terms**.

3. Choose the site folder in which you'd like to install phpMyAdmin. Select an existing directory or create a new directory, then choose the corresponding **Install phpMyAdmin** link to begin the installation.

Congratulations! Your phpMyAdmin tool is now installed.

Step 3. Connect to a MySQL database

1. Start PHPRunner.

2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

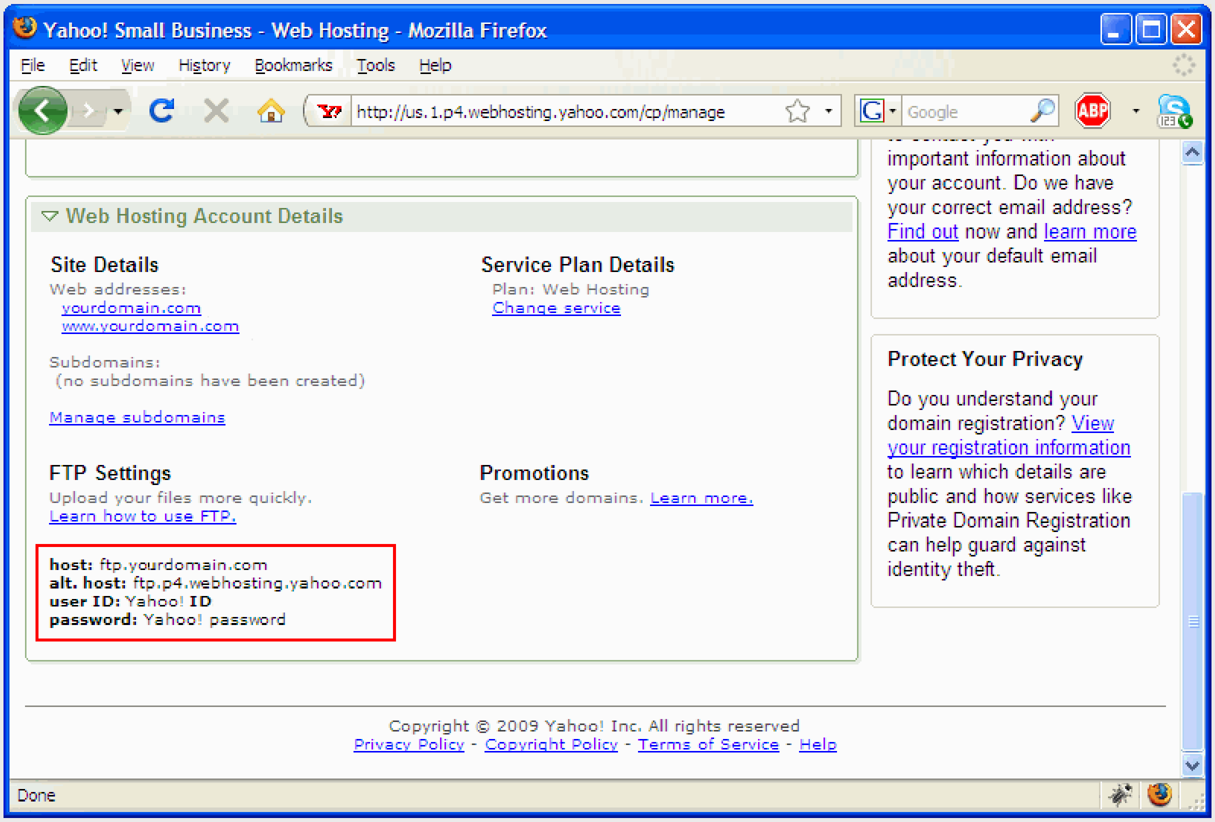
4. Type **mysql** as a Host/Server Name, enter your database administrator user name and password.

5. Select the **Connect using PHP** check box and click **Upload phprunner.php**.

6. Configure built-in FTP client.

How to configure FTP

> To obtain your FTP settings, log in to your account at webhosting.yahoo.com. Click the **Web Hosting Control Panel** link, then select **Manage** tab. In the **Web Hosting Account Details** section you will see your account FTP settings.


Yahoo! Small Business - Web Hosting - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://us.1.p4.webhosting.yahoo.com/cp/manage
Google
Done

important information about your account. Do we have your correct email address? Find out now and learn more about your default email address.

Web Hosting Account Details

Site Details
Web addresses:
yourdomain.com
www.yourdomain.com

Subdomains:
(no subdomains have been created)

Manage subdomains

Service Plan Details
Plan: Web Hosting
Change service

FTP Settings
Upload your files more quickly.
Learn how to use FTP.

host: ftp.yourdomain.com
alt. host: ftp.p4.webhosting.yahoo.com
user ID: Yahoo! ID
password: Yahoo! password

Promotions
Get more domains. Learn more.

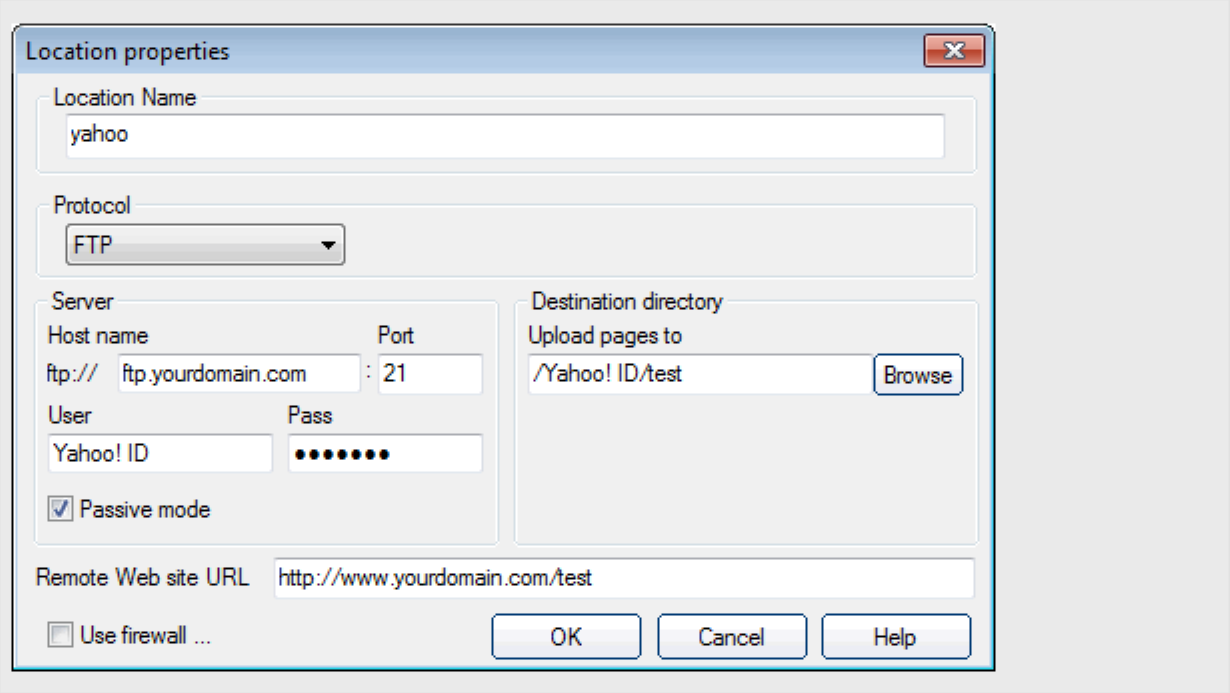
Protect Your Privacy

Do you understand your domain registration? View your registration information to learn which details are public and how services like Private Domain Registration can help guard against identity theft.

Copyright © 2009 Yahoo! Inc. All rights reserved
Privacy Policy - Copyright Policy - Terms of Service - Help

Return to the PHPRunner and enter Host name, User ID and Password. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK.**



Location properties

Location Name: yahoo

Protocol: FTP

Server

Host name: ftp:// ftp.yourdomain.com

Port: 21

User: Yahoo! ID

Pass: •••••••

☑ Passive mode

Destination directory

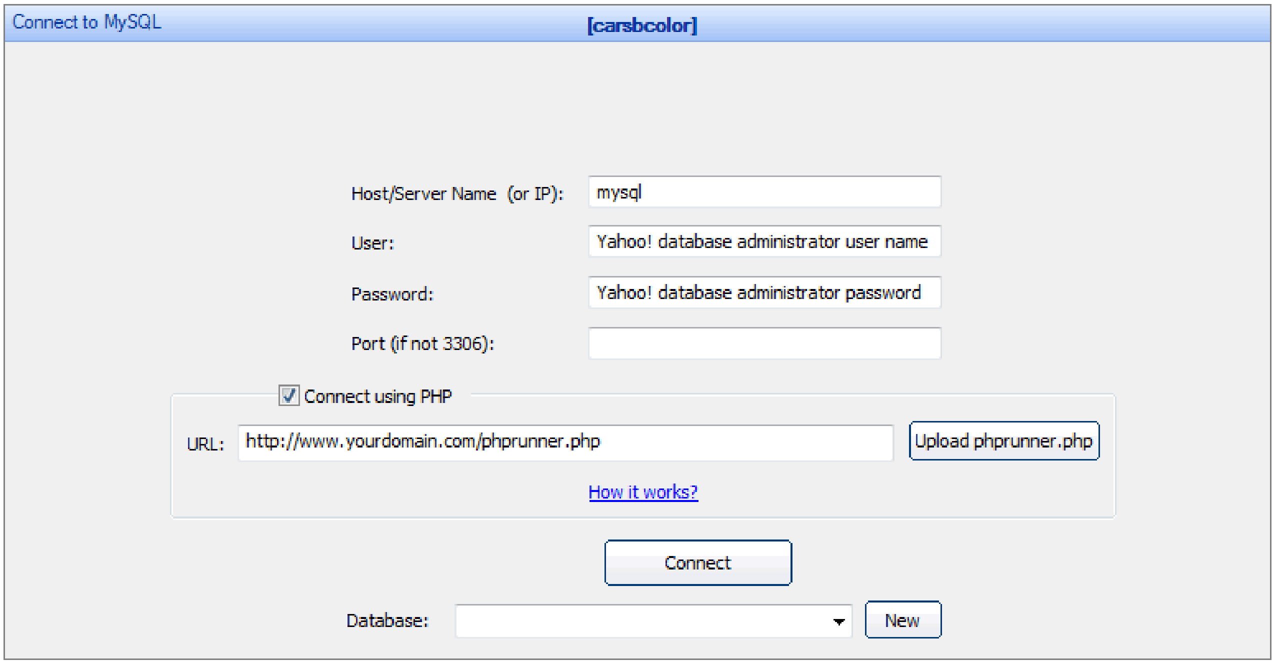
Upload pages to: /Yahoo! ID/test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

7. Click **Upload!** button. Wait while PHPRunner uploads phprunner.php file to your Yahoo! account, then click **Close**.

8. Click **Connect**.



Connect to MySQL [carsbcolor]

Host/Server Name (or IP): mysql

User: Yahoo! database administrator user name

Password: Yahoo! database administrator password

Port (if not 3306):

☑ Connect using PHP

URL: http://www.yourdomain.com/phprunner.php

Database:

9. To create a new database click **New**. After database is created, you will proceed to datasource tables page.

If you receive an error while creating a database, you should do that in phpMyAdmin. So proceed to the next step.

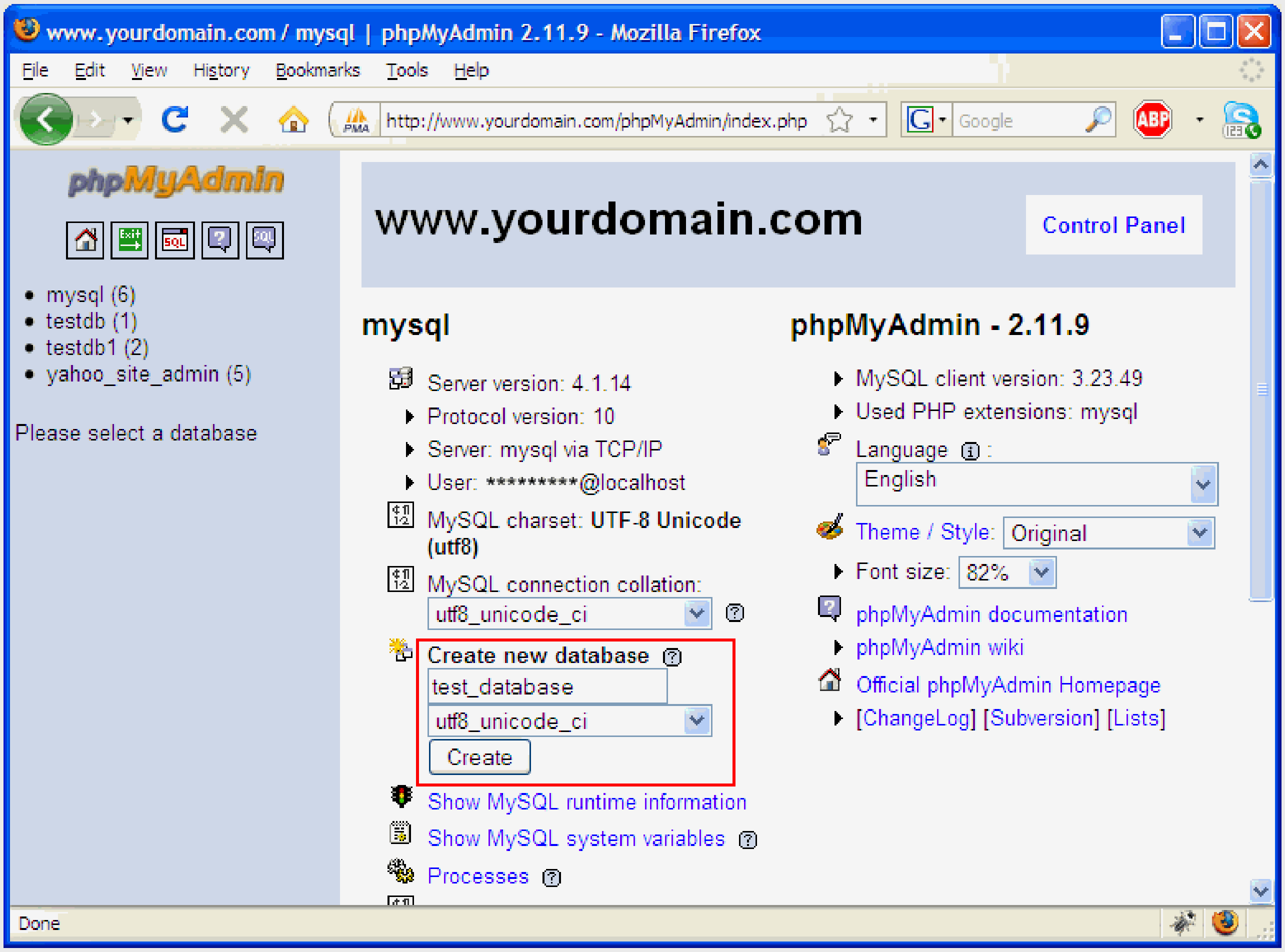
Step 4. Create a new MySQL database in phpMyAdmin

How to create new MySQL database in phpMyAdmin

1. Go to your phpMyAdmin homepage. You can access the tool by visiting your phpMyAdmin home page at http://www.yourdomain.com/subdirectory/, replacing yourdomain.com with your own domain name and subdirectory with the name of the site folder in which your phpMyAdmin files are installed.

2. Enter a database administrator user name and password. Click **Go**.

3. Enter a database name and select the Character Sets and Collations under the **Create new database**. Click **Create**.



www.yourdomain.com / mysql | phpMyAdmin 2.11.9 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.yourdomain.com/phpMyAdmin/index.php
Google
phpMyAdmin
Control Panel
Done

www.yourdomain.com

- mysql (6)
- testdb (1)
- testdb1 (2)
- yahoo_site_admin (5)

Please select a database

mysql

Server version: 4.1.14
Protocol version: 10
Server: mysql via TCP/IP
User: **********@localhost
MySQL charset: **UTF-8 Unicode (utf8)**
MySQL connection collation: utf8_unicode_ci
Create new database: test_database, utf8_unicode_ci, Create
Show MySQL runtime information
Show MySQL system variables
Processes

phpMyAdmin - 2.11.9

MySQL client version: 3.23.49
Used PHP extensions: mysql
Language: English
Theme / Style: Original
Font size: 82%
phpMyAdmin documentation
phpMyAdmin wiki
Official phpMyAdmin Homepage
[ChangeLog] [Subversion] [Lists]

4. Return to the PHPRunner and click **Connect**.

5. Select your database and click **Next**>>.

Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

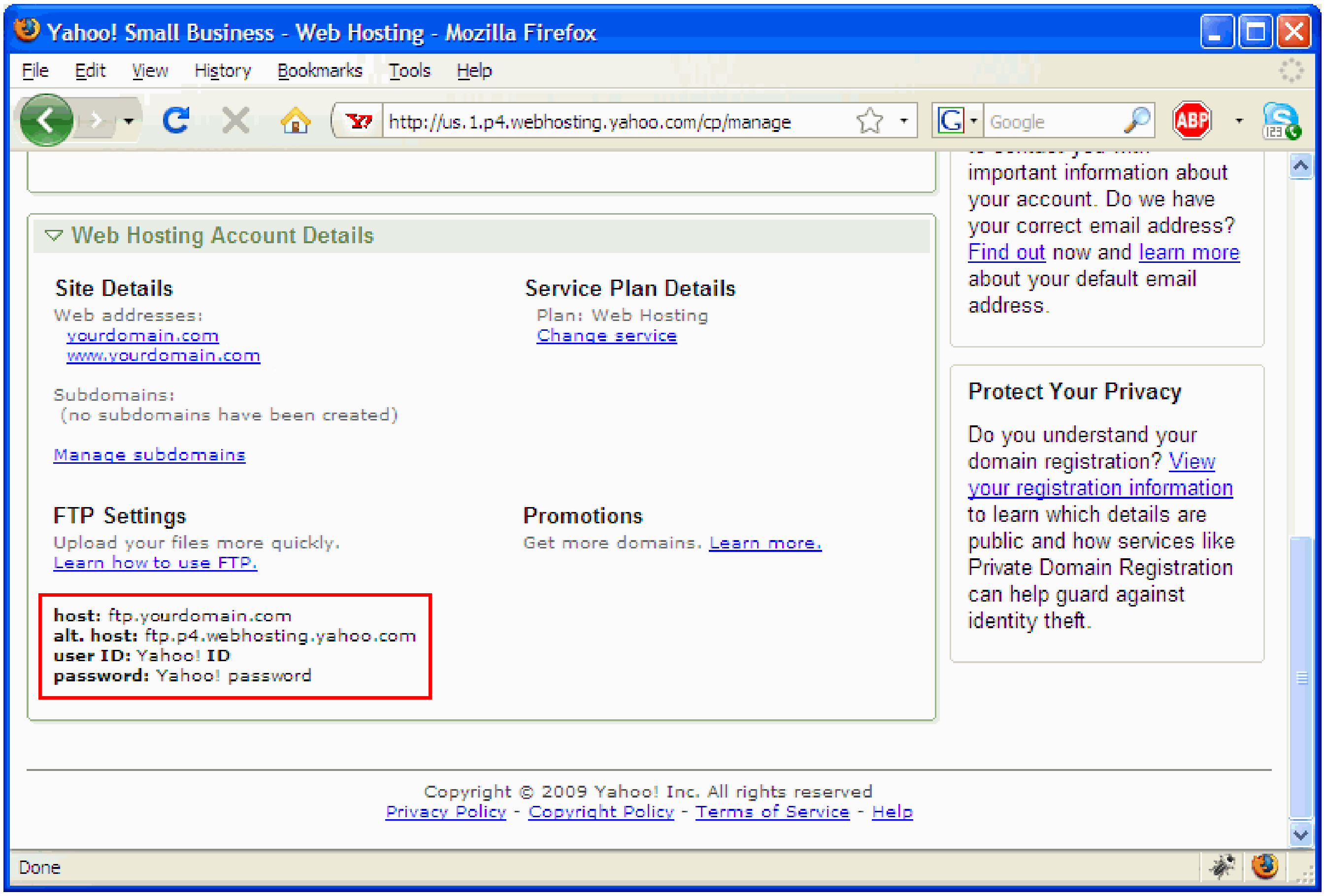
3.6.1.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your Yahoo! account, follow the instructions below.

1. Log in to your account at webhosting.yahoo.com.

2. Click the **Web Hosting Control Panel** link, then select **Manage** tab.

3. In the **Web Hosting Account Details** section you will see your account FTP settings.


Yahoo! Small Business - Web Hosting - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://us.1.p4.webhosting.yahoo.com/cp/manage
Google
Done

important information about your account. Do we have your correct email address? Find out now and learn more about your default email address.

Web Hosting Account Details

Site Details

Web addresses:
yourdomain.com
www.yourdomain.com

Subdomains:
(no subdomains have been created)

Manage subdomains

Service Plan Details

Plan: Web Hosting
Change service

FTP Settings

Upload your files more quickly.
Learn how to use FTP.

host: ftp.yourdomain.com
alt. host: ftp.p4.webhosting.yahoo.com
user ID: Yahoo! ID
password: Yahoo! password

Promotions

Get more domains. Learn more.

Protect Your Privacy

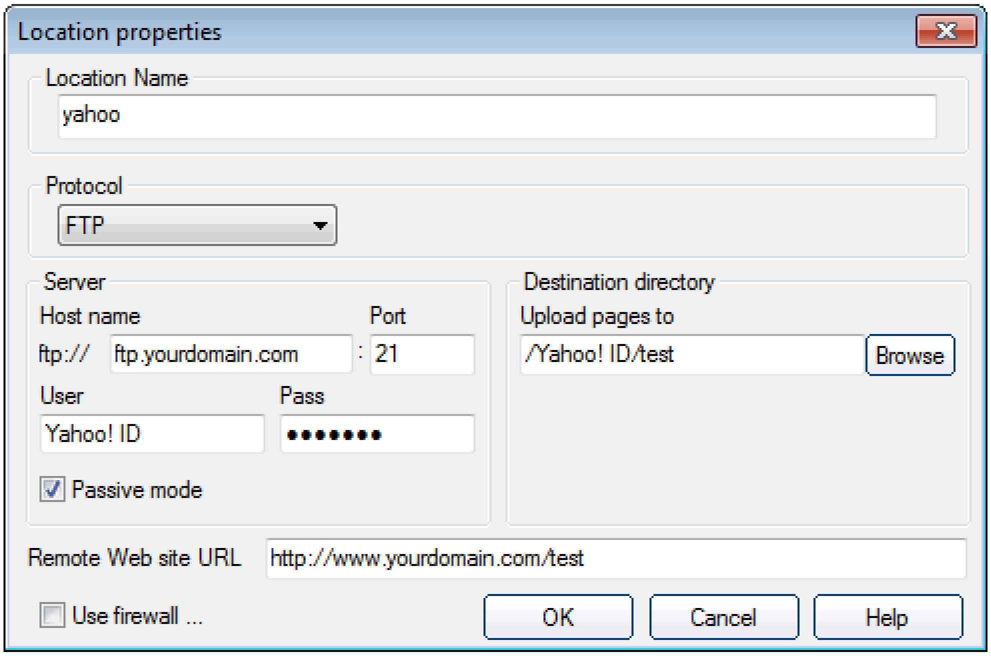
Do you understand your domain registration? View your registration information to learn which details are public and how services like Private Domain Registration can help guard against identity theft.

Copyright © 2009 Yahoo! Inc. All rights reserved
Privacy Policy - Copyright Policy - Terms of Service - Help

4. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

5. Enter Host name, User ID and Password. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.



Location properties

Location Name: yahoo

Protocol: FTP

Server

Host name: ftp:// ftp.yourdomain.com

Port: 21

User: Yahoo! ID

Pass: •••••••

☑ Passive mode

Destination directory

Upload pages to: /Yahoo! ID/test

Remote Web site URL: http://www.yourdomain.com/test

□ Use firewall ...

6. Click **OK**.

Congratulations! FTP client is now configured to upload files to your Yahoo! account.

3.6.1.3 Publishing project via FTP

To publish generated project files to your Yahoo! hosting account using built into PHPRunner FTP client, follow the instructions below.

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

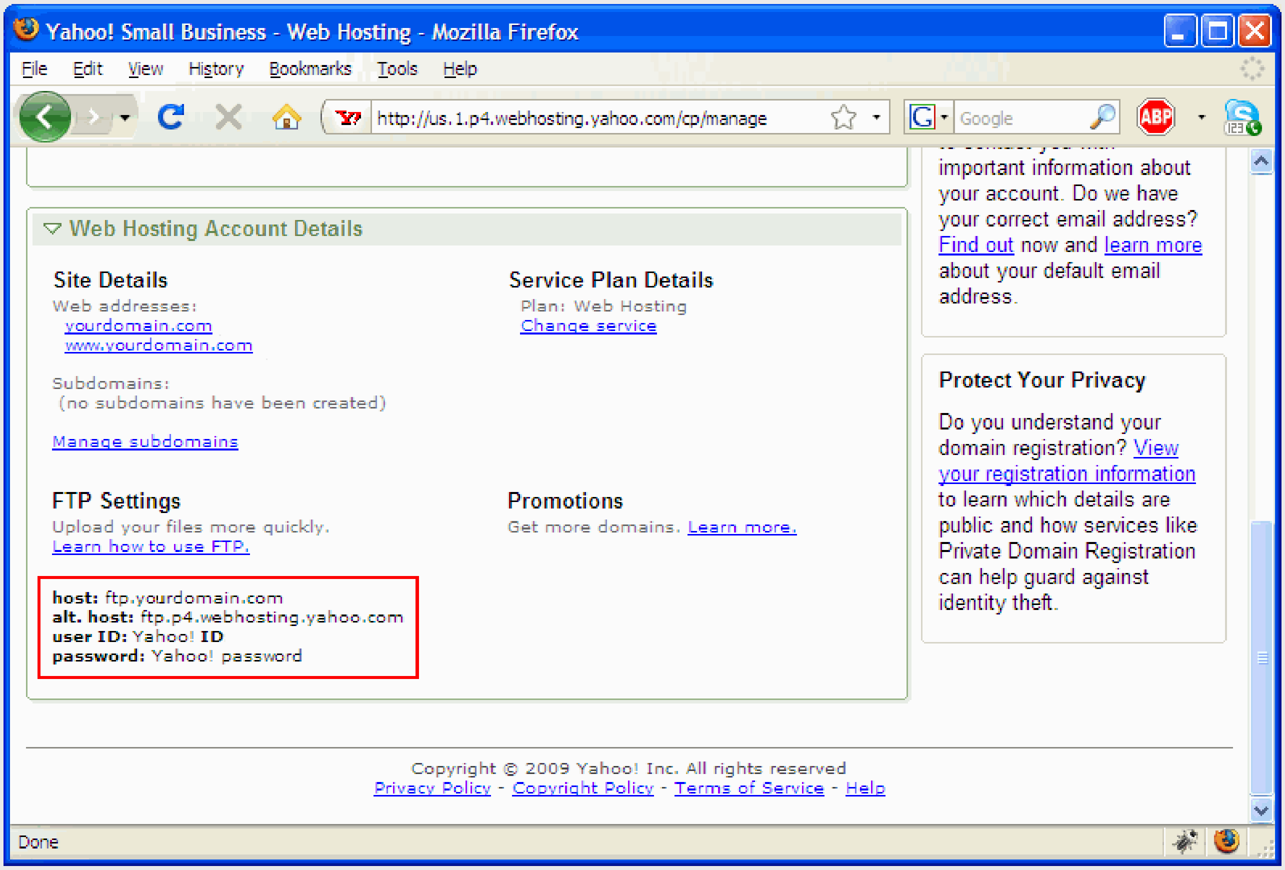
3. Click **Publish via FTP**. Built-in FTP client opens.

4. If you have already configured FTP client, choose FTP location and click **Properties**. Click **Browse** to select a directory to upload your project files to. Update the **Remote web site URL** textbox accordingly to selected site folder. E.g. if you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Click **OK**.

If you have not configured FTP client yet, follow the instructions below.

How to configure FTP

To obtain yout FTP settings, log in to your account at webhosting.yahoo.com. Click the **Web Hosting Control Panel** link, then select **Manage** tab. In the **Web Hosting Account Details** section you will see your account FTP settings.


Yahoo! Small Business - Web Hosting - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://us.1.p4.webhosting.yahoo.com/cp/manage
Google
Done

important information about your account. Do we have your correct email address? Find out now and learn more about your default email address.

Web Hosting Account Details

Site Details
Web addresses:
yourdomain.com
www.yourdomain.com

Subdomains:
(no subdomains have been created)

Manage subdomains

Service Plan Details
Plan: Web Hosting
Change service

FTP Settings
Upload your files more quickly.
Learn how to use FTP.

host: ftp.yourdomain.com
alt. host: ftp.p4.webhosting.yahoo.com
user ID: Yahoo! **ID**
password: Yahoo! password

Promotions
Get more domains. Learn more.

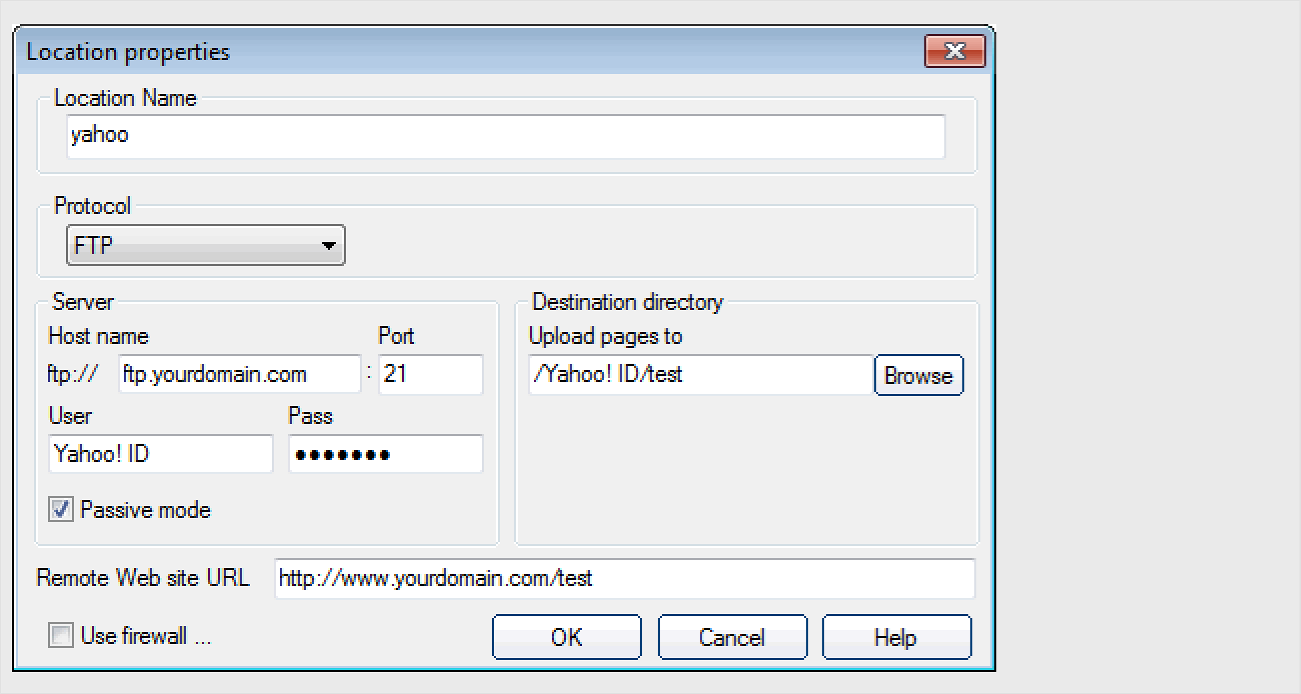
Protect Your Privacy

Do you understand your domain registration? View your registration information to learn which details are public and how services like Private Domain Registration can help guard against identity theft.

Copyright © 2009 Yahoo! Inc. All rights reserved
Privacy Policy - Copyright Policy - Terms of Service - Help

Return to the PHPRunner and enter Host name, User ID and Password. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



Location Name: yahoo

Protocol: FTP

Host name: ftp:// ftp.yourdomain.com : Port: 21

User: Yahoo! ID

Pass: •••••••

☑ Passive mode

Upload pages to: /Yahoo! ID/test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your Yahoo! account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your Yahoo! account.

3.6.2 1&1

3.6.2.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your 1&1 account, follow the instructions below.

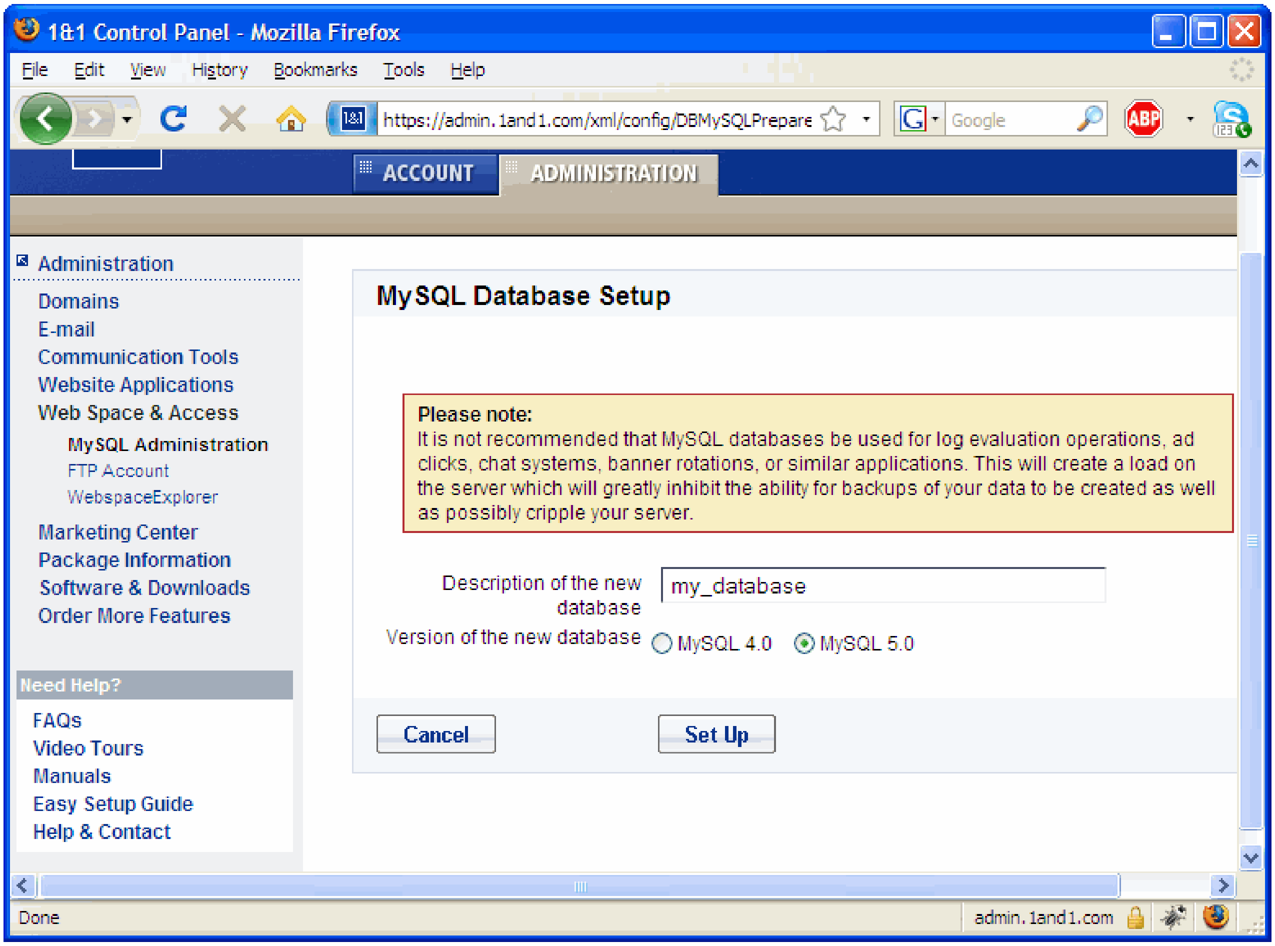
Step 1. Set up a database

1. Log in to the 1&1 Control Panel at https://admin.1and1.com. If you have only one package, you will land on the Administration page. If you have more than one package, select the relevant package to reach its Administration page.

2. Select **MySQL Administration** from the **Web Space & Access** section.

3. If you haven't set up any databases click **New Database** button.

4. Enter a description, select **MySQL 5.0** and click **Set Up**.



MySQL Database Setup

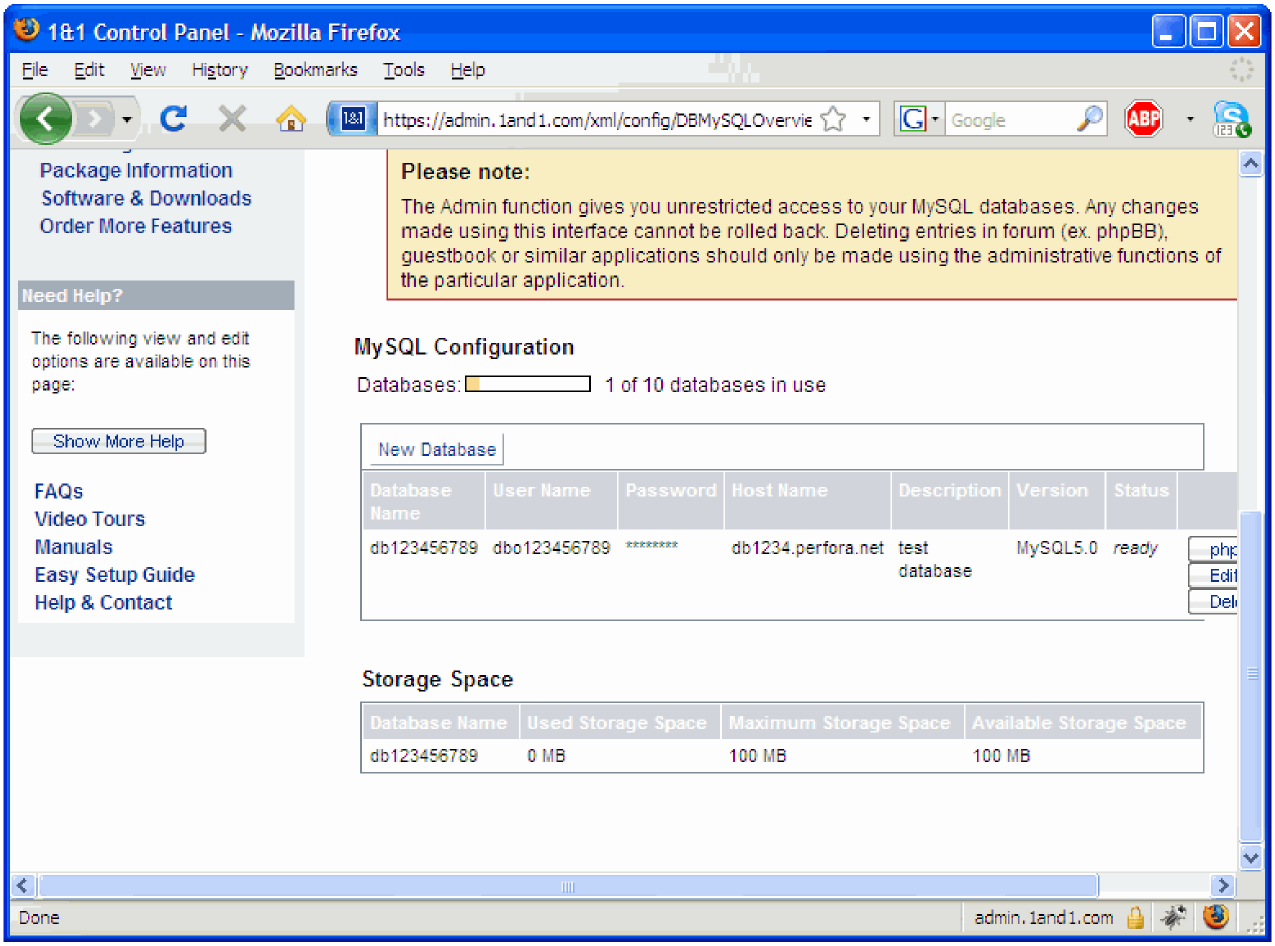
Please note:
It is not recommended that MySQL databases be used for log evaluation operations, ad clicks, chat systems, banner rotations, or similar applications. This will create a load on the server which will greatly inhibit the ability for backups of your data to be created as well as possibly cripple your server.

Description of the new database: my_database

Version of the new database: ○ MySQL 4.0 ● MySQL 5.0

5. Your MySQL database details are displayed. Click **Go To Overview**.

6. The complete MySQL database information is displayed. You will use this information while connecting to MySQL database from PHPRunner. It may take some time to set up your database. When the status of your database became **ready**, you can go forward and connect to your database from PHPRunner.



The following view and edit options are available on this page:

Please note:

The Admin function gives you unrestricted access to your MySQL databases. Any changes made using this interface cannot be rolled back. Deleting entries in forum (ex. phpBB), guestbook or similar applications should only be made using the administrative functions of the particular application.

MySQL Configuration

Databases: 1 of 10 databases in use

Database Name	User Name	Password	Host Name	Description	Version	Status
db123456789	dbo123456789	********	db1234.perfora.net	test database	MySQL5.0	*ready*

Storage Space

Database Name	Used Storage Space	Maximum Storage Space	Available Storage Space
db123456789	0 MB	100 MB	100 MB

Step 2. Connect to a MySQL database

1. Start PHPRunner.

2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

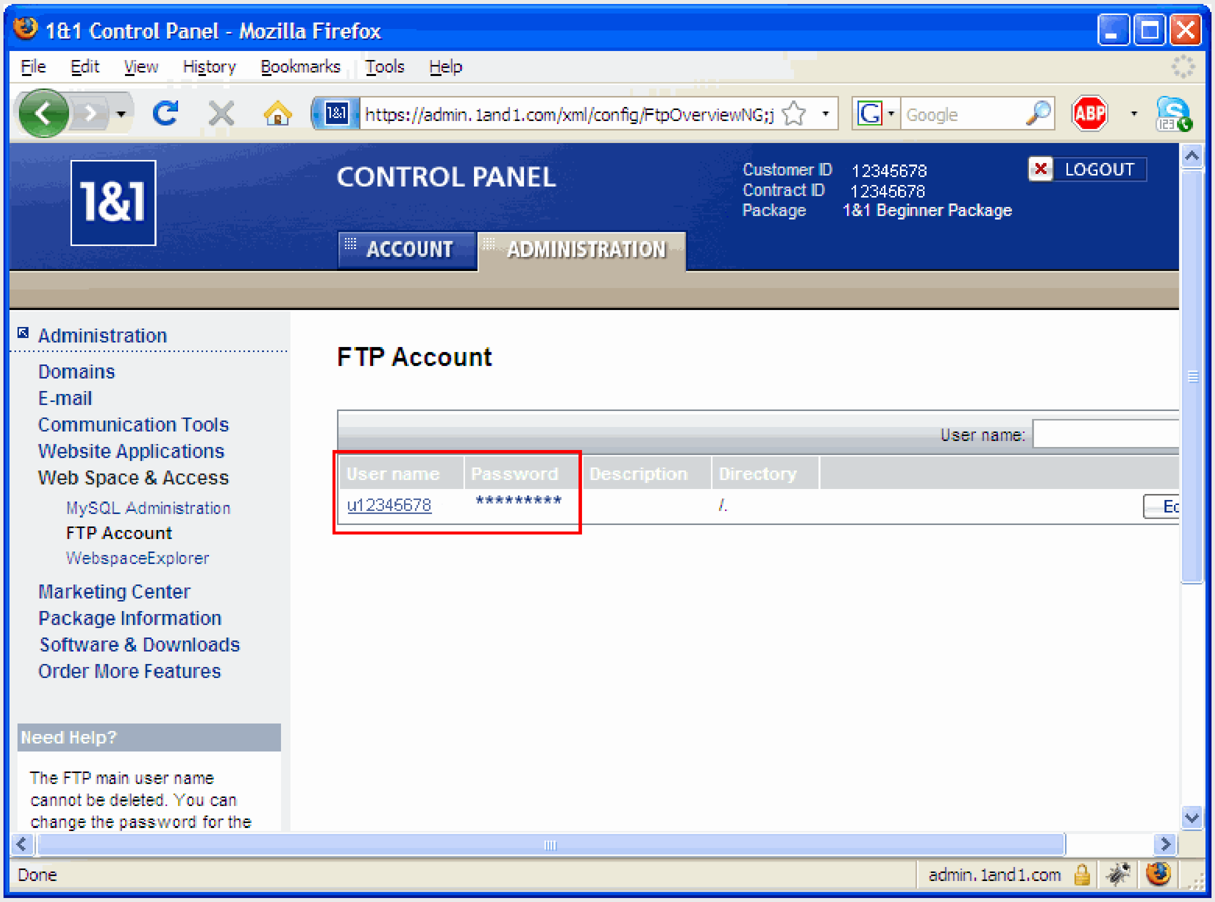
4. Enter a host name, user name and password indicated for your database in the 1&1 Control panel.

5. Select the **Connect using PHP** check box and click **Upload phprunner.php**.

6. Configure built-in FTP client.

How to configure FTP

To obtain yout FTP settings, log in to the 1&1 Control Panel at https://admin.1and1.com. If you have only one package, you will land on the Administration page. If you have more than one package, select the relevant package to reach its Administration page. Select **FTP Account** from the **Web Space & Access** section. You can see the FTP settings for your account.



Customer ID 12345678
Contract ID 12345678
Package 1&1 Beginner Package

FTP Account

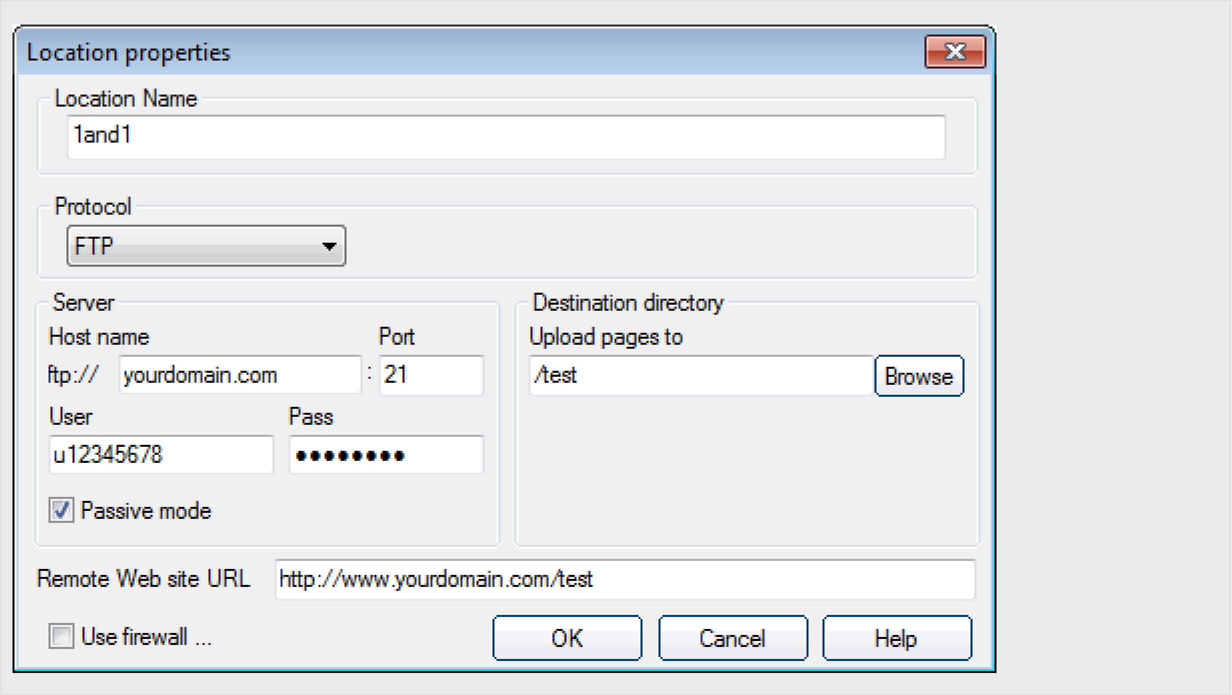
User name:

User name	Password	Description	Directory
u12345678	*********		/.

The FTP main user name cannot be deleted. You can change the password for the

Return to the PHPRunner and type **yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password indicated for your FTP account in the 1&1 Control panel. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.


Location properties

Location Name: 1and1

Protocol: FTP

Server
Host name: ftp:// yourdomain.com
Port: 21
User: u12345678
Pass: ●●●●●●●●●
☑ Passive mode

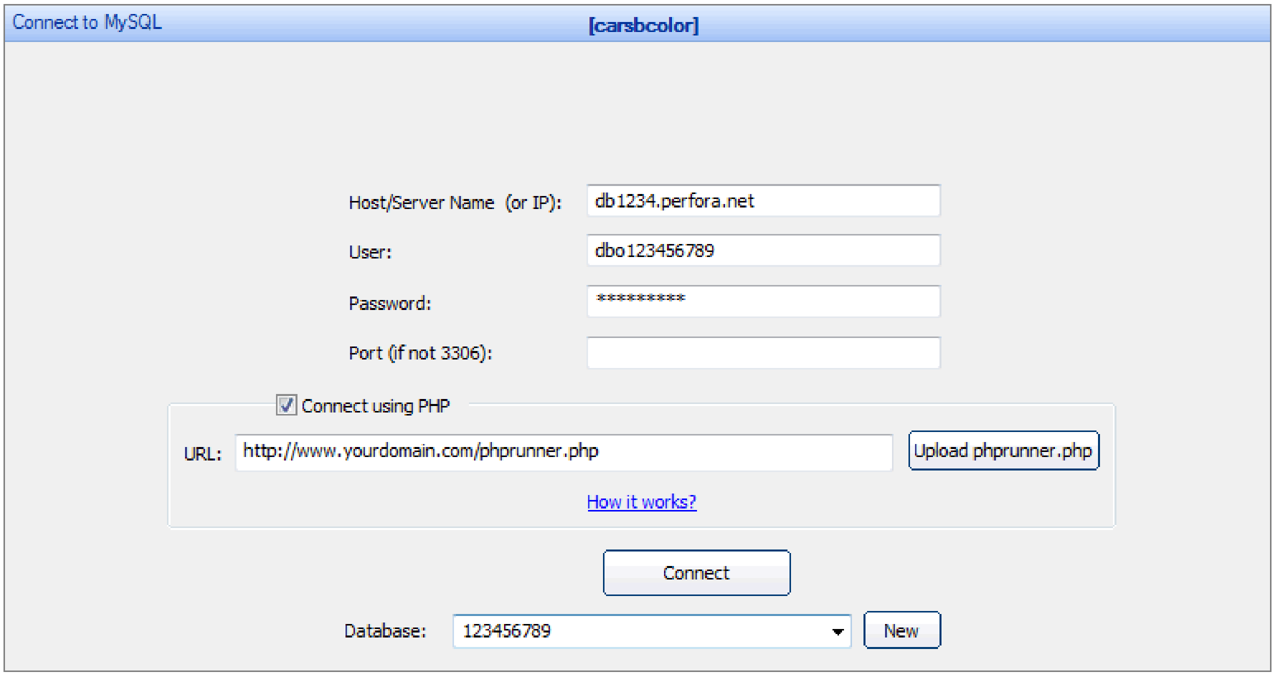
Destination directory
Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

7. Click **Upload!** button. Wait while PHPRunner uploads phprunner.php file to your 1&1 account, then click **Close**.

8. Click **Connect**. If only one database is stored at the indicated server, you will proceed to datasource tables page. If there are several databases on the server, you need to select a database and click **Next**>>.


Connect to MySQL [carsbcolor]

Host/Server Name (or IP): db1234.perfora.net
User: dbo123456789
Password: **********
Port (if not 3306):

☑ Connect using PHP
URL: http://www.yourdomain.com/phprunner.php

Database: 123456789

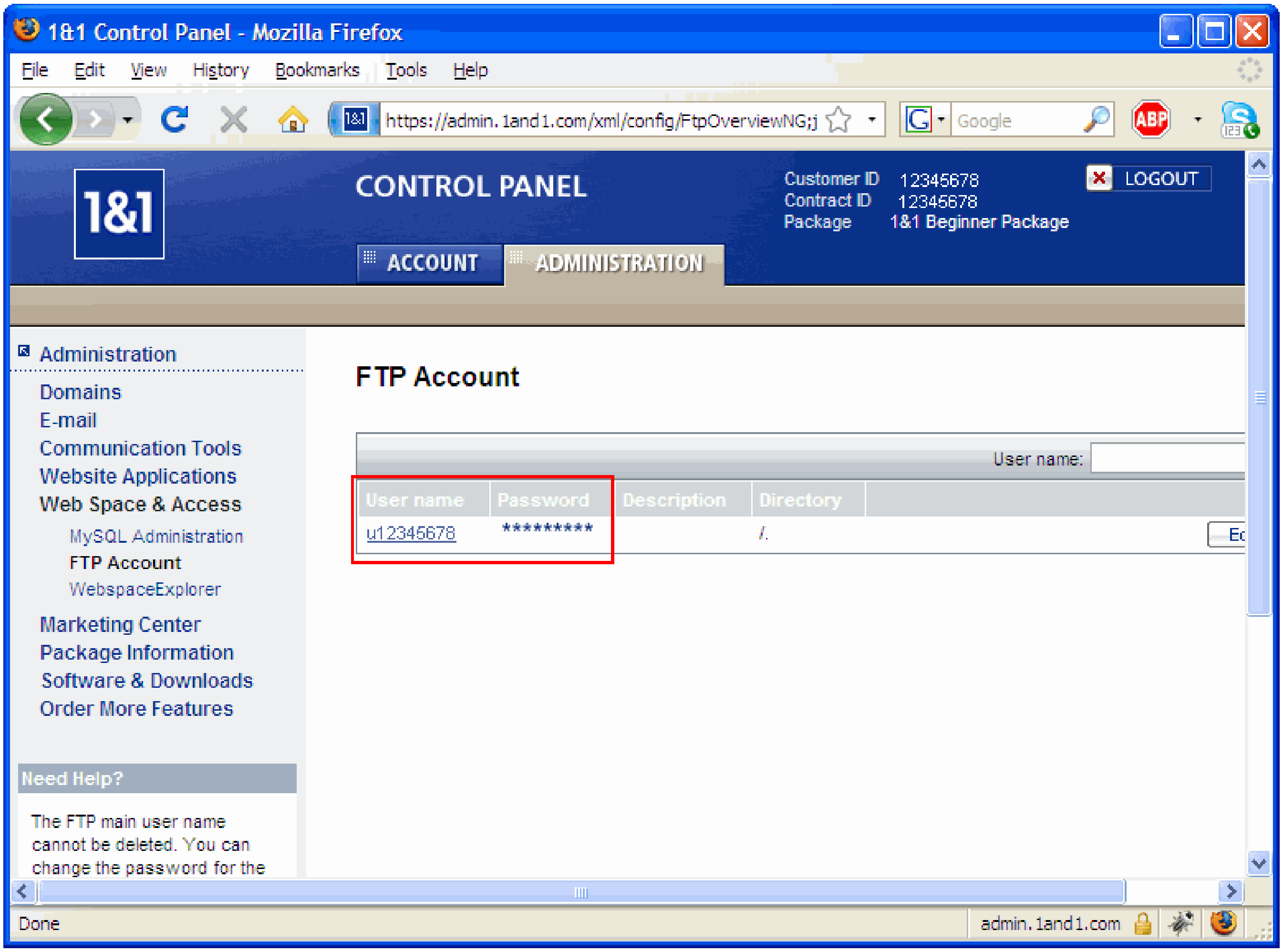
Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

3.6.2.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your 1&1 account, follow the instructions below.

1. Log in to the 1&1 Control Panel at https://admin.1and1.com. If you have only one package, you will land on the Administration page. If you have more than one package, select the relevant package to reach its Administration page.

2. Select **FTP Account** from the **Web Space & Access** section. You can see the FTP settings for your account. You will use this information later.



Customer ID 12345678
Contract ID 12345678
Package 1&1 Beginner Package

FTP Account

User name:

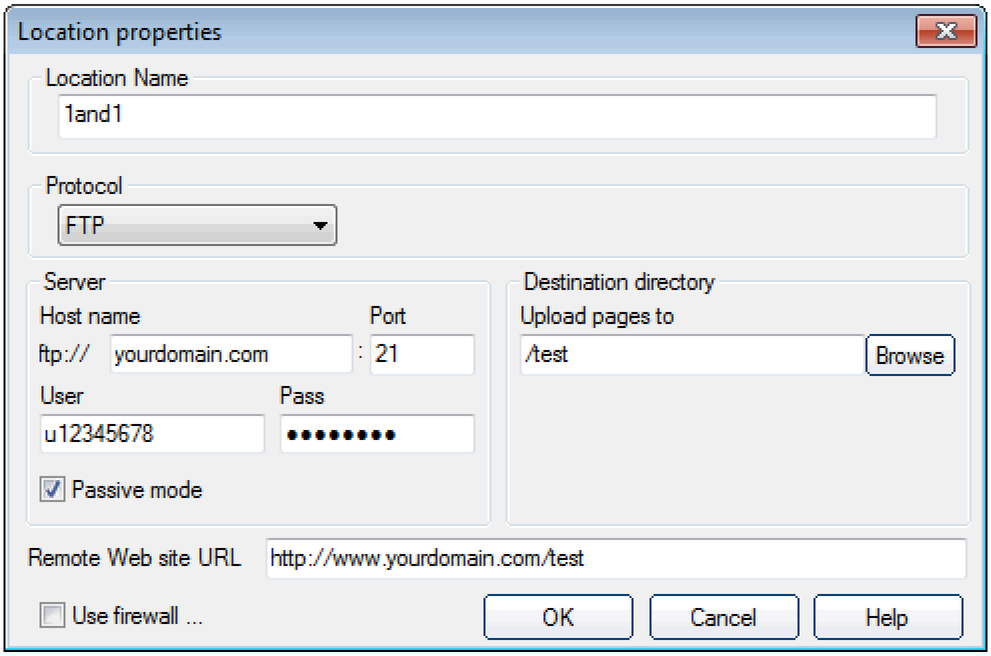
User name	Password	Description	Directory
u12345678	*********		/.

The FTP main user name cannot be deleted. You can change the password for the

3. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

4. Type **yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password indicated for your FTP account in the 1&1 Control panel. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.


Location properties

Location Name: 1and1

Protocol: FTP

Server
Host name: ftp:// yourdomain.com
Port: 21
User: u12345678
Pass: ••••••••
☑ Passive mode

Destination directory
Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

5. Click **OK**.

Congratulations! FTP client is now configured to upload files to your 1&1 account.

3.6.2.3 Publishing project via FTP

To publish generated project files to your 1&1 hosting account using built into PHPRunner FTP client, follow the instructions below.

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Publish via FTP**. Built-in FTP client opens.

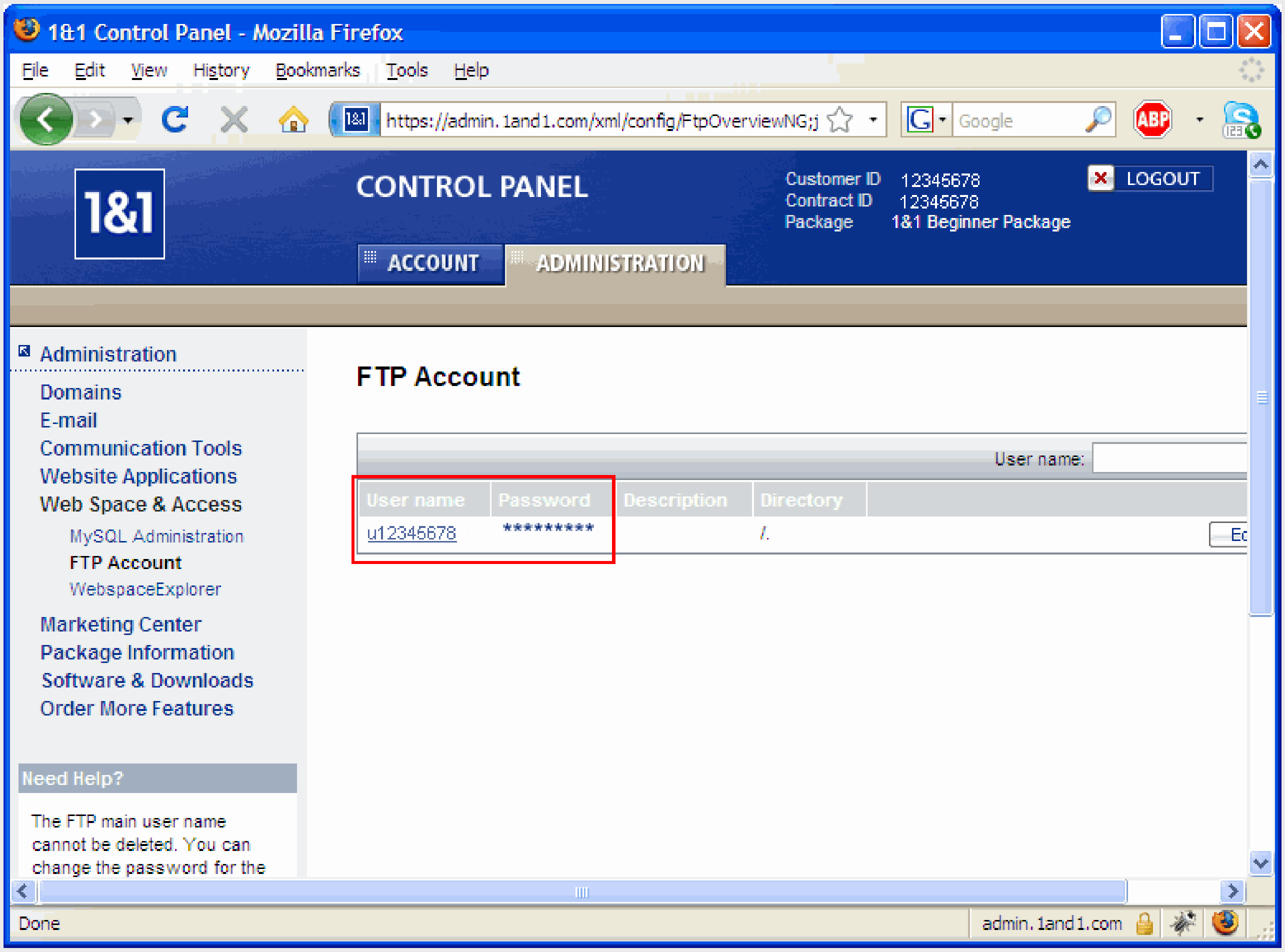
4. If you have already configured FTP client, choose FTP location and click **Properties**. Click **Browse** to select a directory to upload your project files to. Update the **Remote web site URL**

textbox accordingly to selected site folder. E.g. if you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Click **OK**.

If you have not configured FTP client yet, follow the instructions below.

How to configure FTP

To obtain yout FTP settings, log in to the 1&1 Control Panel at https://admin.1and1.com. If you have only one package, you will land on the Administration page. If you have more than one package, select the relevant package to reach its Administration page. Select **FTP Account** from the **Web Space & Access** section. You can see the FTP settings for your account.



Customer ID 12345678
Contract ID 12345678
Package 1&1 Beginner Package

FTP Account

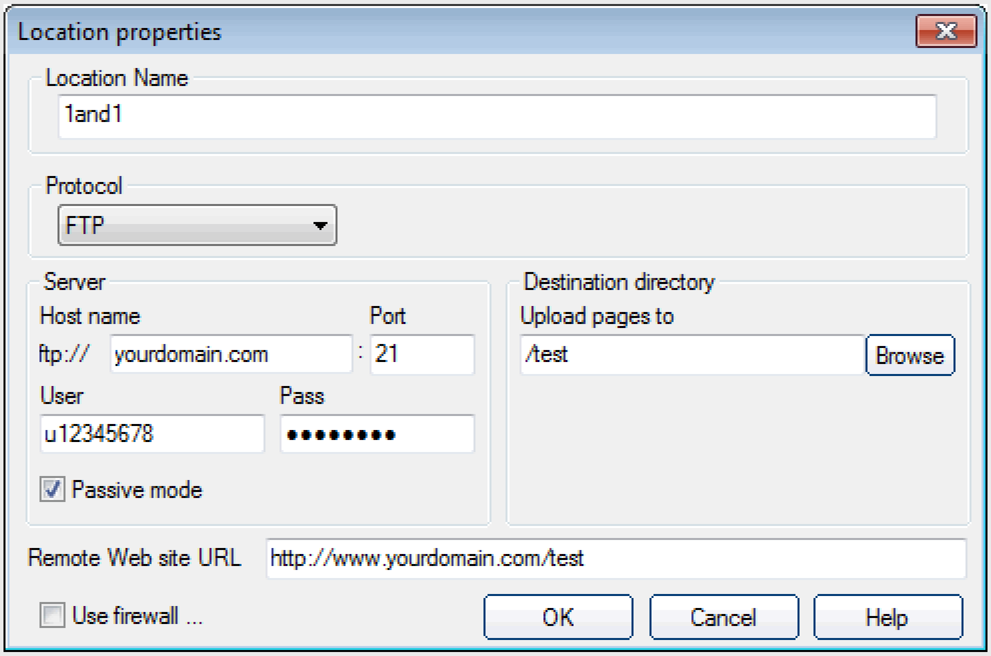
User name:

User name	Password	Description	Directory
u12345678	**********		/.

The FTP main user name cannot be deleted. You can change the password for the

Return to the PHPRunner and type **yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password indicated for your FTP account in the 1&1 Control panel. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



Location properties

Location Name: 1and1

Protocol: FTP

Server

Host name: ftp:// yourdomain.com Port: 21

User: u12345678 Pass: •••••••••

☑ Passive mode

Destination directory

Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your 1&1 account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your 1&1 account.

3.6.3 GoDaddy.com

3.6.3.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your GoDaddy account, follow the instructions below.

Step 1. Set up a MySQL Database

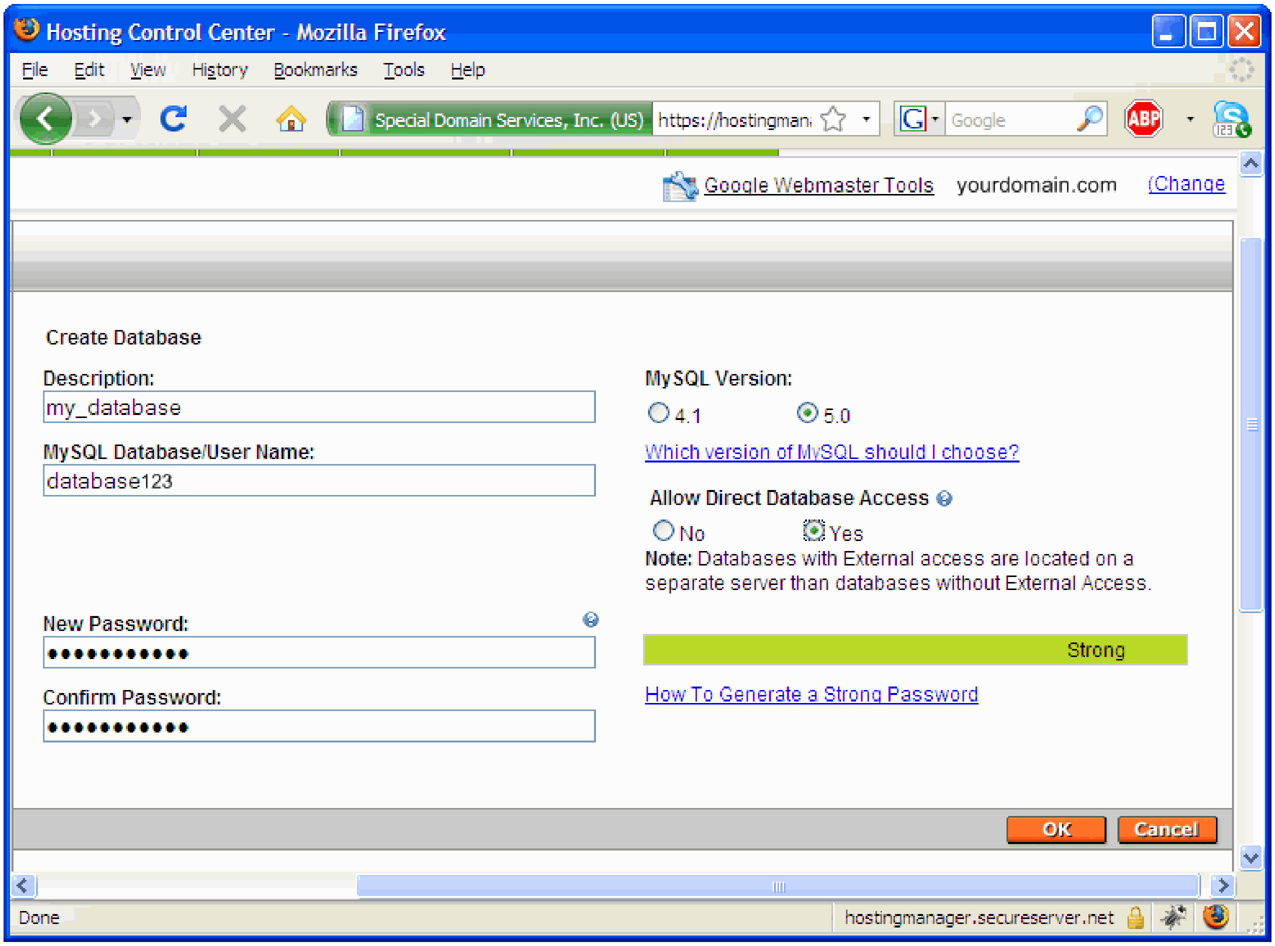
1. Log in to your GoDaddy Hosting Control Center at https://hostingmanager.secureserver.net/Login.aspx.

2. Click the domain name you want to work with.

3. In the **Databases** section, click the **MySQL** icon.

4. Click **Create Database**.

5. Enter a database description, database name (that will also serve as a user name) and password. Select 5.0 as MySQL version. Select **Yes** under **Allow Direct Database Access**.



Create Database

Description: my_database

MySQL Database/User Name: database123

New Password: ●●●●●●●●●●●

Confirm Password: ●●●●●●●●●●●

MySQL Version: ○ 4.1 ● 5.0

Which version of MySQL should I choose?

Allow Direct Database Access

○ No ● Yes

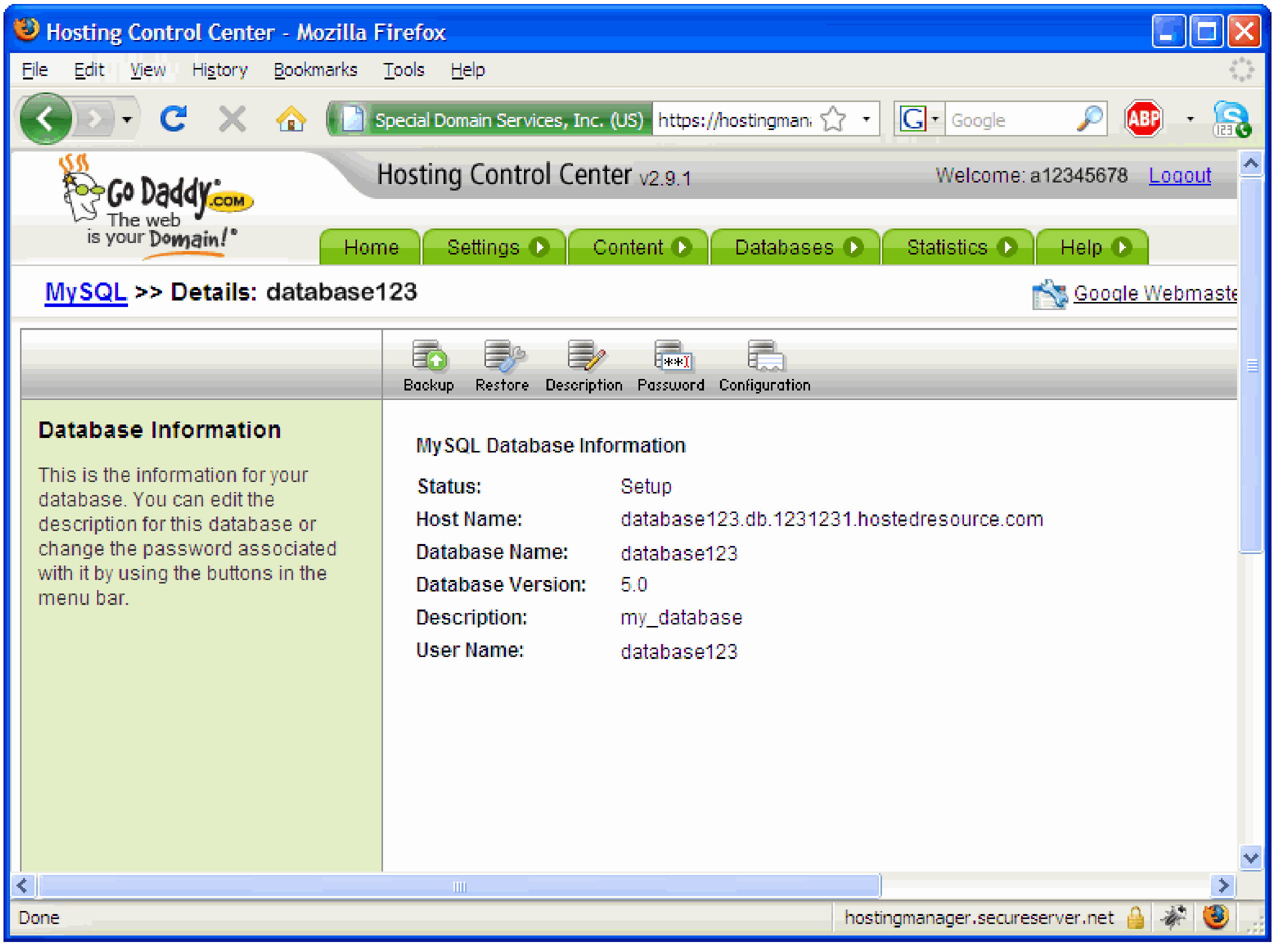
Note: Databases with External access are located on a separate server than databases without External Access.

Strong

How To Generate a Strong Password

6. Click **OK**. It may take a few minutes to create a database. Click your browser's **Refresh** button to check if your database is ready.

7. Once you have set up a database for your account, click the **Pencil** icon next to the name of your database. The complete MySQL database information will be displayed. You will use this information while connecting to MySQL database from PHPRunner.


Hosting Control Center - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Special Domain Services, Inc. (US) https://hostingman
Google
Go Daddy .com The web is your Domain!
Hosting Control Center v2.9.1
Welcome: a12345678 Logout
Home Settings Content Databases Statistics Help
Google Webmaste
Backup Restore Description Password Configuration
Done
hostingmanager.secureserver.net

MySQL >> Details: database123

Database Information

This is the information for your database. You can edit the description for this database or change the password associated with it by using the buttons in the menu bar.

MySQL Database Information

Status:	Setup
Host Name:	database123.db.1231231.hostedresource.com
Database Name:	database123
Database Version:	5.0
Description:	my_database
User Name:	database123

Step 2. Connect to a MySQL database

1. Start PHPRunner.

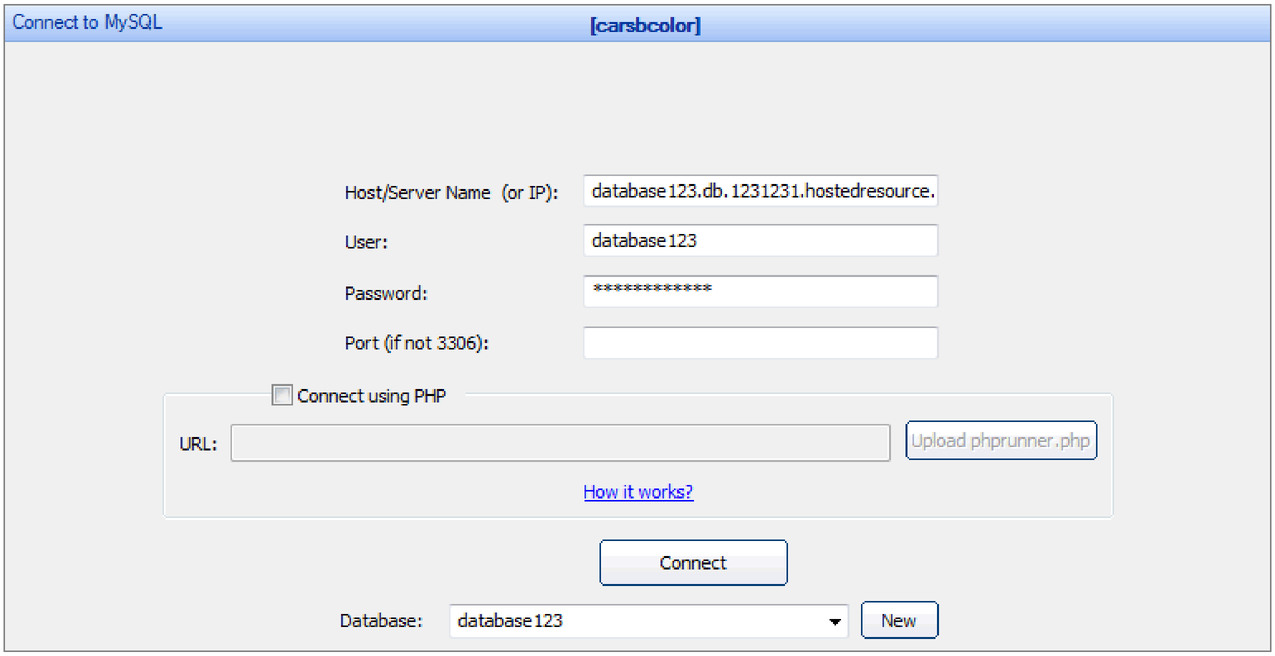
2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

4. Enter a Host Name, User Name and Password indicated for your database in the GoDaddy Hosting Control Center.

5. Click **Connect**.

6. Select a database and click **Next**>>.



Host/Server Name (or IP): database123.db.1231231.hostedresource.

User: database123

Password: *************

Port (if not 3306):

□ Connect using PHP

URL:

Database: database123

Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

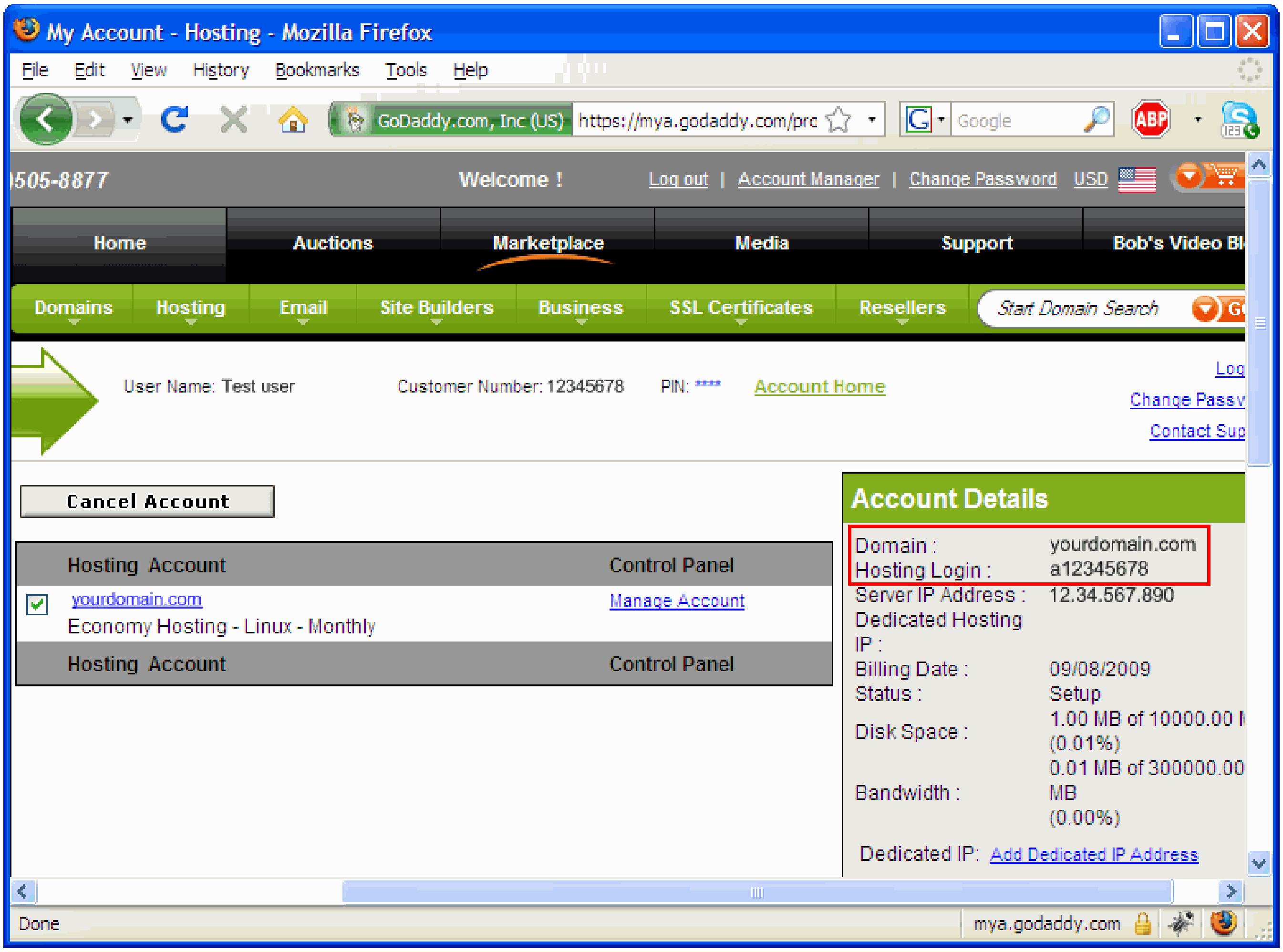
3.6.3.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your GoDaddy account, follow the instructions below.

1. This is the FTP information you will need to use:

- **FTP User Name** - the user name for your hosting account.
- **FTP Password** - the password for your hosting account.
- **FTP Host Name** - ftp://www.yourdomain.com

If you are missing any of this information, log in to your GoDaddy Account Manager at http://mya.godaddy.com/default.aspx?prog_id=GoDaddy. In the **My Products** section, click **Hosting**. In your **Hosting Account** list, click the name of a hosting account you want to work with. In the **Account Details** section, your **Hosting Login** displays.


My Account - Hosting - Mozilla Firefox
File Edit View History Bookmarks Tools Help
GoDaddy.com, Inc (US) https://mya.godaddy.com/prc
Google
)505-8877
Welcome !
Log out | Account Manager | Change Password USD
Home Auctions Marketplace Media Support Bob's Video Bl
Domains Hosting Email Site Builders Business SSL Certificates Resellers
Start Domain Search
Log
Change Passv
Contact Sup
Cancel Account
Done
mya.godaddy.com

User Name: Test user Customer Number: 12345678 PIN: **** Account Home

Hosting Account	Control Panel
☑ yourdomain.com Economy Hosting - Linux - Monthly	Manage Account
Hosting Account	Control Panel

Account Details

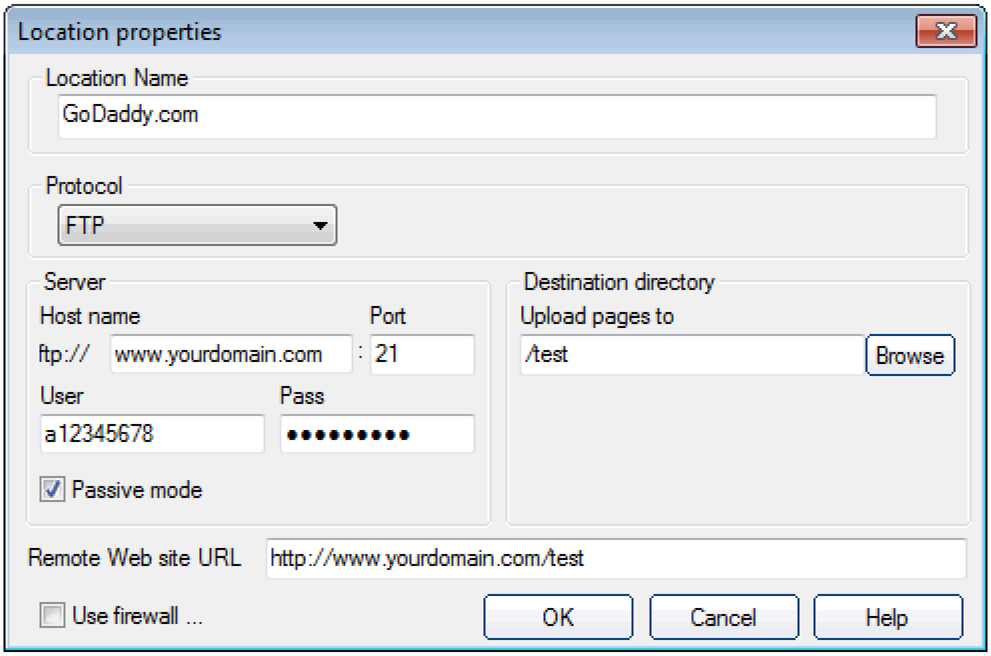
Domain :	yourdomain.com
Hosting Login :	a12345678
Server IP Address :	12.34.567.890
Dedicated Hosting IP :	
Billing Date :	09/08/2009
Status :	Setup
Disk Space :	1.00 MB of 10000.00 I (0.01%)
Bandwidth :	0.01 MB of 300000.00 MB (0.00%)

Dedicated IP: Add Dedicated IP Address

2. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

3. Type **www.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your hosting account. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.



Location properties

Location Name: GoDaddy.com

Protocol: FTP

Server

Host name: ftp:// www.yourdomain.com

Port: 21

User: a12345678

Pass: ●●●●●●●●●●

☑ Passive mode

Destination directory

Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

4. Click **OK**.

Congratulations! FTP client is now configured to upload files to your GoDaddy account.

3.6.3.3 Publishing project via FTP

To publish generated project files to your GoDaddy hosting account using built into PHPRunner FTP client, follow the instructions below.

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Publish via FTP**. Built-in FTP client opens.

4. If you have already configured FTP client, choose FTP location and click **Properties**. Click **Browse** to select a directory to upload your project files to. Update the **Remote web site URL** textbox accordingly to selected site folder. E.g. if you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Click **OK**.

If you have not configured FTP client yet, follow the instructions below.

How to configure FTP

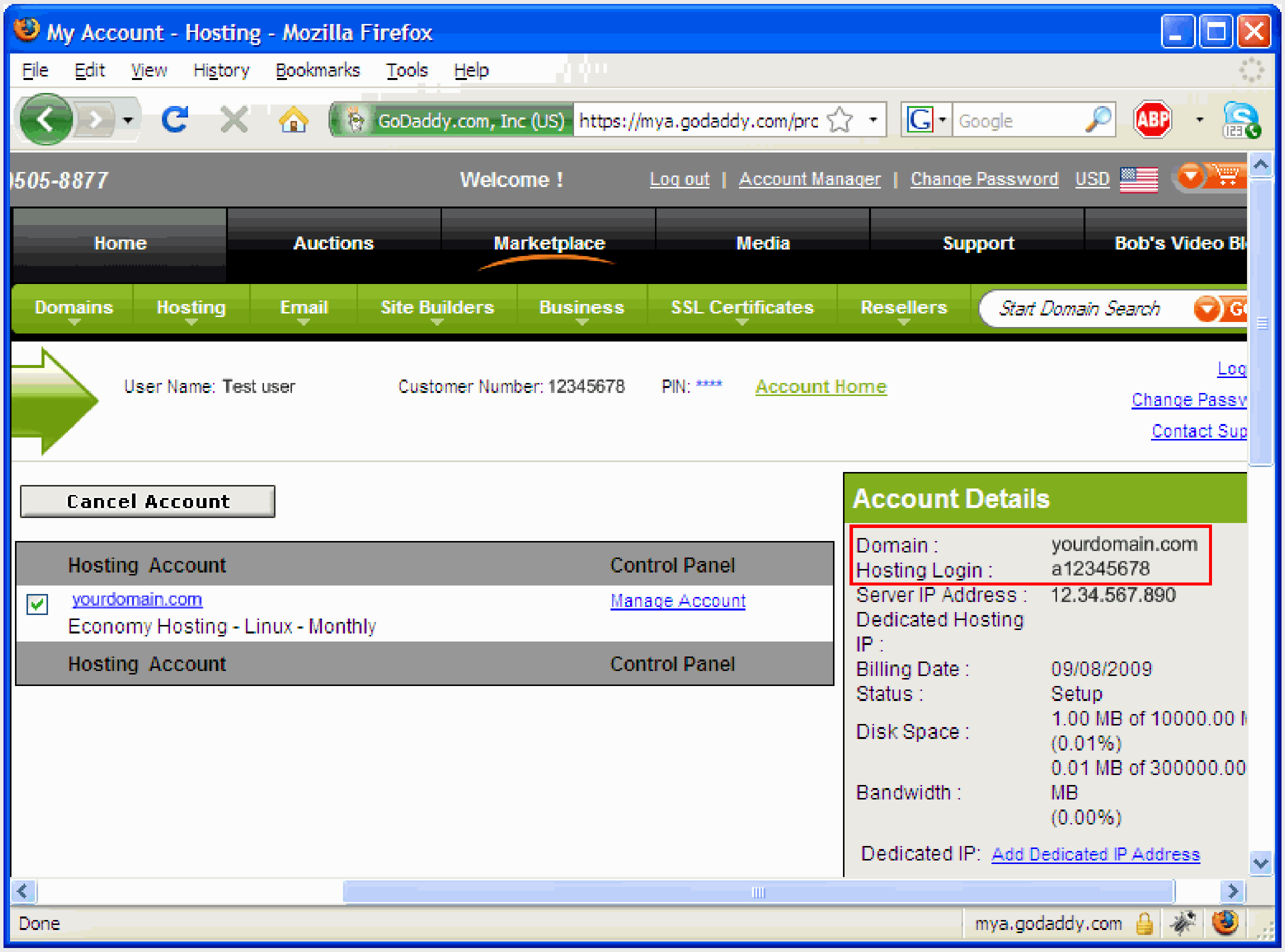
This is the FTP information you will need to use:

- **FTP User Name** - the user name for your hosting account.

- **FTP Password** - your password for your hosting account.

- **FTP Host Name** - ftp://www.yourdomain.com

If you are missing any of this information, log in to your GoDaddy Account Manager at http://mya.godaddy.com/default.aspx?prog_id=GoDaddy. In the **My Products** section, click **Hosting**. In your **Hosting Account** list, click the name of a hosting account you want to work with. In the **Account Details** section, your **Hosting Login** displays.



User Name: Test user Customer Number: 12345678 PIN: ****

Hosting Account	Control Panel
☑ yourdomain.com Economy Hosting - Linux - Monthly	Manage Account
Hosting Account	Control Panel

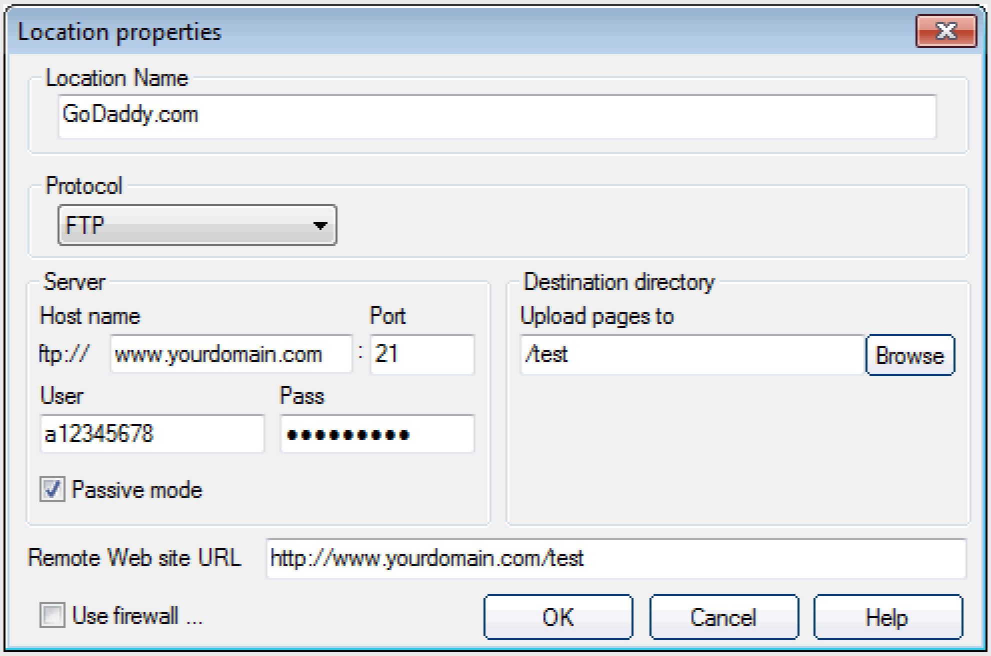
Account Details

Domain :	yourdomain.com
Hosting Login :	a12345678
Server IP Address :	12.34.567.890
Dedicated Hosting IP :	
Billing Date :	09/08/2009
Status :	Setup
Disk Space :	1.00 MB of 10000.00 M (0.01%)
Bandwidth :	0.01 MB of 300000.00 MB (0.00%)

Dedicated IP: Add Dedicated IP Address

Return to the PHPRunner and type **www.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your hosting account. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



Location Name: GoDaddy.com

Protocol: FTP

Host name: ftp:// www.yourdomain.com Port: 21

User: a12345678 Pass: ••••••••••

☑ Passive mode

Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your GoDaddy account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your GoDaddy account.

3.6.4 WebHost4Life.com

3.6.4.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your WebHost4Life account, follow the instructions below.

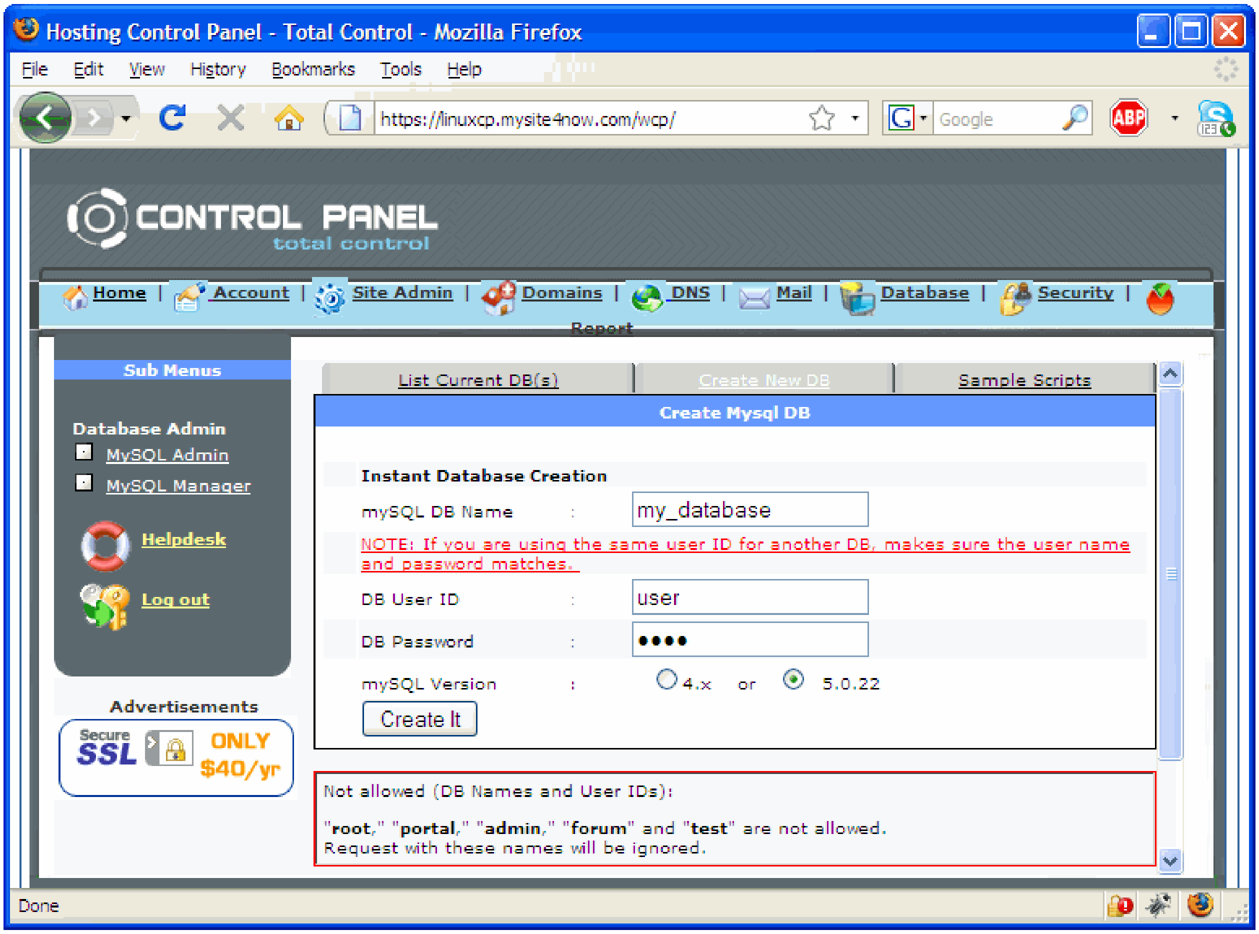
Step 1. Set up a MySQL Database

1. Log in to the Hosting Control Panel at https://linuxcp.mysite4now.com/wcp/ for Linux hosting or https://controlpanel.mysite4now.com/wcp/ for Windows hosting.

2. Click **Database**.

3. Click **Create new DB**.

4. Enter a database name, database user name and password. Select 5.0.22 as MySQL version.


Hosting Control Panel - Total Control - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://linuxcp.mysite4now.com/wcp/
Google
CONTROL PANEL
total control
Home | Account | Site Admin | Domains | DNS | Mail | Database | Security | Report
Sub Menus
Database Admin
MySQL Admin
MySQL Manager
Helpdesk
Log out
Advertisements
Secure SSL ONLY $40/yr
List Current DB(s)
Create New DB
Sample Scripts
Done

Create Mysql DB

Instant Database Creation

mySQL DB Name	:	my_database

NOTE: If you are using the same user ID for another DB, makes sure the user name and password matches.

DB User ID	:	user
DB Password	:	••••
mySQL Version	:	○ 4.x or ● 5.0.22

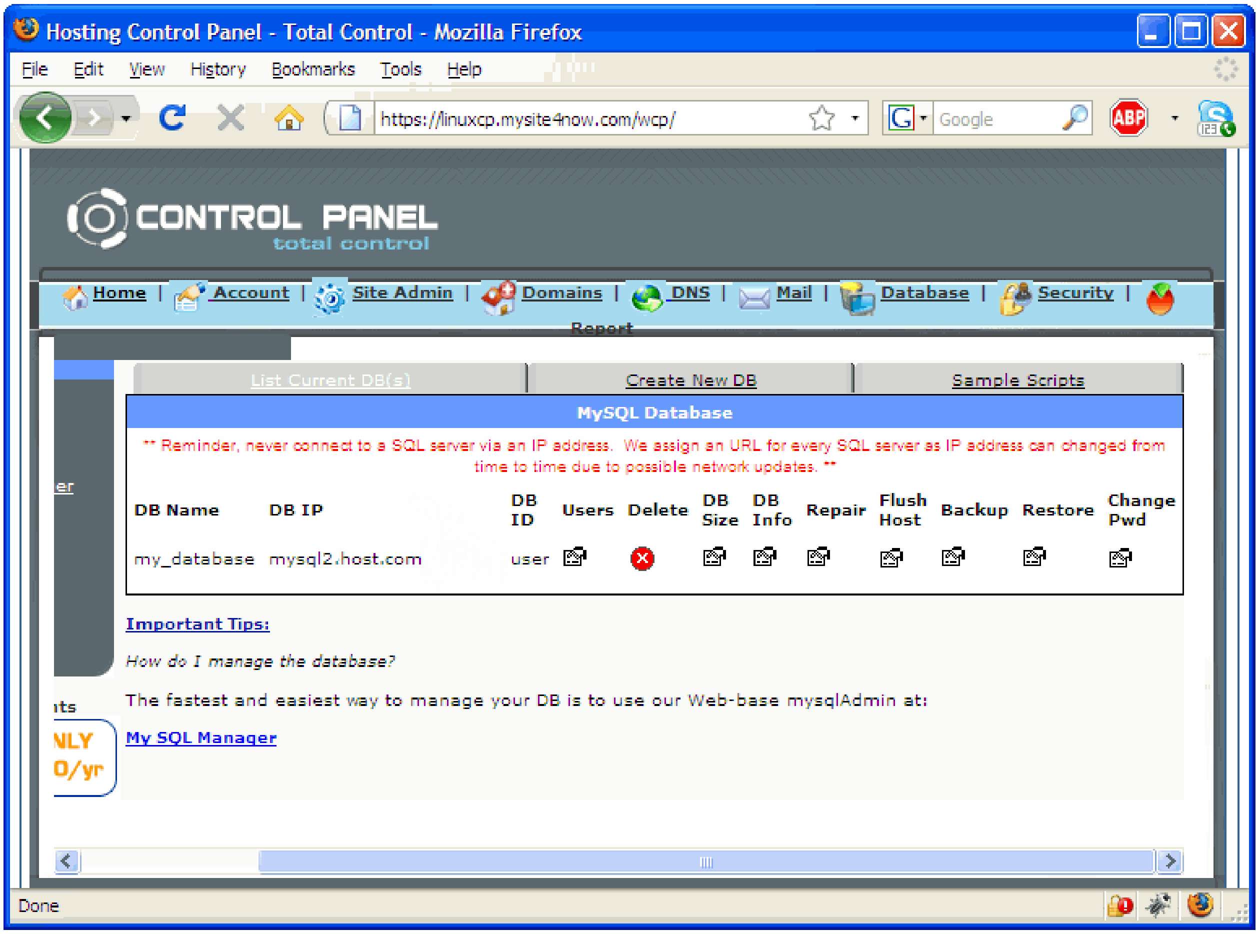
Create It

Not allowed (DB Names and User IDs):

"**root**," "**portal**," "**admin**," "**forum**" and "**test**" are not allowed.
Request with these names will be ignored.

5. Click **Create It**.

6. Once your database is set up, you will see the complete MySQL database information. You will use this information while connecting to MySQL database from PHPRunner.


Hosting Control Panel - Total Control - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://linuxcp.mysite4now.com/wcp/
Google
CONTROL PANEL
total control
Home | Account | Site Admin | Domains | DNS | Mail | Database | Security |
Report
List Current DB(s)
Create New DB
Sample Scripts
er
ıts
NLY
0/yr
Done

MySQL Database

** Reminder, never connect to a SQL server via an IP address. We assign an URL for every SQL server as IP address can changed from time to time due to possible network updates. **

DB Name	DB IP	DB ID	Users	Delete	DB Size	DB Info	Repair	Flush Host	Backup	Restore	Change Pwd
my_database	mysql2.host.com	user	✎	✖	✎	✎	✎	✎	✎	✎	✎

Important Tips:

How do I manage the database?

The fastest and easiest way to manage your DB is to use our Web-base mysqlAdmin at:

My SQL Manager

Step 2. Connect to a MySQL database

1. Start PHPRunner.

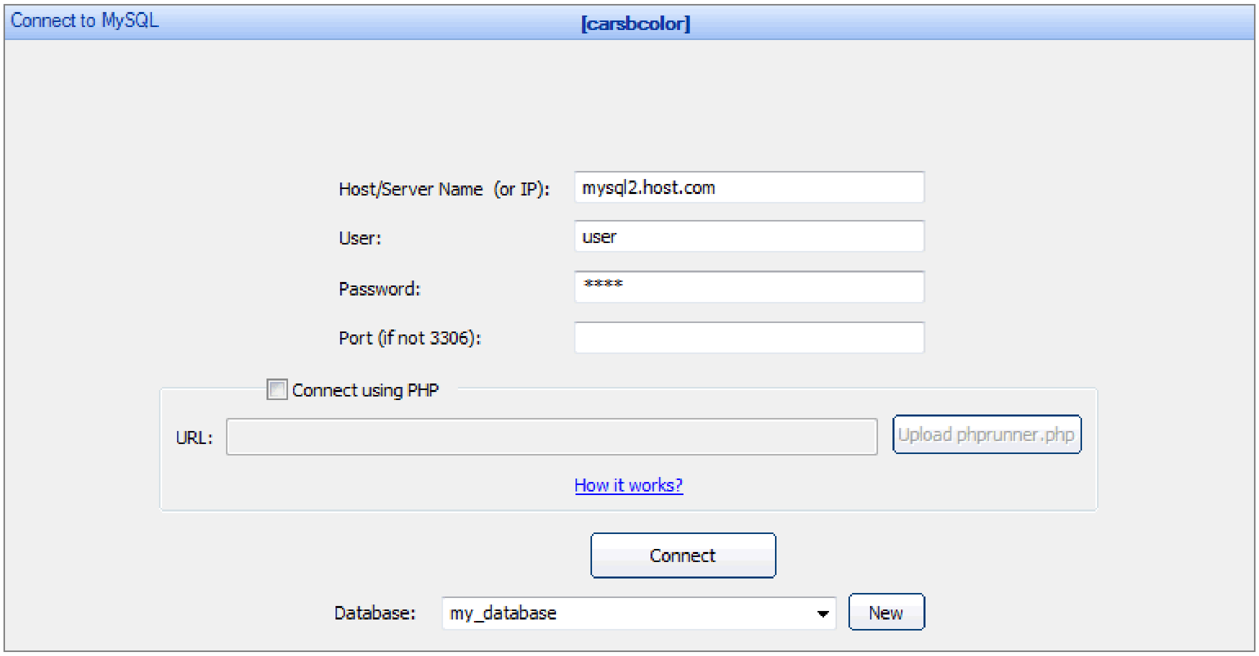
2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

4. Enter a Host Name, User Name and Password indicated for your database in the Hosting Control Panel.

5. Click **Connect**.

6. Select a database and click **Next**>>.



Host/Server Name (or IP): mysql2.host.com

User: user

Password: *****

Port (if not 3306):

□ Connect using PHP

URL:

Database: my_database

Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

3.6.4.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your WebHost4Life account, follow the instructions below.

1. This is the FTP information you will need to use:

- **FTP User Name** - your account ID.
- **FTP Password** - your account password.
- **FTP Host Name** - ftp.yourdomain.com

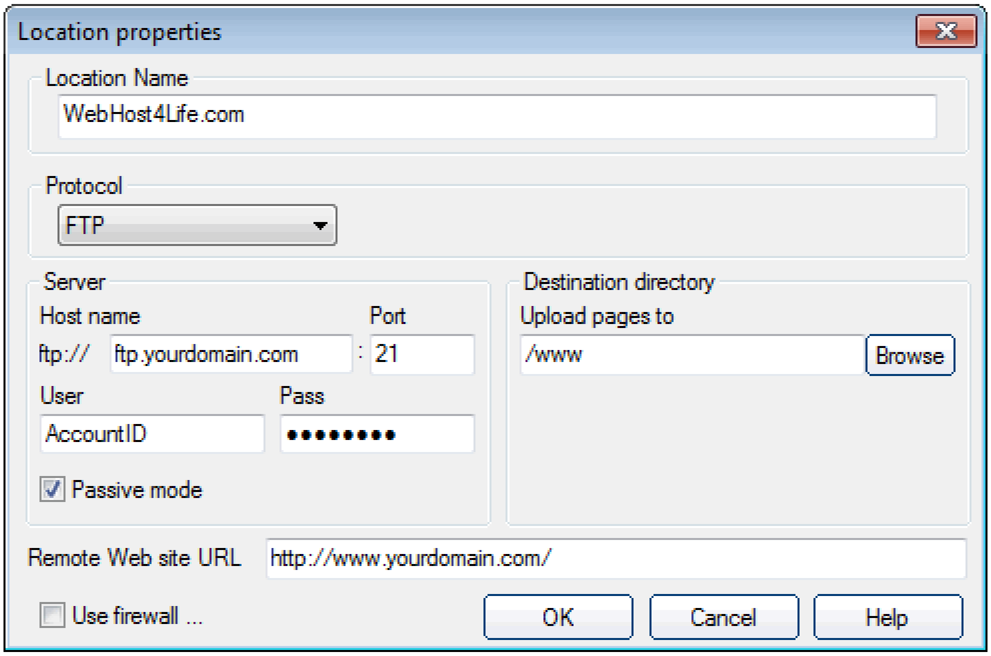
If you are missing any of this information, log in to the Hosting Control Panel at https://linuxcp.mysite4now.com/wcp/ for Linux hosting or https://controlpanel.mysite4now.com/wcp/ for Windows hosting.

2. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

3. Type **ftp.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your WebHost4Life account. Then click **Browse** to

select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.



WebHost4Life.com

FTP

ftp:// ftp.yourdomain.com : 21

AccountID ●●●●●●●●●
☑ Passive mode

/www

Remote Web site URL http://www.yourdomain.com/
□ Use firewall ...

4. Click **OK.**

Congratulations! FTP client is now configured to upload files to your WebHost4Life account.

3.6.4.3 Publishing project via FTP

To publish generated project files to your WebHost4Life hosting account using built into PHPRunner FTP client, follow the instructions below.

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Publish via FTP**. Built-in FTP client opens.

4. If you have already configured FTP client, choose FTP location and click **Properties**. Click **Browse** and choose the **www** site folder. In the **Remote web site URL** textbox type **http://www.yourdomain.com/**, replacing yourdomain.com with your domain name. Click **OK.**

If you have not configured FTP client yet, follow the instructions below.

How to configure FTP

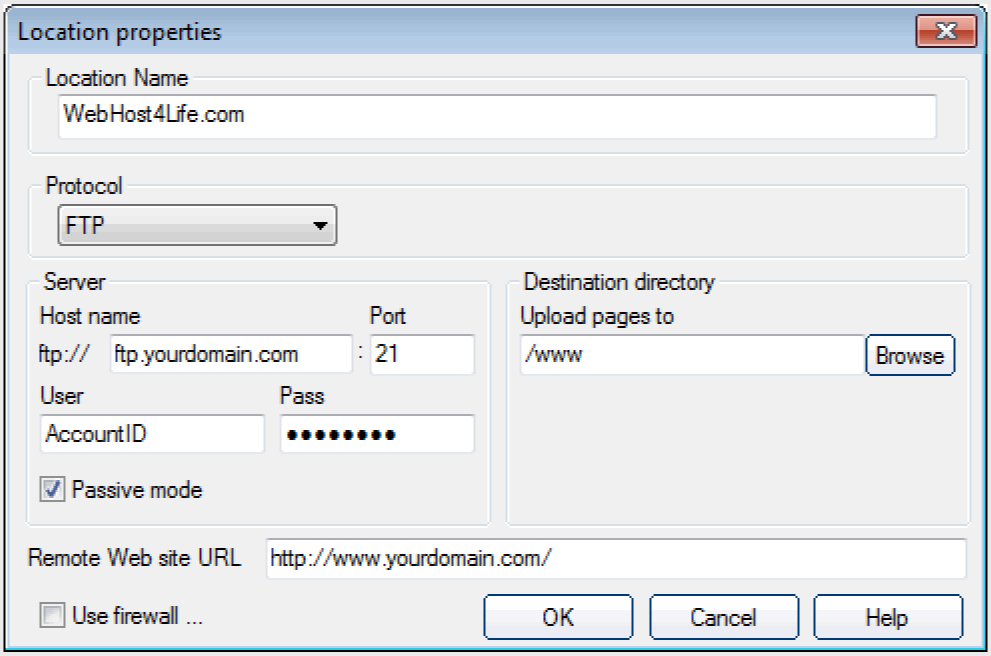
This is the FTP information you will need to use:

- **FTP User Name** - your account ID.
- **FTP Password** - your account Password.
- **FTP Host Name** - http://ftp.yourdomain.com

If you are missing any of this information, log in to the Hosting Control Panel at https://linuxcp.mysite4now.com/wcp/ for Linux hosting or https://controlpanel.mysite4now.com/wcp/ for Windows hosting.

Return to the PHPRunner and type **ftp.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your WebHost4Life account. Then click **Browse** and choose the **www** site folder.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/**, replacing yourdomain.com with your domain name. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



Location properties

Location Name: WebHost4Life.com

Protocol: FTP

Server
Host name: ftp:// ftp.yourdomain.com : Port 21
User: AccountID
Pass: ●●●●●●●●●
☑ Passive mode

Destination directory
Upload pages to: /www

Remote Web site URL: http://www.yourdomain.com/

5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your WebHost4Life account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your WebHost4Life account.

3.6.5 MyHosting.com

3.6.5.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your MyHosting.com account, follow the instructions below.

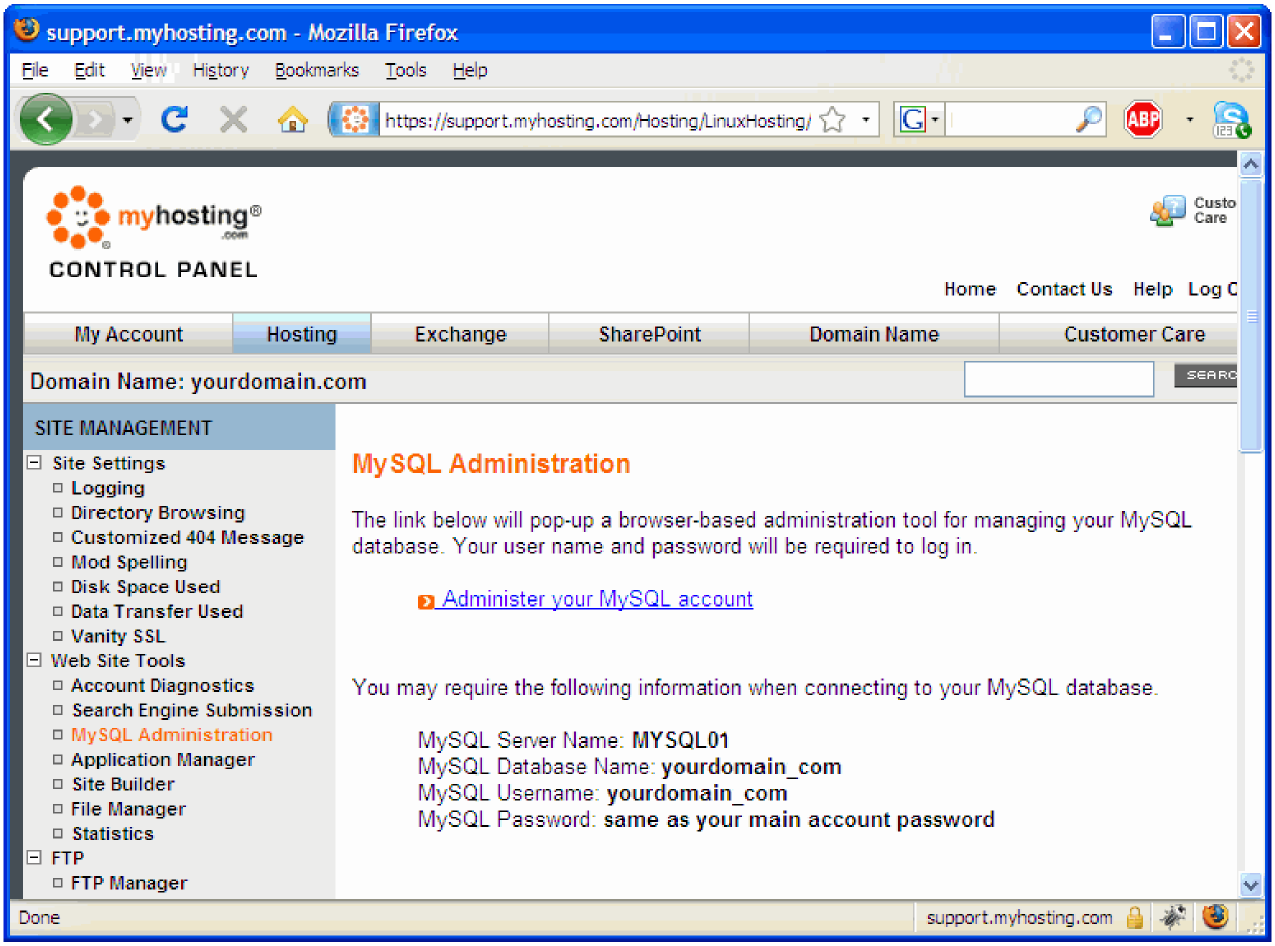
Step 1. Set up a database

1. Log in to the MyHosting.com Control Panel at https://support.myhosting.com/.

2. Select **Application Manager** in the **Linux Hosting** section.

3. Select **MySQL Administration** from the left-side menu.

4. Your MySQL database details are displayed. You will use this information while connecting to MySQL database from PHPRunner.



Domain Name: yourdomain.com

MySQL Administration

The link below will pop-up a browser-based administration tool for managing your MySQL database. Your user name and password will be required to log in.

Administer your MySQL account

You may require the following information when connecting to your MySQL database.

MySQL Server Name: **MYSQL01**
MySQL Database Name: **yourdomain_com**
MySQL Username: **yourdomain_com**
MySQL Password: **same as your main account password**

Step 2. Connect to a MySQL database

1. Start PHPRunner.

2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

4. Enter the host/server name, user name and password indicated for your database in the Control panel.

5. Select the **Connect using PHP** check box and click **Upload phprunner.php**.

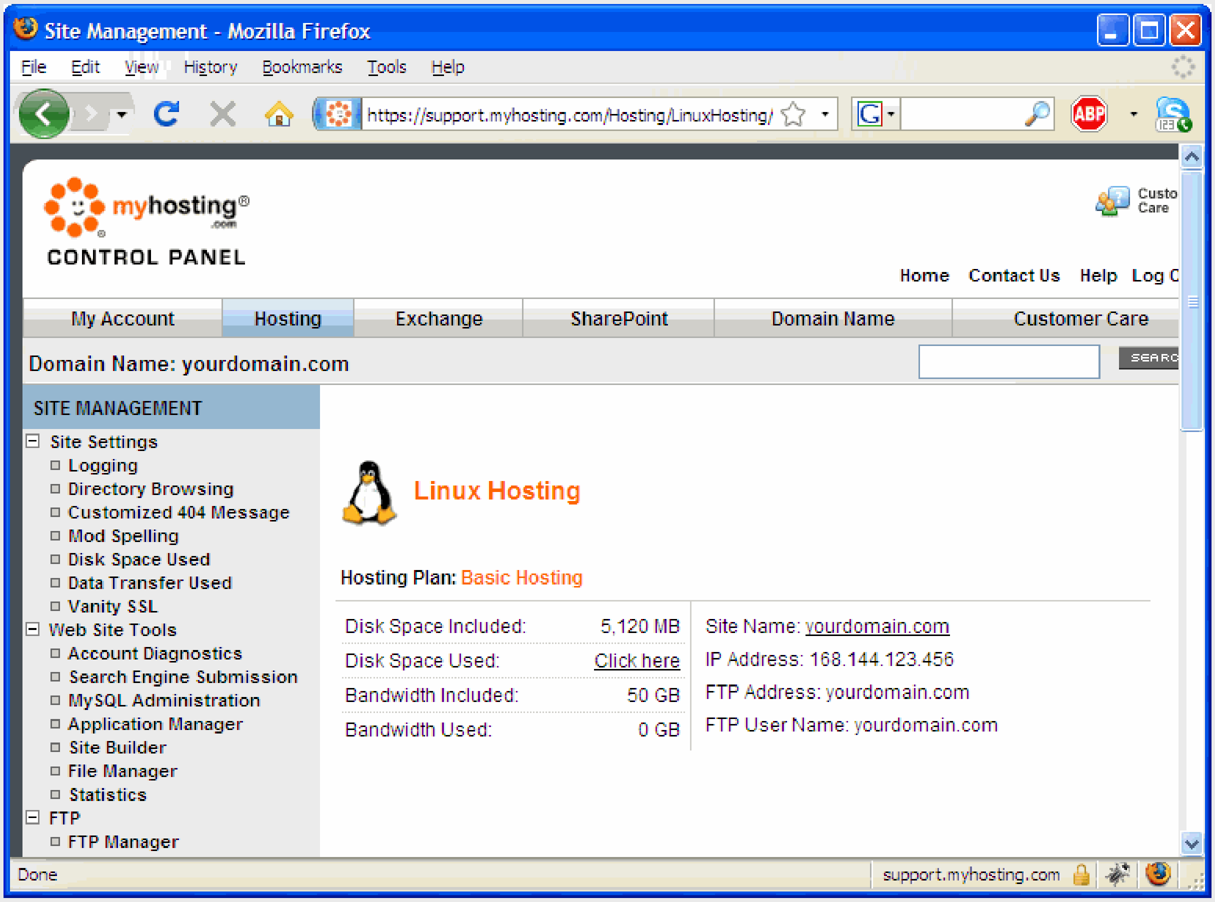
6. Configure built-in FTP client.

How to configure FTP

This is the FTP information you will need to use:

- **FTP User Name** - yourdomain.com.
- **FTP Password** - your account password.
- **FTP Host Name** - yourdomain.com

If you are missing any of this information, log in to the MyHosting.com Control Panel at https://support.myhosting.com/ and click **Hosting**. You can see the FTP settings for your account.



Domain Name: yourdomain.com

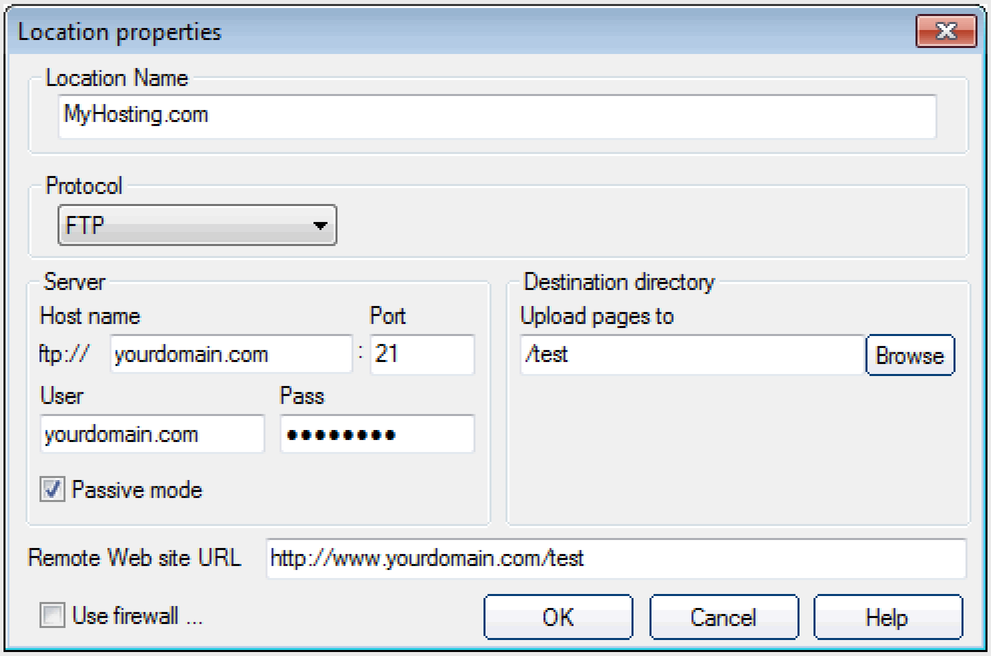
Hosting Plan: Basic Hosting

Disk Space Included:	5,120 MB
Disk Space Used:	Click here
Bandwidth Included:	50 GB
Bandwidth Used:	0 GB

Site Name: yourdomain.com
IP Address: 168.144.123.456
FTP Address: yourdomain.com
FTP User Name: yourdomain.com

Return to the PHPRunner and type **yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password indicated for your FTP account in the Control panel. Then click **Browse** to select a directory to upload the files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK.**



Location properties

Location Name: MyHosting.com

Protocol: FTP

Server

Host name: ftp:// yourdomain.com

Port: 21

User: yourdomain.com

Pass: ●●●●●●●●●

☑ Passive mode

Destination directory

Upload pages to: /test

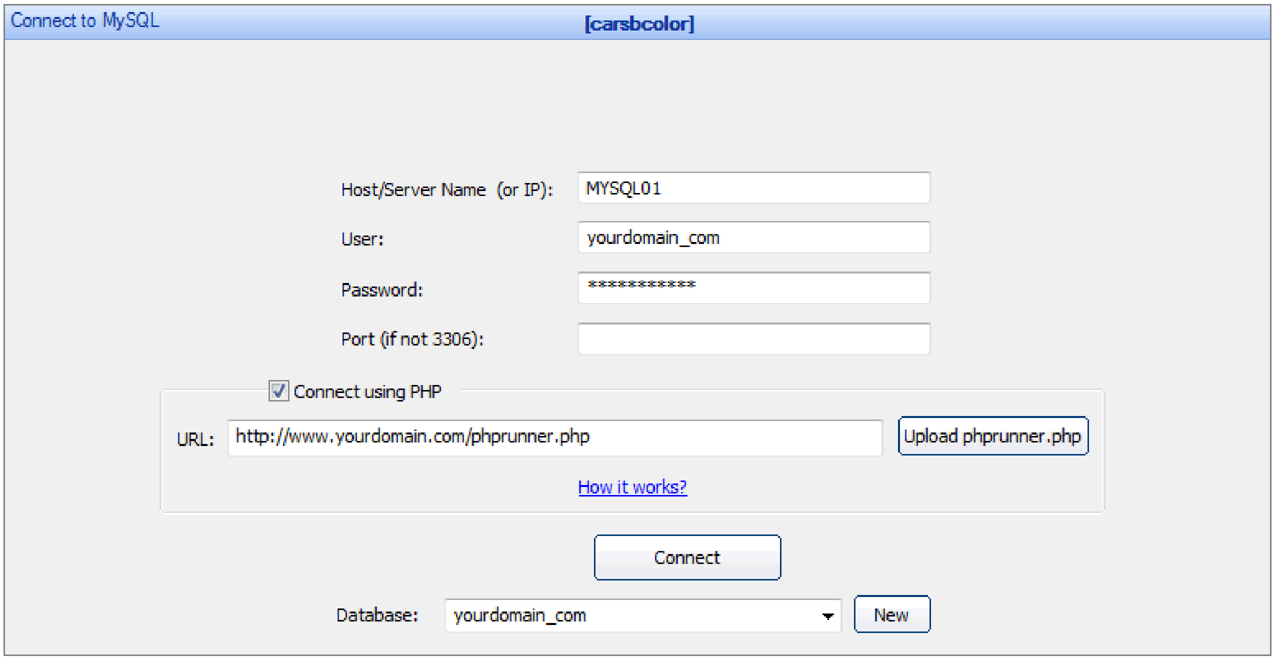
Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

7. Click **Upload!** button. Wait while PHPRunner uploads phprunner.php file to your MyHosting.com account, then click **Close**.

8. Click **Connect**.

9. Select a database and click **Next**>>.



Connect to MySQL [carsbcolor]

Host/Server Name (or IP): MYSQL01

User: yourdomain_com

Password: ************

Port (if not 3306):

☑ Connect using PHP

URL: http://www.yourdomain.com/phprunner.php

Database: yourdomain_com

Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

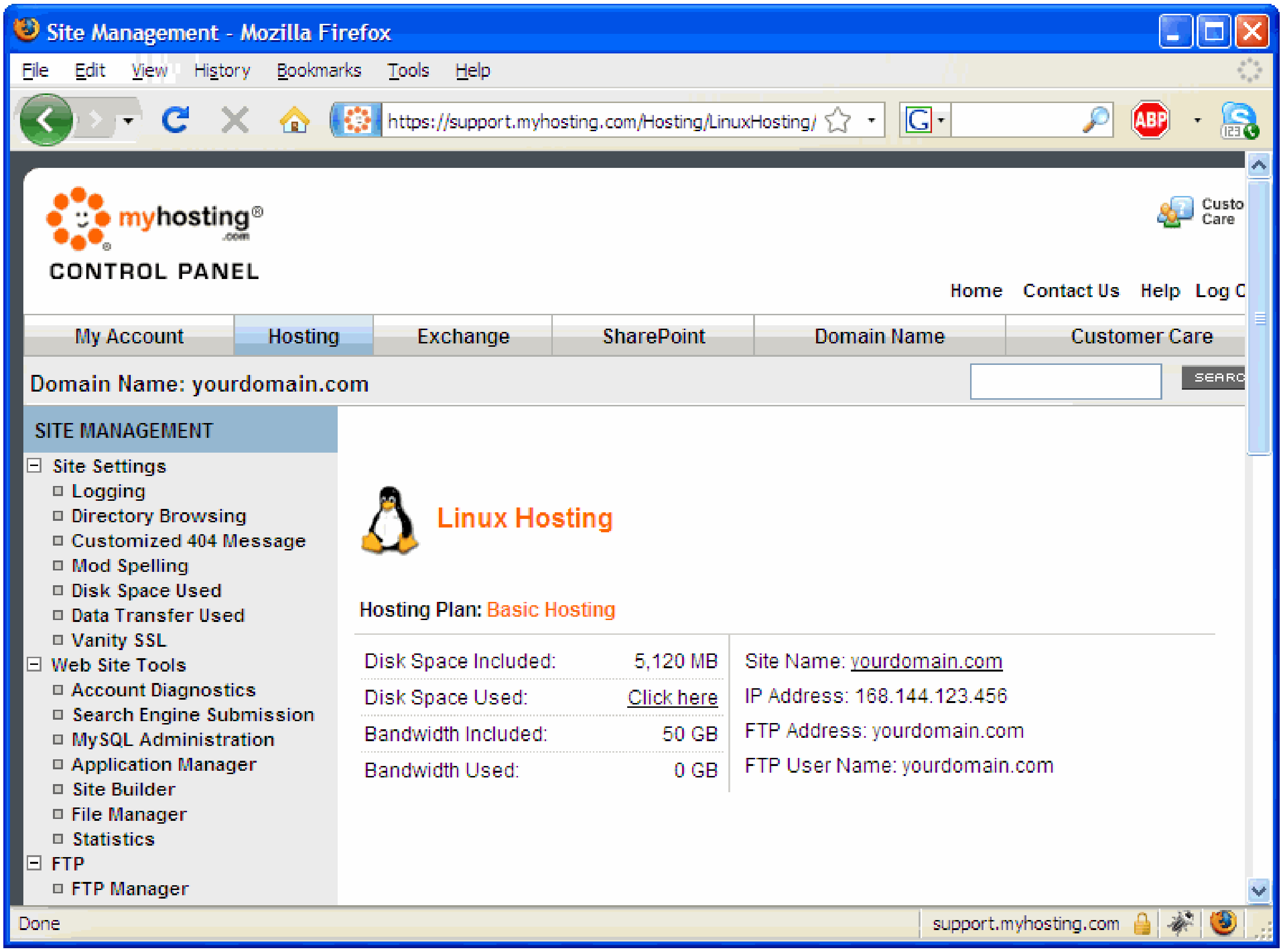
3.6.5.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your MyHosting.com account, follow the instructions below.

1. This is the FTP information you will need to use:

- **FTP User Name** - yourdomain.com.
- **FTP Password** - your account password.
- **FTP Host Name** - yourdomain.com

If you are missing any of this information, log in to the MyHosting.com Control Panel at https://support.myhosting.com/ and click **Hosting**. You can see the FTP settings for your account.



Domain Name: yourdomain.com

Hosting Plan: Basic Hosting

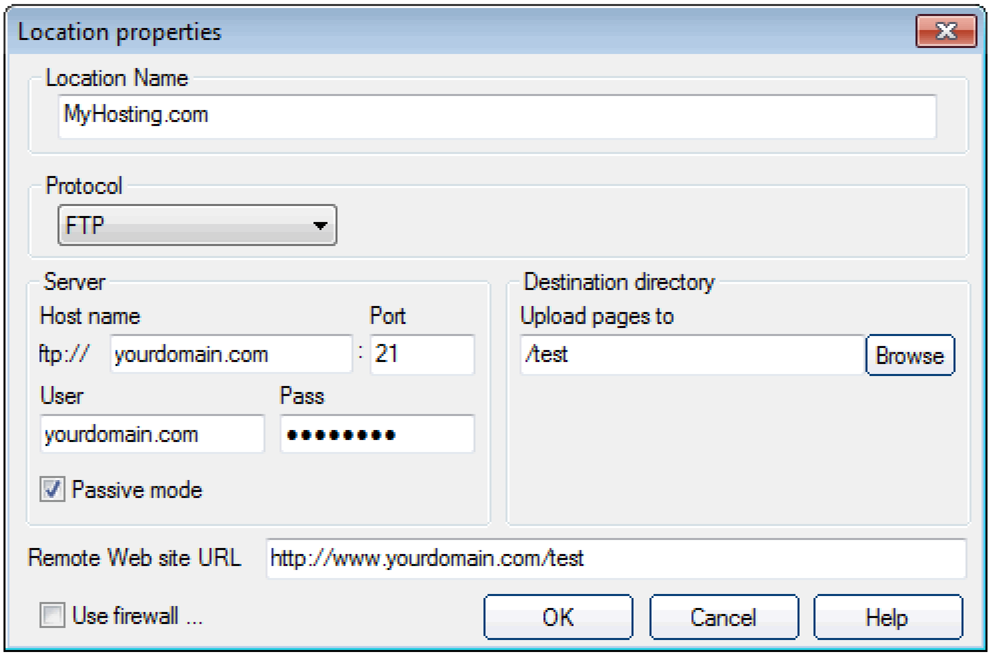
Disk Space Included:	5,120 MB
Disk Space Used:	Click here
Bandwidth Included:	50 GB
Bandwidth Used:	0 GB

Site Name: yourdomain.com
IP Address: 168.144.123.456
FTP Address: yourdomain.com
FTP User Name: yourdomain.com

2. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

3. Type **yourdomain.com** as a host name and user name, replacing yourdomain.com with your domain name. Enter password for your MyHosting.com account. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.



Location properties

Location Name: MyHosting.com

Protocol: FTP

Server

Host name: ftp:// yourdomain.com : Port: 21

User: yourdomain.com

Pass: ●●●●●●●●●

☑ Passive mode

Destination directory

Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

□ Use firewall ...

4. Click **OK**.

Congratulations! FTP client is now configured to upload files to your MyHosting.com account.

3.6.5.3 Publishing project via FTP

To publish generated project files to your MyHosting.com hosting account using built into PHPRunner FTP client, follow the instructions below.

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Publish via FTP**. Built-in FTP client opens.

4. If you have already configured FTP client, choose FTP location and click **Properties**. Click **Browse** and choose the **docs** site folder. In the **Remote web site URL** textbox type **http://www.yourdomain.com/**, replacing yourdomain.com with your domain name. Click **OK**.

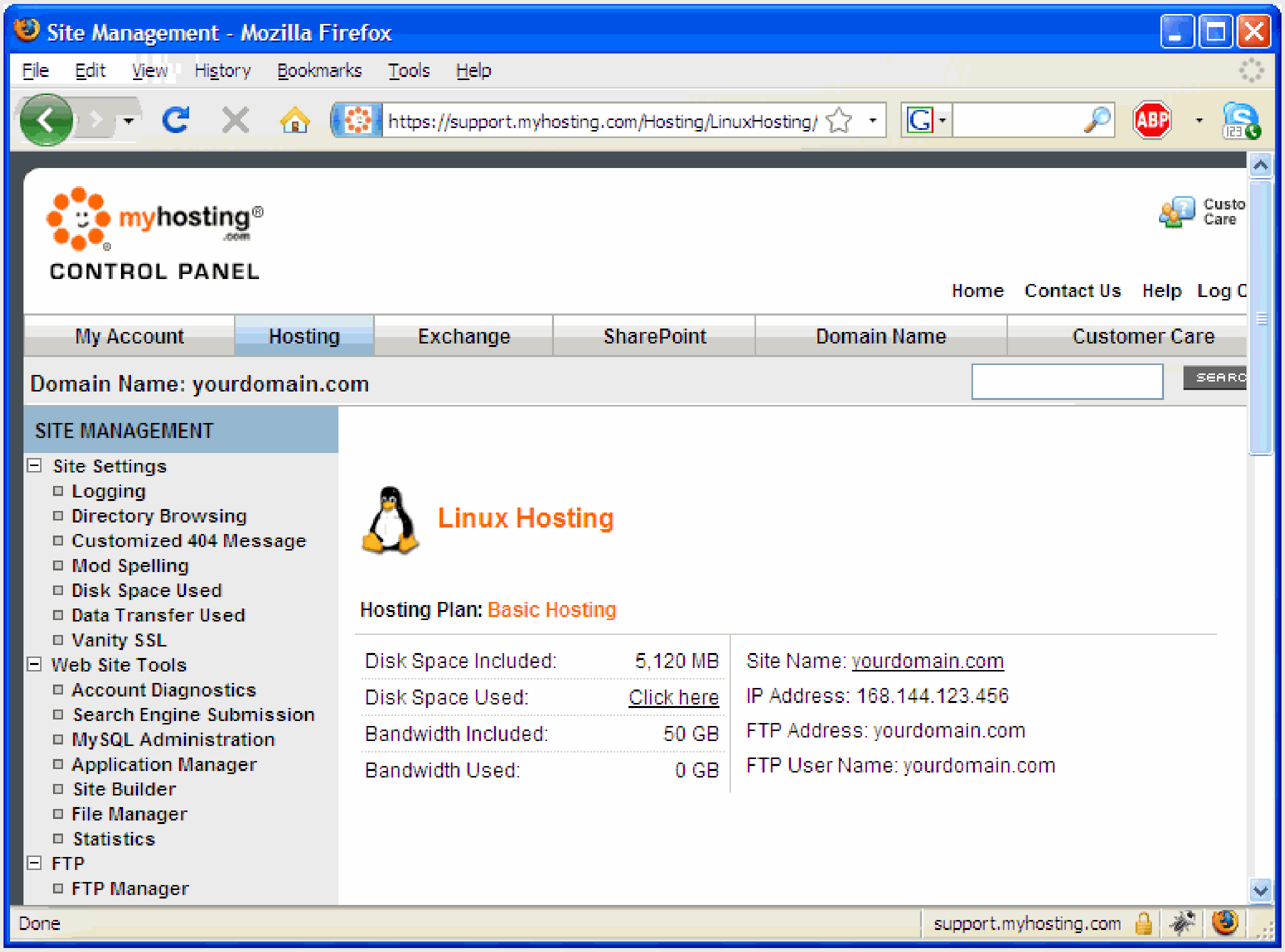
If you have not configured FTP client yet, follow the instructions below.

How to configure FTP

> This is the FTP information you will need to use:

- **FTP User Name** - yourdomain.com.
- **FTP Password** - your account password.
- **FTP Host Name** - yourdomain.com

If you are missing any of this information, log in to the MyHosting.com Control Panel at https://support.myhosting.com/ and click **Hosting**. You can see the FTP settings for your account.



Domain Name: yourdomain.com

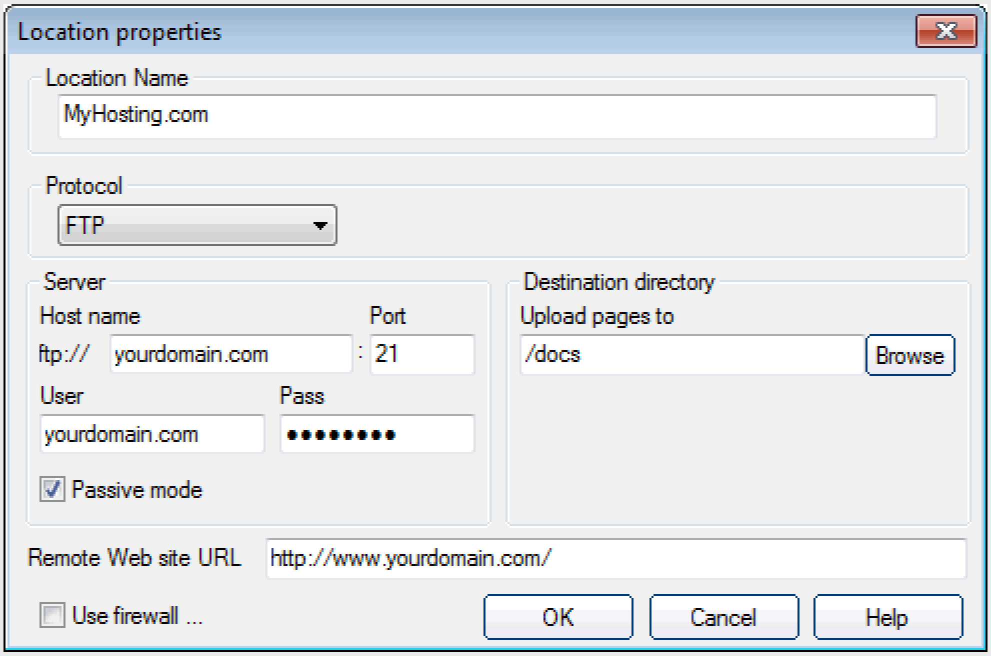
Hosting Plan: Basic Hosting

Disk Space Included:	5,120 MB
Disk Space Used:	Click here
Bandwidth Included:	50 GB
Bandwidth Used:	0 GB

Site Name: yourdomain.com
IP Address: 168.144.123.456
FTP Address: yourdomain.com
FTP User Name: yourdomain.com

Return to the PHPRunner and type **yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your MyHosting.com account. Then click **Browse** and choose the **docs** site folder.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/**, replacing yourdomain.com with your domain name. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your MyHosting.com account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your MyHosting.com account.

3.6.6 InspiRunner.com

3.6.6.1 Connecting to MySQL

To connect from PHPRunner to the remote MySQL database stored at your InspiRunner.com account, follow the instructions below.

Step 1. Set up a database

1. Log in to the InspiRunner.com Control Panel at https://www.inspirunner.com:8443/login.php3.

2. Click the domain name you need.

3. Click **Databases**.

4. Click **Add New Database**.

5. Enter a name for the database. Select the **MySQL** database type. Click **OK**.

6. To set up database administrator's credentials, click **Add New Database User**.

7. Type a user name and a password that will be used for accessing the database.

8. Click **OK**.

Step 2. Connect to a MySQL database

1. Start PHPRunner.

2. Select an existing project or create a new one. Click **Next**>>.

3. Select MySQL as a database type. Click **Next**>>.

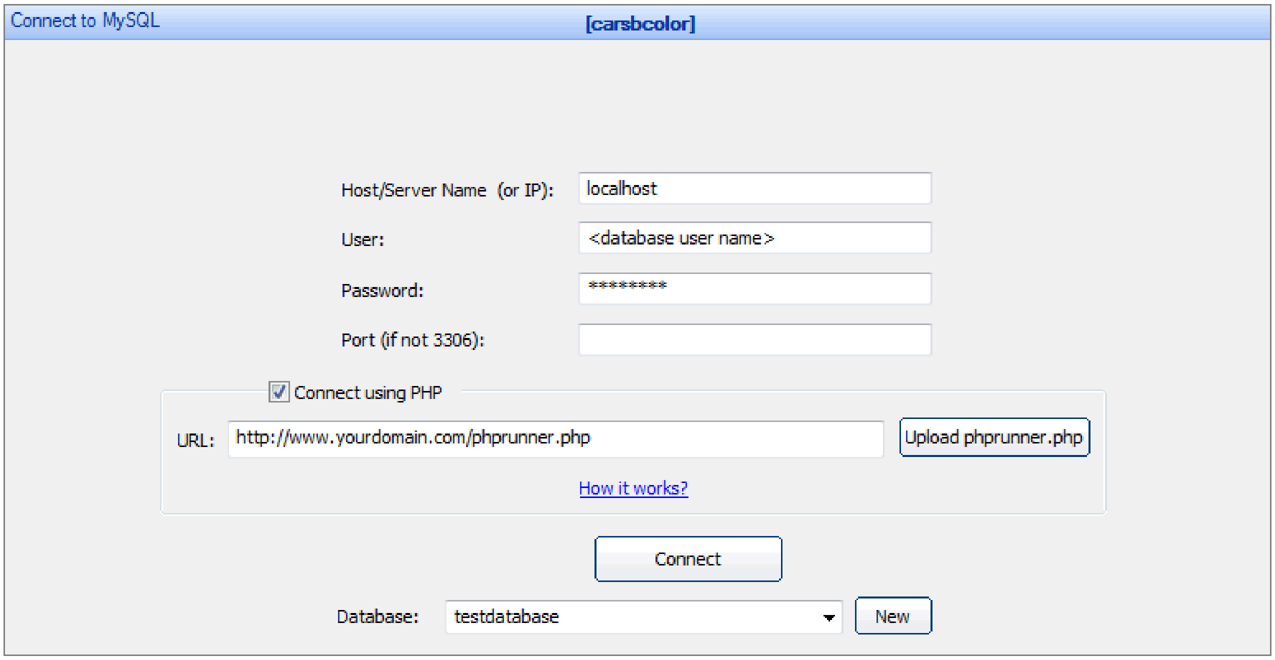
4. Enter **localhost** as the host/server name, database user name and password.

5. Select the **Connect using PHP** check box.

6. In the **URL** textbox type **http://www.yourdomain.com/phprunner.php**, replacing yourdomain.com with your domain name.

8. Click **Connect**.

9. Select a database and click **Next**>>.



Host/Server Name (or IP): localhost

User: <database user name>

Password: *********

Port (if not 3306):

☑ Connect using PHP

URL: http://www.yourdomain.com/phprunner.php

Database: testdatabase

Congratulations! You have connected to the MySQL database. Now it is time to create/modify datasource tables.

3.6.6.2 Configuring FTP

To configure FTP client built into PHPRunner to upload files to your InspiRunner.com account, follow the instructions below.

1. This is the FTP information you will need to use:

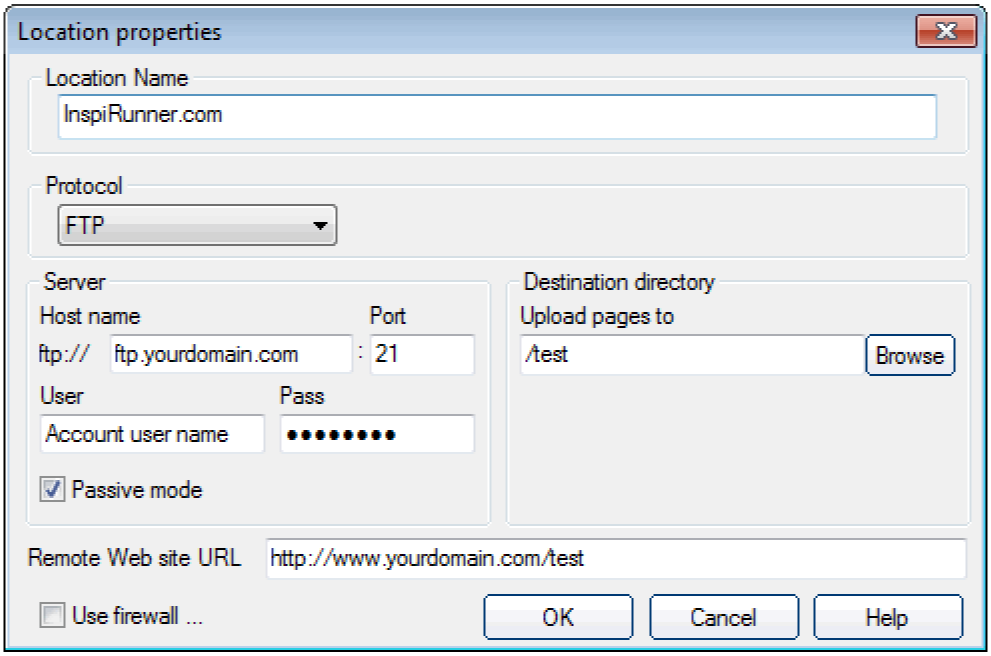
- **FTP User Name** - your account username.
- **FTP Password** - your account password.
- **FTP Host Name** - ftp.yourdomain.com

If you are missing any of this information, log in to the InspiRunner.com Control Panel at https://www.inspirunner.com:8443/login.php3.

2. Start PHPRunner and open **FTP location properties** dialog. This dialog is available while connecting to MySQL under **Upload phprunner.php** button or when a project is built on the last program screen under **Publish via FTP** button.

Type **ftp.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your InspiRunner.com account. Then click **Browse** to select a directory to upload files to. If you receive an error while creating new directory, select the **Passive mode** check box.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/subdirectory/**, replacing yourdomain.com with your domain name and subdirectory with the name of the site folder you selected to upload files to. If you selected a root site folder, the remote web site URL will be http://www.yourdomain.com/. Remote Web site URL is used to open downloaded pages in a browser for test purposes.



Location Name: InspiRunner.com

Protocol: FTP

Host name: ftp.yourdomain.com

Port: 21

User: Account user name

Pass: ●●●●●●●●●

☑ Passive mode

Upload pages to: /test

Remote Web site URL: http://www.yourdomain.com/test

☐ Use firewall ...

3. Click **OK**.

Congratulations! FTP client is now configured to upload files to your InspiRunner.com account.

3.6.6.3 Publishing project via FTP

To publish generated project files to your InspiRunner.com hosting account using built into PHPRunner FTP client, follow the instructions below.

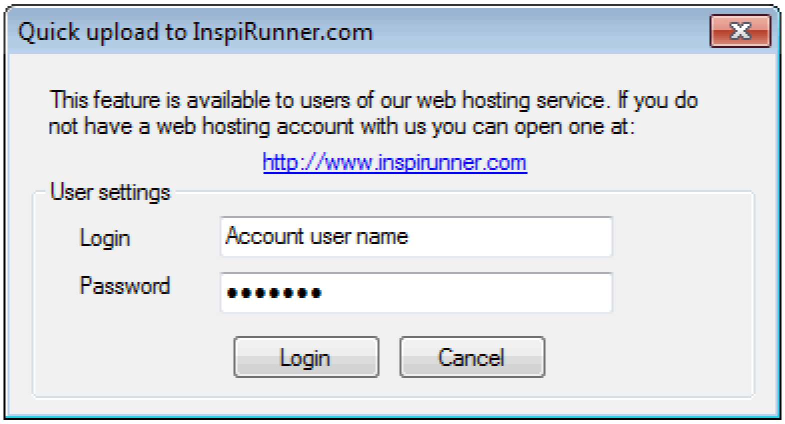
Quick upload to InspiRunner.com

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Quick upload to InspiRunner.com**.

4. Enter your InspiRunner.com account login and password. Click **Login**.



Quick upload to InspiRunner.com

This feature is available to users of our web hosting service. If you do not have a web hosting account with us you can open one at:

http://www.inspirunner.com

User settings

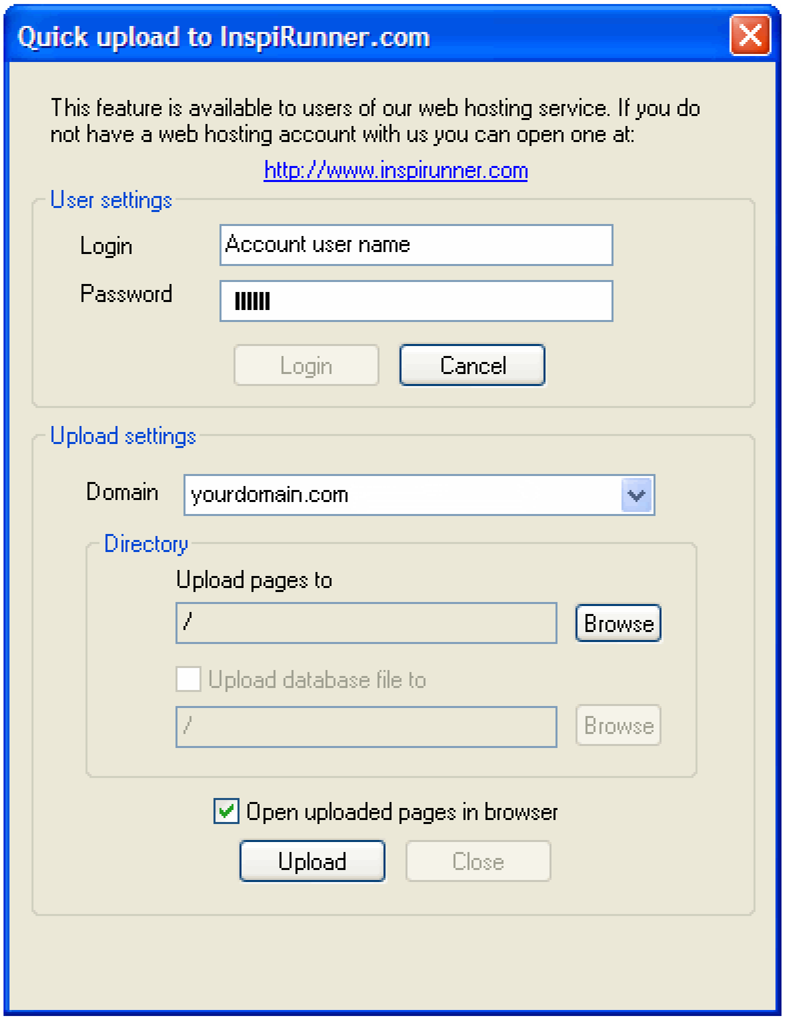
Login: Account user name

Password: •••••••

5. Select the domain name you need.

6. If you wish to upload the generated files to a website root folder, leave the directory settings as is. Otherwise, click **Browse** to choose another directory.

7. Select the **Open uploaded pages in browser** check box.



Quick upload to InspiRunner.com

This feature is available to users of our web hosting service. If you do not have a web hosting account with us you can open one at:

http://www.inspirunner.com

User settings

Login: Account user name

Password: ||||||

Upload settings

Domain: yourdomain.com

Directory

Upload pages to: /

□ Upload database file to: /

☑ Open uploaded pages in browser

8. Click **Upload**.

9. Wait while PHPRunner uploads the project files to your InspiRunner.com account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your InspiRunner.com account.

Publishing via FTP

1. Start PHPRunner.

2. Make all necessary changes to your project and click **Build**.

3. Click **Publish via FTP**. Built-in FTP client opens.

4. To configure FTP client follow the instructions below.

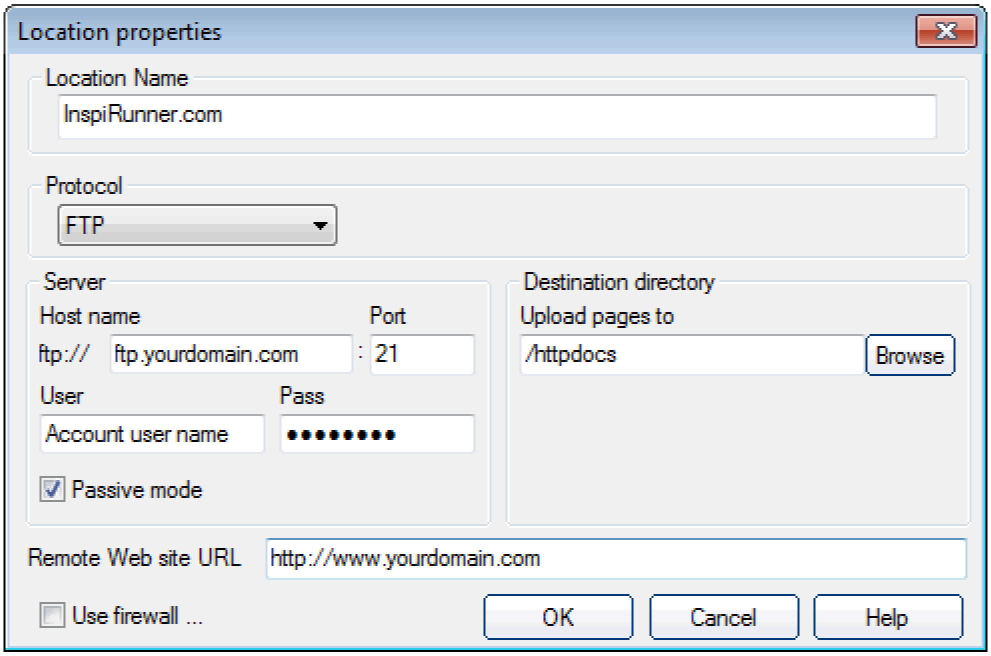
This is the FTP information you will need to use:

- **FTP User Name** - your account username.
- **FTP Password** - your account password.
- **FTP Host Name** - ftp.yourdomain.com

If you are missing any of this information, log in to the InspiRunner.com Control Panel at https://www.inspirunner.com:8443/login.php3.

Return to the PHPRunner and type **ftp.yourdomain.com** as a host name, replacing yourdomain.com with your domain name. Enter User name and Password for your InspiRunner.com account. Then click **Browse** and choose the **httpdocs** site folder.

In the **Remote web site URL** textbox type **http://www.yourdomain.com/**, replacing yourdomain.com with your domain name. Remote Web site URL is used to open downloaded pages in a browser for test purposes. Click **OK**.



Location properties

Location Name: InspiRunner.com

Protocol: FTP

Server

Host name: ftp:// ftp.yourdomain.com : Port: 21

User: Account user name

Pass: ●●●●●●●●●

☑ Passive mode

Destination directory

Upload pages to: /httpdocs

Remote Web site URL: http://www.yourdomain.com

□ Use firewall ...

5. Choose **Upload all files** to upload all files generated by PHPRunner. More info about FTP upload.

6. Click **Upload!** button.

7. Wait while PHPRunner uploads the project files to your InspiRunner.com account, then click **Close**. After that your web site opens in a browser.

Congratulations! Your project was successfully published to your InspiRunner.com account.

3.7 How to install local web server (XAMPP)

PHPRunner comes with built-in web server. However you may want to install your own web server. This page explains how to install a local web server (XAMPP) and configure PHPRunner to use it.

This guide uses Windows XP and XAMPP v5.6.15. XAMPP is a free package that includes an easy-to-set up web server (Apache), database server (MySQL), and a server-side scripting language (PHP). More about XAMPP.

Download XAMPP

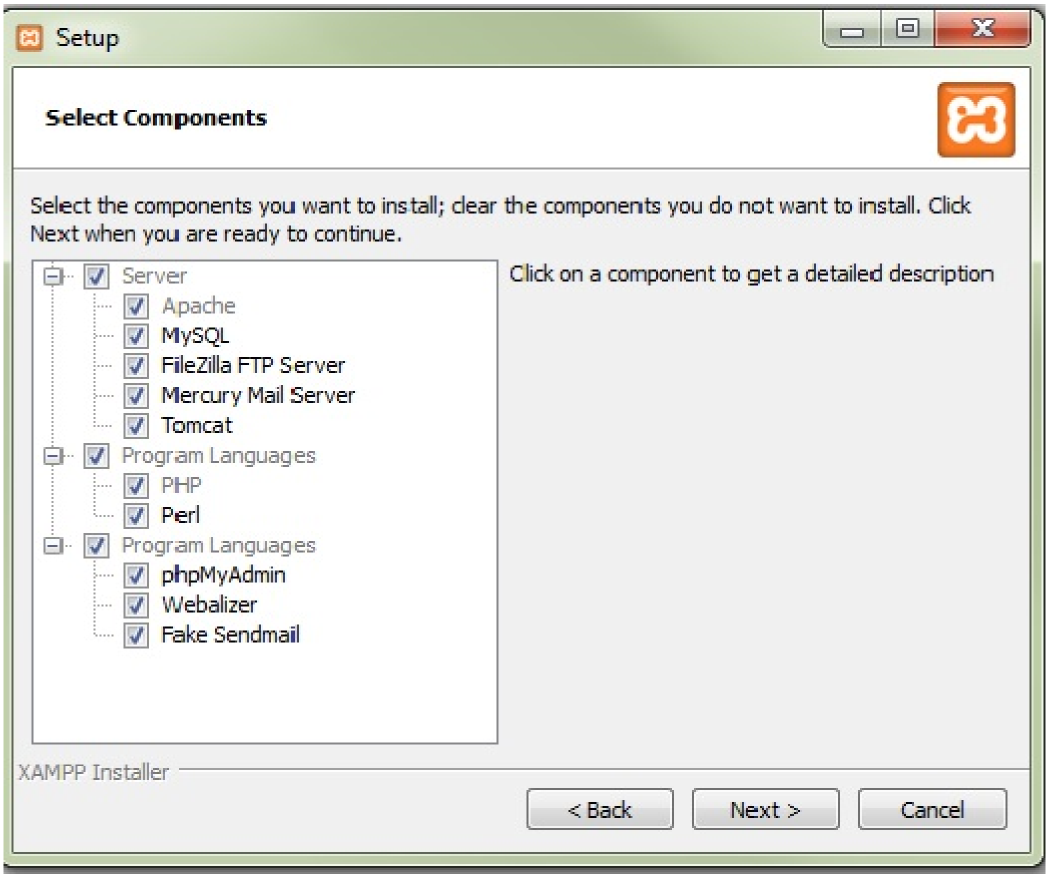
1. Visit http://www.apachefriends.org/en/xampp.html.
2. Select the Installer version of XAMPP for Windows (we download the Installer version of the basic XAMPP package).

Install XAMPP

Note: If you are running the Skype VOIP application, or if you are running IIS Server, you will need to exit them before proceeding. Apache cannot start as a service if Skype or IIS is also running though when XAMPP is not in use, you can exit XAMPP to use Skype or IIS.

1. Run the installer xampp-win32-5.6.15-installer.exe.

2. On the next screen you can select to install Apache and MySQL as services, which will make them start automatically every time you start Windows and will allow other applications you may have, such as Skype, to run simultaneously. If you don't choose this option, you will need to use the XAMPP Control Panel application to start the servers individually each time you need them. This may be desirable if you don't intend to use your servers that often.



Select Components

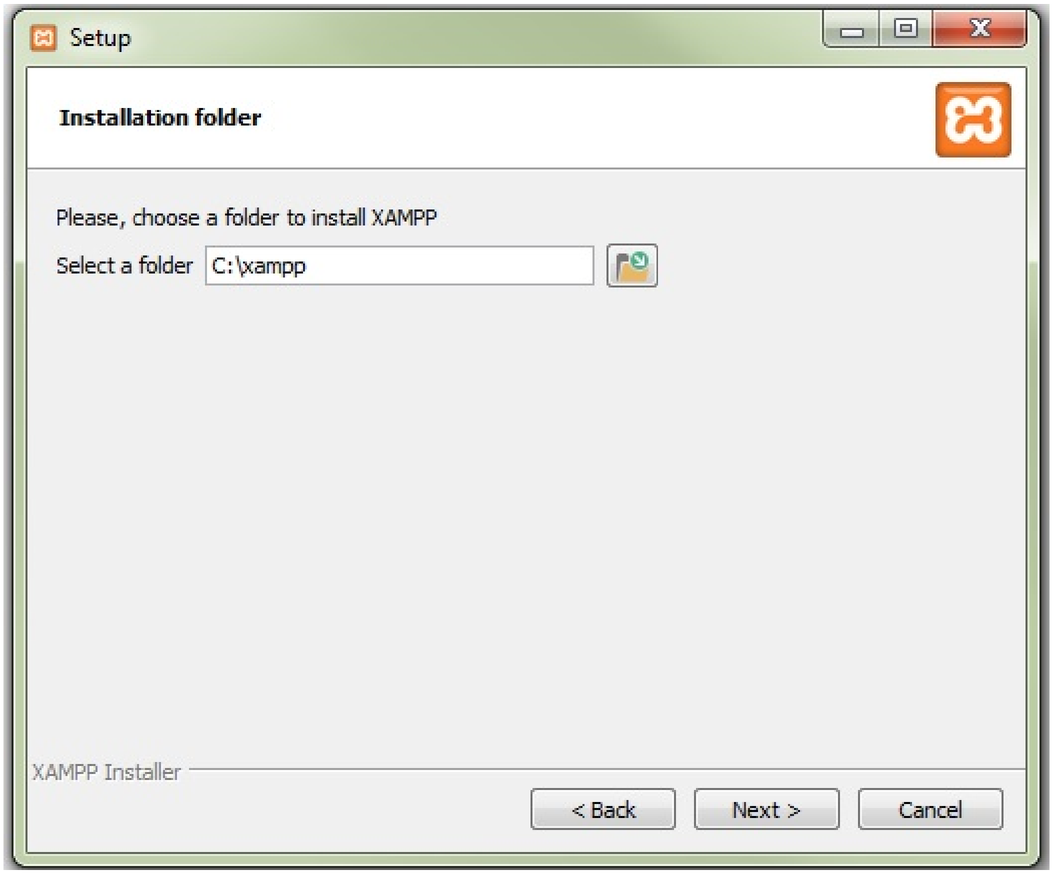
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - ☑ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - ☑ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - ☑ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - ☑ Webalizer
  - ☑ Fake Sendmail

Click on a component to get a detailed description

3. Choose a place to install XAMPP.

Note: for Windows Vista, Apache Friends recommends not installing XAMPP into your c:\Program Files\ folder.



Setup

Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

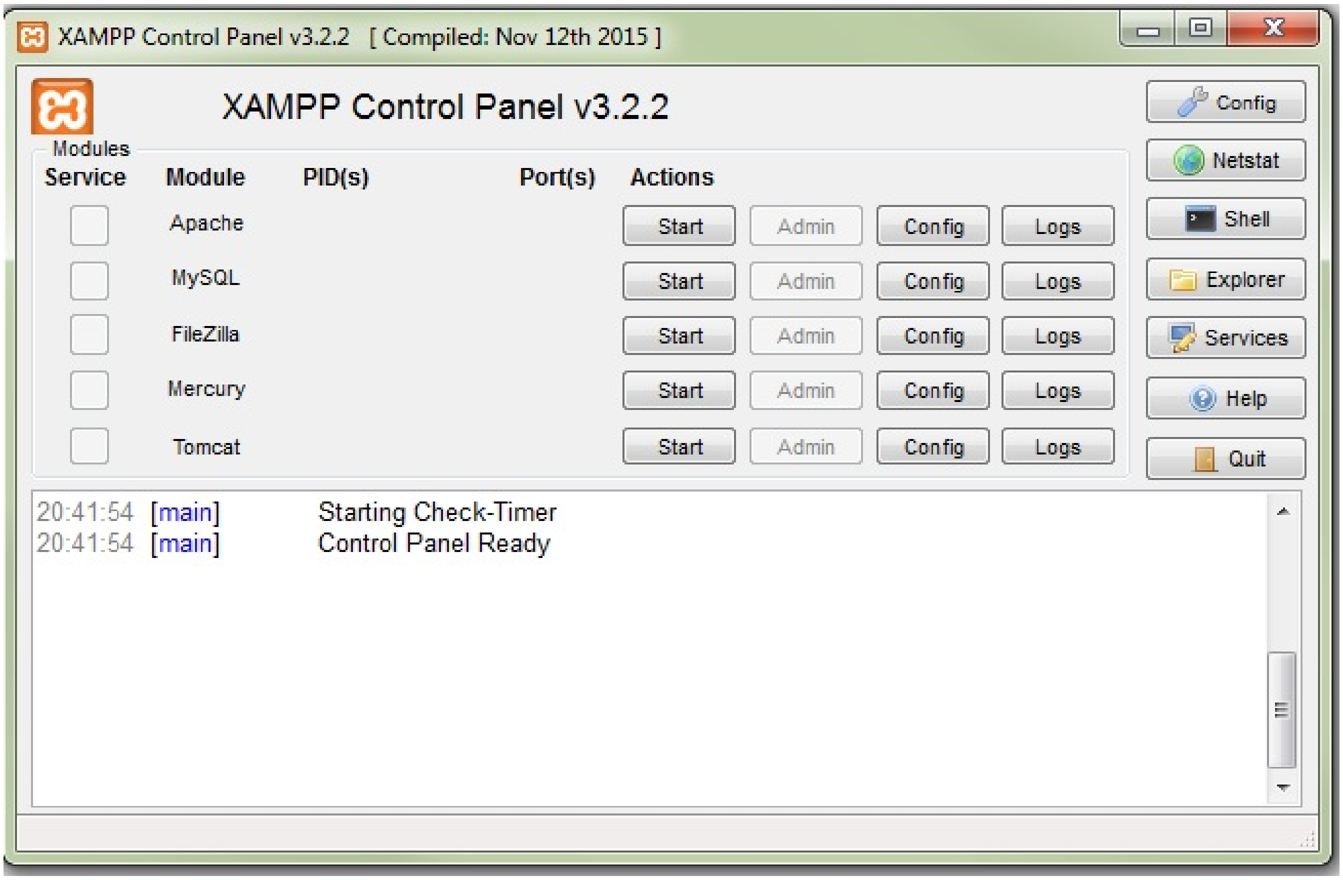
4. XAMPP will then install. At one point, a command-line window will open; do not be alarmed this is normal! After the installation is complete, you will get a message indicating so. Click **Finish**.

5. Next, you are prompted to start the XAMPP control panel, which can also be opened by selecting Start » All Programs » XAMPP » XAMPP Control Panel, or by running c:\xampp\xampp-control.exe. This tool lets you start and stop the various servers installed as part of XAMPP. Since Apache and MySQL already started, select **No**.

Now the installation finished.

Using XAMPP

The XAMPP control panel is used to control and monitor the status of services that XAMPP has installed. When the control panel is running, the following icon will be visible in your system tray . Double-clicking on this icon will bring up the Control Panel.



```
20:41:54  [main]     Starting Check-Timer
20:41:54  [main]     Control Panel Ready
```

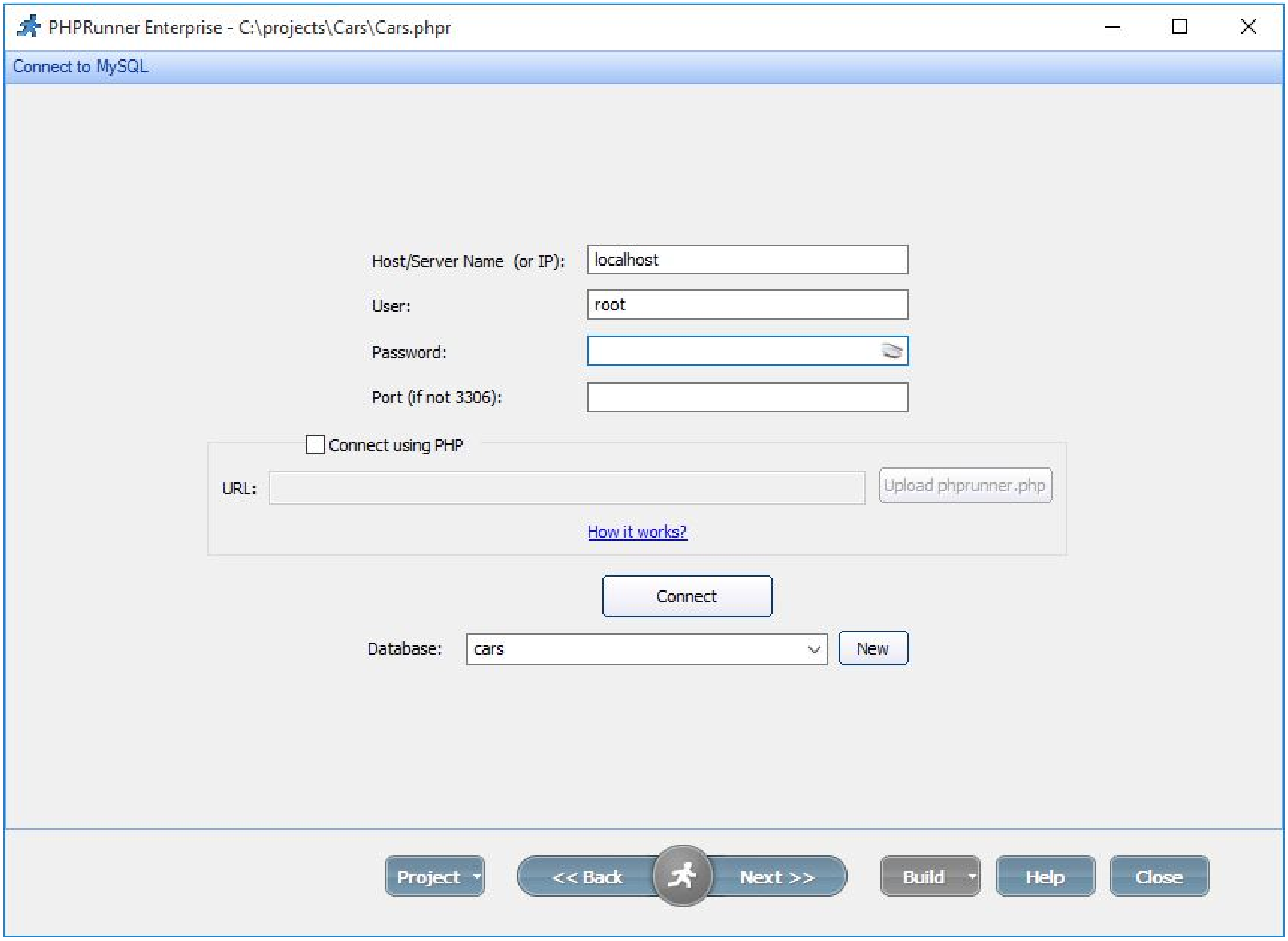
Since Apache and MySQL installed as services you do not need to start/stop them in the Control panel. They start automatically every time you start Windows.

Go to http://localhost/ in a browser. If you are directed to a page with the XAMPP logo, your installation was successful.

You can add or change the files in C:\xampp\htdocs to change what you see at http://localhost/. Right after installation, this folder contains all the files for the XAMPP welcome web page. You can remove or backup these files so that they do not conflict in anyway. XAMPP configuration interface is available at http://localhost/xampp/index.php and is located at C:\xampp\htdocs\xampp.

Note: To send mails via PHP you need to modify php.ini file. In XAMPP it is stored at C:\xampp\php. Change the line *;sendmail_from = me@example.com* to *sendmail_from = your_email_address.*

When connecting to MySQL, type "localhost" in the **Host/Server name (or IP)** field and "root" in the **User** field. Leave **Password** field blank.



Host/Server Name (or IP): localhost

User: root

Password:

Port (if not 3306):

□ Connect using PHP

URL:

Database: cars

On the **Output directory** page switch to **I have my own web server** option and enter URL manually (i.e. *http://localhost/project1*). Change output directory as well to one of web server subdirectories (i.e. *C:\xampp\htdocs\project1*).

3.8 How to add external css/php/js files

To add your own files with JavaScript/CSS/PHP code to a project, follow the instructions below.

1. Copy your files to the project folder

Copy your files to the *source* folder in the project directory (e.g. ..*\Documents\PHPRunnerProjects\Project1\source*). Organize files into subfolders if needed.

For example, you want to add two images *right.png* and *wrong.png* to the *images* folder, new HTML template *quiz.htm* to the *templates* folder and simple CSS file *quiz.css*. To perform this, create the folder *images* inside the *source* folder and copy files *right.png* and *wrong.png* there. Create the folder *templates* inside the *source* folder and copy the file *quiz.htm* there. Copy the file *quiz.css* directly into the *source* folder.

While building the project all files from the *source* folder will be copied to the *output* folder keeping the subfolders' structure.

2. Link your files to pages

JavaScript

To link a JavaScript file to a page, select the page on the Visual Editor tab and switch to the HTML mode. Then add the following code after the <END body> tag. In case of Add/Edit/View pages shown in popup add this code snippet to the List page:

```
<script type="text/javascript" src="include/customfile.js"></script>
```

To link a JavaScript file to all project pages, select the **Header** item on the **Visual Editor** tab and switch to the HTML mode. Then add the code as stated above.

CSS

To link a CSS file to a page, switch to the HTML mode on the **Visual Editor** tab and add your code just after the following line:

```
<link REL="stylesheet" href="style.css" type="text/css">
```

To link a CSS file to all project pages, select the **Header** item on the **Visual Editor** tab and switch to the HTML mode. Then add the code as stated above.

PHP

To add your PHP code, it is recommended to use events. Add the following code to the After table initialized event to link a PHP file to a page or After application initialized event to link a PHP file to all pages:

```
include("customfile.php");
```

3. Accessing PHPRunner session variables from external PHP files

If you need to access PHPRunner session variables from your custom file, add the following as a first line in your PHP file:

```
include("include/dbcommon.php");
```

This code snippet assumes that your external PHP file is located in the main PHPRunner folder.

3.9 Working with MS Access databases

1. If you need to work with Microsoft Access databases we suggest installing 32-bit version of PHPRunner.

2. You also need to install Microsoft Access Database Engine to be able to connect your database. You need to install it on both local machine and on the web server.

Download 32-bit version (AccessDatabaseEngine.exe file) from here:

Microsoft Access Database Engine

and install it.

If it doesn't install, i.e. you see a warining related to 64bit Office components conflict try running it from the command prompt with "/passive" parameter:

```
AccessDatabaseEngine.exe /passive
```

3. On the web server make sure you provided full permissions to web server user on the folder where database file resides.

4. Make sure that your php.ini file contains the following line:

```
extension=php_com_dotnet.dll
```

3.10 Working with Oracle databases

If you need to work with Oracle databases you need to download and install the Oracle Instant Client from this page: http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html

You need to choose 32 or 64-bit client version depending on PHPRunner version that is installed on your computer.

If you are using MS IIS and the PHPRunner wizard has successfully connected to the database but when you run it the generated pages it fail to connect to Oracle, this might be the case that IIS user doesn't have access to tnsnames.ora file.

The suggestion is to use server:port/instance format while connecting to Oracle. This way it will always work in generated app to. Here is the sample tnsnames.ora file

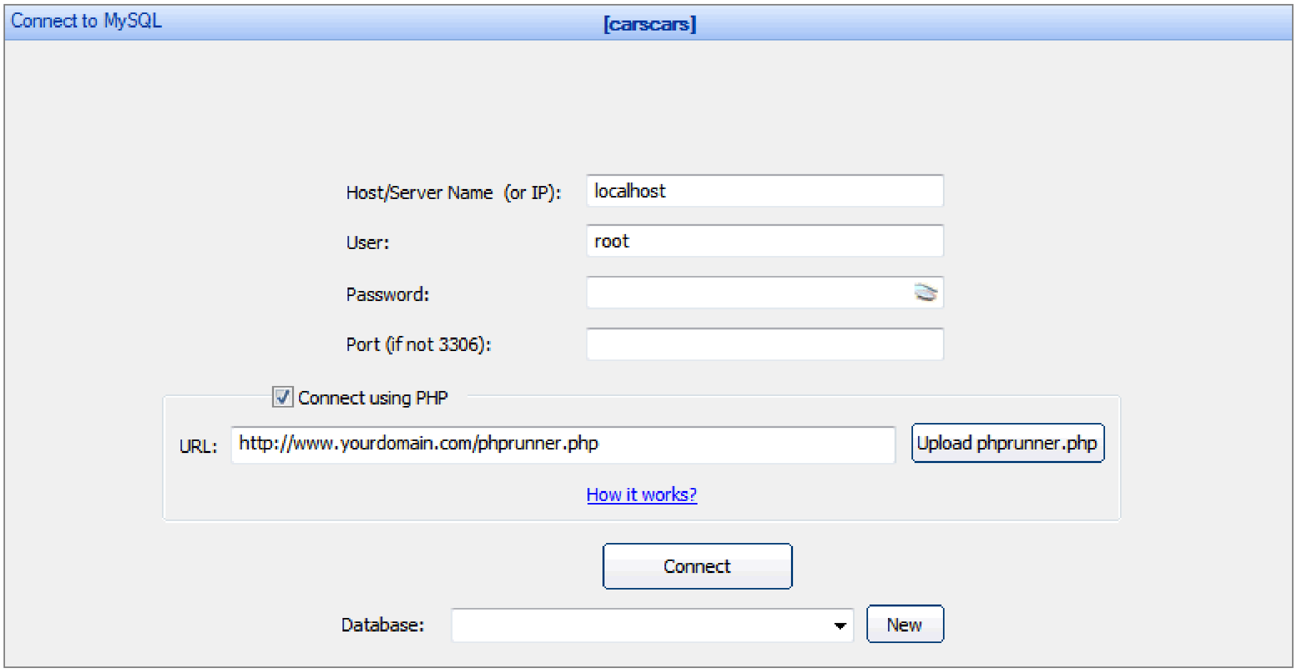
```
AADEV =
(DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = TestOracle.server.gov)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SID = AAINST)
      (SERVER = DEDICATED)
    )
  )
  extension=php_com_dotnet.dll
```

Instead of using AADEV in PHPRunner use TestOracle.server.gov:1521/AAINST

3.11 Connect to remote MySQL database via PHP

If your MySQL server doesn't allow you to connect directly, you can use PHP to connect to it. You should perform the following steps to proceed.

1. On the **Connect to MySQL** page enable 'Connect using PHP' option and upload the connection script *phprunner.php* to your site using **Upload phprunner.php** button or manually. The *phprunner.php* file can be found in the installation folder, usually *C:\Program Files\PHPRunner10.0*.



Host/Server Name (or IP): localhost

User: root

Password:

Port (if not 3306):

☑ Connect using PHP

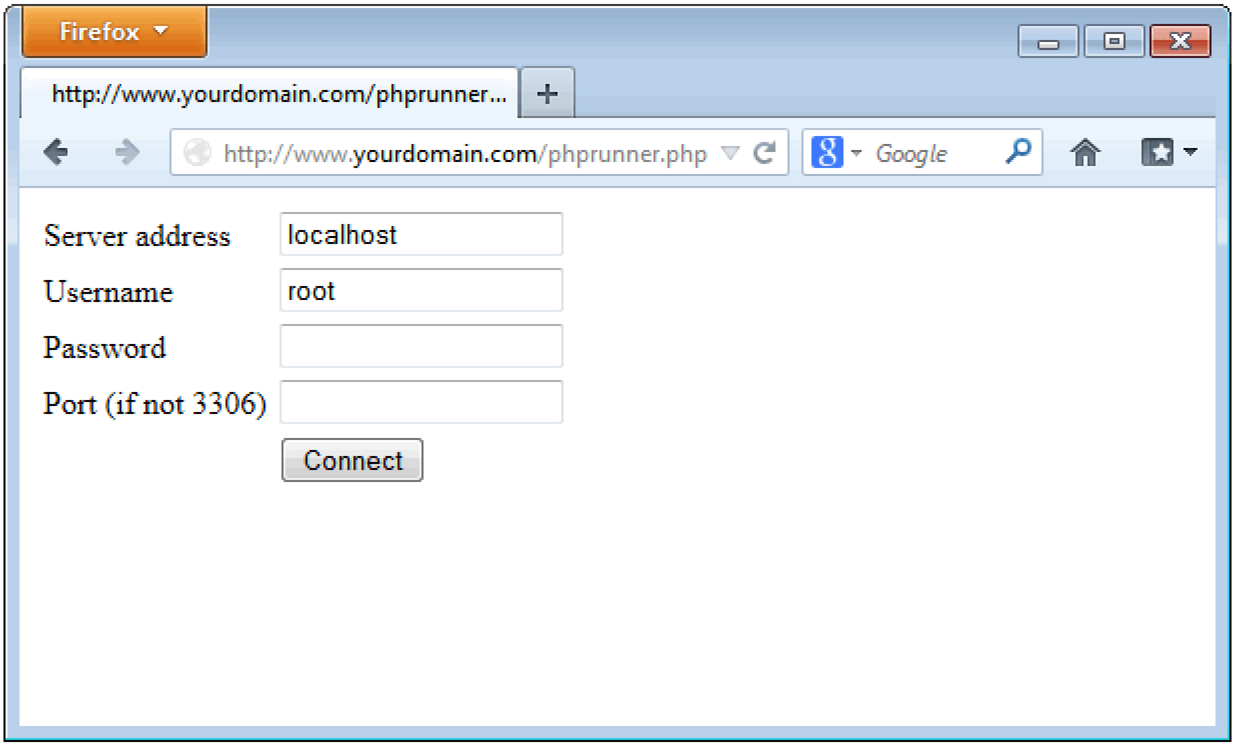
URL: http://www.yourdomain.com/phprunner.php

Database:

2. Test *phprunner.php* in browser:

   - Open the file *phprunner.php* in browser. You can access it at *http://www.yourdomain.com/subdirectory/phprunner.php*, replacing *yourdomain.com* with your own domain name and subdirectory with the name of the site folder in which your *phprunner.php* file were uploaded.
   - Type Host/Server Name (*localhost* if your Web and Mysql server are located at the same host), user name, password and click **Connect**.

Note: Username and password are encrypted before sending.


Firefox
http://www.yourdomain.com/phprunner...
http://www.yourdomain.com/phprunner.php
Google

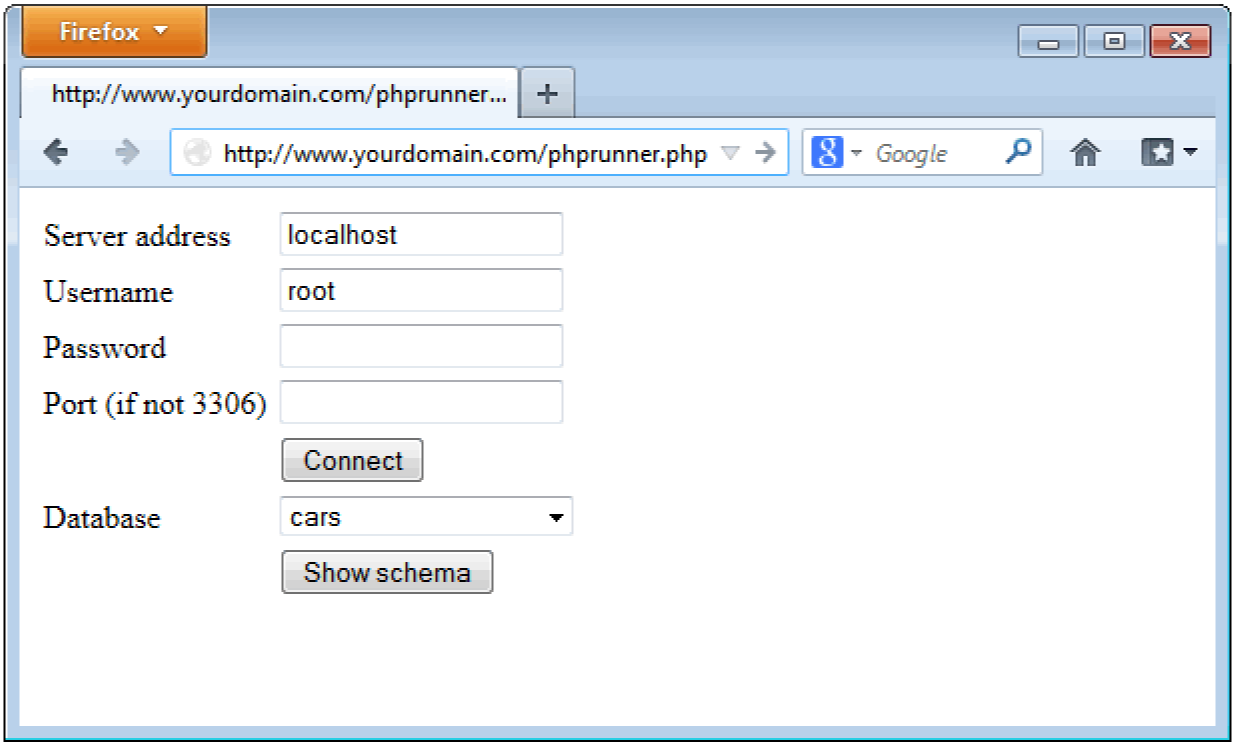
Server address: localhost

Username: root

Password:

Port (if not 3306):

- Select the **Database** you wish to connect.


Firefox
http://www.yourdomain.com/phprunner...
http://www.yourdomain.com/phprunner.php
Google

Server address: localhost

Username: root

Password:

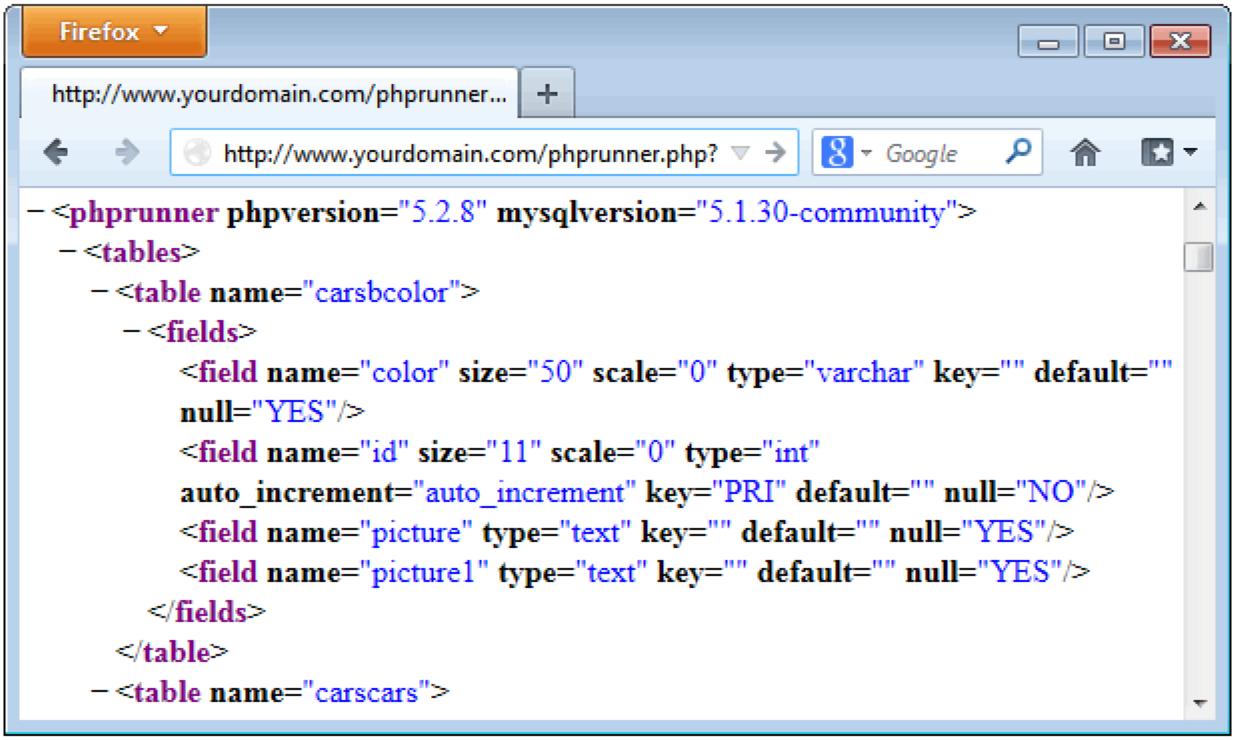
Port (if not 3306):

Connect

Database: cars

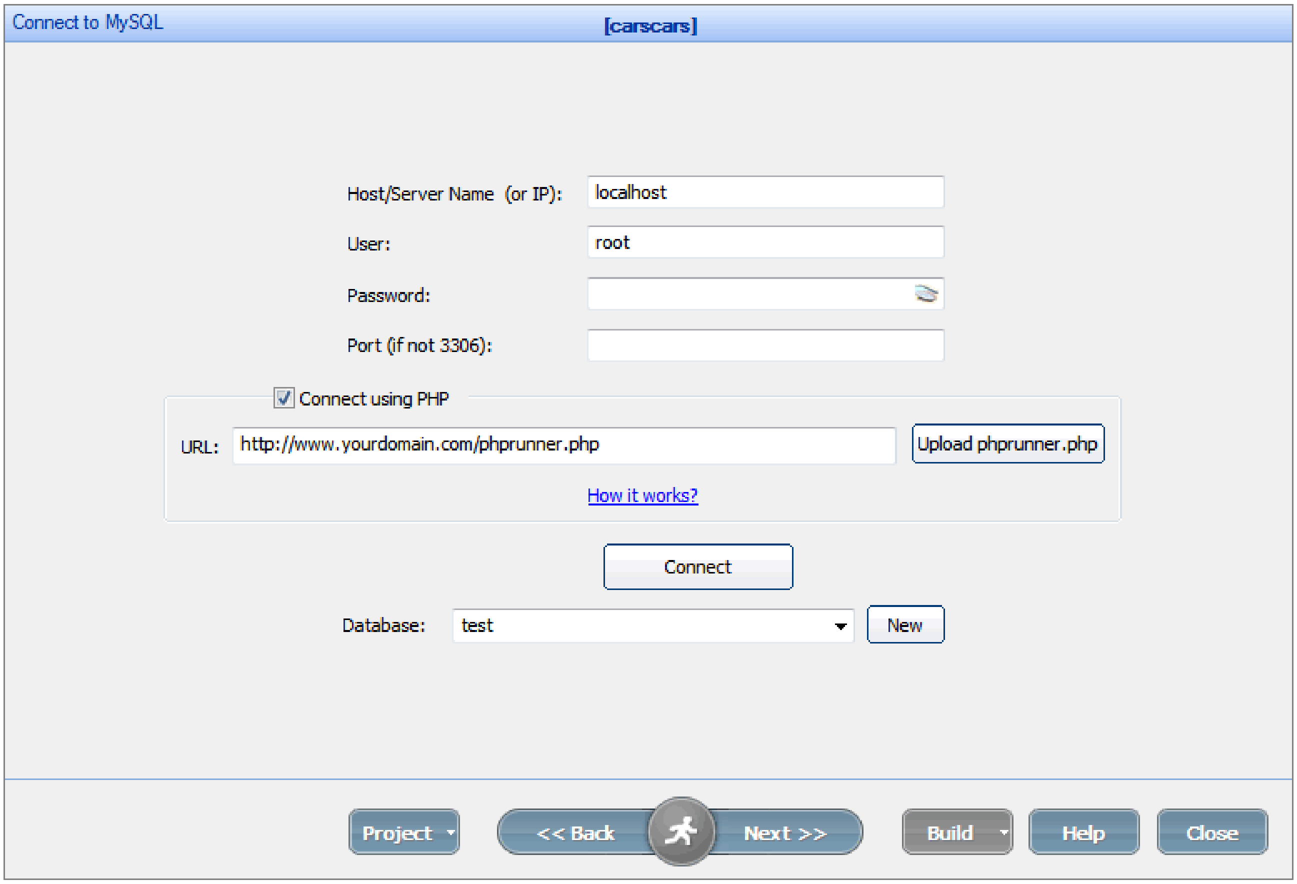
Show schema

- Click **Show schema** to get the XML representation of your database's structure.


Firefox
http://www.yourdomain.com/phprunner...
http://www.yourdomain.com/phprunner.php?
Google

```
- <phprunner phpversion="5.2.8" mysqlversion="5.1.30-community">
  - <tables>
    - <table name="carsbcolor">
      - <fields>
          <field name="color" size="50" scale="0" type="varchar" key="" default=""
          null="YES"/>
          <field name="id" size="11" scale="0" type="int"
          auto_increment="auto_increment" key="PRI" default="" null="NO"/>
          <field name="picture" type="text" key="" default="" null="YES"/>
          <field name="picture1" type="text" key="" default="" null="YES"/>
        </fields>
      </table>
    - <table name="carscars">
```

3. On the **Connect to MySQL** page enter the URL of the connection script *phprunner.php*, type Host/Server Name, user name, password and click **Connect**. All connection settings are the same as you've used on the previous step while testing *phprunner.php* in browser. Then select database and click **Next >>**.



Connect to MySQL [carscars]

Host/Server Name (or IP): localhost

User: root

Password:

Port (if not 3306):

☑ Connect using PHP

URL: http://www.yourdomain.com/phprunner.php

Database: test

Note: some MySQL server won't allow to get a list of databases. In this case you need to type in database name manually.

After you successfully connected to the database you can Select datasource table, list fields and searchable fields.

3.12 AJAX-based Functionality

Quick jump

AJAX-based Auto-Suggest

AJAX-based dependent dropdown boxes

Details Records Preview

Lookup wizard as Edit box with AJAX popup

AJAX-based pagination/sorting/search

PHPRunner comes with AJAX-based functionality added. Now you can search information more easily than ever with google-like auto-suggest feature. Web pages with AJAX driven dependent dropdown boxes are loaded faster. If you want to see details records you just need to mouse over the link, and you don't need to proceed to the Details page. Also new feature for lookup fields was added. Now you can choose the variant from suggest list which is refreshed each time while you are typing in the text box instead of searching through all values in the dropdown boxes.

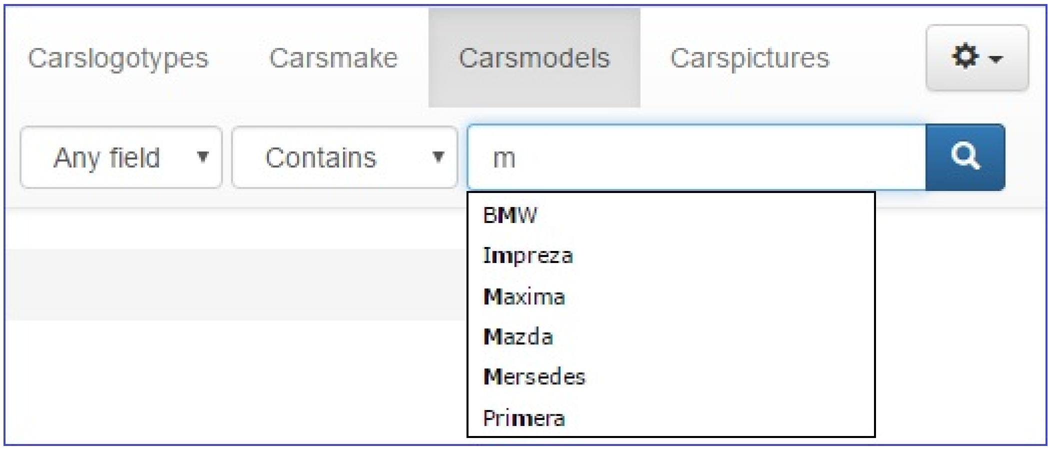
Note: If for some reasons you don't want to use AJAX-based functionality then all you need to do is to change the **$useAJAX** and **$suggestAllContent** variables value in the **include\appsettings.php** file to **false**.

```
$useAJAX = false;
$suggestAllContent = false;
```

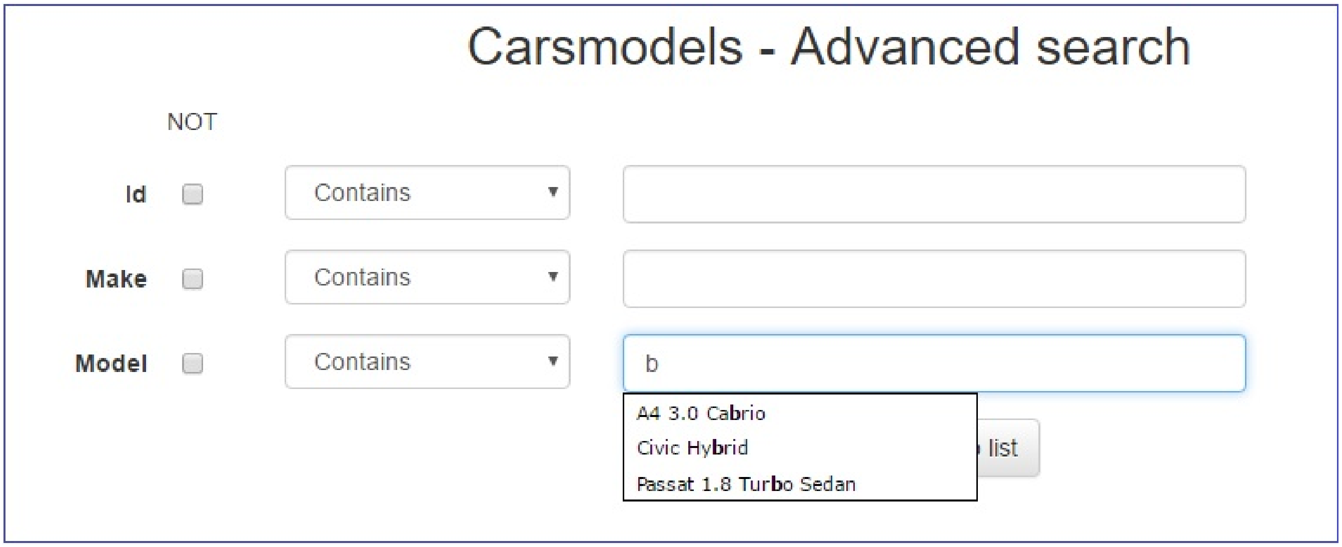
AJAX-based Auto-Suggest

This feature comes in two versions and is enabled on the Basic and Advanced search pages. It is similar in fashion to that of Google Suggest using AJAX technologies.

Search suggest makes your site much more user friendly. To see how it will look like just begin to type in the search box. The picture below demonstrates the search suggest feature on the basic search page:



On the Advanced search page search suggest looks like this:



Carsmodels - Advanced search

NOT

Id □ Contains

Make □ Contains

Model □ Contains b

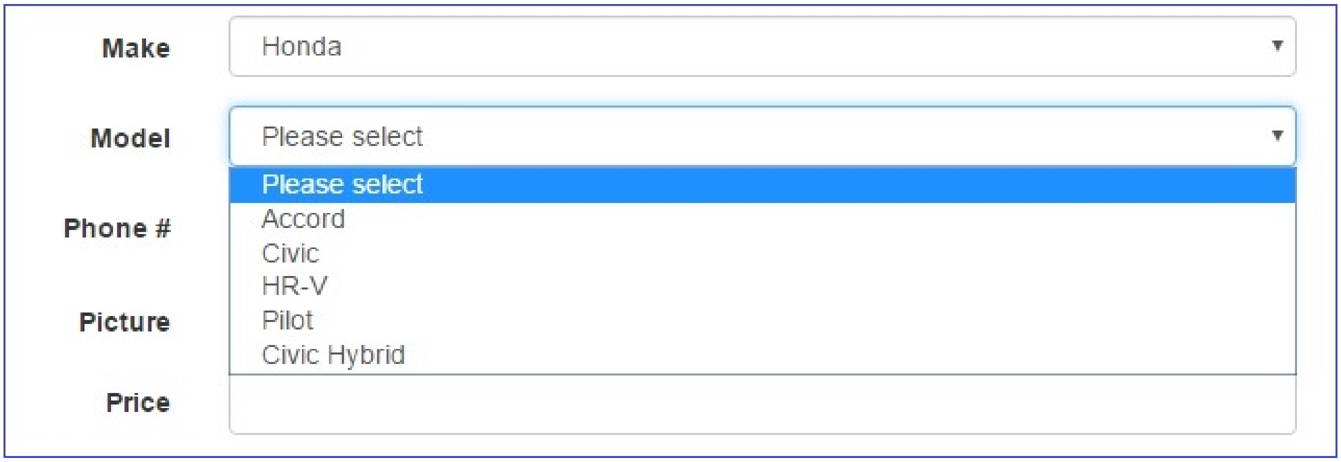
A4 3.0 Cabrio
Civic Hybrid
Passat 1.8 Turbo Sedan

By default, search suggest results include all values in which the search phrase presents. If you want to be shown only those values that begin with the search phrase you should change the **$suggestAllContent** variable value in the **include\appsettings.php** file to **false**.

Note: if a field is not shown at least on one page (list/view/edit/export etc.), then the field values are not shown in search suggest results. This is done to secure the confidential data like passwords and credit card numbers.

AJAX-based dependent dropdown boxes

In PHPRunner dependent dropdown boxes are AJAX Driven. Dropdown boxes content is loaded in the real time using AJAX technologies instead of loading all the content on the web page load. This means web pages are loaded more faster.



Make: Honda

Model: Please select

Please select
Accord
Civic
HR-V
Pilot
Civic Hybrid

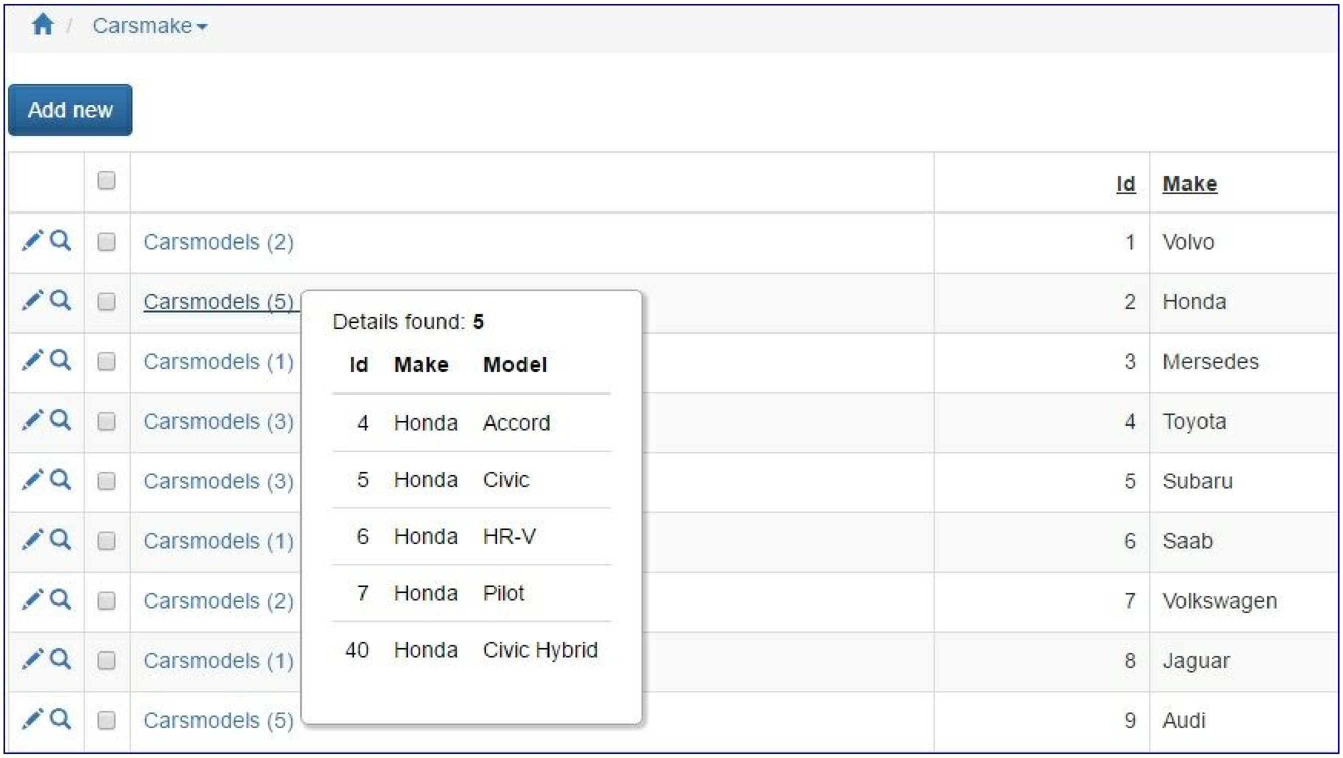
Phone #

Picture

Price

Details Records Preview

In PHPRunner you can see details records preview directly on the list page. All you need to do is to mouse over the link. The following picture demonstrates how the details preview looks like:



	□		Id	Make
✎🔍	□	Carsmodels (2)	1	Volvo
✎🔍	□	Carsmodels (5)	2	Honda
✎🔍	□	Carsmodels (1)	3	Mersedes
✎🔍	□	Carsmodels (3)	4	Toyota
✎🔍	□	Carsmodels (3)	5	Subaru
✎🔍	□	Carsmodels (1)	6	Saab
✎🔍	□	Carsmodels (2)	7	Volkswagen
✎🔍	□	Carsmodels (1)	8	Jaguar
✎🔍	□	Carsmodels (5)	9	Audi

Details found: **5**

Id	Make	Model
4	Honda	Accord
5	Honda	Civic
6	Honda	HR-V
7	Honda	Pilot
40	Honda	Civic Hybrid

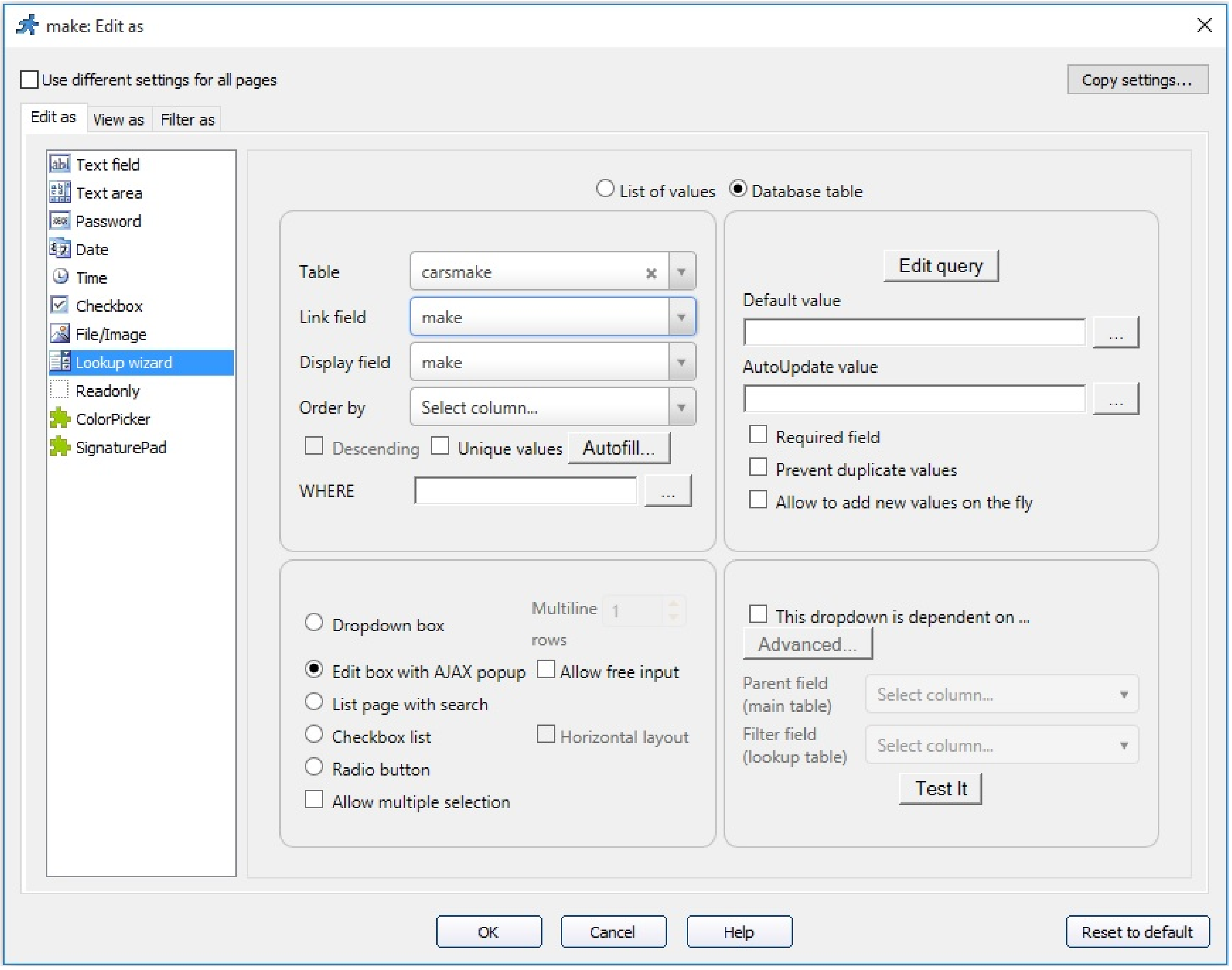
The number of records in the details preview is limited. Only first ten details are displayed. If there is an image in the details then only first five details are displayed.

For more information about how to enable details records preview, see Master-details relationship.

Lookup wizard as Edit box with AJAX popup

This feature is added to lighten the search among the large amount of data. Now you can choose the variant from AJAX popup which is refreshed each time you type in the text box, instead of searching through all the values in the dropdown boxes.

To turn on this feature, you should select the appropriate check box on the **"Edit as" settings** tab.



make: Edit as

□ Use different settings for all pages

Edit as | View as | Filter as

○ List of values ● Database table

Table: carsmake

Link field: make

Display field: make

Order by: Select column...

□ Descending □ Unique values

WHERE:

Default value:

AutoUpdate value:

□ Required field

□ Prevent duplicate values

□ Allow to add new values on the fly

○ Dropdown box Multiline rows: 1

● Edit box with AJAX popup □ Allow free input

○ List page with search

○ Checkbox list □ Horizontal layout

○ Radio button

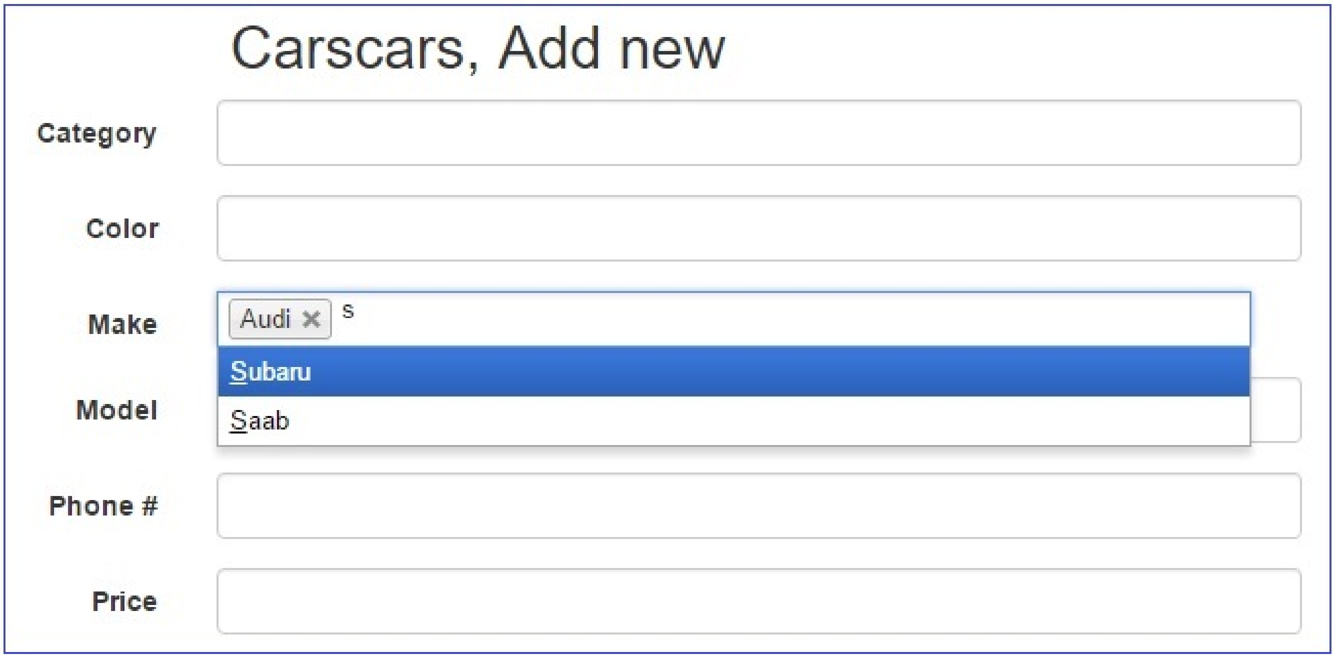
□ Allow multiple selection

□ This dropdown is dependent on ...

Parent field (main table): Select column...

Filter field (lookup table): Select column...

After you build your project, you can see how it works on the Edit, Add, and Search pages. When you begin to type in the edit box AJAX popup appears and you can choose the needed value from the suggest list. If you type in the text box the value which doesn't exist in the database and move to another form control then the border of the text box will change color into red. When you correct the value, it will change back to usual.



The application will look for the occurrence of the typed in string anywhere in the list. For example, when you enter 'co', it will show 'Corolla' and 'Accord'. If you want to change this behavior and make it look for the values starting with the entered value, i.e. 'Corolla' only, add the following code to the AfterAppInit:

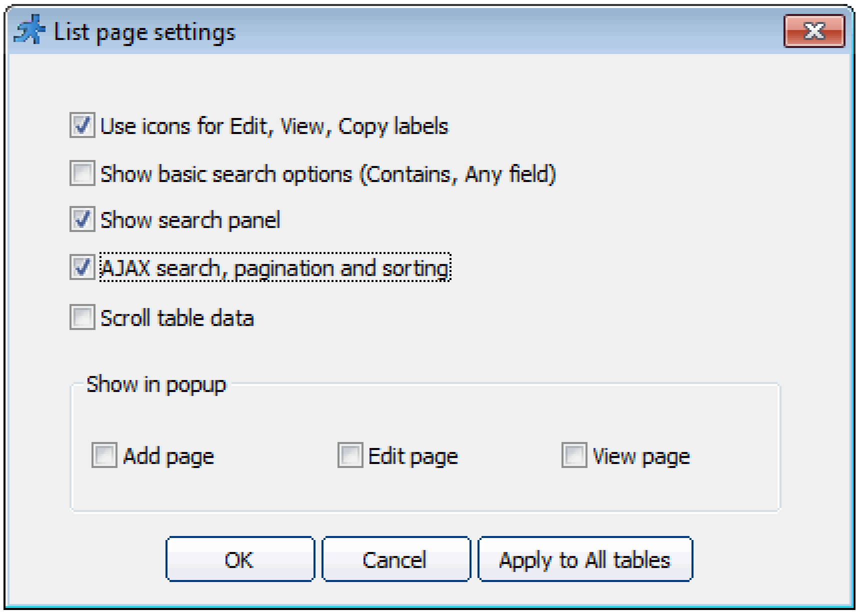
```
$ajaxSearchStartsWith = true;
```

If you try to submit the form with wrong value entered in the text box then the form will be submitted with the previous correct value entry.

AJAX-based pagination/sorting/search

This option enables AJAX search, pagination and sorting that allows updating data without loading the whole page again.

To turn this feature on proceed to the **Choose pages** screen and click '...' button next to the **List page** check box. Then select the **AJAX search, pagination and sorting** check box.



List page settings

☑ Use icons for Edit, View, Copy labels

□ Show basic search options (Contains, Any field)

☑ Show search panel

☑ AJAX search, pagination and sorting

□ Scroll table data

Show in popup

□ Add page □ Edit page □ View page

3.13 Localizing PHPRunner applications

In this article we will cover all the aspects of creating multilingual websites with the help of PHPRunner. This process includes the following steps:

- Translation of system messages
- Translation of table/field names and custom labels
- Translation of data from the database

Translation of system messages

First of all, you need to define the language(s) of standard texts in the website interface or "system messages".

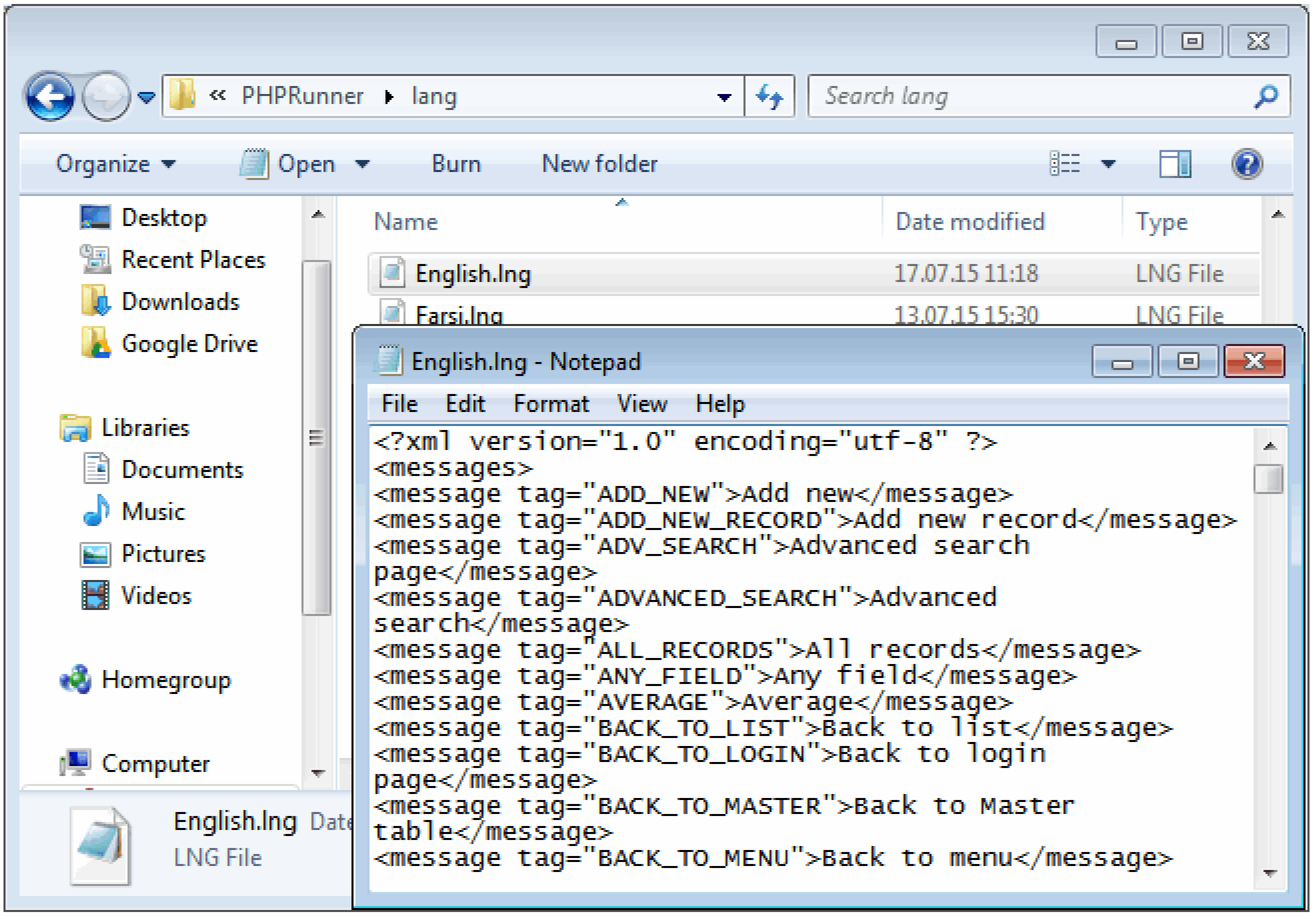
On the Miscellaneous page you can choose one or more languages which your website will support. Use the **Language** drop-down box if you like to choose one language. By clicking **Multiple languages** button you can select several languages and give the user ability to choose language while logging in.

PHPRunner includes translations of system messages into the following languages:

Afrikaans	Arabic	Bosnian	Bulgarian	Catalan	Chinese
Chinese (Hong Kong S.A.R.)	Chinese (Taiwan)	Croatian	Czech	Danish	Dutch

English	Farsi	French	Georgian	German	Greek
Hebrew	Hungarian	Indonesian	Italian	Japanese	Malay
Norwegian (Bokmal)	Polish	Portuguese (Brazil)	Portuguese (Standart)	Romanian	Russian
Slovak	Spanish	Swedish	Tagalog (Phillipines)	Thai	Turkish
Urdu	Welsh				

Translations of system messages are stored in the language files (*.lng) located in the *lang* directory (*C:\Program Files\PHPRunner10.0\lang*). For example, system messages in English language is stored in the *English.lng* file:



```
<?xml version="1.0" encoding="utf-8" ?>
<messages>
<message tag="ADD_NEW">Add new</message>
<message tag="ADD_NEW_RECORD">Add new record</message>
<message tag="ADV_SEARCH">Advanced search
page</message>
<message tag="ADVANCED_SEARCH">Advanced
search</message>
<message tag="ALL_RECORDS">All records</message>
<message tag="ANY_FIELD">Any field</message>
<message tag="AVERAGE">Average</message>
<message tag="BACK_TO_LIST">Back to list</message>
<message tag="BACK_TO_LOGIN">Back to login
page</message>
<message tag="BACK_TO_MASTER">Back to Master
table</message>
<message tag="BACK_TO_MENU">Back to menu</message>
```

To change the translation of system messages in some language, modify the corresponding language file (*.lng).

To add translation of system messages in new language, create a copy of file *English.lng* and translate all phrases there. Then modify the file *languages.cfg* that is also located in the *lang* directory by adding this line (change the values listed in red to match your specific needs):

```
<language filename="YourLanguageFile.lng" name="YourLanguageName"
lcid="YourLocaleID" codepage="YourCodepage" charset="YourCharset" />
```

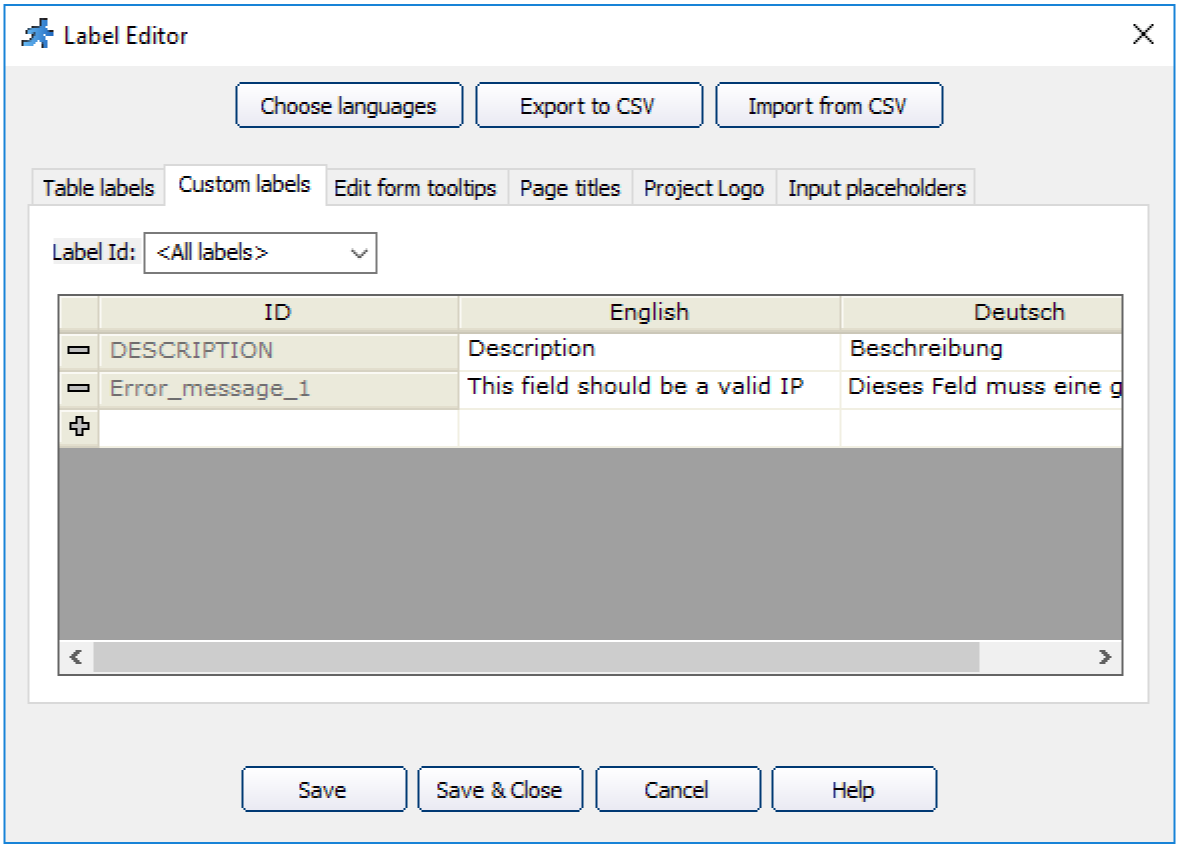
, where

- *filename* - name of the new *.lng* file.
- *name* - language name as it would appear in the PHPRunner wizard.
- *lcid* - locale ID. The list of Locale IDs can be found at http://xlinesoft.com/articles/asp_regional_settings.htm.
- *codepage* - codepage code. The list of codepage codes can be found at http://en.wikipedia.org/wiki/Code_page.
- *charset* - charset code. The list of charset codes can be found at http://www.webcheatsheet.com/html/character_sets_list.php.

Translation of table/field names and custom labels

Use Label Editor on the **Miscellaneous** page to translate table and field names/labels (**Table labels** tab). Also you can add and translate custom labels there (**Custom labels** tab) and add tooltips to the Edit forms (**Edit form tooltips** tab).

Use custom labels to translate menu items, tab/section names, error messages in regular expressions and your own validation plugins etc. When creating menu item, new tab/section or regular expression, use the **Multilanguage** button to create and translate custom label. Use Label Editor (**Custom labels** tab) to create custom labels for error messages in your own validation plugins. In Label Editor you can also create your own custom labels to display some messages to a user.



Label Id: <All labels>

ID	English	Deutsch
DESCRIPTION	Description	Beschreibung
Error_message_1	This field should be a valid IP	Dieses Feld muss eine g

Use the methods listed below to access labels from the PHP code:

Method	Description
GetTableCaption($table)	Returns table caption.
GetFieldLabel($table, $field)	Returns field label.
GetCustomLabel("LABEL_ID")	Returns custom label. *LABEL_ID* - custom label identifier.
GetFieldToolTip($table, $field)	Returns field edit tooltip.
mlang_message($tag)	Returns standard message. You can find the list of message tags in the file *English.lng* (*C:\Program Files\PHPRunner10.0\lang\English.lng*).

Use the methods listed below to access labels from the Javascript code:

Method	Description

Runner.getCustomLabel("LABEL_ID ")	Returns custom label. *LABEL_ID* - custom label identifier.
Runner.lang.constants.CONSTANT _ID	Returns standard message. *CONSTANT_ID* - message constant identifier. You can find the list of constants in the file *RunnerLang.js* (*C:\Program Files\PHPRunner10.0\source\include\common\runnerJS\constants\language\RunnerLang.js)*.

Example 1. Use of custom label in PHP code

If you have created custom label *Message1* and want to place this message on a page, which will vary depending on the selected language, add PHP code snippet to the page and use the following code in it:

```
echo GetCustomLabel ("Message1");
```

Example 2. Use of custom label in Javascript code

If you have created custom label *Error_Message_1* for your validation plugin, use `GetCustomLabel("Error_Message_1")` code in your javascript function. Note that *GetCustomLabel* function is applicable only for editing fields like Add/Edit/Register/List with inline add/edit.

Example 3. Use of field label instead of field name

In this PHP code snippet that sends email with new data we use *GetFieldLabel* method to get field label. Don't forget to replace *table_name* with the correct name of the table.

```
//**********  Send email with new data  ************

$email="test@test.com";
$from="admin@test.com";
$msg="";
$subject="New data record";

foreach($values as $field=>$value)
{
  if(!IsBinaryType(GetFieldType($field)))
    $msg.= GetFieldLabel("table_name", $field) ." : ".$value."\r\n";
}

$ret=runner_mail(array('to' => $email, 'subject' => $subject, 'body' => $msg,
'from'=>$from));
if(!$ret["mailed"])
  echo $ret["message"];
```

Translating data in the database

We can propose two approaches to the translation data from the database:

1. To store data for each language in a separate table and redirect users to the page, depending on the selected language.

Example. To implement this approach we use the following code in the BeforeProcessList event of *mydata_english* table:

```
if ($_SESSION["language"]=="Spanish")
  {
    header("Location: mydata_spanish_list.php");
    exit();
  }
  else if ($_SESSION["language"]=="French")
    {
      header("Location: mydata_french_list.php");
      exit();
    }
```

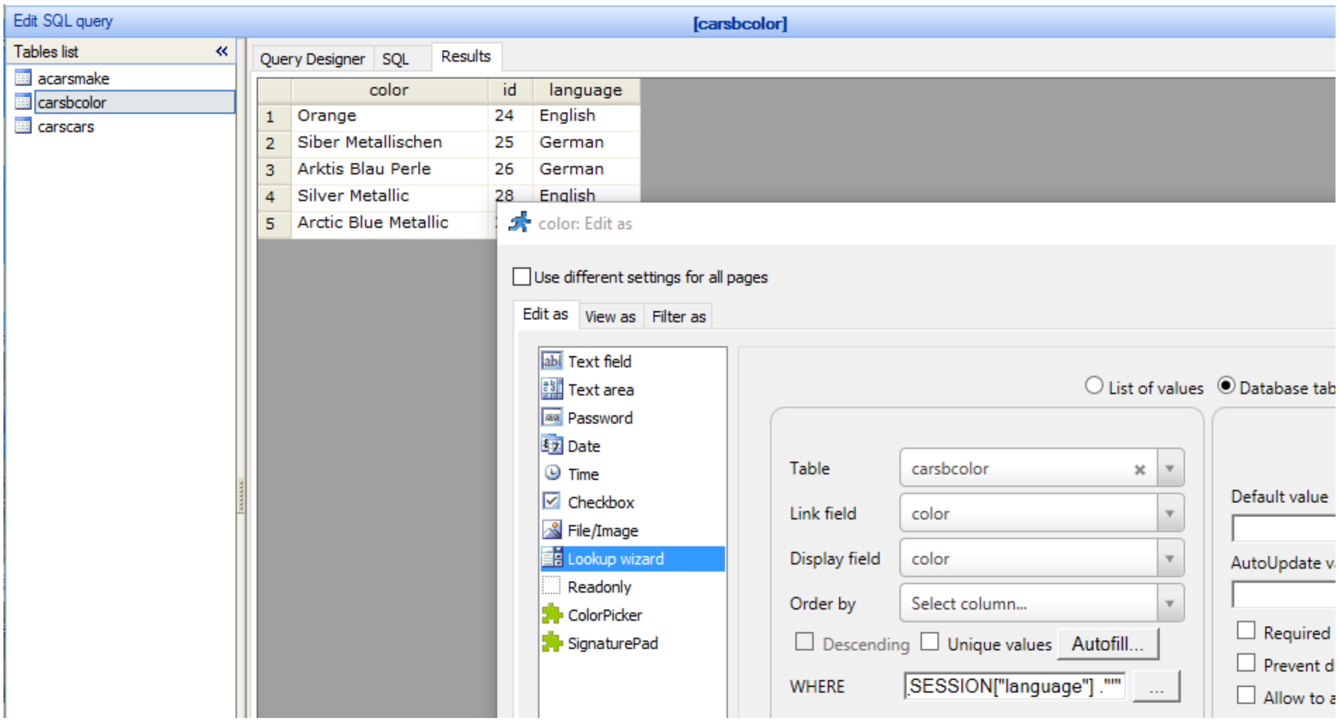
2. To store data for all languages in one table by adding new field for storing language. This approach is easier to configure.

To implement this approach, you can use the method described in the Dynamic SQL query topic and the following code:

```
$strWhereClause = whereAdd($strWhereClause, "language='".
$_SESSION["language"] ."'");
```

If you have lookup table were data is stored on multiple languages, use the following code in the WHERE clause in the Lookup wizard to display the data only for the currently selected language:

```
"language='". $_SESSION["language"] ."'"
```



Edit SQL query [carsbcolor]

Tables list: acarsmake, carsbcolor, carscars

Query Designer | SQL | Results

	color	id	language
1	Orange	24	English
2	Siber Metallischen	25	German
3	Arktis Blau Perle	26	German
4	Silver Metallic	28	English
5	Arctic Blue Metallic		

color: Edit as

☐ Use different settings for all pages

Edit as | View as | Filter as

○ List of values ● Database tab

Table: carsbcolor

Link field: color

Display field: color

Order by: Select column...

☐ Descending ☐ Unique values Autofill...

WHERE: SESSION["language"] ."'"

Default value

AutoUpdate v

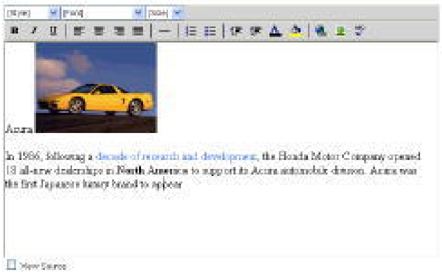
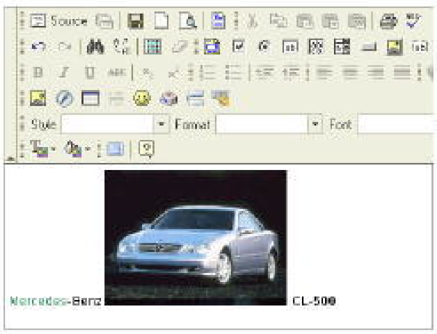
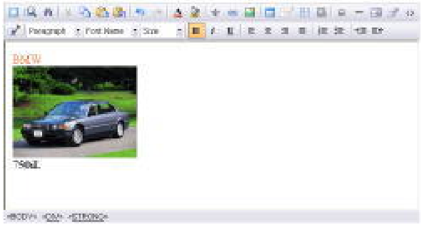
☐ Required

☐ Prevent d

☐ Allow to a

3.14 Rich Text Editor plugins

To choose Rich Text Editor type set Edit type of any text field to **Textarea** and select the **Use RTE** check box. PHPRunner supports the following third party Rich Text Editors:

Basic Rich Text Editor	**CKEditor**	**InnovaStudio Editor**
http://www.kevinroth.com/rte	http://www.ckeditor.com	http://www.innovastudio.com
		

For more info on each editor, documentation, examples etc visit vendor's website.

Configuration

Basic Rich Text Editor

This editor support is built-in into PHPRunner. Nothing to download or configure. Just select it from the list of available Rich Text Editors.

This editor is lightweight and footprint is very small. Documentation, support forum and examples are available at http://www.kevinroth.com/rte/

InnovaStudio Editor

Download InnovaStudio Editor from our website http://www.asprunner.com/files/innovaeditor.zip. Create a folder named *innovaeditor* under *C:\Program files\PHPRunner10.0\source\plugins*.

Unzip InnovaStudio Editor files to this folder - *C:\Program files\PHPRunner10.0\source\plugins\innovaeditor*. After that InnovaStudio Editor will be available for selection on Textarea dialog.

InnovaStudio Editor adds about 1000 files to the generated application, footprint is about 3.5Mb.

InnovaStudio Editor documentation and examples can be found in *documentation* folder.

CKEditor

Proceed to http://www.ckeditor.com and download the latest version of CKEditor. Unzip it into *C:\Program Files\PHPRunner10.0\source\plugins* folder. After that CKEditor will be available for selection on Textarea dialog.

CKEditor adds 400+ files to the generated application, footprint is about 2.5Mb.

CKEditor documentation is available on the Web at http://docs.cksource.com.

To enable the image upload feature **CKFinder** do the following (the instruction below applies to CKFinder 2.5.0):

1. Download CKFinder from http://ckfinder.com/download and unzip it into *C:\Program Files\PHPRunner10.0\source\plugins* folder.

2. Edit file *C:\Program Files\PHPRunner10.0\source\plugins\ckfinder\config.php* in the following way:

2.1 Find the *CheckAuthentication()* function and insert this line just before it (to be able to use session variables):

```
include("../../include/dbcommon.php");
```

2.2 Modify the *CheckAuthentication()* function to perform session validation.

Replace

```
return false;
```

with

```
if (isset($_SESSION['UserID']) && $_SESSION['UserID'] &&
$_SESSION["UserID"]!="Guest")
return true;
```

2.3 Define *$baseDir* variable. *$baseDir* - the path to the local directory (in the server). This is the path used by CKFinder to handle the files in the server. Full write permissions must be granted to this directory. You may point it to a directory directly:

```
$baseDir="C:\\Inetpub\\WWWROOT\\mywebsite.com\\project1\\plugins\\ckfinder\
\userfiles\\";
```

or

```
$baseDir="/home/var/www/mywebsite.com/project1/plugins/ckfinder/userfiles/";
```

You can get more information about CKFinder configuration settings at http://docs.cksource.com/CKFinder_2.x/Developers_Guide/PHP/Configuration.

3. Edit the file *C:\Program Files\PHPRunner10.0\source\plugins\ckeditor\config.js*.

Replace

```
CKEDITOR.editorConfig = function( config )
{
   // Define changes to default configuration here. For example:
   // config.language = 'fr';
   // config.uiColor = '#AADC6E';
};
```

with

```
CKEDITOR.editorConfig = function( config )
{
config.filebrowserBrowseUrl = 'plugins/ckfinder/ckfinder.html';
config.filebrowserImageBrowseLinkUrl =
'plugins/ckfinder/ckfinder.html?type=Images';
config.filebrowserFlashBrowseUrl =  'plugins/ckfinder/ckfinder.html?type=Flash';
config.filebrowserUploadUrl =
'plugins/ckfinder/core/connector/php/connector.php?
command=QuickUpload&type=Files';
config.filebrowserImageUploadUrl =
'plugins/ckfinder/core/connector/php/connector.php?
command=QuickUpload&type=Images';
config.filebrowserFlashUploadUrl =
'plugins/ckfinder/core/connector/php/connector.php?
command=QuickUpload&type=Flash';
};
```

4. Edit file *C:\Program Files\PHPRunner10.0\source\files.txt*.

Replace

```
##if @BUILDER.m_strRTEType==RTE_CK##
plugins\ckeditor plugins\ckeditor
##endif##
```

with

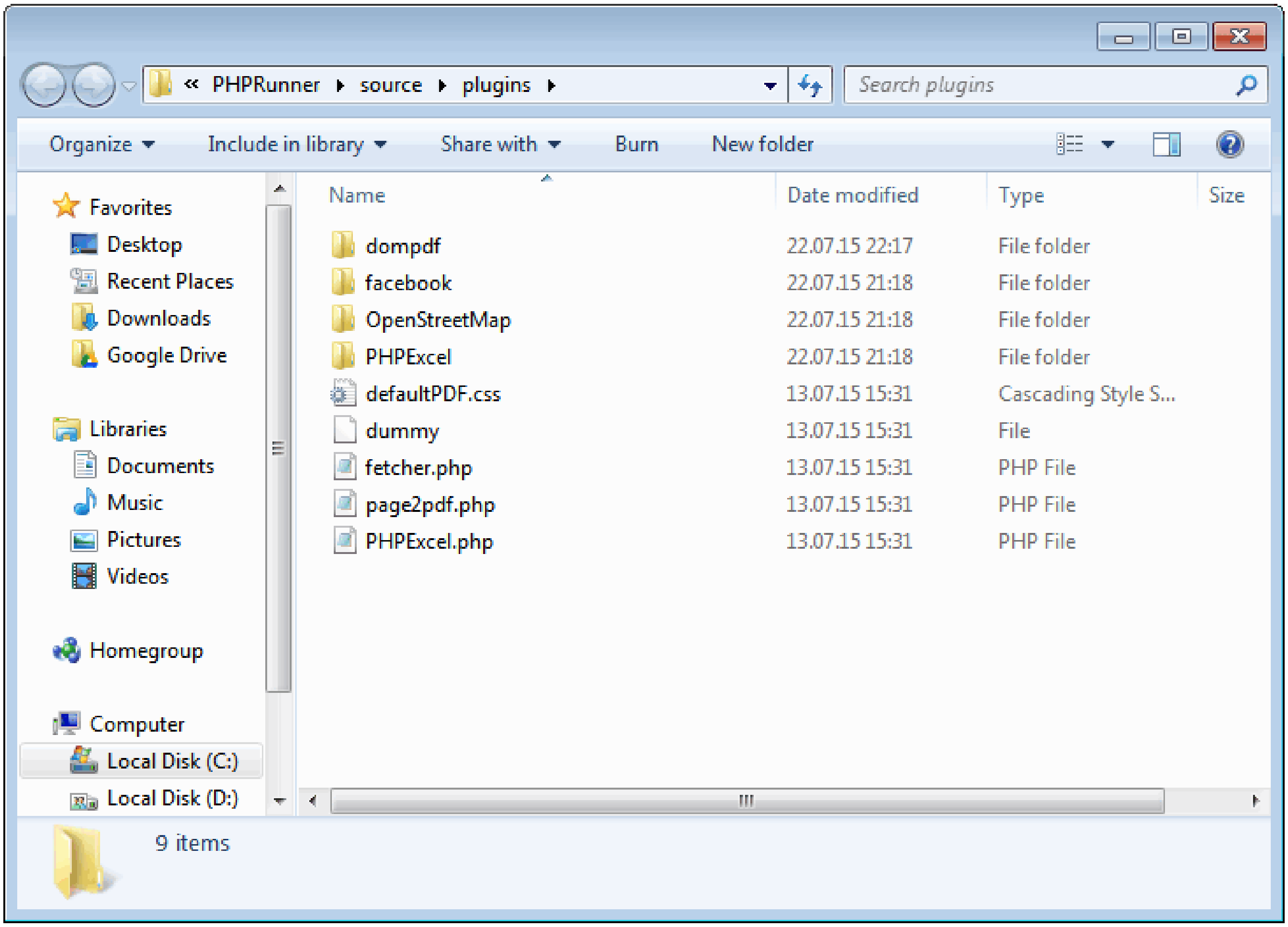
```
##if @BUILDER.m_strRTEType==RTE_CK##
plugins\ckeditor plugins\ckeditor
plugins\ckfinder plugins\ckfinder
##endif##
```

5. Build your project enabling CKEditor option.

CKFinder documentation is available on the Web at http://docs.cksource.com.

3.15 PDF view settings

To enable **PDF view** option download PDF View plugin at http://www.asprunner.com/files/plugins/dompdf.zip. Unzip it to *C:\Program files\PHPRunner10.0\source\plugins* directory.



Name	Date modified	Type	Size
dompdf	22.07.15 22:17	File folder	
facebook	22.07.15 21:18	File folder	
OpenStreetMap	22.07.15 21:18	File folder	
PHPExcel	22.07.15 21:18	File folder	
defaultPDF.css	13.07.15 15:31	Cascading Style S...	
dummy	13.07.15 15:31	File	
fetcher.php	13.07.15 15:31	PHP File	
page2pdf.php	13.07.15 15:31	PHP File	
PHPExcel.php	13.07.15 15:31	PHP File	

Then proceed to the **Choose pages** screen in PHPRunner and select the **PDF View** check box.

This option is global and will be applied to all tables, views and reports in your project. If this option turned on "PDF view" link appears on **View** and **Print All** pages.

This plugin requires about 46 Mb on your hard drive and on the web server.

Note: make sure *templates_c* directory in the output directory is writable by the web server user. On Windows server add full permissions on this directory for IUSR_... user. On Unix server set 766 permissions on this directory. This can be done via your website control panel or via shell (chmod 766 templates_c).

3.16 Web interface guide

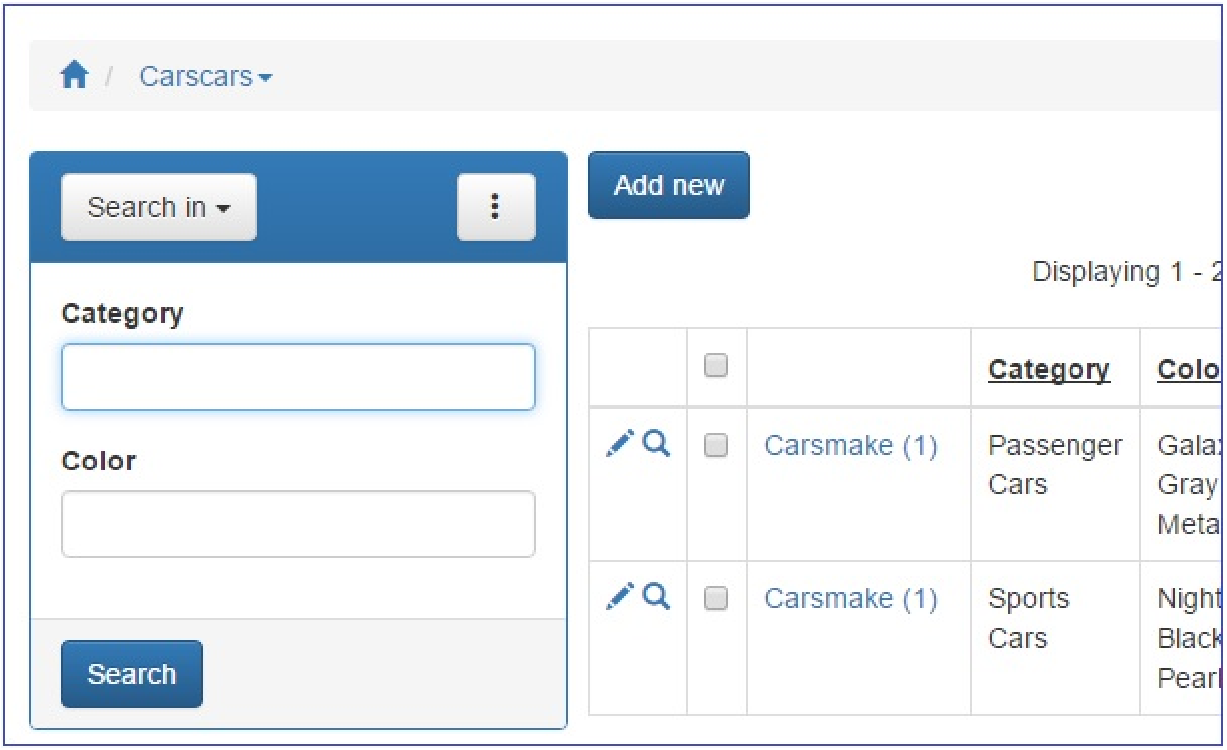
Quick jump

Search panel on the List page

Column resizing

Search panel on the List page

The search panel on the List page allows you to manage the search criteria. To display the search options, click the button with double arrows . You can make the search panel floating clicking the pin button .



Category

Color

Displaying 1 - 2

	□		Category	Colo
	□	Carsmake (1)	Passenger Cars	Gala Gray Meta
	□	Carsmake (1)	Sports Cars	Night Black Pearl

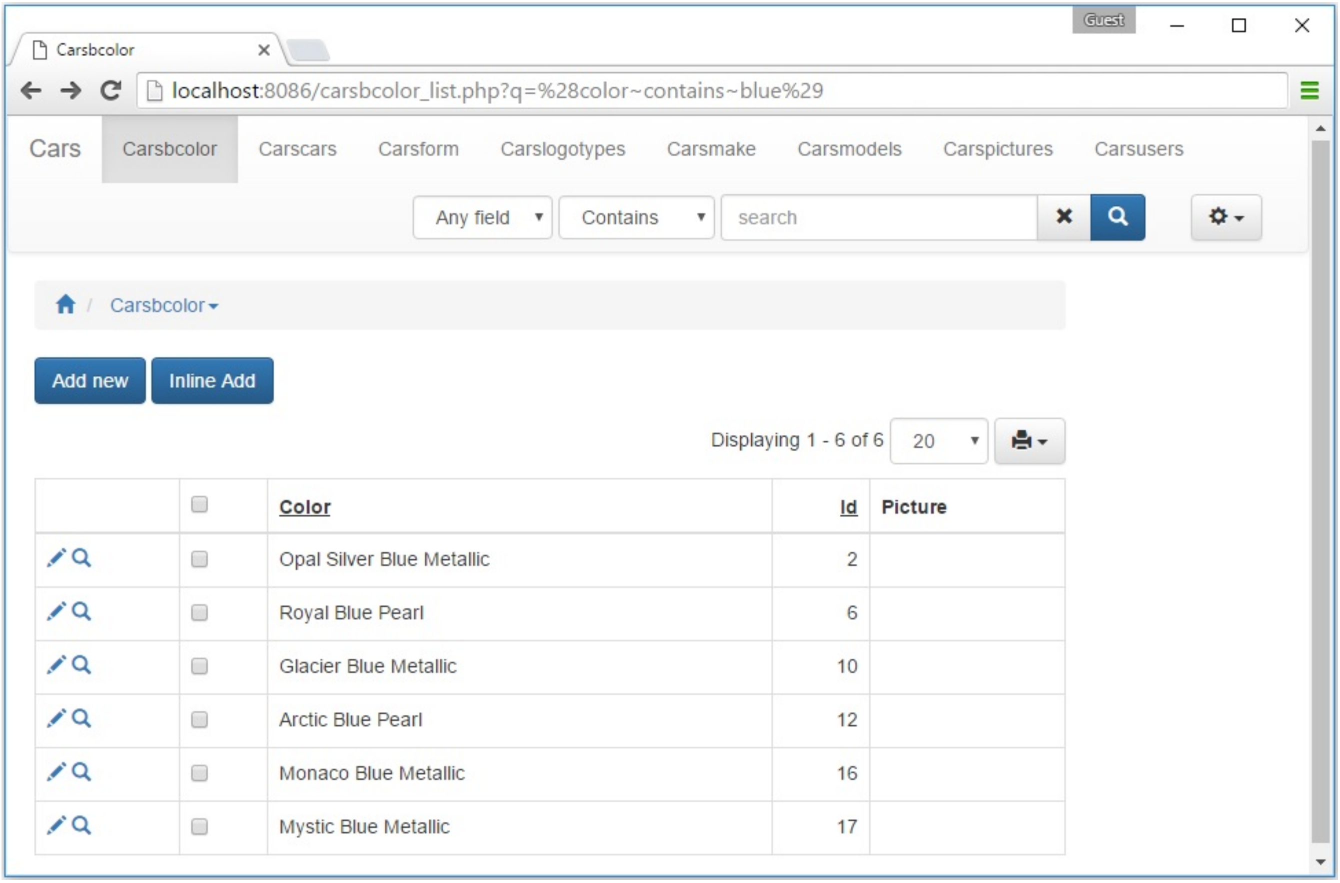
To display fields on the search panel, go to the Choose fields page, click the **Search settings** button and select the corresponding check boxes under **Search Panel**.

Use **Add field** button to add fields to the search criteria. To remove a field from the search panel, move the mouse to that field and click the close button ☒.

You can add the same field twice which will produce *OR* search (e.g. username='admin' or username='test'). By default *Contains* search is implemented. To change this click **Show options** link.

Search panel and simple search box can be used simultaneously that will produce *AND* search.

You can use convenient data filtering via URL parameters. Sample search URL: *carsmodels_list.php?q=(make~equals~Volvo)(<model~contains~70)*. This sort of parameters can be added to any List, Print, Report or Chart pages.



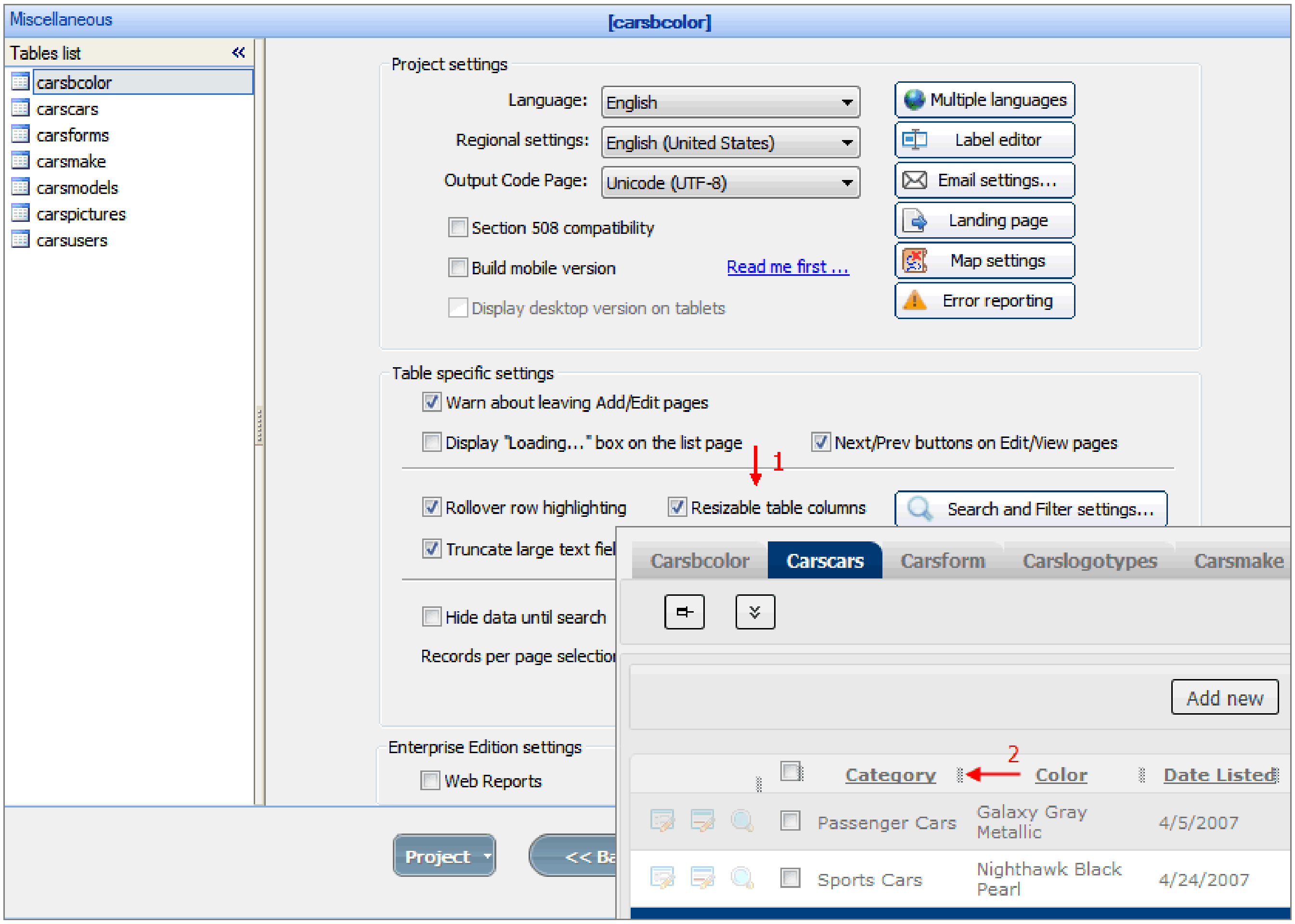
localhost:8086/carsbcolor_list.php?q=%28color~contains~blue%29

Displaying 1 - 6 of 6 20

	□	Color	Id	Picture
	□	Opal Silver Blue Metallic	2	
	□	Royal Blue Pearl	6	
	□	Glacier Blue Metallic	10	
	□	Arctic Blue Pearl	12	
	□	Monaco Blue Metallic	16	
	□	Mystic Blue Metallic	17	

Column resizing

You can resize any column on the **List** page by dragging its edge. To enable this option, proceed to the **Miscellaneous** page and select the **Resizable table columns** check box.



Miscellaneous

[carsbcolor]

Tables list

- carsbcolor
- carscars
- carsforms
- carsmake
- carsmodels
- carspictures
- carsusers

Project settings

Language: English

Regional settings: English (United States)

Output Code Page: Unicode (UTF-8)

☐ Section 508 compatibility

☐ Build mobile version

☐ Display desktop version on tablets

Table specific settings

☑ Warn about leaving Add/Edit pages

☐ Display "Loading..." box on the list page

☑ Next/Prev buttons on Edit/View pages

☑ Rollover row highlighting

☑ Resizable table columns

☑ Truncate large text fiel

☐ Hide data until search

Records per page selectio

Enterprise Edition settings

☐ Web Reports

	Category	Color	Date Listed
☐	Passenger Cars	Galaxy Gray Metallic	4/5/2007
☐	Sports Cars	Nighthawk Black Pearl	4/24/2007

3.17 Mobile Template

Mobile template allows you to build mobile version of your web application (requires PHPRunner 6 or later). To see how it works, you can open mobile-enabled live demos available at http://xlinesoft.com/phprunner/livedemo4.htm from your mobile device.

Mobile Template is a paid add-on and can be purchased for $50. Buy now. *PHPRunner10.0* requires Mobile template v2. Upgrade from previous version of Mobile template $25. Buy now.

You need Mobile template if you use "Old style" layouts. For Bootstrap-based layouts Mobile template is not required.

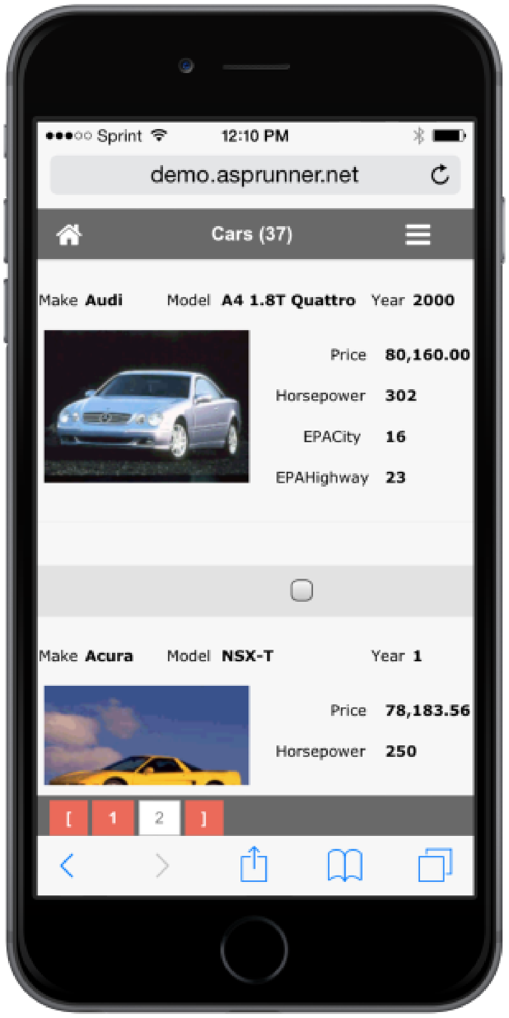
To build a mobile version of your website, proceed to **Miscellaneous** page in PHPRunner and select **Build Mobile version** check box. PHPRunner detects mobile devices automatically and redirects to the mobile version of your website. To display desktop version on tablets, clear **Build Mobile version** check box.

When **Build mobile version** option is enabled on **Miscellaneous** page, **Desktop/Mobile** radio-buttons appear above the list of tables in Visual Editor. You can switch between normal (desktop) or mobile version of the current page and make changes to each version independently.

You can customize mobile pages appearance for any chosen page in Visual Editor i.e. add a button or change the page layout.

Here is the mobile pages on iPhone:

List page



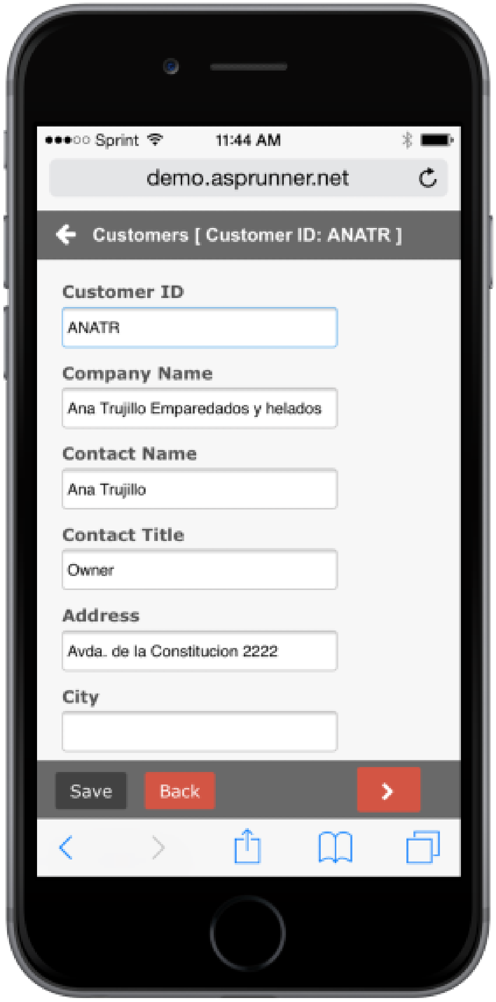
Make **Audi** Model **A4 1.8T Quattro** Year **2000**

Price **80,160.00**
Horsepower **302**
EPACity **16**
EPAHighway **23**

Make **Acura** Model **NSX-T** Year **1**

Price **78,183.56**
Horsepower **250**

Edit page



Customer ID
ANATR

Company Name
Ana Trujillo Emparedados y helados

Contact Name
Ana Trujillo

Contact Title
Owner

Address
Avda. de la Constitucion 2222

City

Installation

To install mobile template for 32-bit version of PHPRunner unzip the template to *C:\Program Files (x86)\PHPRunner10.0\userLayouts\Mobile*

To install mobile template for 64-bit version of PHPRunner unzip the template to *C:\Program Files\PHPRunner10.0\userLayouts\Mobile*

Notes

How do I find if application is being accessed from mobile device?

PHPRunner provides *MobileDetected()* function that returns true if application is accessed from mobile device. For example, you can hide certain tables from mobile users. Use *ModifyMenuItem* event for this purpose:

```
if ($menuItem->getTable() == 'Cars' && MobileDetected()) {
    return false;
}
return true;
```

How do I display different header when application runs on mobile device?

MobileDetected() function comes to rescue again. Proceed to the **Visual Editor**, open Header page and switch to HTML mode. Here is the sample code:

```
<?php

if (MobileDetected())
   echo "<font size='3'>My website</font>";
else
   echo "<font size='6'>My website</font>";

?>
```

Testing

You can test test your mobile app on your mobile device as well as on the desktop: Testing web apps in mobile mode.

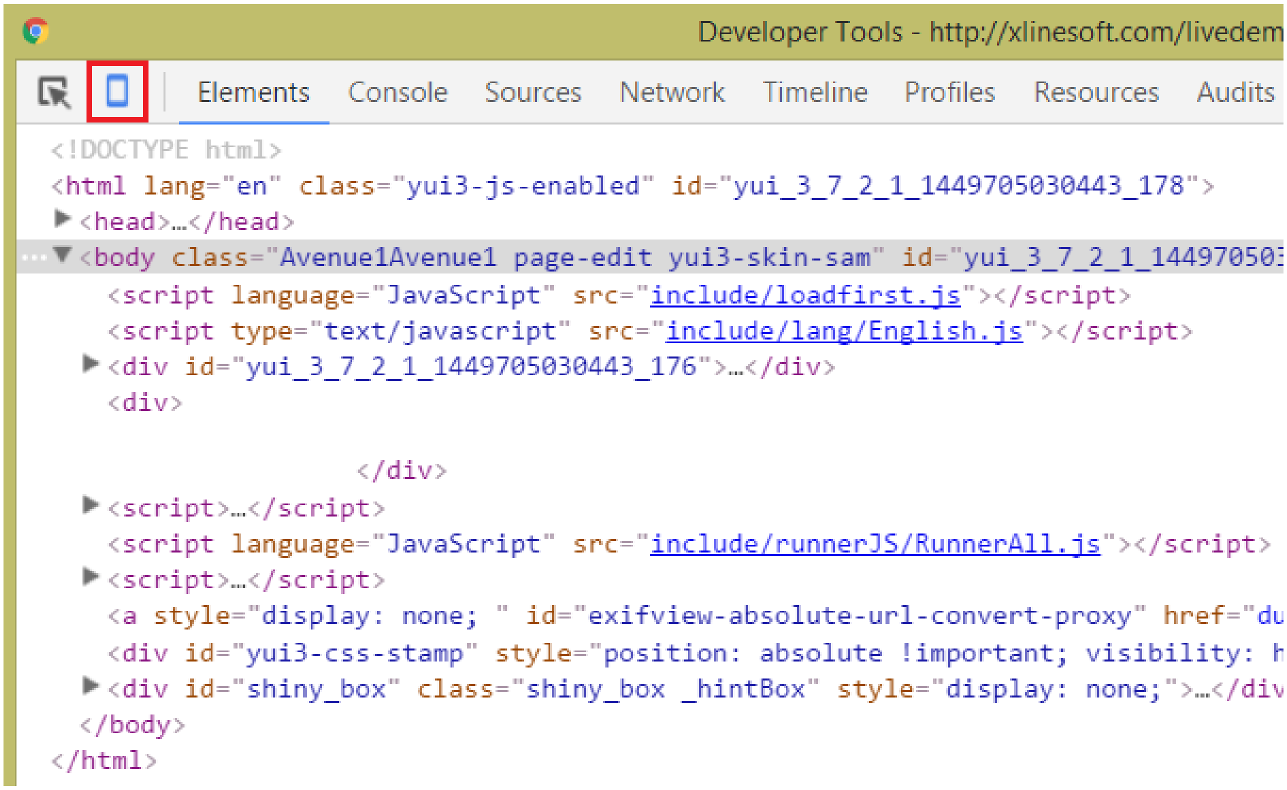
3.18 Testing web apps in mobile mode

There are a few different ways of how you can quickly test the mobile version of your application.

1. Using your web browser

All modern browsers have an option to show any page the way it will appear on mobile device. There is a way how to do this using Chrome web browser.

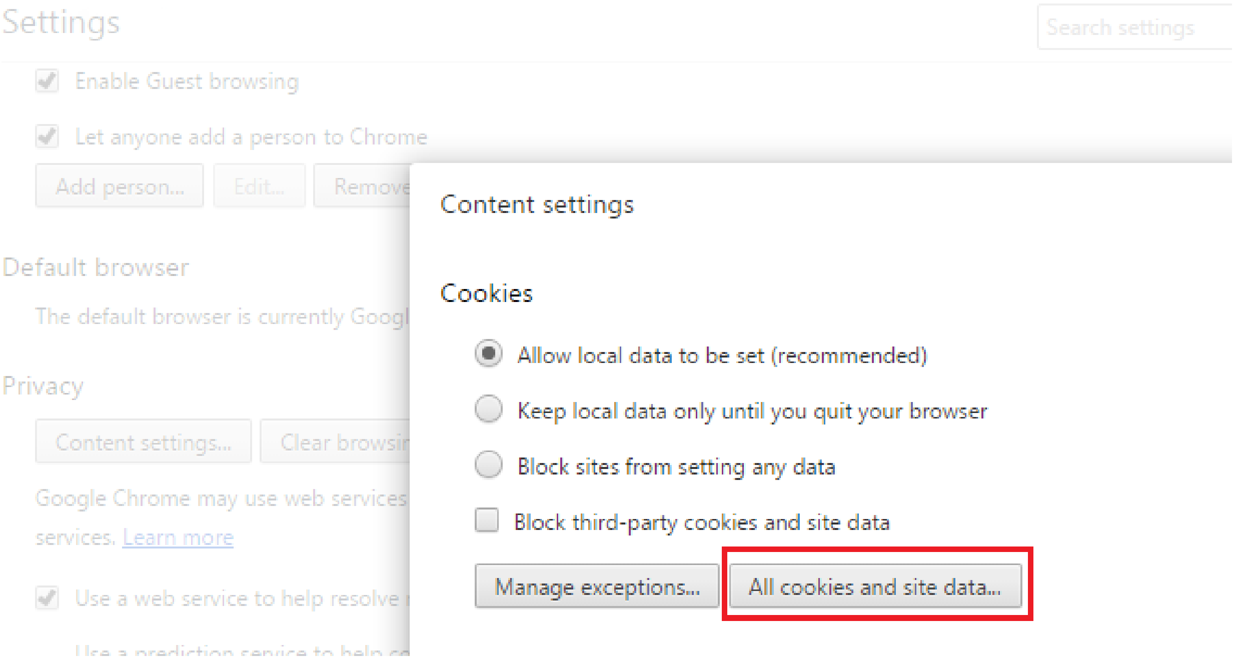
1.1 Hit F12 to open Chrome Developers Tools. Click Mobile device icon in Chrome Developer tools.



```
<!DOCTYPE html>
<html lang="en" class="yui3-js-enabled" id="yui_3_7_2_1_1449705030443_178">
▶<head>…</head>
▼<body class="Avenue1Avenue1 page-edit yui3-skin-sam" id="yui_3_7_2_1_14497050:
    <script language="JavaScript" src="include/loadfirst.js"></script>
    <script type="text/javascript" src="include/lang/English.js"></script>
  ▶<div id="yui_3_7_2_1_1449705030443_176">…</div>
    <div>

                    </div>
  ▶<script>…</script>
    <script language="JavaScript" src="include/runnerJS/RunnerAll.js"></script>
  ▶<script>…</script>
    <a style="display: none; " id="exifview-absolute-url-convert-proxy" href="du
    <div id="yui3-css-stamp" style="position: absolute !important; visibility: h
  ▶<div id="shiny_box" class="shiny_box _hintBox" style="display: none;">…</div
  </body>
</html>
```

1.2 Your app now switches to mobile mode but since we store device info in cookies/session you need to reset cookies for this app. The best thing is to use **EditThisCookie Chrome** plugin as it only takes a couple of mouse clicks to clear cookies for the current website.

You can also do that using built-in tools. You can delete all cookies for selected website. Just make sure you are not deleting all cookies for all websites.



Content settings

Cookies

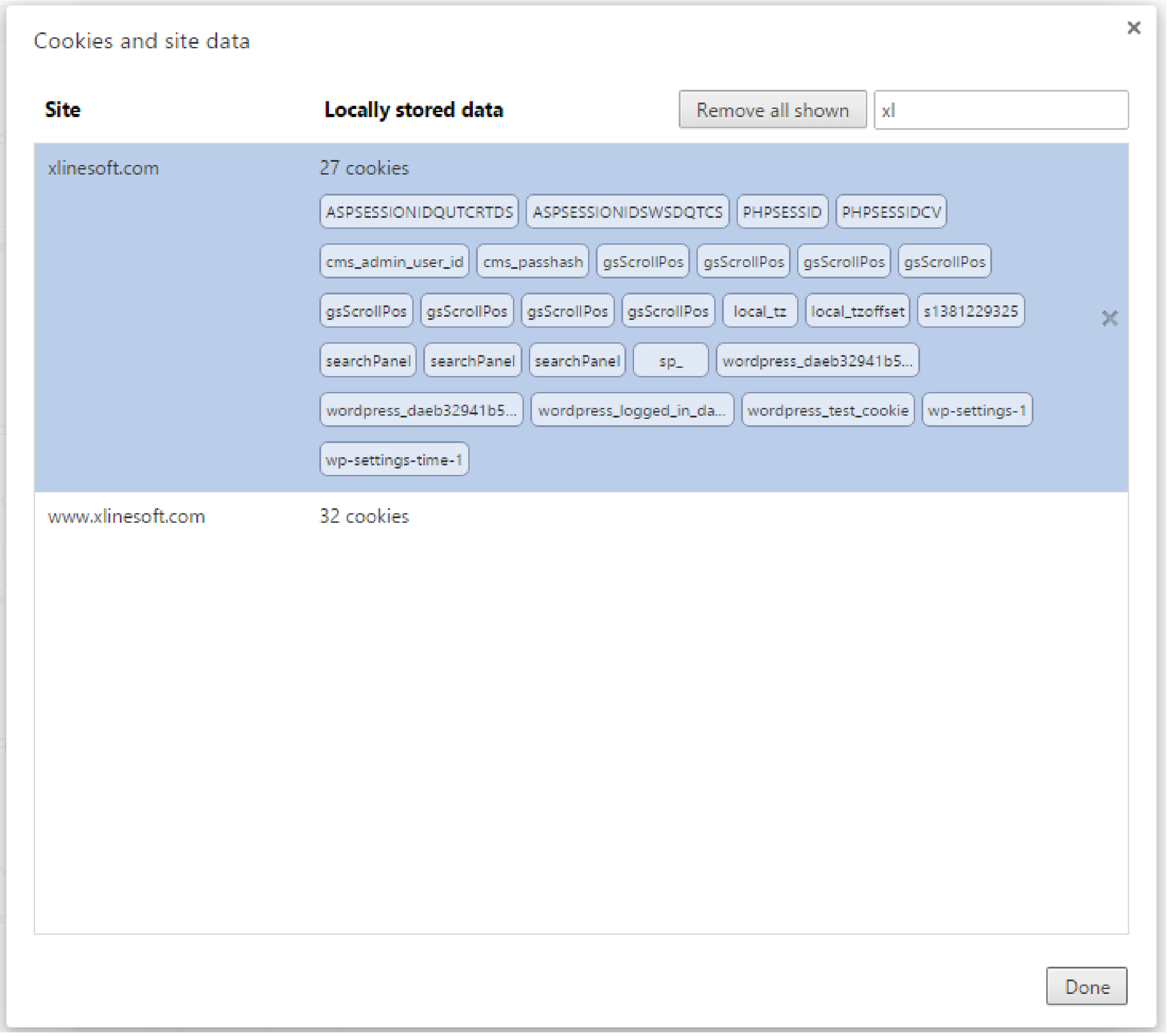
● Allow local data to be set (recommended)

○ Keep local data only until you quit your browser

○ Block sites from setting any data

□ Block third-party cookies and site data

Manage exceptions...
All cookies and site data...
Settings
Search settings
Enable Guest browsing
Let anyone add a person to Chrome
Add person...
Edit...
Remove
Default browser
The default browser is currently Googl
Privacy
Content settings...
Clear browsi
Google Chrome may use web services
services. Learn more
Use a web service to help resolve

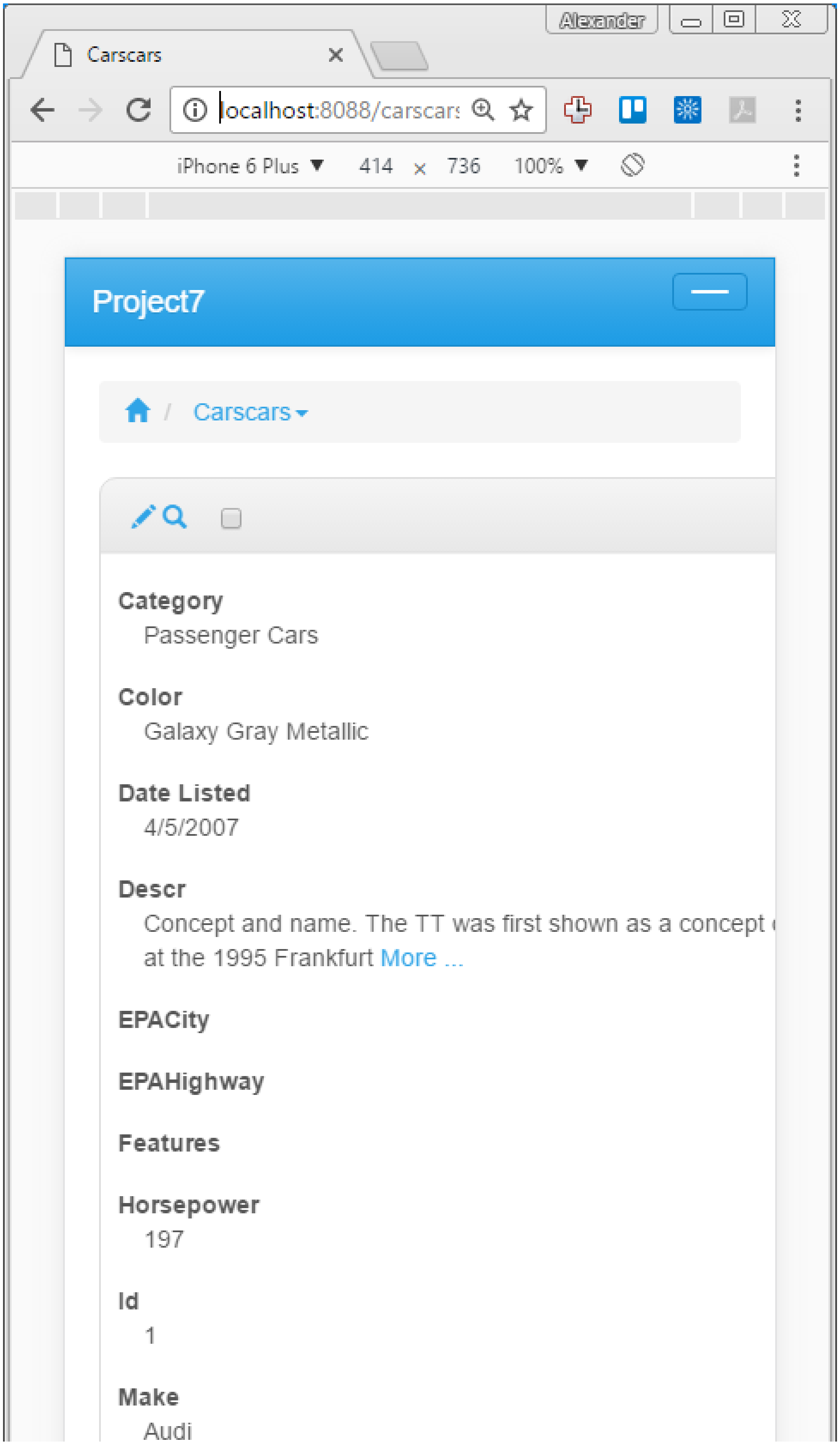


Cookies and site data

Site	Locally stored data
xlinesoft.com	27 cookies ASPSESSIONIDQUTCRTDS, ASPSESSIONIDSWSDQTCS, PHPSESSID, PHPSESSIDCV cms_admin_user_id, cms_passhash, gsScrollPos, gsScrollPos, gsScrollPos, gsScrollPos gsScrollPos, gsScrollPos, gsScrollPos, gsScrollPos, local_tz, local_tzoffset, s1381229325 searchPanel, searchPanel, searchPanel, sp_, wordpress_daeb32941b5... wordpress_daeb32941b5..., wordpress_logged_in_da..., wordpress_test_cookie, wp-settings-1 wp-settings-time-1
www.xlinesoft.com	32 cookies

xl

1.3 If you done everything right this is what you going to see. You can choose different device types, horizontal or vertical orientation etc.



Category
Passenger Cars

Color
Galaxy Gray Metallic

Date Listed
4/5/2007

Descr
Concept and name. The TT was first shown as a concept at the 1995 Frankfurt More ...

EPACity

EPAHighway

Features

Horsepower
197

Id
1

Make
Audi

2. Using your mobile device

If your mobile device and desktop computer are on the same network you can test your app using your smartphone or tablet. Note that built-in web server that comes with PHPRunner is only designed for local testing. You will need to use your own web server like XAMPP or IIS.

Choose output folder under your web server root folder i.e. *C:\xampp\htdocs\project1* then open a URL like *http://desktop_computer_name/project1* in your browser on mobile device.

Another option - post your project to Demo Account, copy the link, email it to yourself. On your mobile device open that email and click the link.

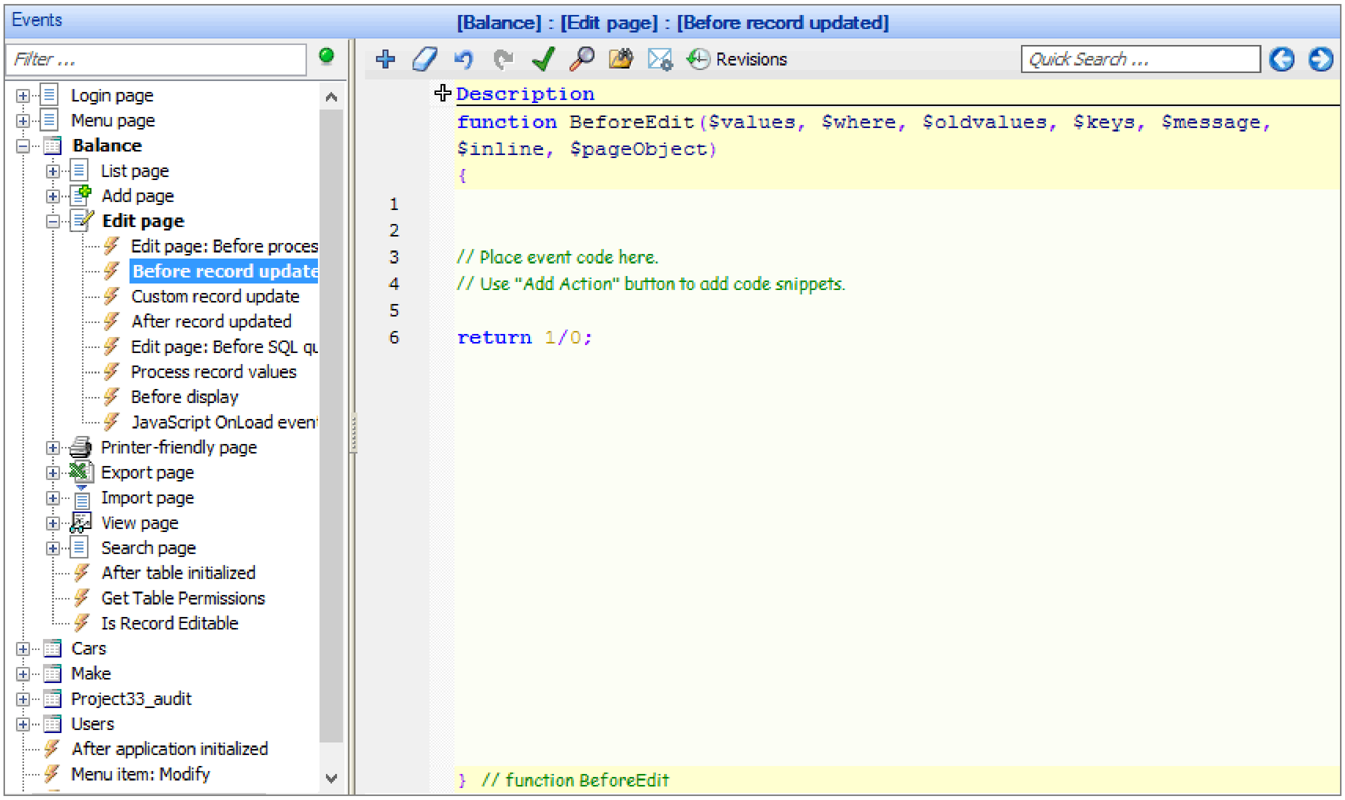
3.19 Error reporting

You can display your custom error message instead of detailed error messages. To enable this option, go to **Miscellaneous** page -> Error reporting.

Displaying server errors in AJAX mode

From now on you can view the details of server errors that occur in AJAX mode while browsing generated web pages. Here is an example.

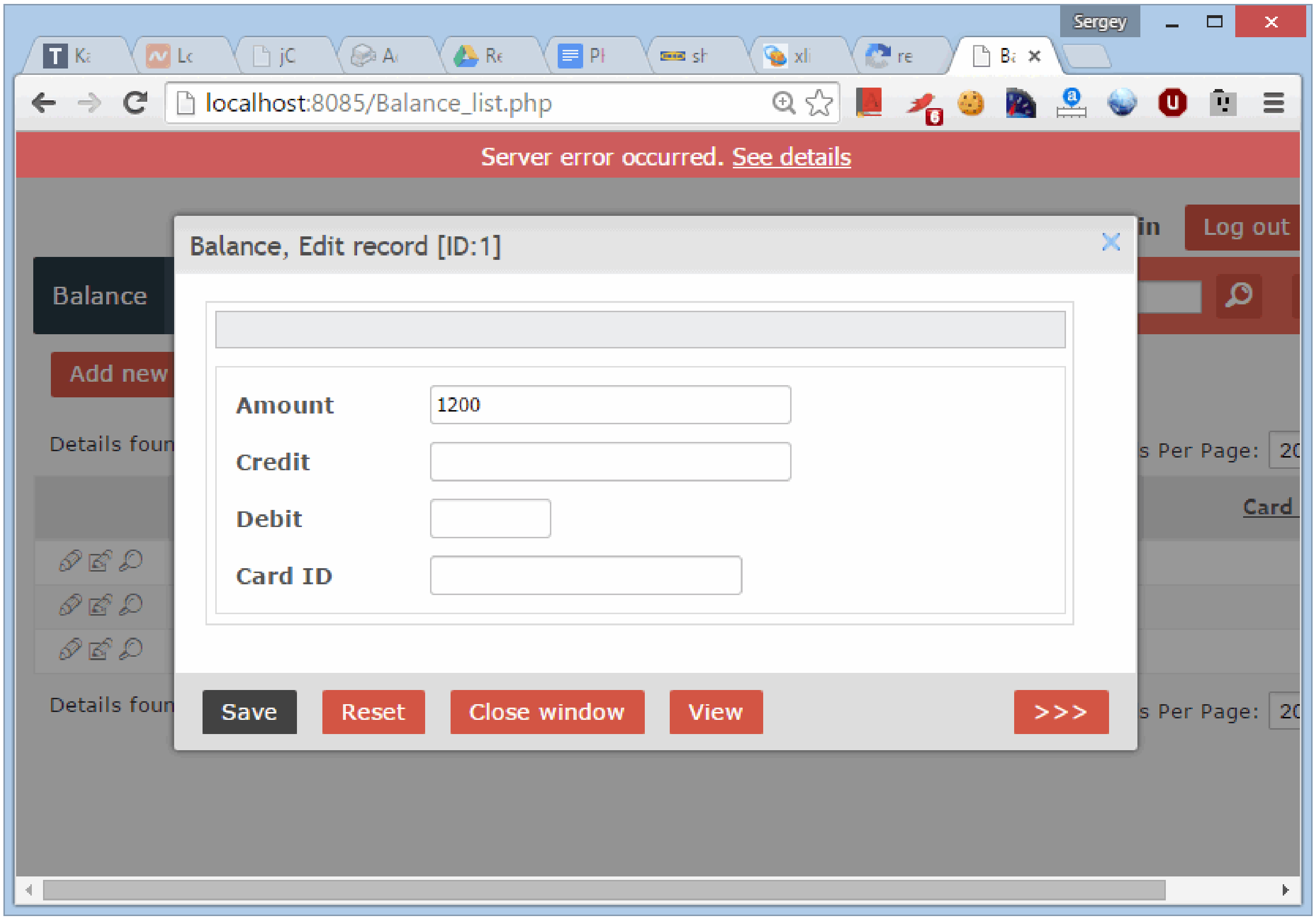
There is an Edit page, which is displayed in popup. BeforeEdit event of this page contains an error - division by 0.



[Balance] : [Edit page] : [Before record updated]

```
function BeforeEdit($values, $where, $oldvalues, $keys, $message,
$inline, $pageObject)
{


// Place event code here.
// Use "Add Action" button to add code snippets.

return 1/0;
} // function BeforeEdit
```

When trying to save the updated record, you will see an 'error occurred' message at the top of the page.



Server error occurred. See details

Balance, Edit record [ID:1]

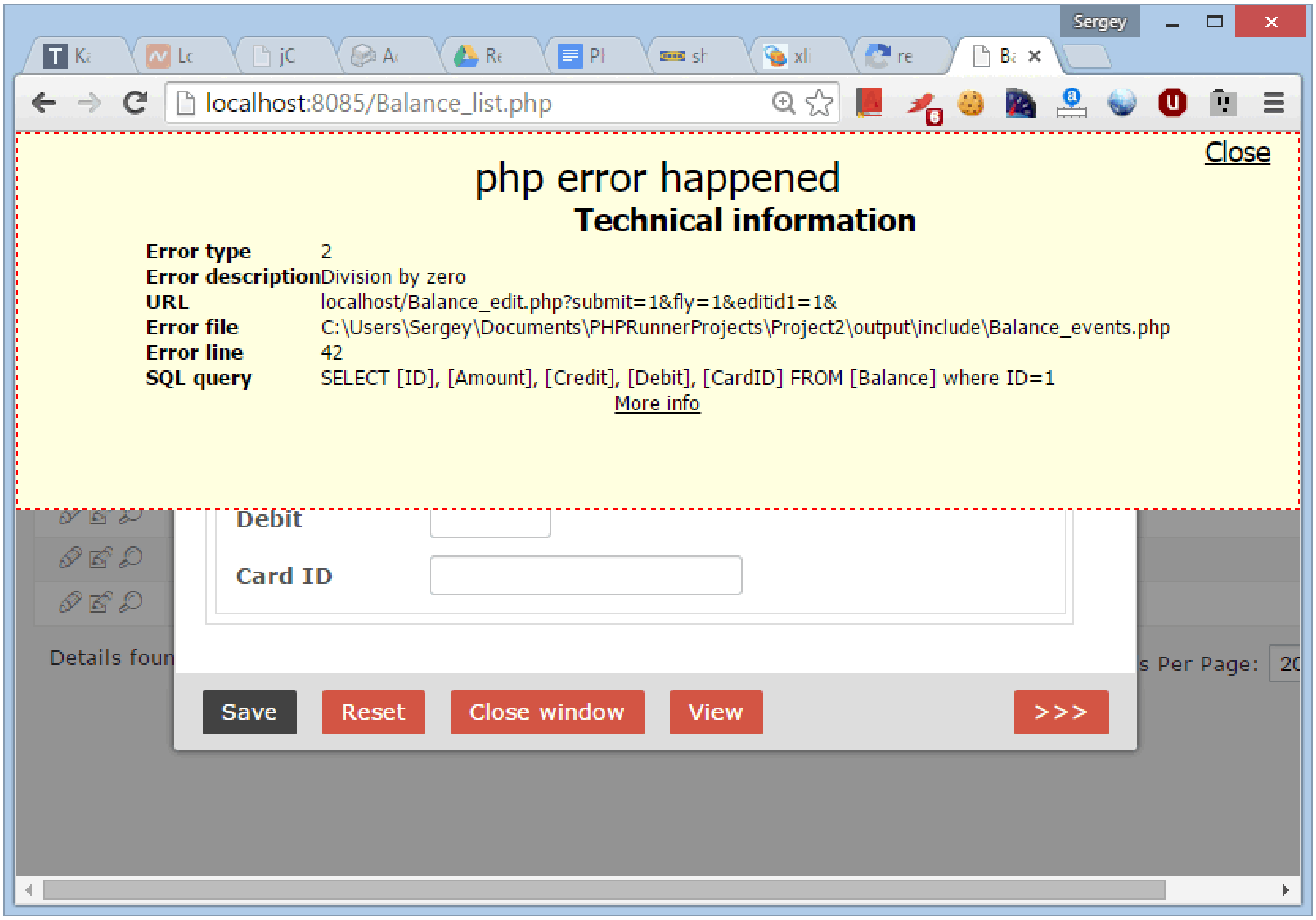
Amount: 1200

Credit:

Debit:

Card ID:

Click **See details** to view extended error information.



php error happened

Technical information

```
Error type         2
Error description  Division by zero
URL                localhost/Balance_edit.php?submit=1&fly=1&editid1=1&
Error file         C:\Users\Sergey\Documents\PHPRunnerProjects\Project2\output\include\Balance_events.php
Error line         42
SQL query          SELECT [ID], [Amount], [Credit], [Debit], [CardID] FROM [Balance] where ID=1
```

4 Order PHPRunner online

PHPRunner may be ordered through our online store seven days a week, 24 hours a day.

Click here to order your copy now!

Visit phprunner online

XLineSoft software development