

# SQL Manager.net™

EMS® Database Management Solutions



## SQL Administrator User's Manual

© 2025 EMS Software Development

# SQL Administrator User's Manual

© 2025 EMS Software Development

All rights reserved.

This manual documents EMS SQL Administrator.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Use of this documentation is subject to the following terms: you may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way.

Document generated on: 20.01.2025

# Table of Contents

<b>Part I Welcome to EMS SQL Administrator!</b>	<b>14</b>
What's new .....	15
Feature Matrix .....	17
System requirements .....	18
Installation .....	19
Purchasing .....	20
How to register SQL Administrator .....	22
Version history .....	24
EMS SQL Administrator FAQ .....	28
Other EMS Products .....	31
<b>Part II Getting Started</b>	<b>38</b>
Work area .....	40
Server Explorer .....	45
Database Explorer .....	47
Working with servers .....	49
Main menu & Toolbars .....	50
Working with windows .....	54
Working with grid .....	59
Filter Builder dialog .....	63
<b>Part III Servers</b>	<b>66</b>
Register server/Registration properties .....	69
Configuring general connection settings .....	70
WMI Connection .....	72
Managing alarms .....	73
Setting statistics collection parameters .....	78
Register servers .....	80
Alarm summary .....	81
Alarm Log .....	85
SQL Administrator Repository .....	86
WMI Connection .....	89
Statistics Collector .....	91
Server properties .....	94
Memory .....	96
Processor .....	98
Parallelism .....	100
Security .....	102

Surface area configuration .....	104
Connections .....	106
Database options .....	108
Full-text search .....	110
Miscellaneous .....	112
Network .....	115
<b>Compare server configurations .....</b>	<b>116</b>
<b>Selecting operation .....</b>	<b>117</b>
<b>Saving configuration .....</b>	<b>118</b>
Setting saving configuration options.....	118
View ing saving configuration information.....	119
Executing saving configuration operation.....	120
<b>Comparing configurations .....</b>	<b>121</b>
Specifying configurations for comparing.....	121
Setting comparing configurations options.....	122
View ing comparing configurations information.....	124
Executing comparing operation.....	125
View ing comparison results.....	126
Selecting server to update.....	127
Specifying destination file name.....	128
Setting passw ords.....	130
View ing update information.....	131
Executing updating operation.....	132
<b>Deleting configuration .....</b>	<b>133</b>
Selecting server for deleting configuration.....	133
Selecting configurations to delete.....	134
View ing delele configuration information.....	135
Executing delete configuration operation.....	136

## Part IV Database Maintenance

## 138

<b>Maintenance plans .....</b>	<b>141</b>
<b>Maintenance Plan Editor .....</b>	<b>143</b>
Using Navigation bar.....	144
Setting general options.....	146
Managing tasks.....	147
Backup database.....	149
Advanced .....	151
Check database .....	154
Rebuild indexes .....	156
Reorganize indexes.....	158
Shrink database .....	160
Update statistics .....	162
Execute job .....	164
Execute T-SQL statement.....	166
Notify operator .....	168
Clean up history .....	170
Clean up maintenance log.....	172
Precedence constraints.....	174
Adding connections.....	176
Managing schedules.....	178
Setting reporting options.....	179
View ing maintenance plan history.....	181
<b>Maintenance plan History .....</b>	<b>182</b>

<b>Backup database .....</b>	<b>184</b>
<b>Backup Database Wizard .....</b>	<b>186</b>
Selecting server.....	187
Setting backup type.....	188
Selecting a single database to backup.....	190
Selecting several databases to backup.....	191
Selecting database files to backup.....	192
Selecting backup devices.....	193
Specifying backup sets.....	196
Setting backup media options.....	198
Setting backup options.....	200
Specifying transaction log backup options.....	202
Setting backup encryption options.....	204
Setting schedule options.....	206
Viewing summary information.....	208
Executing backup database task.....	209
<b>Backup History .....</b>	<b>210</b>
<b>Failed backups .....</b>	<b>211</b>
<b>Backup devices .....</b>	<b>213</b>
Backup Device Editor.....	215
Setting general properties.....	216
Viewing device contents.....	217
Viewing SQL definition.....	218
<b>Restore database .....</b>	<b>219</b>
<b>Selecting server and setting restore type .....</b>	<b>220</b>
<b>Setting database .....</b>	<b>222</b>
<b>Specifying point in time .....</b>	<b>224</b>
<b>Selecting backup set to restore .....</b>	<b>225</b>
<b>Selecting files and filegroups to restore .....</b>	<b>227</b>
<b>Setting restore options .....</b>	<b>228</b>
<b>Setting recovery options .....</b>	<b>230</b>
<b>Viewing summary information .....</b>	<b>232</b>
<b>Executing restore database task .....</b>	<b>233</b>
<b>Attach database .....</b>	<b>234</b>
<b>Selecting server .....</b>	<b>235</b>
<b>Defining database files .....</b>	<b>236</b>
<b>Viewing summary information .....</b>	<b>238</b>
<b>Executing attach database task .....</b>	<b>239</b>
<b>Detach database .....</b>	<b>240</b>
<b>Selecting server .....</b>	<b>241</b>
<b>Selecting databases .....</b>	<b>242</b>
<b>Viewing summary information .....</b>	<b>243</b>
<b>Executing detach database task .....</b>	<b>244</b>
<b>Check database .....</b>	<b>245</b>
<b>Check database wizard .....</b>	<b>247</b>
Selecting server and databases to check.....	248
Setting schedule options.....	249
Specifying checks to perform.....	251
Selecting tables to check.....	253
Setting repair options.....	254
Setting check options.....	256
Viewing summary information.....	258

Executing check database task.....	259
<b>Rebuild/Reorganize Indexes .....</b>	<b>260</b>
<b>Rebuild/Reorganize Indexes Wizard .....</b>	<b>262</b>
Selecting server and databases.....	263
Setting schedule options.....	264
Setting reindex action condition.....	266
Setting index actions and options.....	269
Viewing summary information.....	271
Executing rebuild/reorganize indexes task.....	272
<b>Shrink database .....</b>	<b>273</b>
<b>Selecting server and databases .....</b>	<b>274</b>
<b>Setting shrink type .....</b>	<b>275</b>
<b>Specifying shrink database options .....</b>	<b>276</b>
<b>Specifying shrink file options .....</b>	<b>278</b>
<b>Setting schedule options .....</b>	<b>280</b>
<b>Viewing summary information .....</b>	<b>281</b>
<b>Executing shrink database task .....</b>	<b>282</b>

## Part V Database Objects Management

**284**

<b>Tables .....</b>	<b>286</b>
<b>Table Editor .....</b>	<b>287</b>
Using Navigation bar.....	288
Fields.....	290
Properties.....	292
Filetable options .....	294
Check constraints.....	295
Indexes and primary/unique keys.....	296
Statistics.....	297
Foreign keys .....	298
Triggers.....	299
Information.....	300
Description.....	301
Permissions.....	302
SQL definition.....	304
<b>Field Editor .....</b>	<b>305</b>
Properties.....	306
Default value.....	309
XML options.....	310
Description.....	311
SQL definition.....	312
<b>Check Constraint Editor .....</b>	<b>313</b>
Properties.....	314
Description.....	315
SQL definition.....	316
<b>Index Editor .....</b>	<b>317</b>
Properties.....	318
Storage.....	321
Filter.....	323
XML index options.....	324
Spatial index options.....	326
Description.....	328
SQL Definition.....	329

<b>Statistics Editor .....</b>	<b>330</b>
Properties.....	331
Statistics data.....	333
Density vector .....	334
Histogram .....	335
SQL Definition.....	336
<b>Foreign Key Editor .....</b>	<b>337</b>
Properties.....	338
Description.....	340
SQL Definition.....	341
<b>Trigger Editor .....</b>	<b>342</b>
Properties.....	343
Options.....	345
Description.....	347
SQL Definition.....	348
<b>Views .....</b>	<b>349</b>
<b>View Editor .....</b>	<b>350</b>
Properties.....	351
Fields.....	353
Indexes .....	354
Statistics .....	355
Triggers.....	356
Description.....	357
Permissions.....	358
SQL Definition.....	360
<b>Procedures .....</b>	<b>361</b>
<b>Procedure Editor .....</b>	<b>362</b>
Properties.....	363
Parameters.....	365
Description.....	366
Permissions.....	367
SQL Definition.....	369
<b>Functions .....</b>	<b>370</b>
<b>Function Editor .....</b>	<b>371</b>
Properties.....	372
Parameters.....	374
Description.....	375
Permissions.....	376
SQL Definition.....	377
<b>Data Types .....</b>	<b>378</b>
<b>Data Type Editor .....</b>	<b>380</b>
Properties.....	381
Check constraints.....	384
Primary/unique keys.....	386
Description.....	387
Permissions.....	388
SQL Definition.....	390
<b>Database Users .....</b>	<b>391</b>
<b>Database Roles .....</b>	<b>392</b>
<b>Application Roles .....</b>	<b>394</b>
<b>Partition Functions .....</b>	<b>396</b>

<b>Partition Function Editor</b> .....	<b>397</b>
Properties.....	398
Description.....	401
SQL Definition.....	402
<b>Partition Schemes</b> .....	<b>403</b>
<b>Partition Scheme Editor</b> .....	<b>404</b>
Properties.....	405
Description.....	406
SQL Definition.....	407
<b>Sequences</b> .....	<b>408</b>
<b>Sequence Editor</b> .....	<b>409</b>
Properties.....	410
Description.....	412
Permissions.....	413
SQL Definition.....	415
<b>Synonyms</b> .....	<b>416</b>
<b>Synonym Editor</b> .....	<b>417</b>
Properties.....	418
Description.....	420
Permissions.....	421
SQL Definition.....	422
<b>Schemas</b> .....	<b>423</b>
<b>Schema Editor</b> .....	<b>424</b>
Properties.....	425
Description.....	426
Permissions.....	427
SQL Definition.....	429
<b>Part VI Storage</b> .....	<b>432</b>
<b>Database Editor</b> .....	<b>434</b>
Using Navigation bar .....	435
Setting general options .....	437
Managing filegroups .....	441
Editing database options .....	443
Configuring change tracking .....	450
Viewing database properties .....	451
Setting permissions .....	452
Viewing SQL definition .....	454
<b>Databases</b> .....	<b>455</b>
<b>Database Snapshots</b> .....	<b>457</b>
Using Navigation bar .....	459
Database snapshot editor .....	460
Viewing SQL definition .....	462
<b>Disk Space Usage</b> .....	<b>463</b>
<b>Space Usage by Objects</b> .....	<b>465</b>
<b>Part VII Security</b> .....	<b>468</b>
<b>Logins</b> .....	<b>470</b>
Login Editor .....	472

Using Navigation bar.....	473
Setting login properties.....	475
Assigning server roles.....	477
User mapping.....	478
Setting permissions.....	479
Viewing SQL definition.....	480
<b>Database users .....</b>	<b>481</b>
<b>User Editor .....</b>	<b>483</b>
Using Navigation bar.....	484
Setting database user properties.....	486
Editing description.....	488
Setting permissions on user.....	489
Setting user permissions.....	490
Viewing SQL definition.....	492
<b>Database roles .....</b>	<b>493</b>
<b>Database Role Editor .....</b>	<b>495</b>
Using Navigation bar.....	496
Setting database role properties.....	498
Defining role membership.....	499
Editing description.....	500
Setting permissions on role.....	501
Setting role permissions.....	502
Viewing SQL definition.....	504
<b>Application Roles .....</b>	<b>505</b>
<b>Application Role Editor .....</b>	<b>507</b>
Using Navigation bar.....	508
Setting application role properties.....	510
Editing description.....	512
Setting permissions on role.....	513
Setting permissions.....	514
Viewing SQL definition.....	516
<b>Server Roles .....</b>	<b>517</b>
<b>Server Roles Editor .....</b>	<b>519</b>
Using Navigation bar.....	520
Setting server role properties.....	522
Defining server role membership.....	523
Setting permissions.....	524
Viewing SQL definition.....	525
<b>Server Permissions .....</b>	<b>526</b>
<b>Database Permissions .....</b>	<b>528</b>
<b>Column permissions .....</b>	<b>530</b>
<b>Effective permissions .....</b>	<b>531</b>
<b>Credentials .....</b>	<b>532</b>
<b>Credential Editor .....</b>	<b>534</b>
Using Navigation bar.....	535
Setting credential properties.....	536
Viewing SQL definition.....	537
<b>Part VIII SQL Server Agent .....</b>	<b>540</b>
<b>Jobs .....</b>	<b>542</b>

<b>Job Editor</b> .....	<b>544</b>
Using Navigation bar .....	545
Setting job properties .....	547
Managing job steps .....	549
Managing job schedules .....	553
Schedule Editor .....	554
Managing notifications .....	556
Managing target servers .....	558
Managing alerts .....	559
Viewing job history .....	560
Viewing SQL definition .....	561
<b>Job history</b> .....	<b>562</b>
<b>Alerts</b> .....	<b>564</b>
<b>Alert Editor</b> .....	<b>566</b>
Using Navigation bar .....	567
Setting alert properties .....	568
Setting response .....	571
Setting advanced properties .....	572
Viewing SQL definition .....	573
<b>Operators</b> .....	<b>574</b>
<b>Operator Editor</b> .....	<b>576</b>
Using Navigation bar .....	577
Setting operator properties .....	578
Managing alert notifications .....	580
Managing job notifications .....	581
Viewing notification history .....	582
Viewing SQL definition .....	583
<b>Proxies</b> .....	<b>584</b>
<b>Proxy Editor</b> .....	<b>586</b>
Using Navigation bar .....	587
Setting proxy properties .....	588
Managing principals .....	590
Managing job steps .....	591
Viewing SQL definition .....	592
Credential Editor .....	593
<b>Part IX Performance</b> .....	<b>596</b>
<b>OS Processes</b> .....	<b>599</b>
<b>Server Processes</b> .....	<b>600</b>
<b>Locks</b> .....	<b>601</b>
<b>Performance Counters</b> .....	<b>603</b>
<b>Collected Statistics</b> .....	<b>607</b>
Collected Statistics Trend .....	612
<b>Index Physical Stats</b> .....	<b>616</b>
<b>Missing Indexes</b> .....	<b>618</b>
<b>Part X SQL Server Log</b> .....	<b>622</b>
<b>Part XI Query Editor</b> .....	<b>626</b>

Using toolbar .....	627
Working with the Query Editor area .....	630
Using the context menu .....	634
Executing queries .....	636
Using BLOB Editor .....	638
Editing as Hexadecimal .....	639
Editing as Text .....	640
Editing as Rich Text .....	641
Editing as Image .....	642
Viewing messages .....	643
<b>Part XII Options</b> .....	<b>646</b>
Environment .....	647
Server Explorer .....	649
Statistics Collector .....	650
Query tool .....	651
SQL Editor .....	653
Highlight .....	655
Quick code .....	656
Appearance .....	657
<b>Part XIII Appendix</b> .....	<b>660</b>
Shortcuts .....	660
Find & Replace text dialog .....	662
Print preview .....	664
Database states .....	666
Select databases dialog .....	667
Select objects dialog .....	668
Using templates .....	669



**Part**



# 1 Welcome to EMS SQL Administrator!

**SQL Administrator** is a toolset for efficient SQL Server administration. It provides with a complete and easy-to-use set of tools for administering SQL Server. SQL Administrator is designed specifically for database administrators and allows performing administrative tasks in the fastest, easiest and most efficient way.

## **Key features:**

- Performance monitoring in real-time
- Statistics Collector to collect statistics of the system and SQL Server performances
- Graphical comparison of statistics over different time periods to find performance bottlenecks
- Analysing trends of performance metrics over time
- Checking SQL Server health with Alarm Summary tool
- Well-described wizard system to backup/restore, attach/detach, check, shrink databases, rebuild/reorganize indexes
- Getting complete information about all database maintenance tasks including their schedule and status
- Storing, comparing and applying server configurations including server properties, configuration options, logins, database options, objects' permissions
- Forecasting disk space exhaustion
- Analysing space utilization by drive, database or database objects
- Creating/editing/deleting SQL Server objects
- Viewing information about current server processes and locks of specific processes and objects
- Handy Query Management tool for SQL queries and script execution
- Ability to print almost any data of the program
- Support of SQL Server versions from 2000 to 2014

## 1.1 What's new

### Version

**SQL Administrator 2.0**

### Release date

April 1, 2015

### What's new in SQL Administrator 2.0?

1. [Database Explorer](#) is added: new tool for viewing information of database and databases objects and managing them.
2. [Database Explorer](#) | [Database tab](#). Main information on the database state is displayed here, namely: values of its properties and options, dates and results of service tasks execution, location of the latest database backup, database size and growth, information on the files and latency at their access and many other. Clamant values are being lighted up. Here one can perform various actions with the database: make a backup, verify its integrity, refresh indices, create an object etc.
3. [Server Explorer](#) | [Server tab](#). Information volume concerning the server has been considerably increased. Now it is possible to create/edit/delete objects right here with no need for opening the appropriate tabs of the Server Explorer. For user's convenience alarms can be browsed in the form of vertical grid.
4. [Server Explorer](#) | [Performance](#). Now all the most important information on the current server load is gathered here in the form of graphs and tables.
5. New freeware version with limited features is released.
6. [Server Explorer](#) | [Maintenance](#). Now the date and the status of the last execution of each server maintenance task for all server databases are displayed here.
7. New [alarms](#) have been added:
  - High Disk Transfer Latency
  - Low Page Life Expectancy
  - Databases in Restricted User Mode
  - Databases in Non-Online State
  - Errors in Server Error Log
8. For [WMI connection](#) to server, which is used for receiving the system information and counter values, now one can set Windows login/password different from the current one.
9. [Statistics Collector](#). The following counters are also collected now:
  - Paging File. % Usage
  - Physical Disk. Avg. Disk sec/Read
  - Physical Disk. Avg. Disk sec/Write
  - Physical Disk. Disk Reads/sec
  - Physical Disk. Disk Writes/sec
  - SQL Server: Buffer Manager. Page life expectancy
  - SQL Server: Memory Manager. Memory Grants Pending
  - SQL Server: SQL Statistics. Batch Requests/sec
10. [Server Explorer](#) | [Server group tab](#). For more convenient data scanning the data is presented in the form of vertical grid. Here additional information on the servers is also displayed.
11. [User](#), [Role](#), [Application Role](#) Editors. In addition to the possibility to browse and edit permissions of the given principal, there has appeared the possibility to set permissions for the given principal.
12. Default size of objects editors has been enlarged.
13. [Server Explorer](#) | [Alarm Summary](#). The representative width has been limited, and other visual amendments have been made.
14. [Maintenance Plan Editor](#). Now it has become more convenient to create tasks for one

and the same database - there is no need to open database picker dialog every time, databases from the previous tasks are added to the combobox.

15. The height of popup menus for comboboxes has been increased – it used to be inconvenient to search the required element in the long list.
16. Large number of other improvements and bug fixes.

## 1.2 Feature Matrix

The **FREE version** of SQL Administrator does not include all features of the **FULL version** and has some limitations concerning the set of data manipulation, server and database maintenance tools. The detailed feature matrix is given below.

To browse the **Feature Matrix**, visit the following page: <http://www.sqlmanager.net/products/mssql/sqladmin/matrix>

## 1.3 System requirements

- Microsoft Windows 2000, Microsoft Windows 2000 Server, Microsoft Windows XP, Microsoft Windows 2003 Server, Microsoft Windows 2008 Server, Microsoft Windows Vista, Microsoft Windows 7, Microsoft Windows 8, Microsoft Windows 2012 Server;
- .Net Framework 2.0;
- .Net Framework 4.0 (for maintenance plans of SQL Server 2012)
- 512 MB RAM or more;
- 70MB of available HD space for program installation;
- Super VGA (800x600) or higher-resolution video adapter and monitor; Super VGA (1024x768) or higher-resolution recommended;
- Microsoft Mouse or compatible pointing device;
- Microsoft Data Access Components (MDAC) or SQL Server Native Client;
- Local or remote SQL Server 2000, SQL Server 2005, SQL Server 2008, SQL Server 2012 or SQL Server 2014 (We do not support SQL Server Compact Edition);

## 1.4 Installation

If you are **installing EMS SQL Administrator for the first time** on your PC:

- download the SQL Administrator distribution package from the [download page](#) available at our site;
- unzip the downloaded file to any local directory, e.g. *C:\unzipped*;
- run *SQLAdminSetup.exe* from the local directory and follow the instructions of the installation wizard;
- after the installation process is complete, find the SQL Administrator shortcut in the corresponding group of Windows Start.

If you want to **upgrade an installed copy of SQL Administrator** to the latest version:

- download the full installation package of SQL Administrator from the [download page](#) available at our site;
- close SQL Administrator application if it is running;
- repeat the steps you have made for the first-time installation;
- run SQL Administrator using its shortcut in the corresponding group of Windows Start menu.

---

**See also:**

[EMS SQL Administrator FAQ](#)

## 1.5 Purchasing

To make it easier for you to purchase our products, we have contracted with PayPro Global registration service. The PayPro Global order process is protected via a secure connection and makes on-line ordering by credit/debit card quick and safe. The following information about PayPro Global is provided for your convenience.

**PayPro Global** is a global e-commerce provider for software and shareware sales via the Internet. It accepts payments in US Dollars, Euros, Pounds Sterling, Japanese Yen, Australian Dollars, Canadian Dollars or Swiss Franks by Credit Card (Visa, MasterCard/ EuroCard, American Express, Diners Club), Bank/Wire Transfer.

If you want to review your order information, or you have questions about ordering or payments please visit our [PayPro Global Shopper Support](#), provided by **PayPro Global**.

Please note that all of our products are delivered via ESD (Electronic Software Delivery) only. After purchase you will be able to immediately download the registration keys. Also you will receive a copy of registration keys by email. Please make sure to enter a valid email address in your order. If you have not received the keys within 2 hours, please, contact us at [sales@sqlmanager.net](mailto:sales@sqlmanager.net).

Product distribution	PayPro Global
<b>SQL Administrator</b> (Business license) + 1-Year Maintenance*	<a href="#">Register Now!</a>
<b>SQL Administrator</b> (Business license) + 2-Year Maintenance*	
<b>SQL Administrator</b> (Business license) + 3-Year Maintenance*	
<b>SQL Administrator</b> (Non-commercial license) + 1-Year Maintenance*	
<b>SQL Administrator</b> (Non-commercial license) + 2-Year Maintenance*	
<b>SQL Administrator</b> (Non-commercial license) + 3-Year Maintenance*	
<b>SQL Administrator</b> (Trial version)	<a href="#">Download Now!</a>
<b>SQL Administrator</b> (Free version)	

\***EMS Maintenance Program** provides the following benefits:

- Free software bug fixes, enhancements, updates and upgrades during the maintenance period
- Free unlimited communications with technical staff for the purpose of reporting Software failures
- Free reasonable number of communications for the purpose of consultation on operational aspects of the software

After your maintenance expires you will not be able to update your software or get technical support. To protect your investments and have your software up-to-date, you need to renew your maintenance.

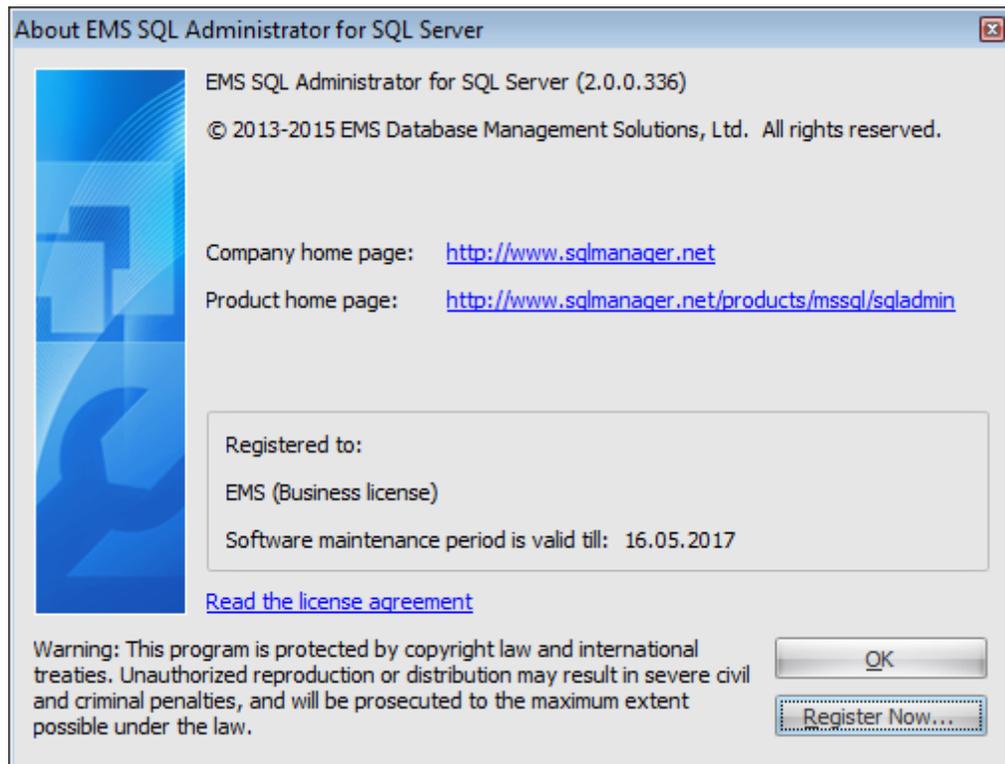
You can easily reinitiate/renew your maintenance with our on-line, speed-through Maintenance Reinstatement/Renewal Interface. After reinitiating/renewal you will receive a confirmation e-mail with all the necessary information.

**See also:**

[How to register SQL Administrator](#)

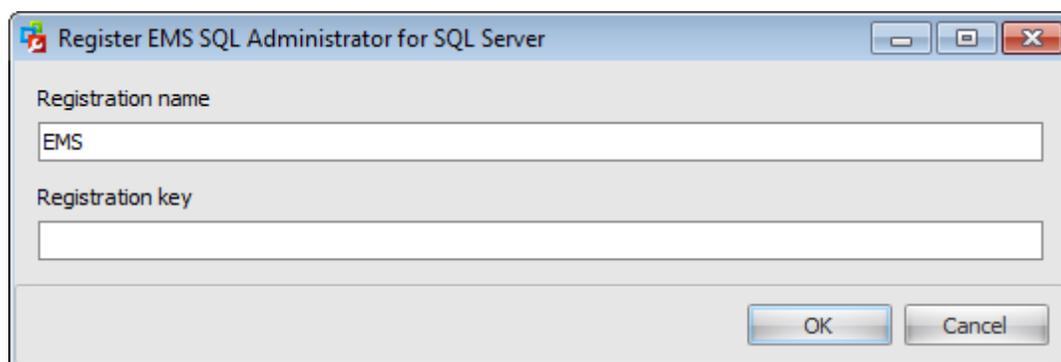
## 1.6 How to register SQL Administrator

If you have not registered your copy of SQL Administrator yet, you can do it by pressing the **Register Now** button and entering your registration information in the **Register SQL Administrator** dialog.



To register your newly purchased copy of EMS SQL Administrator, perform the following steps:

- receive the notification letter from **PayPro Global** with the registration info;
- enter the **Registration name** and the **Registration key** from this letter;
- make sure that the registration process has been completed successfully – check the registration information in the **About SQL Administrator** dialog (use the **Help | About** menu item to open this dialog).





## 1.7 Version history

Product name	Version	Release date
SQL Administrator 1.4	<a href="#">Version 1.4</a>	September 29, 2014
SQL Administrator 1.3	<a href="#">Version 1.3</a>	February 4, 2014
SQL Administrator 1.2	<a href="#">Version 1.2</a>	August 28, 2013
SQL Administrator 1.1	<a href="#">Version 1.1</a>	May 24, 2013
SQL Administrator 1.0	<a href="#">Version 1.0</a>	March 04, 2013

### Version 1.4

- Added support of new SQL Server 2014 features:
  - Support of [backup encryption](#).
  - Support of [backup to Windows Azure](#).
  - Support of [memory-optimized files and filegroups](#).
  - Support of new [server-level permissions](#): CONNECT ANY DATABASE, IMPERSONATE ANY LOGIN, SELECT ALL USER SECURABLES.
  - Support of [maintenance plans](#) on SQL Server 2014.
- Added ability to clone server objects including jobs and maintenance plans to the same or another server.
- [Server Explorer | Security view](#). Added the [Credentials](#) view which allows browsing and managing credentials.
- [Server Explorer | Security view](#). The logins of securityadmin server role (having the ability to assign most server permissions) are displayed in the System and Security Administrators list.
- [Database Editor](#).
  - The support of some previously unsupported SQL Server 2012 and 2014 databases properties has been implemented.
  - The default size of a database file was 3Mb that is lower than the minimum allowed size for SQL Server 2012. It caused an error while saving the database properties. Fixed now.
  - After deleting a database file it appeared in the list but was actually removed from the server. Fixed now.
- [Query tool](#). The height of the grid now automatically fits the height of the parent data view area when it is changed.
- [Backup wizard](#). The ability to set the 'Server default' value for the backup compression option added now.
- Server Explorer | [Database Users](#), [Roles](#), [Application Roles](#) views. The Description field added.
- [Server Explorer | Databases view](#). The ID field added. It is hidden by default and can be shown using Column Chooser dialog.
- [Alarm Summary](#) | Databases without Backup alarm. Backup check interval is now 8 days by default for new server registrations.
- [Server Explorer | Database Users view](#). The Orphaned flag was incorrectly shown for some users on SQL Server 2012 although the corresponding alarm worked right. Fixed now.
- [Alarm Summary](#). If the database name contained the specific characters that should be quoted, the 'Incorrect syntax' error occurred in the alarm status. Fixed now.
- If the error occurred on creating [Maintenance Plan](#) on SQL Server 2012 the Ssis11Adpt.exe process was not finished with the program. Fixed now.
- Lots of other improvements and bug-fixes.

### Version 1.3

1. [Compare Server Configurations wizard](#) is implemented. It allows saving to repository, comparing and restoring configuration of SQL Server. It is allowed to compare saved server configurations as well as configurations of two servers. Configuration includes the following: server information, server properties, server configuration options, database options, database files and filegroups, server and database principals, server and database object permissions.
2. Added [Register Servers](#) form, which allows registering SQL Servers of local area network.
3. In the Server Connection dialog of a [query window](#) it is possible to select the current database for query execution immediately. There is a list of recently selected databases.
4. [Query execution tool](#).
  - Added shortcuts Alt+key for some commands of the toolbar.
  - The list of non-reserved server words for highlighting and Code Completion is enlarged.
  - When editing file without server connection and when opening Code Completion the error "Object reference not set to an instance of an object" occurred. Fixed now.
5. Wizards.
  - After each new start of the wizard log of operation execution result scrolls so that new execution data appear in the upper line of the window.
  - After loading a template, its file is blocked for editing. Fixed now.
6. When opening the [Maintenance plan editor](#) from Alarm Summary and saving the error "Maintenance plan with 'XXX' names already exists" occurred. Fixed now.
7. When opening [Blob Editor](#) of read-only query field, the upload from file button remained active. Fixed now.
8. Permissions. It was impossible to revoke permissions set by the current user. Fixed now.
9. [Server Registration](#), Server Group, [Windows](#) and other dialogs. Added shortcuts Alt+key for focusing some dialog controls. (Shown after pressing Alt).
10. [Windows](#) dialog. Close button overlapped Help button. Fixed now.
11. [Backup wizard](#). When using apostrophe character (') in backup name, backup description, media set name, media set description fields and executing the error occurred. Fixed now.
12. [SQL Editor](#).
  - When deleting text from the end of the line till space, the space was also deleted. Fixed now.
  - Clicking the scrollbar to move page up did not work. Fixed now.
13. [Options](#) dialog. The field where the software log file name is set now contains the open file button.
14. Lots of other improvements and bug-fixes.

### Version 1.2

1. [Query execution tool](#).
  - The [results](#) of the query execution can be edited now. (The table should have the primary key.)
  - [BLOB Editor](#) is implemented to view/edit the values of BLOB fields.
  - The grid navigator to shift the records and edit the query results is added.

1. [Server Explorer](#). It is possible to start/stop server now.
2. Wizards. The context menu of the execution log now contains *Select All* and *Copy*.
3. [Statistics Collector State dialog](#). The progress on start/stop of the service is shown now.
4. Multiple choice dialog (e.g., choice of server role members) contains the context menu with Select All/Unselect All commands.
5. [Maintenance Plans](#). Maintenance Plan History button was not disabled if the plan list was empty. Fixed now.
6. [Collected Statistics Trend](#). Show Details button was not disabled if the list was empty. Fixed now.
7. Lots of other improvements and bug-fixes.

### Version 1.1

1. Data grids and chart views of [Server Explorer](#), [query data](#), [Alarm Summary](#) now can be [printed](#) or exported in several popular formats.
2. To the [Server Explorer](#) data grids the incremental search is added.
3. [Server Explorer](#) | [Current](#) and [Collected](#) Statistics | Chart View Customization. Now the Add Performance Counter dialogue displays its description.
4. [Server Explorer](#) | [Maintenance](#) | [Check Database](#). Added a column of the last successful database check date.
5. [Server Explorer](#) | [Storage](#) | [Databases](#). In the context menu the "Recover without Restoring" command is now available for databases being in restoring state.
6. [Backup Database wizard](#). Database icons are now displayed together with their (database) statement.
7. [Server Explorer](#) | [Maintenance](#) | [Backup](#) | [Failed Backups](#). Now when calling the [Backup Database wizard](#), a database name is inserted from the current view string.
8. [Server Explorer](#). In some cases when opening the Maintenance views the "An element with the same key already exists" error occurred. Fixed now.
9. When registering a server from the main menu of the program, the current server group was not specified. Fixed now.
10. If in [Server Explorer](#) a server group is selected, when calling a New Query this server group is specified in the Connect to Server dialogue.
11. Other improvements and bugfixes.

### Version 1.0

- Managing maintenance tasks, monitoring their performance schedule, periodicity and the last execution result.
- Detecting and informing about possible failures.
- Managing SQL Server storage. Ability to get information about disk space used by databases/separate objects and to forecast disk space exhaustion.
- Collecting and analyzing performance statistics of the system and SQL Server. Ability to find performance bottlenecks and predict their occurrence.
- Easy and intuitive SQL Server environment Security Management.
- Monitoring server activity, analyzing server logs.
- Creating/modifying/dropping different server objects.
- SQL queries and scripts execution.

[Scroll to top](#)

---

**See also:**

[What's new](#)

## 1.8 EMS SQL Administrator FAQ

Please read this page attentively if you have questions about **EMS SQL Administrator**.

### Table of contents

- [What is EMS SQL Administrator?](#)
- [What do I need to start working with EMS SQL Administrator?](#)
- [Why cannot I connect to SQL Server?](#)
- [Why does SQL Administrator use WMI connection to server?](#)
- [I get an error 'Timeout expired' when I try to execute a query in Query tool.](#)
- [Is it possible somehow to restore my program settings and server registrations from the old or faulty HDD, where SQL Administrator was installed?](#)
- [What is the difference between detailed and average statistics in the Performance | Collected Statistics view?](#)
- [I cannot edit a maintenance plan. The editor is set to "read only" mode.](#)
- [I try to execute a maintenance plan, but nothing happens and the maintenance plan history does not contain any information.](#)
- [Alarm Summary | Alarm Log takes too long time to open. What can I do?](#)
- [I perform a database check using Check Database wizard, but in Server Explorer the last maintenance execution date is not updated. Why?](#)

### Question/answer list

Q: *What is EMS SQL Administrator?*

A: EMS SQL Administrator is a powerful graphical tool for Microsoft SQL Server administration. It provides a complete and easy-to-use set of tools for administering SQL Server. SQL Administrator is designed specifically for database administrators and allows performing administrative tasks in the fastest, easiest and most efficient way.

Q: *What do I need to start working with EMS SQL Administrator?*

A: First of all you must have an opportunity to connect to some local or remote Microsoft SQL Server. Besides you need your computer to satisfy the [system requirements](#) of EMS SQL Administrator.

Q: *Why cannot I connect to SQL Server?*

A:

1. If your server is installed as a named instance, use server name in form `computer_name\instance_name`. SQL Server Express Edition has the default `SQLEXPRESS` instance name.
2. If you connect to the remote server, make sure that the connection is not blocked by firewall. Ask your system administrator to correctly open appropriate ports.
3. Make sure that SQL Server is started and you have correct login to access it.
4. Make sure that client protocols are correctly tuned. Use SQL Server Configuration Manager on connection to SQL Server 2005 or higher (it is installed with Native Client) or SQL Server Client Network Utility (`system32\cliconfg.exe`) on connection to earlier versions.
5. Make sure that remote connection is enabled on server and server protocols are correctly tuned. Use SQL Server Surface Area Configuration on SQL Server 2005 or higher and SQL Server Network Utility (Program Files\Microsoft SQL Server\80\Tools\Binn\svrnetcn.exe) on earlier server versions.

6. If you use named instance of server with dynamic TCP/IP port, make sure that SQL Server Browser service is running.
7. If you connect remotely, try using server IP address instead of its name to ensure that DNS works well.

Q: *Why does SQL Administrator use WMI connection to server?*

A: WMI connection is used to:

- ✓ get system information about SQL server, e.g. installed operating system or CPU type;
- ✓ start and stop SQL Server services;
- ✓ view Windows event logs;
- ✓ get system performance counter values to display them in the product, in some alarms and for statistics collection.

Q: *I get an error 'Timeout expired' when I try to execute a query in Query tool.*

A: You need to increase timeout values on the [Query tool](#) tab of the [Options](#) dialog, or set them to 0 - unlimited.

Q: *Is it possible somehow to restore my program settings and server registrations from the old or faulty HDD, where SQL Administrator was installed?*

A: SQL Administrator stores all your settings in the Windows registry. It means that the only way to transfer your database data is to get access to the Windows registry. You can do it by loading OS from your old HDD (if possible) or by opening the registry file with a special editor program. If you managed it, you can unload the 'HKEY\_CURRENT\_USER\Software\EMS\SQL Administrator' and 'HKEY\_LOCAL\_MACHINE\Software\EMS\SQL Administrator' branches to the \*.REG files, transfer these files to a new system and add information to the registry by double-clicking the files. Also you have to grant write permissions for 'HKEY\_LOCAL\_MACHINE\Software\EMS\SQL Administrator' branch to 'Users' Windows group using Registry Editor tool.

Q: *What is the difference between detailed and average statistics in the Performance | Collected Statistics view?*

A: Statistics Collector collects detailed performance statistics every 5 minutes by default. This information is stored 31 days by default and then averaged out. The collected values are averaged out by days of the week every 30 minutes. It is necessary to reduce the total amount of stored information. Statistics collection parameters can be set for each server on the [Statistics Collection](#) tab of the [Server Registration Properties](#) dialogue.

Q: *I cannot edit a maintenance plan. The editor is set to "read only" mode.*

A: To create and edit maintenance plans you should install SQL Server Integration Services on your computer. Otherwise, you can view the existing maintenance plans only. If SQL Server Integration Services are installed on another computer in your local network, using the 'Ssis11Adpt.exe' file and sample configuration files 'Ssis11Adpt.exe.config.example' and 'SsisClient.config.example' from the program installation folder, you can do the following:

- ✓ copy the 'Ssis11Adpt.exe' file to the computer with SQL Server Integration Services installed;
- ✓ create the 'Ssis11Adpt.exe.config' file in the 'Ssis11Adpt.exe' folder and set parameters;
- ✓ launch 'Ssis11Adpt.exe';
- ✓ create the 'SsisClient.config' file in the program folder and set parameters.

If the configuration files are set correctly, maintenance plans can be edited. Do not forget to open a port in Firewall.

Q: *I try to execute a maintenance plan, but nothing happens and the maintenance plan history does not contain any information.*

A: Probably the maintenance plan job cannot be run. Please see the job execution history with <maintenance\_plan\_name> or <maintenance\_plan\_name>.<subplan\_name> names.

Q: *Alarm Summary | Alarm Log takes too long time to open. What can I do?*

A: Please check on the server where SQL Administrator repository database is located that the cleanup\_alarm\_log\_<repository\_database\_name> job is running. It cleans the old log records.

Q: *I perform a database check using Check Database wizard, but in Server Explorer the last maintenance execution date is not updated. Why?*

A: Information on checking database and rebuilding/reorganizing indexes task executions is taken from the maintenance plan and job history. So, the task launches in the wizard and SQL command executions are not reflected in the 'Last Execution Date' column.

[Scroll to top](#)

If you still have any questions, contact us at our [Support Center](#).

## 1.9 Other EMS Products

### Quick navigation



[MySQL](#)



[Microsoft SQL Server](#)



[PostgreSQL](#)



[InterBase / FireBird](#)



[Oracle](#)



[IBM DB2](#)



[Tools & components](#)

### MySQL



#### [SQL Management Studio for MySQL](#)

EMS SQL Management Studio for MySQL is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



#### [SQL Manager for MySQL](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



#### [Data Export for MySQL](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



#### [Data Import for MySQL](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



#### [Data Pump for MySQL](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, DB2, InterBase/Firebird, etc.) to MySQL.



#### [Data Generator for MySQL](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



#### [DB Comparer for MySQL](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



#### [DB Extract for MySQL](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



#### [SQL Query for MySQL](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



#### [Data Comparer for MySQL](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

## Microsoft SQL Server



### [SQL Management Studio for SQL Server](#)

EMS SQL Management Studio for SQL Server is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



### [EMS SQL Backup for SQL Server](#)

Perform backup and restore, log shipping and many other regular maintenance tasks on the whole set of SQL Servers in your company.



### [SQL Administrator for SQL Server](#)

Perform administrative tasks in the fastest, easiest and most efficient way. Manage maintenance tasks, monitor their performance schedule, frequency and the last execution result.



### [SQL Manager for SQL Server](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



### [Data Export for SQL Server](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



### [Data Import for SQL Server](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



### [Data Pump for SQL Server](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, DB2, InterBase/Firebird, etc.) to Microsoft® SQL Server™.



### [Data Generator for SQL Server](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



### [DB Comparer for SQL Server](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



### [DB Extract for SQL Server](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



### [SQL Query for SQL Server](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



### [Data Comparer for SQL Server](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

## PostgreSQL



### [SQL Management Studio for PostgreSQL](#)

EMS SQL Management Studio for PostgreSQL is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



### [EMS SQL Backup for PostgreSQL](#)

Creates backups for multiple PostgreSQL servers from a single console. You can use automatic backup tasks with advanced schedules and store them in local or remote folders or cloud storages



### [SQL Manager for PostgreSQL](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



### [Data Export for PostgreSQL](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



### [Data Import for PostgreSQL](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



### [Data Pump for PostgreSQL](#)

Migrate from most popular databases (MySQL, SQL Server, Oracle, DB2, InterBase/Firebird, etc.) to PostgreSQL.



### [Data Generator for PostgreSQL](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



### [DB Comparer for PostgreSQL](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



### [DB Extract for PostgreSQL](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



### [SQL Query for PostgreSQL](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



### [Data Comparer for PostgreSQL](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

## **InterBase / Firebird**



### [SQL Management Studio for InterBase/Firebird](#)

EMS SQL Management Studio for InterBase and Firebird is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



### [SQL Manager for InterBase/Firebird](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



### [Data Export for InterBase/Firebird](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more



### [Data Import for InterBase/Firebird](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



### [Data Pump for InterBase/Firebird](#)

Migrate from most popular databases (MySQL, SQL Server, Oracle, DB2, PostgreSQL, etc.) to InterBase/Firebird.



### [Data Generator for InterBase/Firebird](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



### [DB Comparer for InterBase/Firebird](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



### [DB Extract for InterBase/Firebird](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



### [SQL Query for InterBase/Firebird](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



### [Data Comparer for InterBase/Firebird](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

## Oracle



### [SQL Management Studio for Oracle](#)

EMS SQL Management Studio for Oracle is a complete solution for database administration and development. SQL Studio unites the must-have tools in one powerful and easy-to-use environment that will make you more productive than ever before!



### [SQL Manager for Oracle](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



### [Data Export for Oracle](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



### [Data Import for Oracle](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via

user-friendly wizard interface.



#### [Data Pump for Oracle](#)

Migrate from most popular databases (MySQL, PostgreSQL, MySQL, DB2, InterBase/Firebird, etc.) to Oracle



#### [Data Generator for Oracle](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



#### [DB Comparer for Oracle](#)

Compare and synchronize the structure of your databases. Move changes on your development database to production with ease.



#### [DB Extract for Oracle](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



#### [SQL Query for Oracle](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts based on retrieved data quickly and more.



#### [Data Comparer for Oracle](#)

Compare and synchronize the contents of your databases. Automate your data migrations from development to production database.

[Scroll to top](#)

## IBM DB2



#### [SQL Manager for DB2](#)

Simplify and automate your database development process, design, explore and maintain existing databases, build compound SQL query statements, manage database user rights and manipulate data in different ways.



#### [Data Export for DB2](#)

Export your data to any of 20 most popular data formats, including MS Access, MS Excel, MS Word, PDF, HTML and more.



#### [Data Import for DB2](#)

Import your data from MS Access, MS Excel and other popular formats to database tables via user-friendly wizard interface.



#### [Data Pump for DB2](#)

Migrate from most popular databases (MySQL, PostgreSQL, Oracle, MySQL, InterBase/Firebird, etc.) to DB2



#### [Data Generator for DB2](#)

Generate test data for database testing purposes in a simple and direct way. Wide range of data generation parameters.



#### [DB Extract for DB2](#)

Create database backups in the form of SQL scripts, save your database structure and table data as a whole or partially.



#### [SQL Query for DB2](#)

Analyze and retrieve your data, build your queries visually, work with query plans, build charts

based on retrieved data quickly and more.

[Scroll to top](#)

## Tools & components



### [Advanced Data Export for RAD Studio VCL](#)

Advanced Data Export for RAD Studio VCL allows you to save your data in the most popular office programs formats.



### [Advanced Data Export .NET](#)

Advanced Data Export .NET is a component for Microsoft Visual Studio .NET that will allow you to save your data in the most popular data formats for the future viewing, modification, printing or web publication. You can export data into MS Access, MS Excel, MS Word (RTF), PDF, TXT, DBF, CSV and more! There will be no need to waste your time on tiresome data conversion - Advanced Data Export will do the task quickly and will give the result in the desired format.



### [Advanced Data Import for RAD Studio VCL](#)

Advanced Data Import for RAD Studio VCL will allow you to import your data to the database from files in the most popular data formats.



### [Advanced PDF Generator for RAD Studio](#)

Advanced PDF Generator for RAD Studio gives you an opportunity to create PDF documents with your applications written on Delphi or C++ Builder.



### [Advanced Query Builder for RAD Studio VCL](#)

Advanced Query Builder for RAD Studio VCL is a powerful component for Delphi and C++ Builder intended for visual building SQL statements for the SELECT, INSERT, UPDATE and DELETE clauses.



### [Advanced Excel Report for RAD Studio](#)

Advanced Excel Report for RAD Studio is a powerful band-oriented generator of template-based reports in MS Excel.

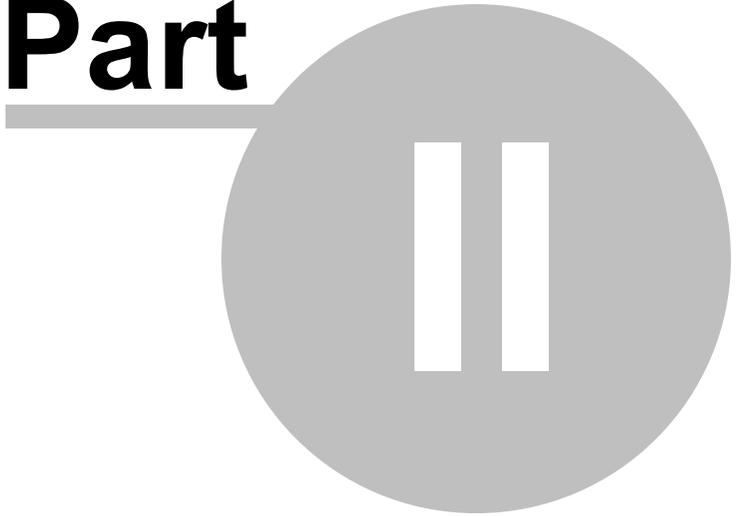


### [Advanced Localizer for RAD Studio VCL](#)

Advanced Localizer for RAD Studio VCL is an indispensable component for Delphi for adding multilingual support to your applications.

[Scroll to top](#)

**Part**



## 2 Getting Started

**SQL Administrator** provides you with a toolset for SQL Server administration with most efficient way. When the program is launched for the first time the **Start Page** is opened. This is how SQL Administrator application looks when you start it.



### Getting started

- [Register a server](#) that will be used as the program repository for collected statistics.
- [Set repository database](#).
- [Register servers](#) to administrate using SQL Administrator and set [alarm summary](#) and [statistics collection](#) options.
- Install and launch [statistics collector](#) service.
- Tune the [program options](#).

### Tools

- Run the [Server Explorer](#) which provides the main set of SQL Administrator tools.
- View [Alarm Summary](#). It provides information about possible failures on registered servers and allows fixing them promptly.
- Launch the [Statistics Collector State](#) dialog allowing you to install/uninstall statistics collector service.
- Tune the [program options](#).

The succeeding chapters of this document are intended to inform you about the tools implemented in SQL Administrator. Please see the instructions below to learn how to perform various operations in the easiest way.

- [Work area](#)
- [Server Explorer](#)
- [Main menu & Toolbars](#)
- [Working with windows](#)
- [Working with grid](#)
- [Shortcuts](#)

---

**See also:**

[Servers](#)

[Database Maintenance](#)

[Storage](#)

[Security](#)

[SQL Server Agent](#)

[Performance](#)

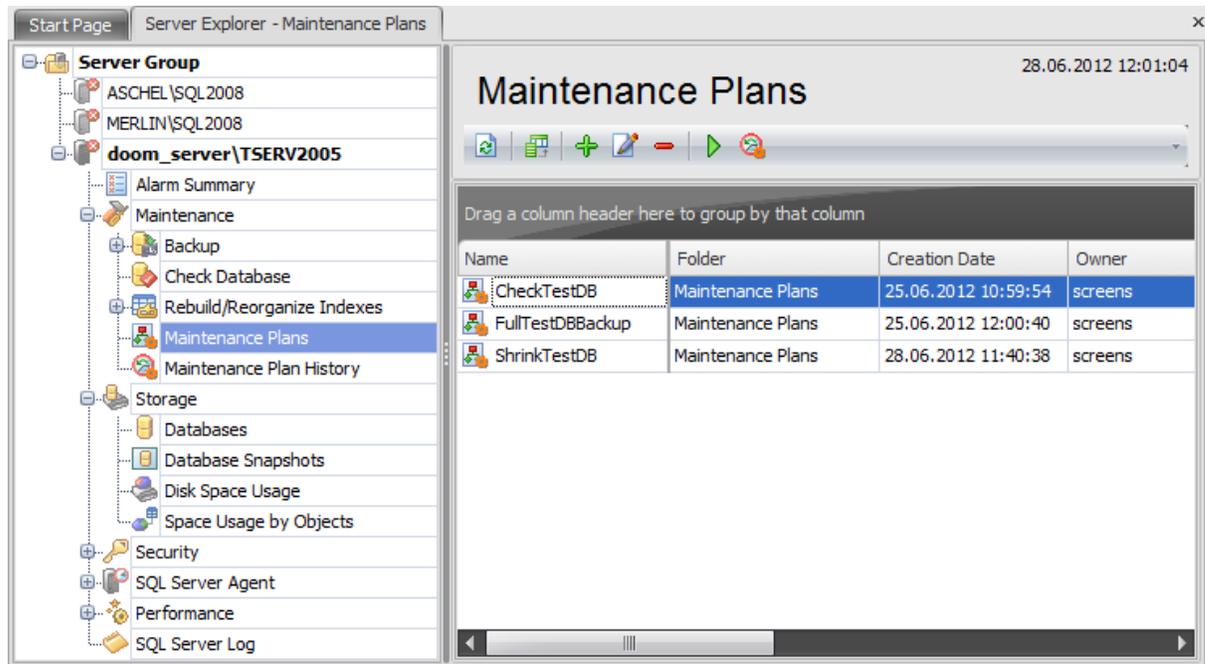
[SQL Server Log](#)

[Query Editor](#)

[Options](#)

## 2.1 Work area

Work area is used to create tabs or windows. The layout type is to be selected in the [Environment Options](#).



Tabs or windows allow switching between work areas.

Every section of the [Server Explorer](#) has the **Refresh** button on the toolbar which allows refreshing the content of the section.

Some sections which grids are displayed in have the **Column Chooser** item on their toolbars which allows you to change the number of displayed columns and their positions.

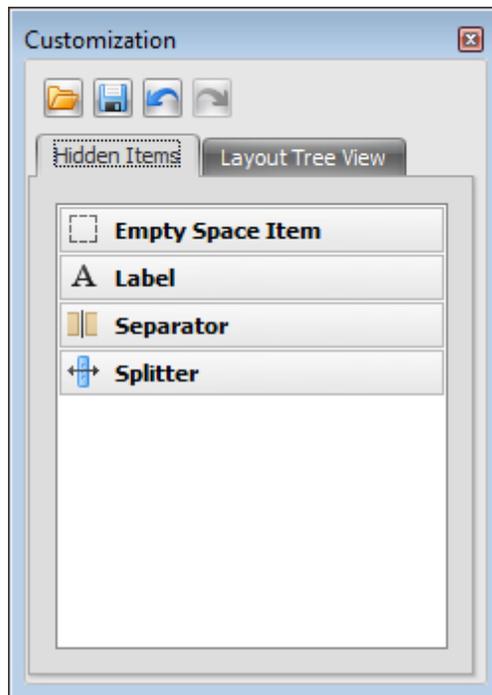
### Customization

The layouts of the following sections of the [Server Explorer](#) can be customized according to your liking:

- [Storage](#)
- [Security](#)
- [SQL Server Agent](#)
- [Performance](#)

Suppose we need to customize the layout of the [Security](#) section.

Click the **Customize Layout** button situated on the toolbar. The **Customization** window appears.



### Separator

Separator is used to separate one section from another.

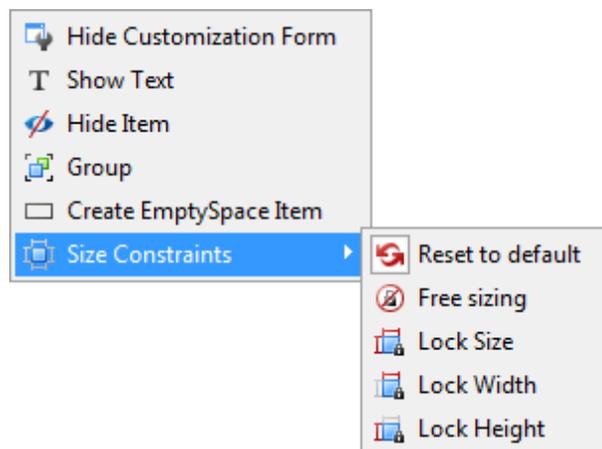


### Splitter

**Splitter** controls are used to resize docked controls at run time.



The context menu of each item looks the following way:



 hide customization window and save changes;

-  delete the selected item from the work area;
-  create a group of items;
-  create an empty section;
-  set item size constraints;
-  reset all size settings to default;
  -  enable/disable free sizing;
  -  fix the current size of the item;
  -  fix the current width of the item;
  -  fix the current height of the item.

In the context menu of the Label item you can also find the  **Rename** item which allows you to set a name for the selected label item.

When you choose the  **Group** context menu item a new group appears in the **Layout Group** list situated on the **Layout Tree View** tab of the **Customization** window. To delete a group select the  **Ungroup** item of its context menu. Also the group context menu allows you to:

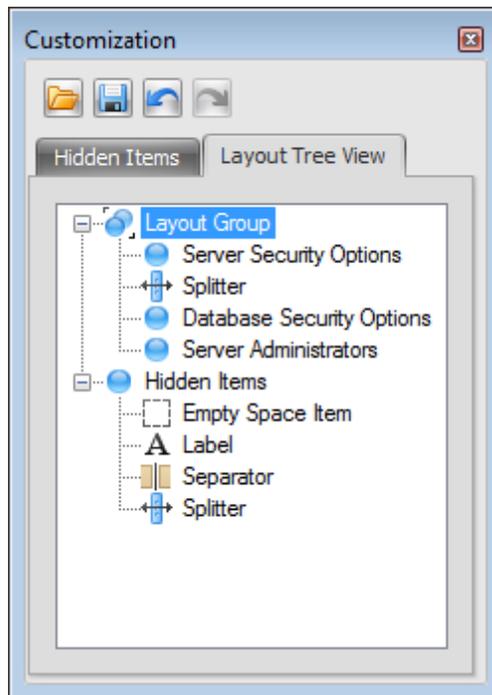
-  hide customization window and save changes;
-  rename the group;
-  hide the group heading;
-  text position (determines where the group name is situated);
-  create an empty section;
-  create tabbed group;

For the tabbed group you can:

-  ungroup tabbed group;
-  add a new tab.

The **Layout Group** node of the **Layout Tree View** tab shows the elements created on the work area. In the [Security](#) section there are three pre-created items:

- Server Security Options
- Database Security Options
- System Administrators



When you click the item name in the list the frame of the corresponding section on the work area is highlighted.  
Then you can drag the section to any place you want and resize it if necessary.

#### Server Security Options

Server authentication	mixed
Cross-database ownership chaining	off
Ad hoc distributed queries	enabled
CLR integration	disabled
Remote dedicated administrator connection (DAC)	disabled
Database Mail	disabled
OLE Automation objects	enabled
xp_cmdshell	enabled
SQL Mail	disabled

#### Database Security Options

Database	DB Chaining	Trust Worthy	Guest Enabled
alendb_logshipping	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
android_dream	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
anna	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AW	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AW_restored	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AW2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBA_222_restored	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dba222_restored2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBASpaceRepository	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBASpaceRepository2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBASpaceRepository2_1_1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBASpaceRepository222	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Server Administrators

Login Name	Login Type	Disabled	Denied	Locked	Credential
EMSDOMAIN\aschel	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EMSDOMAIN\avu	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EMSDOMAIN\pb	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EMSDOMAIN\tio	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NT AUTHORITY\SYSTEM	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
sa	SQL Server login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
tester	SQL Server login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
V-XP-MUI\tester	Windows user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Server...

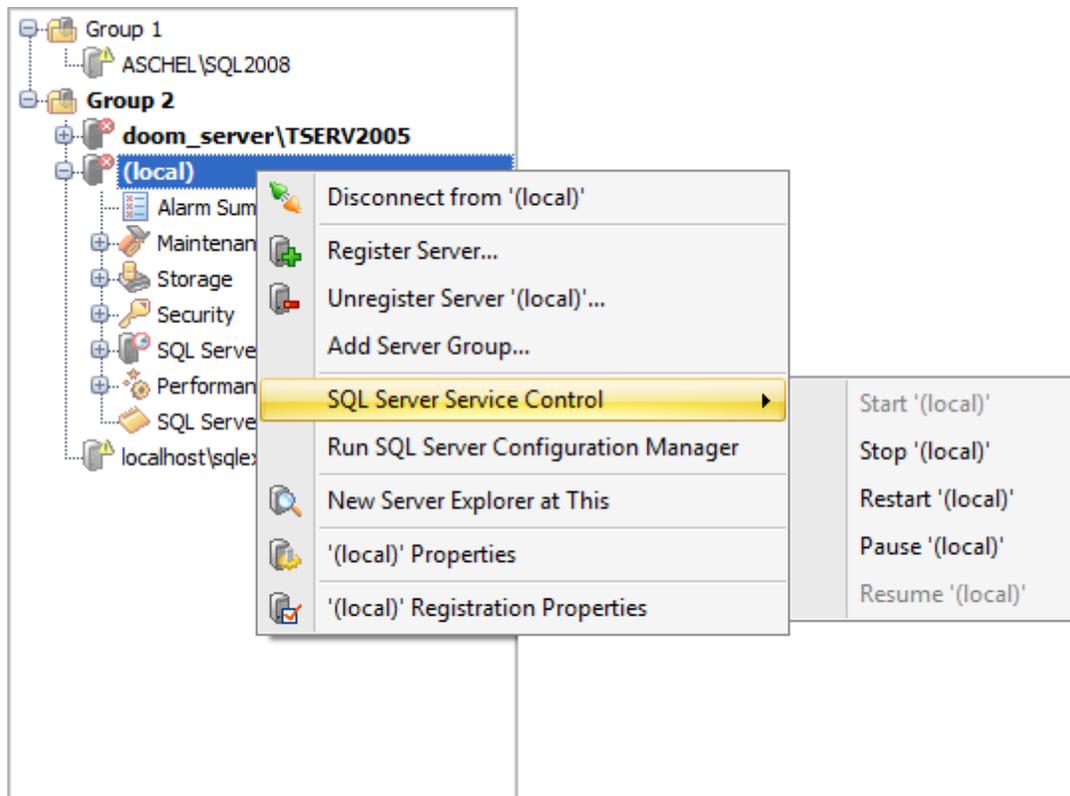
**See also:**[Server Explorer](#)[Working with servers](#)[Main menu & Toolbars](#)[Working with windows](#)[Working with grid](#)

## 2.2 Server Explorer

**Server Explorer** is the main navigation window for managing servers, maintenance tasks, statistics, etc. All objects are available within the corresponding nodes of the tree. It allows performing practically all server tasks. A server appears in the Server Explorer after it is [registered](#).

To launch the Server Explorer choose **Tools | Server Explorer** main menu item.

To expand/collapse a node, you can double-click it or use the +/- icons.

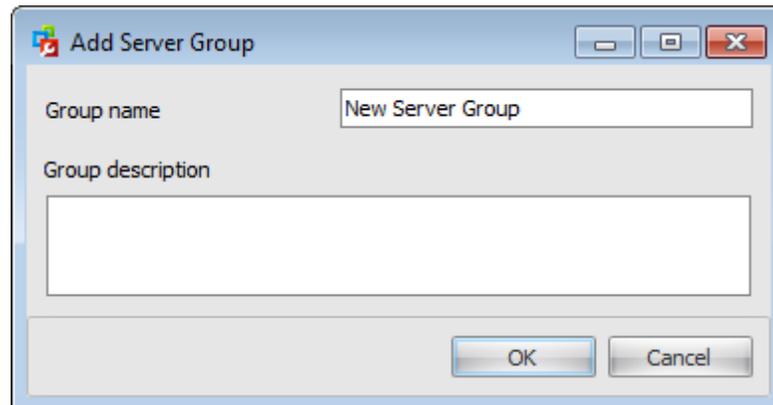


The context menu of the Server in the Server Explorer allows you to:

- connect to the selected server (if connection to the server is not active yet);
- disconnect from the selected server (if connection to the server has been already set);
- [register](#) a new server;
- unregister the selected server;
- add new server group;
- start server;
- stop server;
- restart server;
- pause server;
- resume server;
- run SQL Server Configuration Manager;
- open a new Server Explorer;
- view [server properties](#);
- view [server registration properties](#).

All servers can be grouped into **Server Groups**. To add a server group right-click within

the **Server Explorer** and select the **Add Server Group...** context menu item.



In the opened dialog define the **Group name** and enter the **Group description**.

To move a server from one group to another one just drag and drop it. Alternatively, you can launch the [Server Registration](#) dialog for the selected server and change the **Server Group** in the [General](#) section.

To reorder servers within the server group or groups drag and drop it holding the 'Shift' button.

---

**See also:**

[Work area](#)

[Working with servers](#)

[Database Explorer](#)

[Main menu & Toolbars](#)

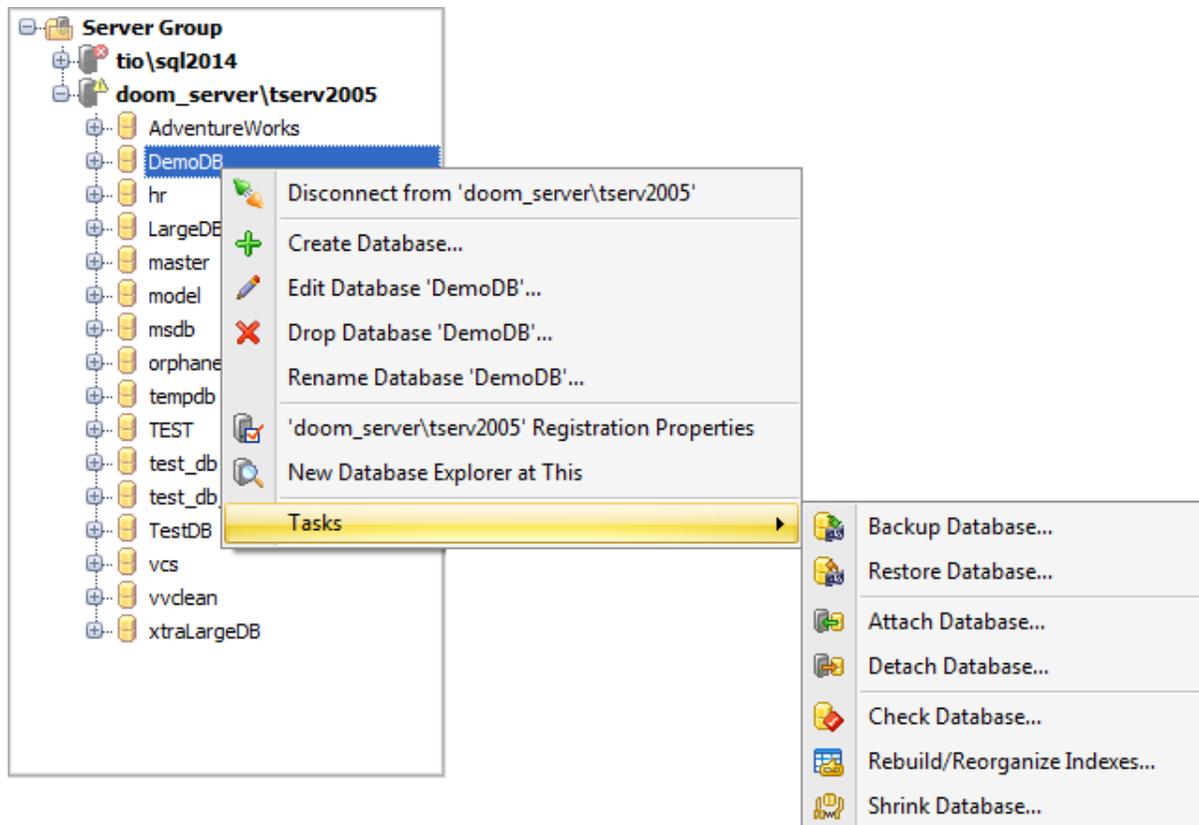
[Working with windows](#)

[Working with grid](#)

## 2.3 Database Explorer

**Database Explorer** is the navigation window for managing databases, database objects and database maintenance tasks. All objects are available within the database node of the tree. It allows viewing practically all database objects. A server appears in the Database Explorer after it is [registered](#).

To launch the Server Explorer choose **Tools | Database Explorer** main menu item. To expand/collapse a node, you can double-click it or use the +/- icons.



The context menu of the Database in the Database Explorer allows you to:

- disconnect from the server;
-  [create](#) database
-  [edit](#) selected database(s)
-  drop selected database(s)
- rename selected database
- view [server registration properties](#);
- open a new Database Explorer;
- perform the following tasks on the selected databases:
  -  [backup database](#)
  -  [restore database](#)
  -  [attach database](#)
  -  [detach database](#)
  -  [check database](#)
  -  [rebuild/reorganize indexes](#)

 [shrink database](#)

---

**See also:**

[Work area](#)

[Working with servers](#)

[Server Explorer](#)

[Database Objects Management](#)

[Main menu & Toolbars](#)

[Working with windows](#)

[Working with grid](#)

## 2.4 Working with servers

### Registering a Server

Before working with a server in SQL Administrator you need to register it. This is done by means of the [Register Server dialog](#). To open this dialog:

- select the **Server |  Register Server [main menu](#)** item;
- use the ** Register Database [toolbar](#)** button;
- right-click within the [Server Explorer](#) and select the ** Register Server...** context menu item.

### Connecting to a Server

To connect to the selected server:

- choose the **Server | Connect to <server\_name> [main menu](#)** item;
- select the **Connect to <server\_name>** item of the server context menu;
- double-click the server alias in the [Server Explorer](#).

### Disconnecting from a Server

To disconnect from the selected server:

- choose the **Server | Disconnect from <server\_name> [main menu](#)** item;
- select the **Disconnect from <server\_name>** item of the server context menu;

### Viewing Server Registration Properties

To view and configure server [registration properties](#) right-click the server alias in the [Server Explorer](#) and select the **<server\_name> Registration Properties...** item of the context menu.

### Managing Server Groups

Servers can be grouped in the [Server Explorer](#).

### Unregistering a Server

To unregister a server:

- choose the **Server |  Unregister server <server\_name> [main menu](#)** item;
- right-click the server alias in the [Server Explorer](#) and select the ** Unregister server <server\_name>** context menu item.

---

#### See also:

[Work area](#)

[Server Explorer](#)

[Main menu & Toolbars](#)

[Working with windows](#)

[Working with grid](#)

## 2.5 Main menu & Toolbars

### Main menu

Main menu provides access to main program tools and settings. It is located on the top of the program window.



### Toolbars

A **toolbar** is a horizontal row or vertical column of selectable image buttons that give the user a constantly visible reminder of and an easy way to select certain application functions.



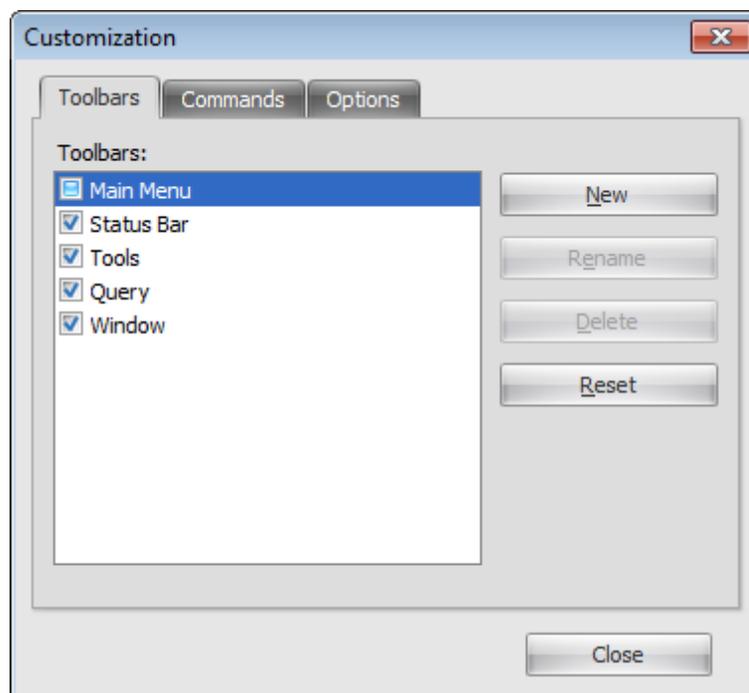
Most SQL Administrator toolbars are dockable, i.e. you can place a toolbar to any available location within the parent window. Toolbars may be changed depending on the active window.

### Customizing toolbars

The **Customize** dialog allows you to create and personalize toolbars.

To call this dialog, select the **Tools | Customize** main menu item or right-click any toolbar and select the **Customize...** popup menu item. Alternatively, click **More buttons...** on the right side of any, then click **Add or Remove Buttons** and select **Customize...** from the drop-down menu.

### Toolbars



This list displays all currently existing toolbars (both *default* and *user-defined* toolbars). Check/uncheck the box at a toolbar name to show/hide the toolbar.

### New

Use this button to add a new user-defined toolbar to the **Toolbars** list. Set a name for the newly created toolbar and dock it by dragging it to any permitted location within the application window.

### Rename

Use this button to rename the selected user-defined toolbar.

### Delete

Use this button to delete the selected user-defined toolbar.

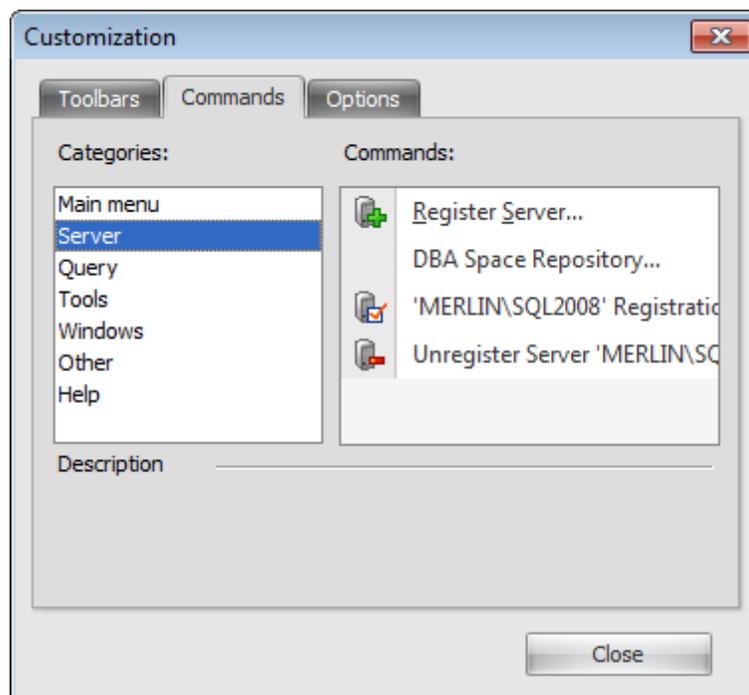
### Reset

Use this button to reset all settings to the initial state.

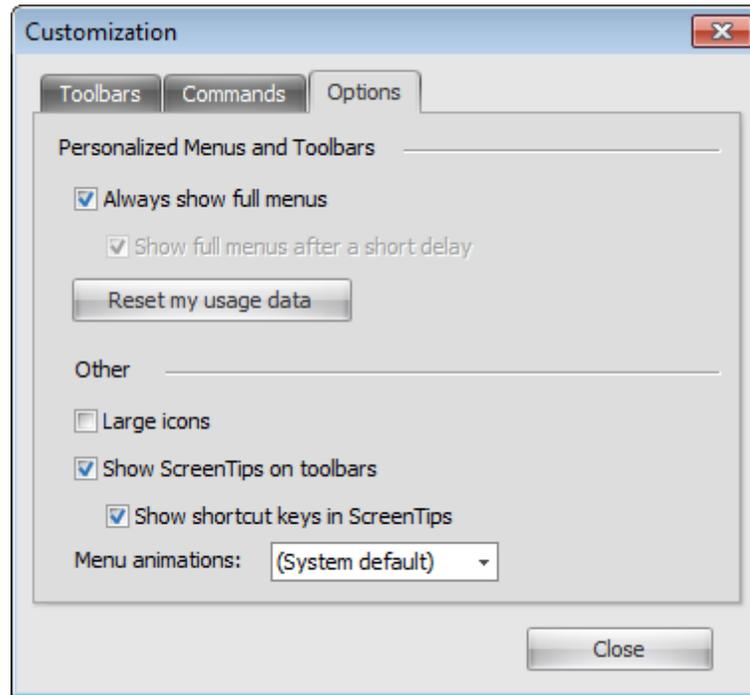
### Commands

This tab allows you to browse the list of all commands available within the menus and toolbars of the application window. Selecting categories in the **Categories** list displays commands of the selected category (e.g. 'Server' or 'Tools') in the **Commands** list.

If necessary, you can pick a command and drag it to any toolbar to create a button for this command.



### Options



**Always show full menus**

If this option is checked, then all menu items (including infrequently used) will be always shown in the menu.

**Show full menus after a short delay**

This option is available only if the **Always show full menus** option is not selected. If this option is enabled, infrequently used menu items (if they appear "collapsed") will be automatically expanded after a delay upon setting mouse cursor (or upon selection with the *Up/Down* keys) on the bottom of the menu. Otherwise, the menu expands only after clicking its bottom-most button.

**Reset my usage data**

Resets the lists of recently used commands in the toolbars and menus.

**Large icons**

This option displays larger icons on the parent window toolbars.

**Show ScreenTips on Toolbars**

If this option is selected, ToolTips (hints) popup when the mouse cursor is positioned over a toolbar button.

**Show shortcut keys in ScreenTips**

If this option is selected, the corresponding shortcuts are displayed in ToolTips (hints) for toolbar buttons.

**Menu animations**

Use the drop-down list to specify the menu animation effects:

*None* (no animation)

*Slide* (sliding menus)

*Fade* (menus fade in when appearing)

*Unfold* (unfolding menus)

*Random* (random choice: *Unfold, Slide, Fade*)

---

**See also:**

[Work area](#)

[Server Explorer](#)

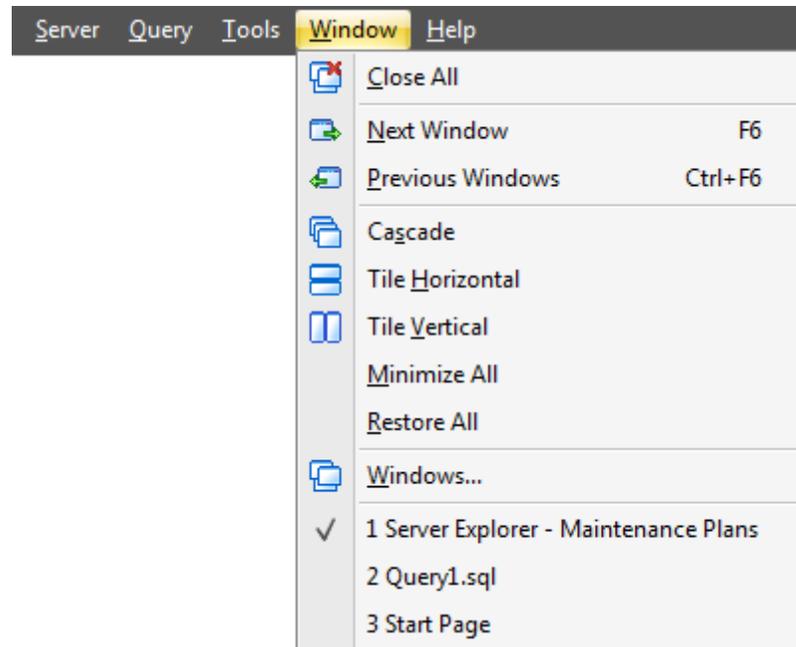
[Working with servers](#)

[Working with windows](#)

[Working with grid](#)

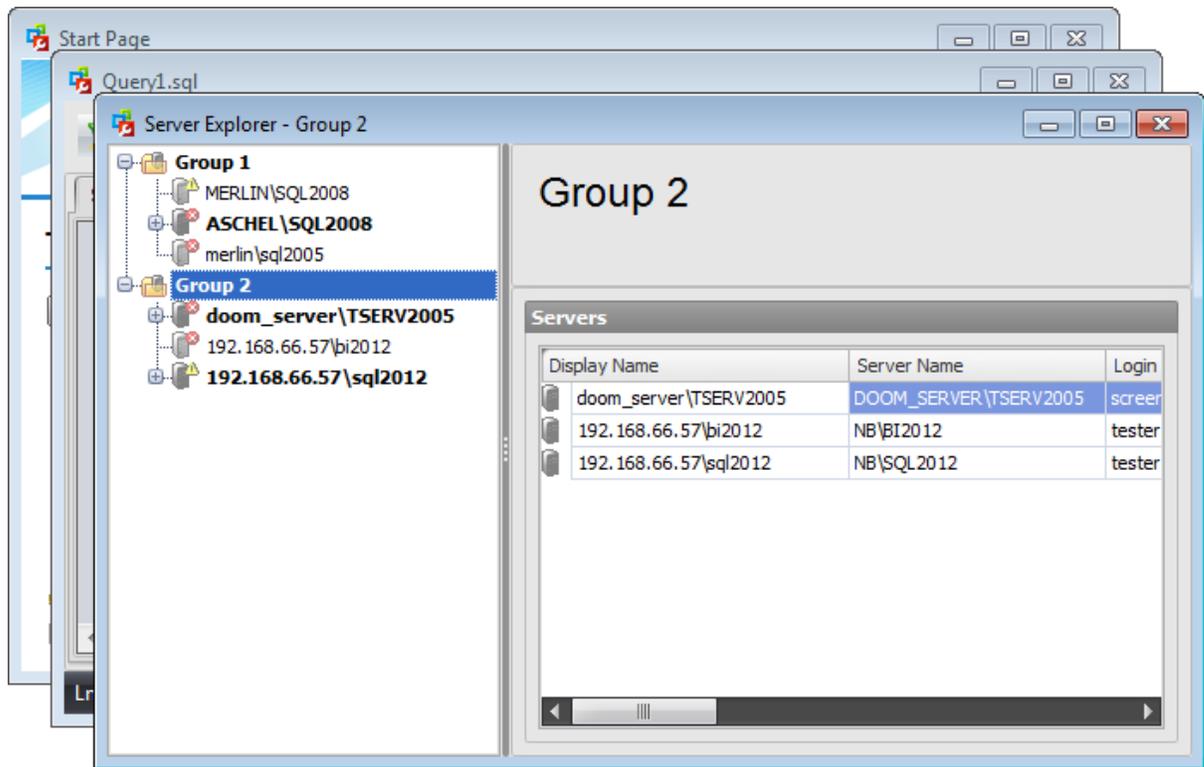
## 2.6 Working with windows

The **Windows** menu facilitates your work with SQL Administrator windows.

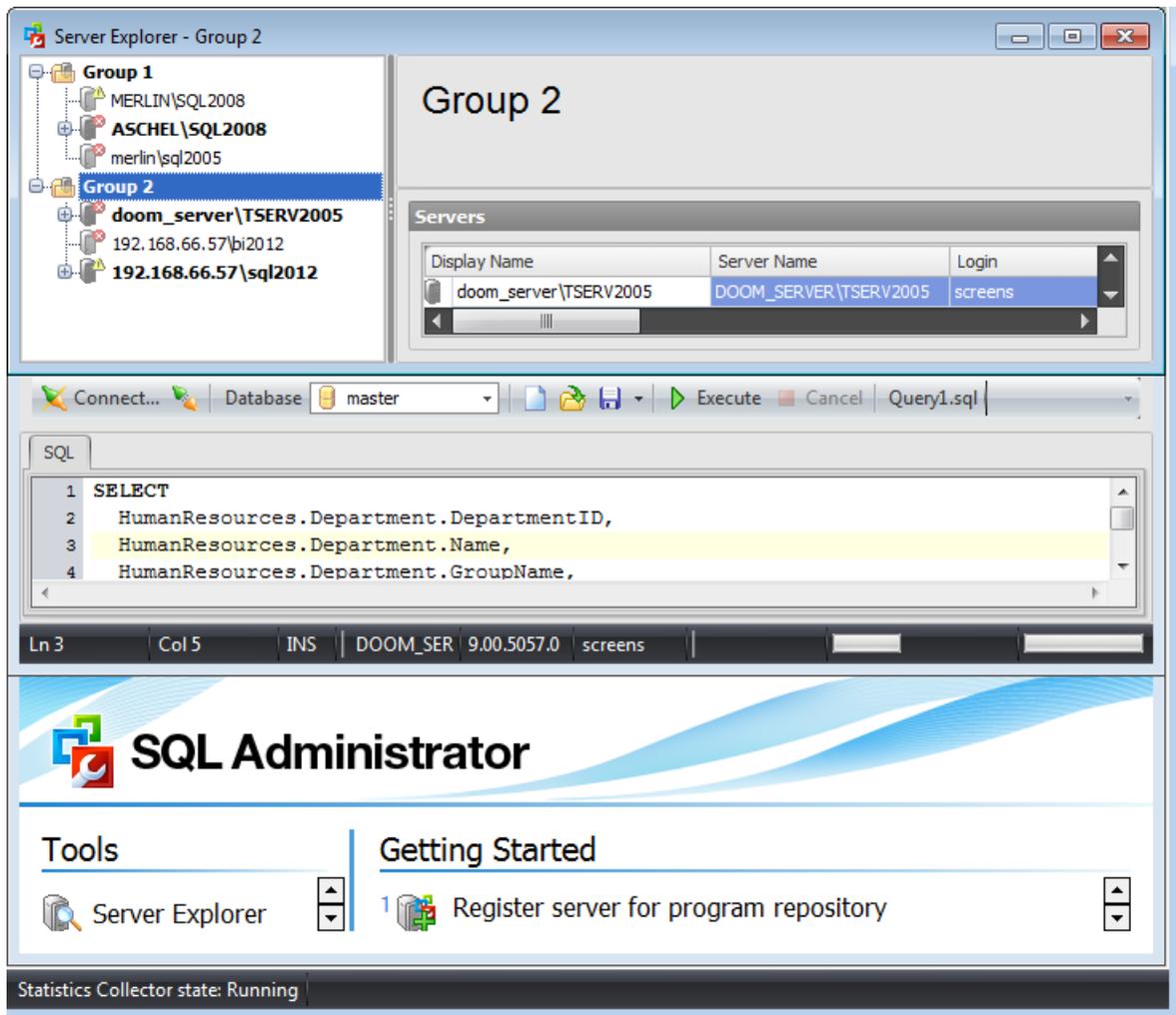


The **Windows** menu allows you to:

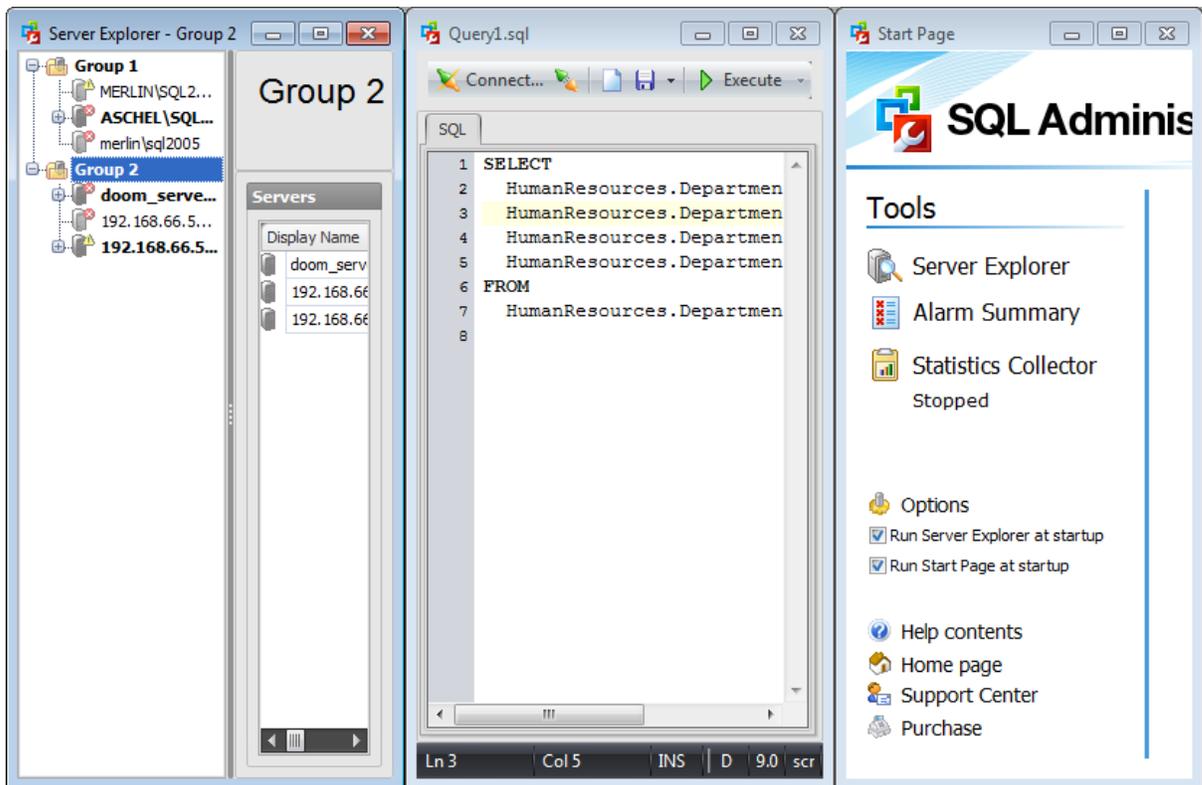
- set all current windows *cascade*:



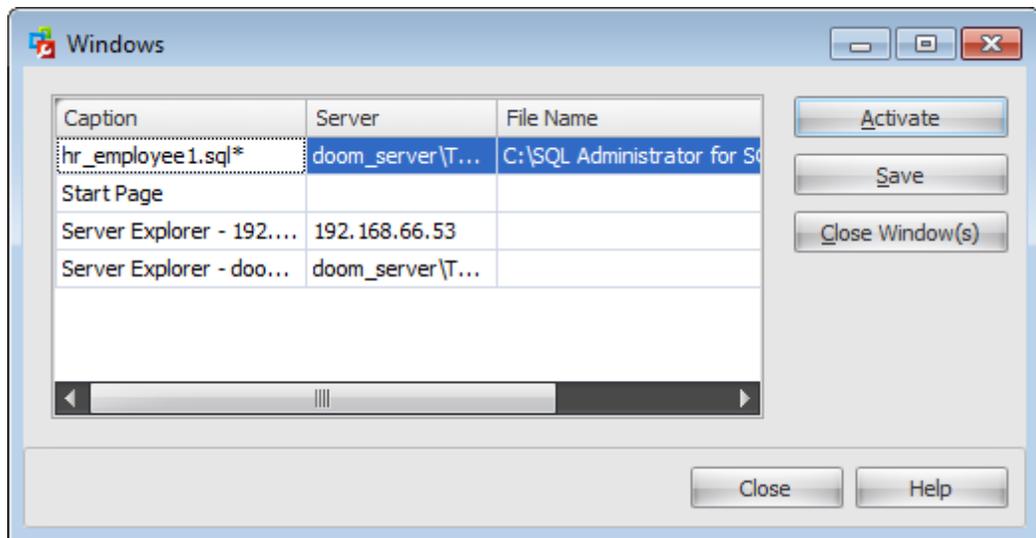
- minimize all windows;
- restore all windows;
- tile all current windows *horizontally*:



- tile all current windows *vertically*:



- close all windows;
- switch to the previous window;
- switch to the next window;
- view the list of opened windows:



Using this dialog you can **Activate** or **Close** any window from the list. Also, it allows you to **Save** queries into the \*.sql files.

To activate selected window you can double-click within the grid.

**Note:** Most of these options are available if the **Windows layout** is set to  *Multiple documents* in the [Options | Environment](#).

---

**See also:**

[Work area](#)

[Server Explorer](#)

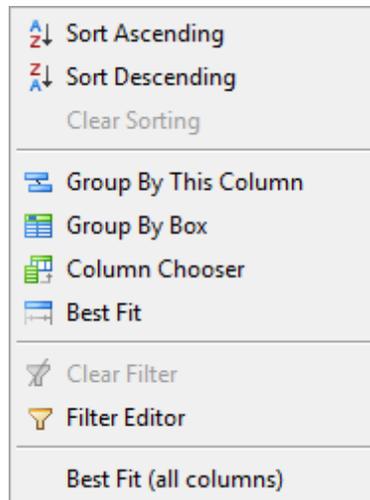
[Working with servers](#)

[Main menu & Toolbars](#)

[Working with grid](#)

## 2.7 Working with grid

The context menu of the column heads in the grid is the following:



### Sorting

To sort records in the grid choose the column by which you want to sort data and click the column title.

If the column was not sorted, the first click will sort it in the ascending order and the second one - in the descending order.

Alternatively, you can choose  **Sort Ascending** or  **Sort Descending** context menu item.

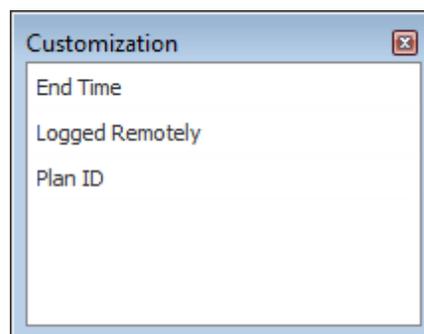
### Clear Sorting

To cancel the sorting, open the context menu by right-clicking the necessary column and choose the **Clear Sorting** item.

### Selecting visible columns

You can specify which columns of the current dataset will be visible. Click the  button available on the toolbar or in the context menu. Drag and drop the column heading from the opened box to the head of the table.

To delete a visible column from the grid drag and drop its heading away from the grid heading.



### Changing columns order

For your convenience the possibility to *change the order* of the columns in the data grid is available. To reorder columns, drag a column caption horizontally to a place in between two other column headers indicated with a black frame.

Maintenance Plan Name	EndTime	Subplan Name	Run Status	End Time
-----------------------	---------	--------------	------------	----------

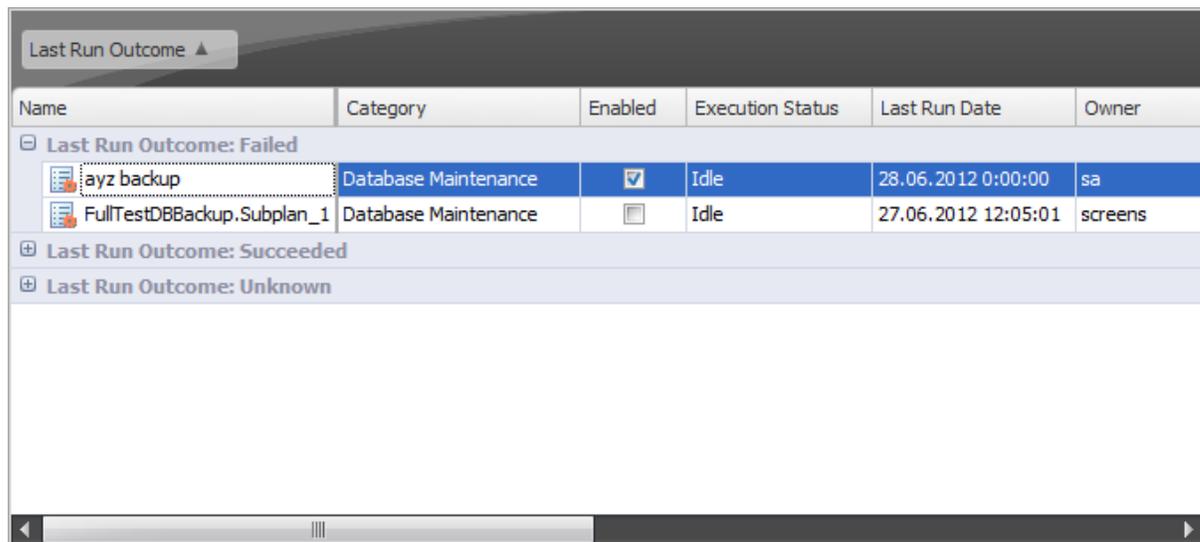
### Grouping

If necessary, you can **group the data in grid** by any of the columns. This operation is performed by dragging the column header to the **"Group by" box** area at the top.

Alternatively you can use the  **Group By This Column** context menu item. To show/hide the **"Group by" box** area use the  **Group By Box** context menu item.

When grouping by a column is applied to the grid, all the rows are displayed as subnodes to the grouping row value. To reverse grouping, just drag the column header back.

**Hint:** While dragging the column header back, you can also change the column position.



Name	Category	Enabled	Execution Status	Last Run Date	Owner
<b>Last Run Outcome: Failed</b>					
ayz backup	Database Maintenance	<input checked="" type="checkbox"/>	Idle	28.06.2012 0:00:00	sa
FullTestDBBackup.Subplan_1	Database Maintenance	<input type="checkbox"/>	Idle	27.06.2012 12:05:01	screens
<b>Last Run Outcome: Succeeded</b>					
<b>Last Run Outcome: Unknown</b>					

If necessary, you can group data by two or more columns. In this case column headers are displayed hierarchically, and data are grouped by these columns in the order the column headers appear in the **"Group by"** area.

Name	Enabled	Execution Status	Last Run Date	Owner	Next Run Date	Sch
Last Run Outcome: Failed						
Last Run Outcome: Succeeded						
Category: [Uncategorized (Multi-Server)]						
Job1_TEST	<input checked="" type="checkbox"/>	Unknown	06.11.2007 18:33:38	sa		
Category: Database Maintenance						
CheckTestDB.Subplan_1	<input type="checkbox"/>	Idle	25.06.2012 11:04:01	screens		
Last Run Outcome: Unknown						

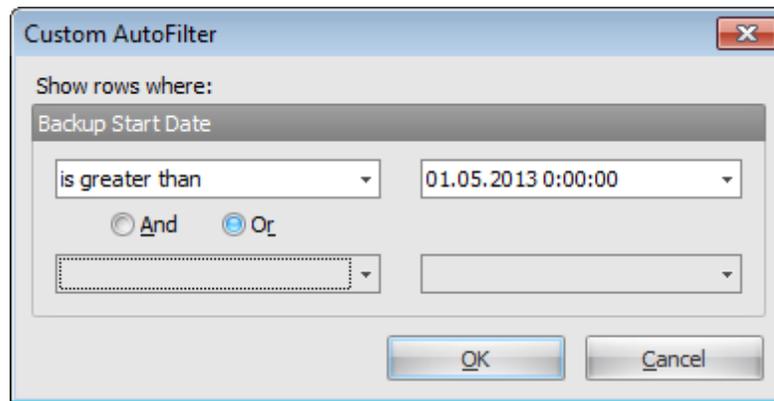
### Filtering

A number of **filtering** facilities are implemented in the grid for your convenience. You can filter records in the grid in either of the following ways:

- click the filter button next to the column caption to display the drop-down list and select any of the column values to filter records by this value of the selected column;

Name	Enabled	Execution Status	Last Run Outcome	Category	Owner
ayz backup	<input checked="" type="checkbox"/>	Idle	Failed	(Custom)	sa
FullTestDBBackup.Subplan_1	<input type="checkbox"/>	Idle	Failed	(Blanks)	screens
CheckTestDB.Subplan_1	<input type="checkbox"/>	Idle	Succeeded	(Non blanks)	screens
Job1_TEST	<input checked="" type="checkbox"/>	Unknown	Succeeded	Failed	screens
ShrinkTestDB.Subplan_1	<input checked="" type="checkbox"/>	Idle	Unknown	Succeeded	screens
MaintenancePlan1	<input checked="" type="checkbox"/>	Idle	Unknown	Unknown	screens
cleanup_stats	<input checked="" type="checkbox"/>	Idle	Unknown	Database Maintenance	screens
cleanup_alarm_log	<input checked="" type="checkbox"/>	Idle	Unknown	SQL Administrator Repository	screens
avg_stats	<input checked="" type="checkbox"/>	Idle	Unknown	SQL Administrator Repository	screens

- click the filter button next to the column caption to display the drop-down list, then select the **Custom** item and build a simple filter using the Custom Filter dialog;



- use the  **Filter Editor** context menu item to invoke the [Filter Builder](#) dialog and create a composite filter using the dialog.

After the filter is set, the gray **filtering panel** becomes visible at the bottom of the grid. This panel allows you to see the active filtering condition and easily enable or disable it using the checkbox on the left. The Arrow-down button opens the drop-down menu which allows you to browse the filter history for this dataset.

If necessary, you can click the **Edit Filter...** button on the right to customize your filter and add more complicated filtering conditions within the [Filter Builder](#) dialog.



To remove the current filter, click the **×** **Close** button.

---

#### See also:

[Work area](#)

[Server Explorer](#)

[Working with servers](#)

[Main menu & Toolbars](#)

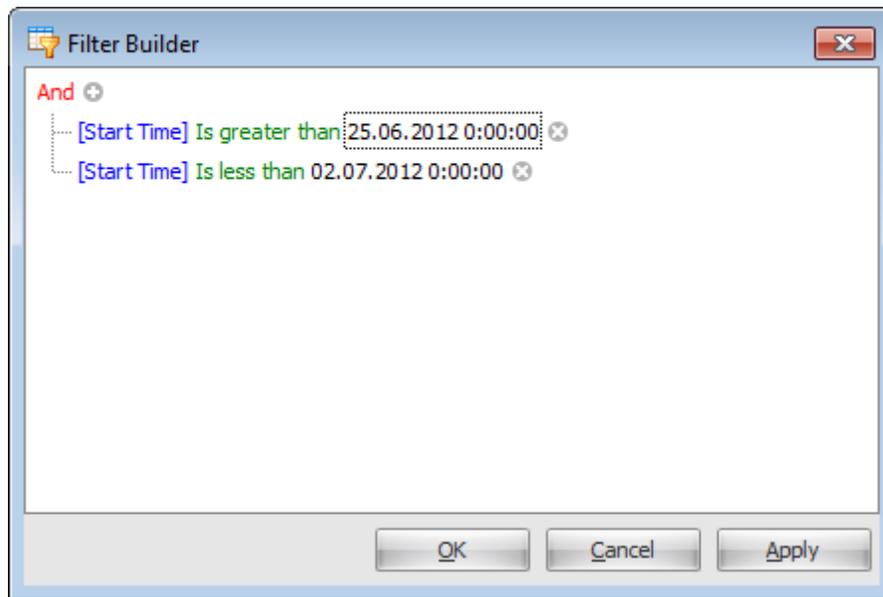
[Working with windows](#)

## 2.7.1 Filter Builder dialog

The **Filter Builder** dialog is a powerful filtering tool. In this dialog you can set as much conditions as you need.

It is aimed at facilitating the procedure of creating and applying complex filter criteria for data.

To open the **Filter Builder** dialog, use the  **Filter Editor** item of the grid heading context menu.



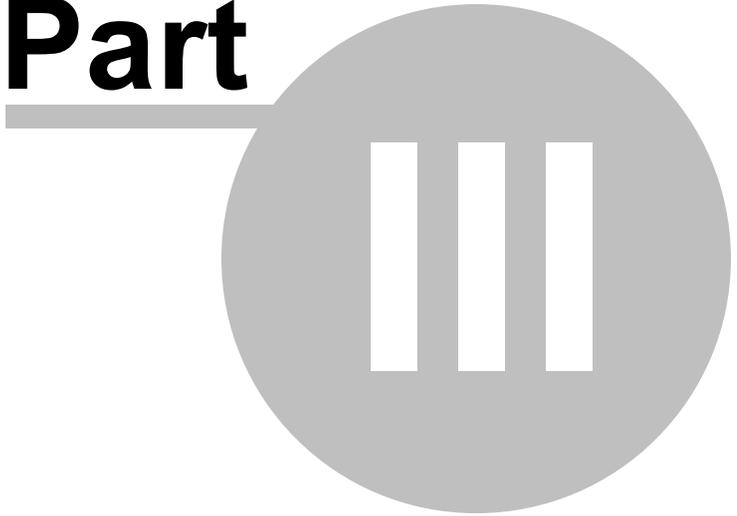
Condition consists of Column, Logical operator and Value. Press any of these elements to change its value.

Two or more conditions form Condition group. Its popup menu allows you to change relations between child conditions ( **And**,  **Or**,  **Not And**,  **Not Or**), to  **Add Condition** or  **Condition Group** or to  **Remove Group**. For root condition group the  **Clear All** operation is also available.

**Note:** You can use the *Insert* keyboard button to add a condition to the selected group and the *Delete* button to clear the filter.

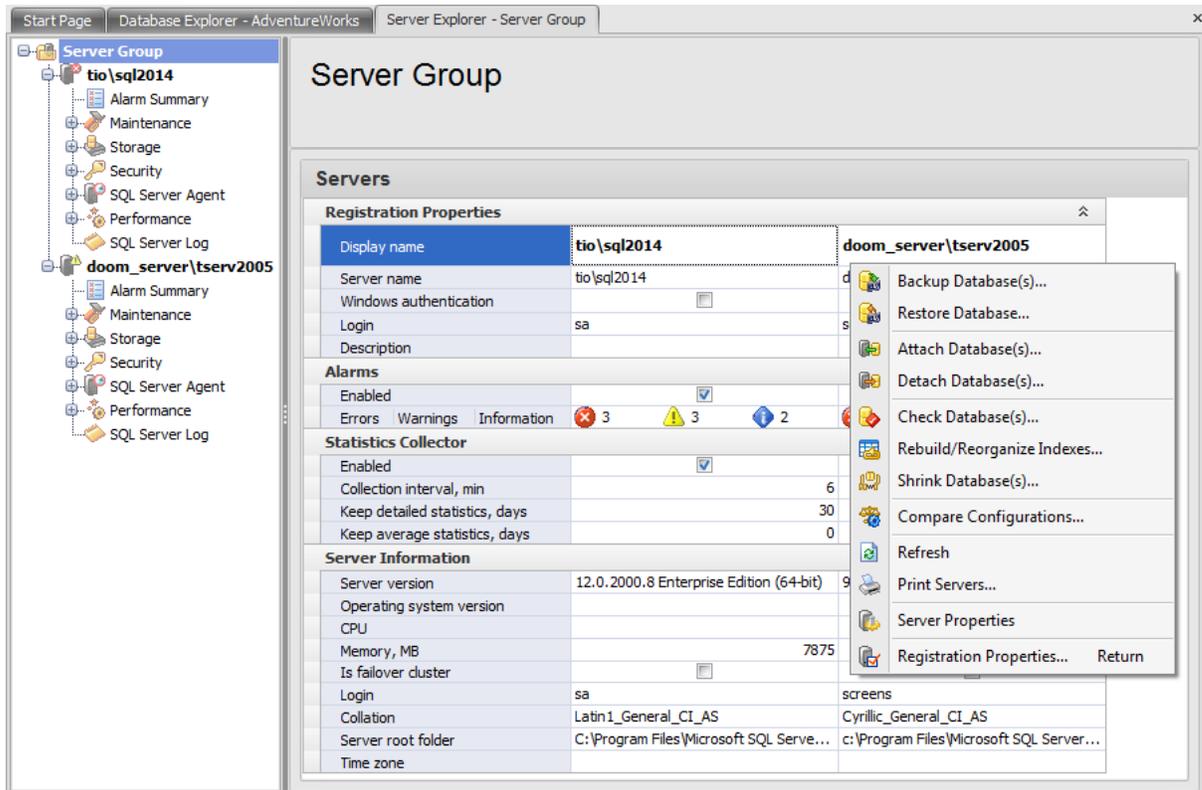


**Part**



### 3 Servers

All [registered](#) servers are grouped into server groups. When you choose a group in the [Server Explorer](#) the [Work area](#) looks the following way:



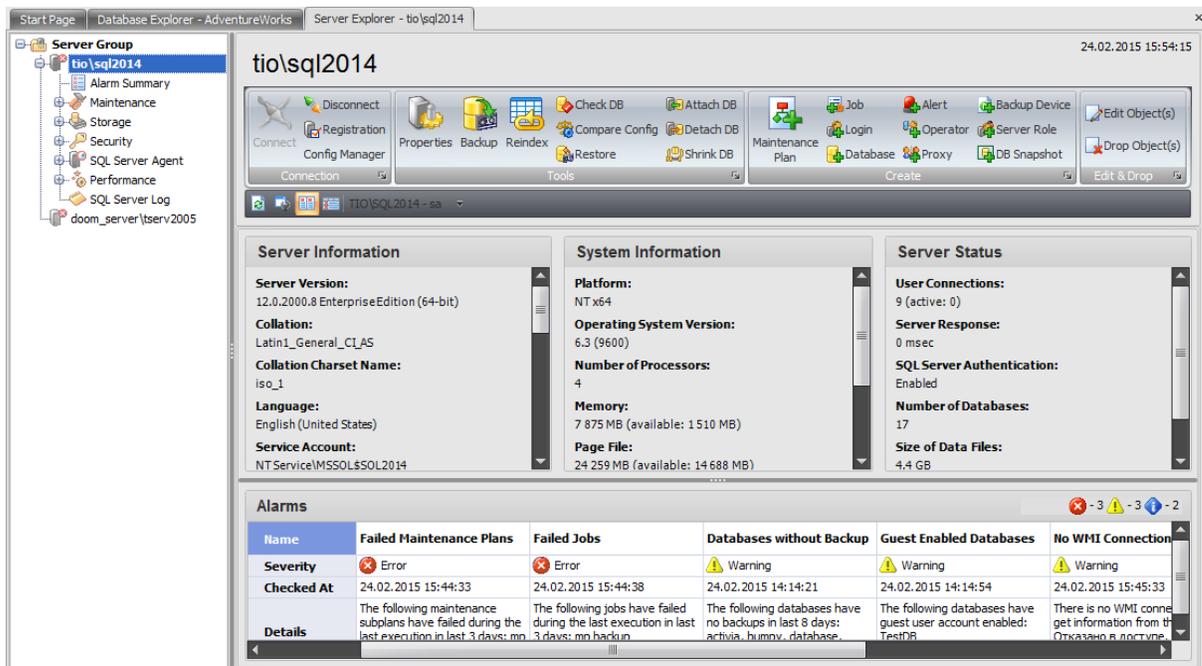
In this grid all registered servers are displayed. Here you can see [Server Registration Properties](#), [Alarms](#) information, [Statistics Collector](#) options and Server Information.

The context menu of the server allows you to launch wizards to perform common [database maintenance](#) operations, [Compare server configurations](#), refresh server information, [print](#) the document with servers information, view [Server Properties](#) and [Registration Properties](#).

Double click the cell with number of alarms to open the [Alarm Summary](#) tab for the selected server.

Double-clicking any other section opens the [Registration properties](#) dialog on the corresponding tab.

When you select a specific server in the [Server Explorer](#) the [Work area](#) looks the following way:



Using the buttons on the ribbon you can perform different operations with the server and objects.

### Connection

- connect to the server/disconnect from the server;
- open the [Server Registration Properties](#) dialog;
- open local SQL Server Configuration Manager.

### Tools

- open [Server Properties](#) dialog;
- run database maintenance wizards:
  - ✓ [Backup database](#)
  - ✓ [Rebuild/Reorganize Indexes](#)
  - ✓ [Check database](#)
  - ✓ [Attach database](#)
  - ✓ [Detach database](#)
  - ✓ [Restore database](#)
  - ✓ [Shrink database](#)
- run [Compare Server Configurations](#) wizard.

### Create

Using buttons on this ribbon you can create new server objects in the corresponding object editors:

- ✓ [Maintenance Plan Editor](#)
- ✓ [Job Editor](#)
- ✓ [Login Editor](#)
- ✓ [Database Editor](#)
- ✓ [Alert Editor](#)
- ✓ [Operator Editor](#)
- ✓ [Proxy Editor](#)
- ✓ [Backup Device Editor](#)

- ✓ [Server Roles Editor](#)
- ✓ [Database Snapshots](#)

### Edit & Drop

Use these buttons to open the Select Objects dialog where you can choose server objects that you want to edit or delete.

The [Work area](#) consists of four sections: Server Information, System Information, Server Status and Alarms.

**System Information** area provides you with the system information where the server selected in the [Server Explorer](#) is installed.

**Server Status** section provides you with the current SQL Server parameters: *User Connections, Server Response (msec), SQL Server Authentication (enabled/disabled), Number of Databases, Size of Data Files, Size of Log Files.*

The **Alarms** area contains information about problems occurred on the server. In the header the number of alarms of each type is shown:  Information,  Warning,  Error.

The alarm options are set on the [Alarm Summary](#) tab of the [Server Registration Properties](#) when registering or configuring the server. To set alarm options you can also proceed to the [Alarm summary](#) section and click the  button.

---

### See also:

[Getting Started](#)  
[Database Maintenance](#)  
[Storage](#)  
[Security](#)  
[SQL Server Agent](#)  
[Performance](#)  
[SQL Server Log](#)  
[Query Editor](#)  
[Options](#)

## 3.1 Register server/Registration properties

Before working with a server in SQL Administrator you need to register it. This is done by means of the Register Server dialog. To open this dialog:

- select the **Server |  Register Server [main menu](#)** item
- use the ** Register Database [toolbar](#)** button
- right-click within the [Server Explorer](#) and select the ** Register Server...** context menu item.

To edit properties of the already registered server right-click its alias and select the **<server\_name> Registration Properties** context menu item.

The list of **Register Server** dialog tabs is displayed on the navigation bar:

- [Configuring general connection settings](#)
- [WMI Connection](#)
- [Managing alarms](#)
- [Setting statistics collection parameters](#)

---

### See also:

[Alarm summary](#)

[SQL Administrator Repository](#)

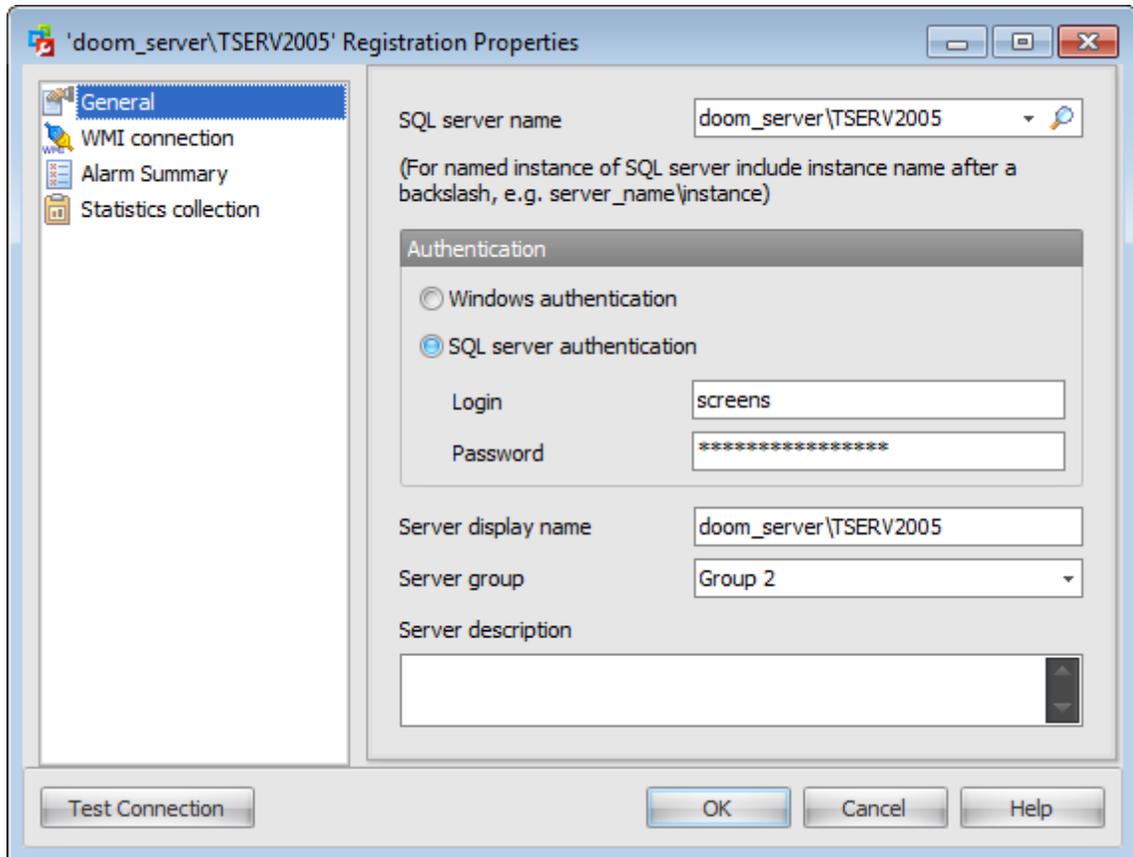
[WMI Connection](#)

[Statistics Collector](#)

[Server properties](#)

### 3.1.1 Configuring general connection settings

The **General** section of the **Register Server Wizard** allows you to view and change database server connection parameters.



#### SQL server name

Select server (host) to connect from the drop-down list. Click the  **Search** button to search for servers in a local network. In the list appeared choose the necessary server and press 'OK' button.

**Note:** If server name is not displayed in this list it can be entered manually.

#### Connection settings

SQL Server can be installed as a named instance. That is, its name is assigned during an installation process by default (e.g. SQL Server 2005 Express is set as a named instance SQLEXPRESS).

In this case you should enter the name of your machine (or IP address) and the instance name in the **SQL server name** field in the following format:

*computer\_name\sqlserver\_instance\_name* (e.g. "MYCOMPUTER\SQLEXPRESS").

If the instance is not named then specify only the name of your machine or its IP address.

#### Authentication

Specify the type of Microsoft® SQL Server™ authentication to be used for the connection:  *Windows authentication* or  *SQL Server authentication*. It is strongly recommended to avoid using SQL Server authentication with "sa" as the login.

**Windows Authentication**

SQL Administrator identifies a user by his domain name and password which are defined automatically according to his Windows domain user account.

**SQL Server authentication**

SQL Administrator identifies user by his server login and password. Login and password management is accomplished by Login Manager.

If  *Server authentication* is selected as the *authentication type*, you should provide *authorization* settings: **Login** and **Password**. Analyzing them a system determines information and therefore object access permissions.

**Note:** It is recommended to use a login that is a sysadmin fixed server role member so as to have an access for using all features of SQL Administrator.

**Server display name**

Enter the server name to be displayed in the [Explorer](#).

**Server group**

Select or enter server group that the server belongs to. Server groups are used to group servers in the [Server Explorer](#) visually.

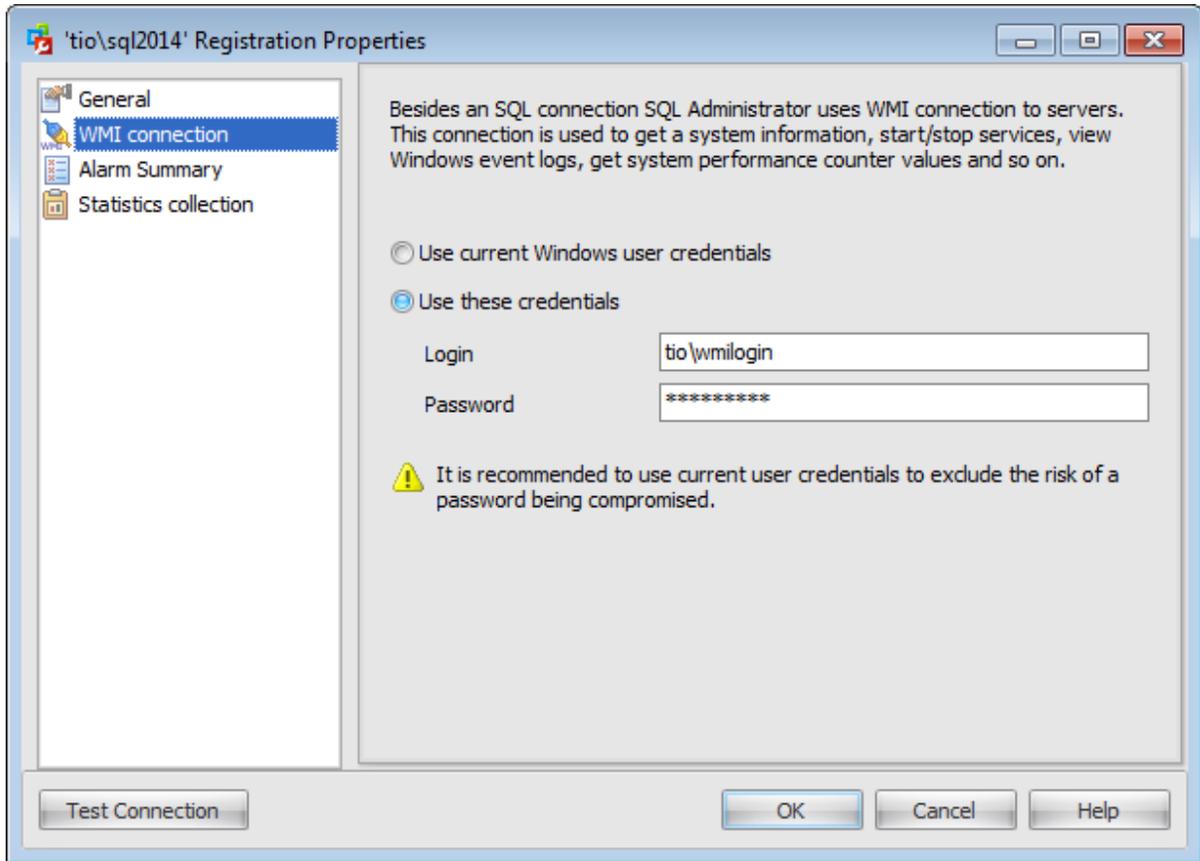
**Server description**

Here you can enter some additional information.

Click the **Test Connection** button to check the ability to connect to the defined server.

### 3.1.2 WMI Connection

Here you are provided with some information on WMI connection.



- **Use current Windows user credentials**

This disabled option specifies that on WMI connection to the server machine current Windows account is used.

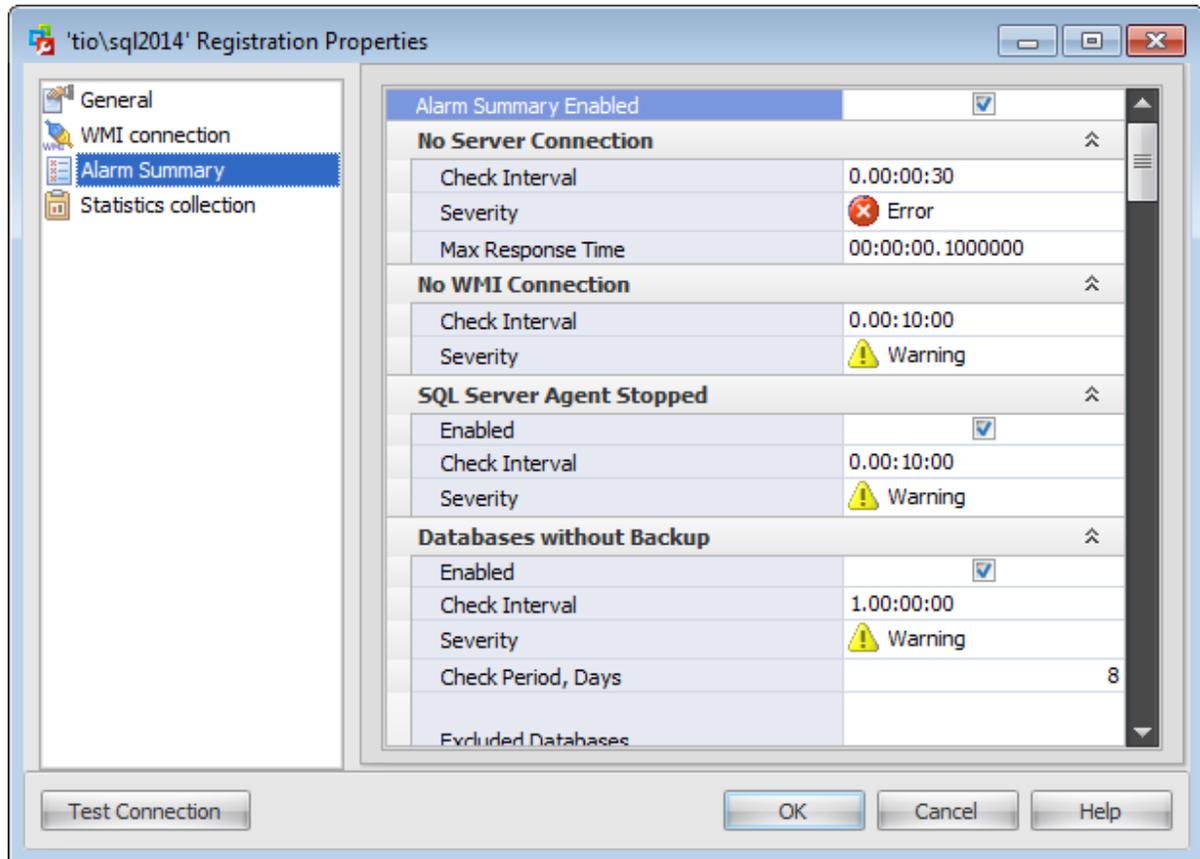
- **Use these credentials**

Specify *Login* and *Password* for a user that can connect to the remote computer via WMI.

To learn more about WMI connection, see the [WMI Connection](#) chapter.

### 3.1.3 Managing alarms

Use the **Alarm Summary** tab to set parameters of different alarms. An alarm is a message that informs you about possible malfunctions and potential problems on the server. All these messages are displayed in the [Alarm summary](#) window.



#### Alarm Summary Enabled

Check this option to enable alarming on all checked events.

**Note:** *No Server Connection* and *No WMI Connection* alarms are always enabled / disabled by this option.

Practically all alarms (except *No Server Connection* and *No WMI Connection*) have three common parameters:

#### Enabled

Check this option to enable alarming on the particular event.

#### Check Interval

Define the interval in the '*d.hh:mm:ss*' format. The server will be checked on alarms occurred in the defined period of time.

#### Severity

Set the preferable severity degree for every alarm:

- Information
- Warning



Error

**Note:** The server status is the highest severity degree of all fired alarms.

Severity degree is configured for your convenience, so as you could see these icons in the [Alarm summary](#).

#### **No Server Connection**

This alarm warns about failed connection to the server.

#### **Max Response Time**

Define the period in the *'d.hh:mm:ss'* format. If a connection timeout exceeds the defined time interval then the *No Server Connection* alarm appears.

#### **No WMI Connection**

This alarm appears when there is no WMI connection to get information from the server. For more details, see [WMI Connection](#).

#### **SQL Server Agent Stopped**

This alarm informs you that [SQL Server Agent](#) is not running and therefore scheduled [jobs](#) are not performed.

#### **Databases without Backup**

This alarm shows databases that have not been [backed up](#) during the *Check Period*.

#### **Check Period, Days**

Type in or use the spinner control to define the number of days to check for not backed up databases.

#### **Excluded Databases**

Enter the databases (one on the line) that you do not want to be checked for backups. The list of databases can be found [here](#).

#### **Failed Backups**

This alarm shows [backup](#) maintenance plans/jobs that have failed during the last execution.

#### **Excluded Maintenance Subplans**

Enter the backup maintenance subplans (one on the line) which you do not want to be alarmed about in case of their failure.

#### **Excluded Jobs**

Enter the backup jobs (one on the line) which you do not want to be alarmed about in case of failure.

#### **Failed Maintenance Plans**

This alarm shows [subplans](#) that have failed up during the last execution within the *Check Period*.

#### **Check Period, Days**

Type in or use the spinner control to define the number of days to check for failed maintenance plans.

**Excluded Maintenance Plans**

Enter the maintenance plans (one on the line) which you do not want to be alarmed about in case of failure.

**Failed Jobs**

This alarm shows [jobs](#) that have failed up during the last execution within the *Check Period*.

**Check Period, Days**

Type in or use the spinner control to define the number of days to check for failed jobs.

**Excluded Jobs**

Enter the jobs (one on the line) which you do not want to be alarmed about in case of failure.

**Filling up of Database Space**

This alarm shows databases that have filled up [filegroups](#).

**Excluded Filegroups (database, filegroups)**

Enter the database filegroups as *<database\_name, filegroup\_name>* (one on the line) that you do not want to be checked for filling up.

**Max Fullness Percent**

Type in or use the spinner control to set the maximum percentage of a filegroup filling. The alarm will appear when the filegroup filling exceeds the defined percentage value.

**Guest Enabled Databases**

This alarm shows databases that have guest [user](#) account enabled.

**Excluded Databases**

Enter the databases (one on the line) that you do not want to be checked for enabled guest user account.

**Orphaned Logins**

This alarm shows orphaned logins. A [login](#) can become orphaned if the corresponding Windows users or groups no longer exist. Also, a login can become orphaned after the master database transfer.

**Excluded Logins**

Enter the logins (one on the line) that you do not want to be alarmed about in case they are orphaned.

**Orphaned Users**

This alarm shows orphaned database users. A [database user](#) can become orphaned if the corresponding SQL Server login is dropped. Also, a database user can become orphaned after a database is restored or attached to a different instance of SQL Server.

**Excluded Users (database, user)**

Enter the database users as *<database\_name, database\_user>* (one on the line) that you do not want to be alarmed about in case they are orphaned.

**Mixed Mode Authentication**

This alarm informs you that the server authentication is set to a mixed mode, i.e.

Windows and SQL Server authentication mode.

#### Ole Automation Procedures Enabled

This alarm informs you that Ole Automation Procedures are enabled on the server. Ole Automation Procedures option specifies whether OLE Automation objects can be instantiated within Transact-SQL batches.

#### xp\_cmdshell Enabled

This alarm appears when *xp\_cmdshell* procedure is enabled on the server. The *xp\_cmdshell* procedure spawns a Windows command shell.

#### CPU Overload

This alarm appears when the CPU loading exceeds the *Max CPU Usage Percent*. It determines the average processor time, that is the percentage of time the processor is busy by measuring the percentage of time the thread of the Idle process is running and then subtracting that from 100 percent. This measurement is the amount of processor utilization.

#### Max CPU Usage Percent

Type in or use the spinner control to define the maximum percent of processor utilization.

#### Lack of Memory

This alarm notifies you that there is a possible lack of memory on the server and displays the current pages per second value and amount of available memory.

#### Max Pages per Second

Set the maximum number of physical pages read to (or written to) disk.

#### Min Available Memory Percent

Define the minimum percentage of available memory for the alarm to appear.

#### Busy Disks

This alarm shows you busy disks that have low idle time (less than *Min Disk Time Percent*).

#### Min Disk Idle Time Percent

Type in or use the spinner control to define the minimum percentage of a disk idle time for alarm to appear.

#### Large Disk Queue

This alarm shows disks that have high queue length (higher than *Max Disk Queue Length*).

#### Max Disk Queue Length

Type in or use the spinner control to define the maximum length of a disk queue for the alarm to appear.

#### High Disk Transfer Latency

This alarm raises when disk transfer latency exceeds the values specified in *Max Read Time* and *Max Write Time*. It shows current values for the read/write time.

#### Max Read Time

Type in or use the spinner control to define the maximum read time (in seconds) latency of a disk for the alarm to appear.

**Max Write Time**

Type in or use the spinner control to define the maximum write time (in seconds) latency of a disk for the alarm to appear.

**Low Page Life Expectancy**

Page Life Expectancy is a number of seconds a page will stay in the buffer pool without references. This alarm notifies that the page life expectancy became lower than the specified *Min Page Life Expectancy*.

**Min Page Life Expectancy**

Type in or use the spinner control to define the minimum page life expectancy (in seconds) for the alarm to appear. Note that the recommended value for this parameter is 300 seconds.

**Databases in Restricted User Mode**

This alarm notifies you that there are databases on the server that have the user access set to the restricted user mode.

**Excluded Databases**

Enter the databases (one on the line) that you do not want to be checked for the restricted user mode.

**Databases in Non-online State**

This alarm informs you that there are databases that are not in *Online* state. It shows databases that are in *Offline* or *Emergency* state.

**Excluded Databases**

Enter the databases (one on the line) that you do not want to be checked for the non-online state.

**Errors in Server Error Log**

This alarm raises when the string specified in the *Search for* field appears in the SQL Server Error Log. It shows the full text of the error occurred.

**Search for**

Enter the substring of the SQL Server error that you want to be notified about (for example, *'login failed'*).

**Low Buffer Cache Hit Ratio**

This alarm informs you that buffer cache hit ration is low (lower than *Min Buffer Hit Ratio*). Buffer Cache Hit Ratio is the percentage of SQL Server pages requested and retrieved from the buffer cache without reading from disk.

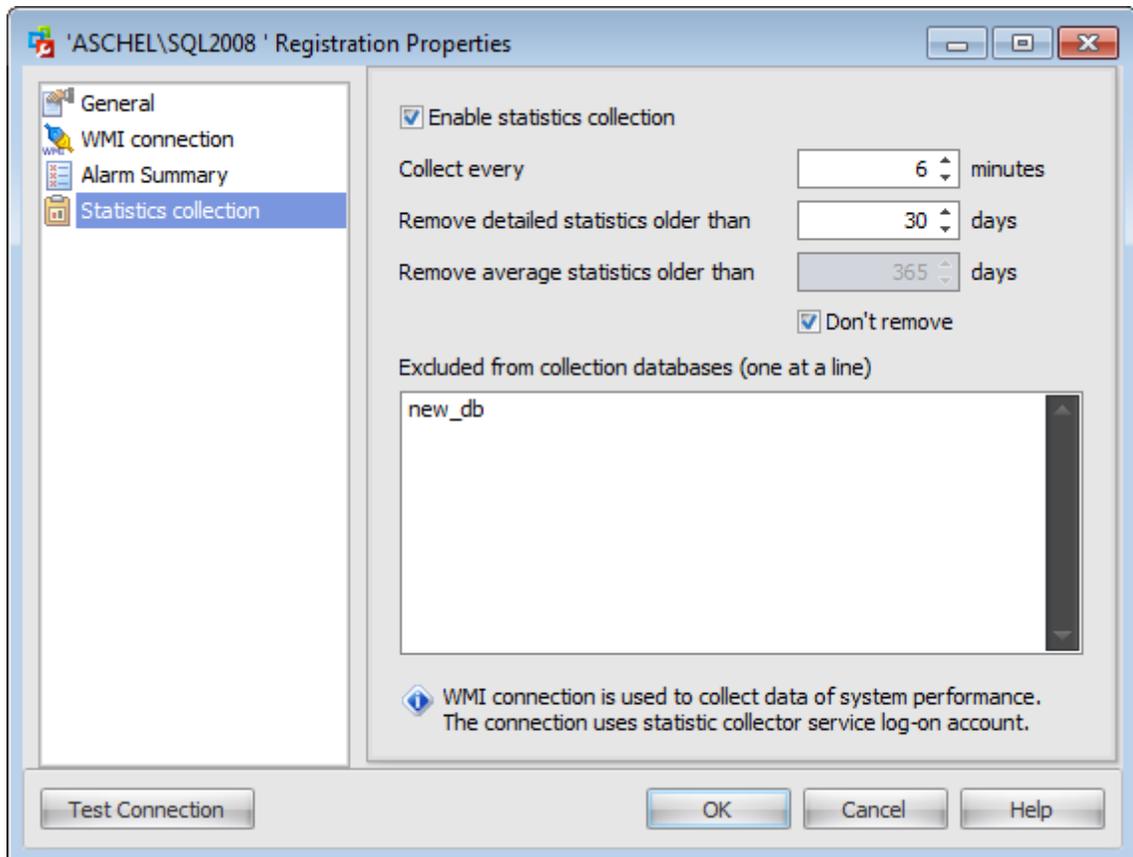
**Min Buffer Hit Ratio**

Type in or use the spinner control to define the minimum percentage of the buffer cache hit ratio.

### 3.1.4 Setting statistics collection parameters

The **Statistics Collection** section of the **Register Server** dialog allows you to set properties of the statistics diagram display. These statistics can be viewed in the [Collected Statistics](#) section of the [Server Explorer](#).

**Note:** Database statistics is collected only if the [Statistics Collector](#) service has been installed and is running during the current session.



#### **Enable Statistics collection**

Check this option to allow statistics collection for the configured server and define its periodicity in **Collect every n minutes** field.

#### **Remove detailed statistics older than n days**

The detailed statistics for the registrable server will be removed on the expiry of the defined number of days.

#### **Remove average statistics older than n days**

Define the number of days after which the average statistics for the registrable server will be removed or check the  **Don't remove** option to store statistics for this server.

#### **Excluded from collection databases**

In this window you can define databases of the registrable server for which the statistics collection must not be performed. The full list of databases can be found [here](#).

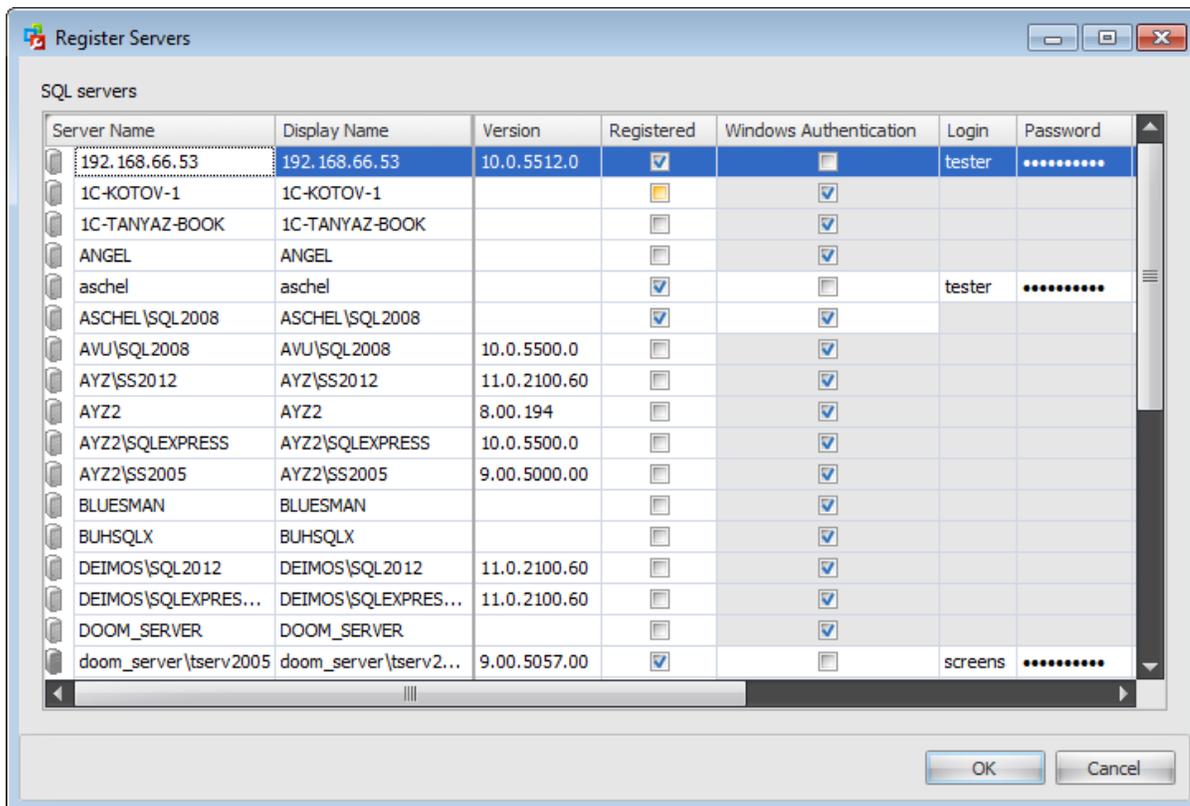


## 3.2 Register servers

**Register servers** dialog allows you to register/unregister several servers at once.

To open this dialog:

- select the **Server | Register Servers...** [main menu](#) item
- or
- right-click within the [Server Explorer](#) and select the **Register Servers...** context menu item.



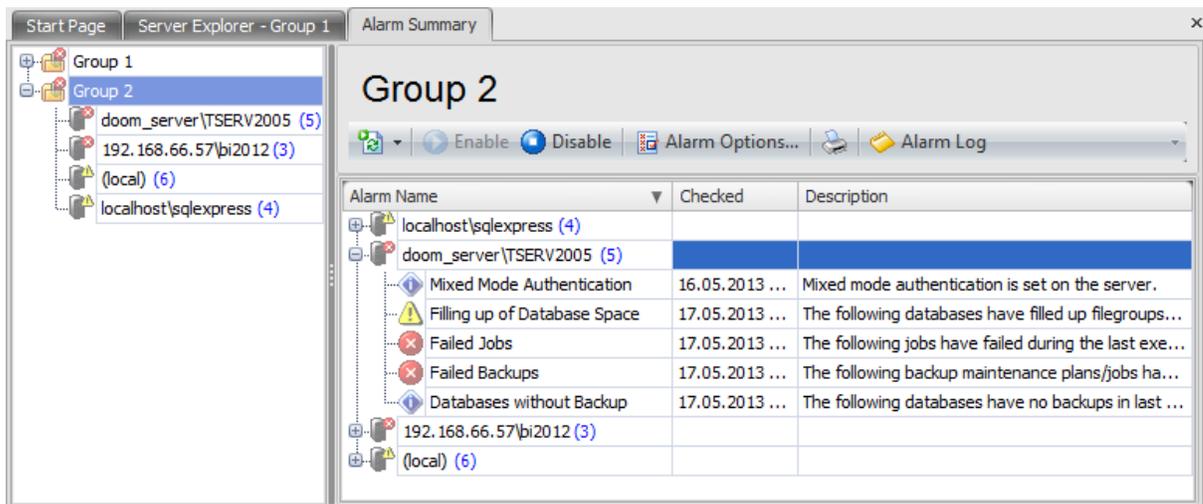
This grid contains the list of SQL servers in the local network that have been found by means of SQL Server client. All registered servers are displayed here as well. Already registered servers are checked in the *Registered* column. To register a new server check the corresponding checkbox in the *Registered* column. To *Unregister* a server you need to uncheck this checkbox.

Also, you can specify the following options for the server being registered: *Display name*, *Server Group*, *Description*, *Alarm Summary enabled*, *Statistics Collection enabled*. You can provide authentication to be used for the connection: select *Windows authentication* or specify *Login/Password* for SQL Server authentication.

To learn more about server registration options, see the [Register server/Registration properties](#) topic.

### 3.3 Alarm summary

This page displays detailed information about all occurred alarms. Alarm summary can be evoked via the **Alarm Summary** item of the [Server Explorer](#) or using the **Tools | Alarm Summary** main menu item. In the latter case it will be opened in the separate **Alarm Summary** tab where you can view a grid containing fired alarms for all servers at once.

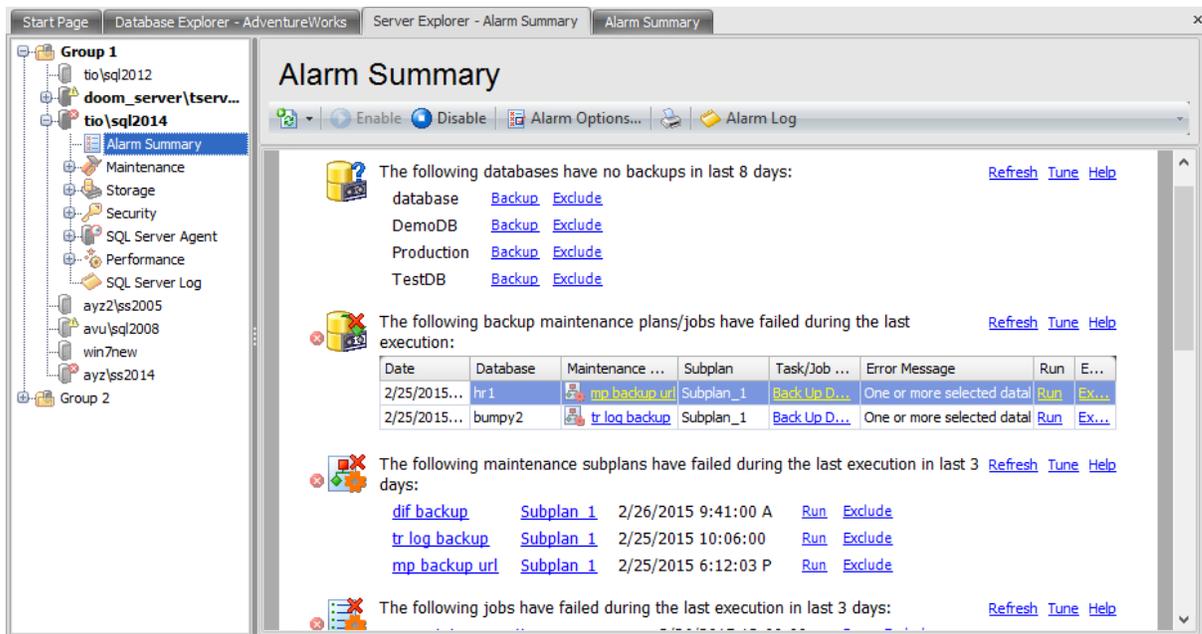


To view a detailed alarm summary for a particular server select it in the Server Explorer.

The **Toolbar** provides quick access to tools implemented for working with alarms.

 mark to refresh the list of alarms. All alarms will be placed in a queue and then refreshed. By pressing the arrow-down button you can choose the alarms you want to refresh.

-  enable alarm summary
-  disable alarm summary
-  view and set [alarm options](#)
-  [print](#) the list of alarms
-  view [alarm log](#)



Every alarm has the following set of buttons:

- Refresh** - refresh the particular alarm;
- Tune** - [set parameters](#) for the particular alarm;
- Help** - view help documentation.

Some alarms contain the list of objects with detected mistakes. For every object there is an action that allows you to eliminate the occurred problem.

Here are some alarms with specific solutions:



### Databases without Backup

It notifies you about databases that had no backups within the defined [check period](#). To solve this problem use the **Backup** button which launches the [Backup Database Wizard](#).



### Failed jobs

This alarm appears when a job has failed during its last execution within the [check period](#). Click the **Run** button to repeat the job execution and see whether the occurred problem has been solved.

**Note:** To execute a job you need to run [SQL Server Agent](#) service on the server.



### Failed backups

This alarm shows a table of backup jobs and maintenance plans that have failed during the last execution. You can launch the [Job Editor](#) or [Maintenance Plan Editor](#) by clicking its name in the *Job / Maintenance Plan* column or launch the [Step Editor](#) / [Task Editor](#) by clicking its name in the *Task/Job Step* column.

Also you can run the necessary job immediately by clicking the **Run** button in the *Run* column.

**Note:** To execute a job you need to run [SQL Server Agent](#) service on the server.



### Failed Maintenance Plans

This alarm lists the [subplans](#) that have failed up during the last execution within the [check period](#). By clicking the maintenance plan name you can open the [Maintenance Plan Editor](#), by clicking the Subplan name you can open the [Managing tasks](#) section of the [Maintenance Plan Editor](#). You can execute the necessary maintenance plan by clicking the **Run** button as well.



#### Filling up of Database Space

This alarm shows databases that have filled up [filegroups](#). It appears when the the filegroup filling exceeds the defined [percentage value](#). By clicking the **Edit** button in the corresponding cell you can edit the filegroup size in the opened [Database Editor](#).



#### Guest Enabled Databases

This alarm shows a list of databases which enable guest user connection. You can disable this connection by pressing the **Disable guest** button.



#### Orphaned Logins

This alarm shows a list of orphaned logins. You can edit a login in the [Login Editor](#) or drop it.



#### Orphaned Users

This alarm appears when a database user becomes orphaned. You can edit a user in the [User Editor](#) so as it will not be orphaned (specify the SQL Server [login](#) for which the database user is being created) or drop the selected user.



#### Mixed Mode Authentication

This alarm informs you that the server authentication is set to a mixed mode, i.e. Windows and SQL Server authentication mode. To configure this setting click the **Modify** button and change the authentication mode in the opened [Server properties](#) dialog.



#### Ole Automation Procedures Enabled

This alarm informs you that Ole Automation Procedures are enabled on the server. Click the **Disable** button to open the [Server properties](#) dialog and disable this option.



#### xp\_cmdshell Enabled

This alarm appears when `xp_cmdshell` procedure is enabled on the server. Click the **Disable** button to open the [Server properties](#) dialog and disable this option.



#### Low Buffer Cache Hit Ratio

This alarm informs you that buffer cache hit ration is low (lower than [Min Buffer Hit Ratio](#)). Tune the [Server Properties](#) to eliminate the problem.



#### Errors in Server Error Log

This alarm appears when the error specified in the [Search for](#) field is found in the SQL Server Error Log. The full text of the error is shown in the **Message** column.



#### Databases in Restricted User Mode

This alarm raises when there are databases on the server that have the user access set to the restricted user mode. By clicking the database name you can open the **Database Editor** and change the user mode option on the [Options](#) tab.



#### Databases in Non-online State

This alarm informs you that there are databases that are not in *Online* state. It shows databases that are in *Offline* or *Emergency* state. By clicking the database name you can open the **Database Editor** and change the database state option on the [Options](#) tab, if possible.

Some alarms have the **Exclude** button which allows you to delete an object from alarm summary in case you do not want to be alarmed about its possible malfunction.

**Note:** The following alarms require [WMI Connection](#): SQL Server Agent Stopped, CPU Overload, Lack of Memory, Busy Disks, Large Disk Queue.  
The full list of alarms can be found [here](#).

---

**See also:**

[Register server/Registration properties](#)

[SQL Administrator Repository](#)

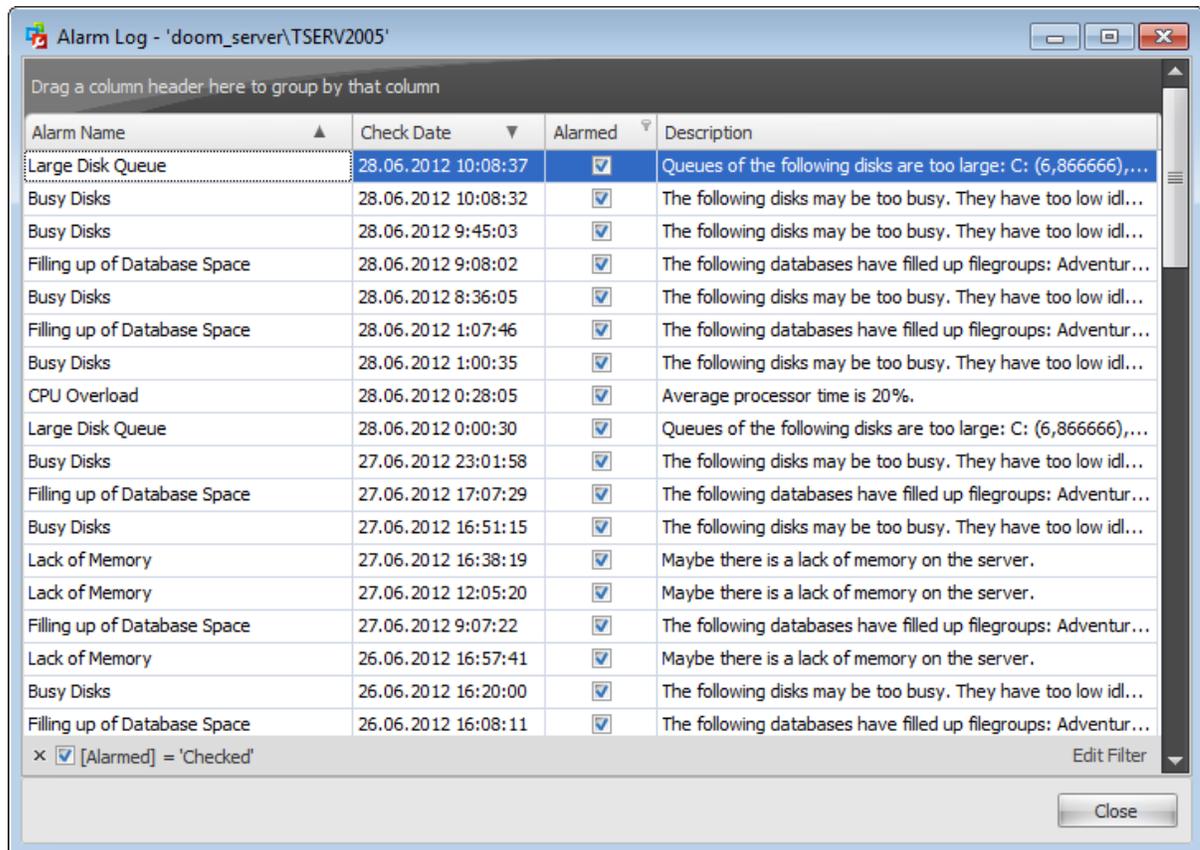
[WMI Connection](#)

[Statistics Collector](#)

[Server properties](#)

### 3.3.1 Alarm Log

Alarm log displays the history of some alarms occurred on the server.



The screenshot shows a window titled "Alarm Log - 'doom\_server\TSERV2005'". The window contains a table with the following columns: Alarm Name, Check Date, Alarmed, and Description. The table lists various alarm events, including "Large Disk Queue", "Busy Disks", "Filling up of Database Space", "CPU Overload", and "Lack of Memory". The "Alarmed" column contains checkboxes, most of which are checked. The "Description" column provides details about the alarm, such as "Queues of the following disks are too large: C: (6,866666),..." and "Average processor time is 20%".

Alarm Name	Check Date	Alarmed	Description
Large Disk Queue	28.06.2012 10:08:37	<input checked="" type="checkbox"/>	Queues of the following disks are too large: C: (6,866666),...
Busy Disks	28.06.2012 10:08:32	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Busy Disks	28.06.2012 9:45:03	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Filling up of Database Space	28.06.2012 9:08:02	<input checked="" type="checkbox"/>	The following databases have filled up filegroups: Adventur...
Busy Disks	28.06.2012 8:36:05	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Filling up of Database Space	28.06.2012 1:07:46	<input checked="" type="checkbox"/>	The following databases have filled up filegroups: Adventur...
Busy Disks	28.06.2012 1:00:35	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
CPU Overload	28.06.2012 0:28:05	<input checked="" type="checkbox"/>	Average processor time is 20%.
Large Disk Queue	28.06.2012 0:00:30	<input checked="" type="checkbox"/>	Queues of the following disks are too large: C: (6,866666),...
Busy Disks	27.06.2012 23:01:58	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Filling up of Database Space	27.06.2012 17:07:29	<input checked="" type="checkbox"/>	The following databases have filled up filegroups: Adventur...
Busy Disks	27.06.2012 16:51:15	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Lack of Memory	27.06.2012 16:38:19	<input checked="" type="checkbox"/>	Maybe there is a lack of memory on the server.
Lack of Memory	27.06.2012 12:05:20	<input checked="" type="checkbox"/>	Maybe there is a lack of memory on the server.
Filling up of Database Space	27.06.2012 9:07:22	<input checked="" type="checkbox"/>	The following databases have filled up filegroups: Adventur...
Lack of Memory	26.06.2012 16:57:41	<input checked="" type="checkbox"/>	Maybe there is a lack of memory on the server.
Busy Disks	26.06.2012 16:20:00	<input checked="" type="checkbox"/>	The following disks may be too busy. They have too low idl...
Filling up of Database Space	26.06.2012 16:08:11	<input checked="" type="checkbox"/>	The following databases have filled up filegroups: Adventur...

At the bottom of the window, there is a filter section with a checkbox for "[Alarmed] = 'Checked'" and an "Edit Filter" button. A "Close" button is located at the bottom right of the window.

The grid consists of the following fields:

*Alarm Name*;

*Check Date* (the date when the server was checked on potential problems);

*Alarmed* (informs whether the alarm went off);

*Description* (the message informing about a potential problem).

**Note:** The following alarms are logged: Filling up of Database Space, No Server Connection, Low Buffer Cache Hit Ratio, CPU Overload, Lack of Memory, Busy Disks, Large Disk Queue.

The full list of alarms can be found [here](#).

### 3.4 SQL Administrator Repository

SQL Administrator uses a repository database on the server to store the service information.

The repository stores:

1. The last [alarm](#) values.
2. [Alarm log](#) (for alarms that support logs).
3. [Statistics collection](#) parameters for all registered servers.
4. Current (detailed) statistics of the server productivity.
5. Average statistics of the server productivity.

Use the **Server | SQL Administrator Repository...** [main menu item](#) to open the dialog window to set repository parameters.

Specify the following settings for the repository:

#### SQL server name

Type in or use the drop-down list to select the host/instance name where the repository database should reside.

**Note:** If Microsoft® SQL Server™ is installed as a named instance the server name must be specified in the following format: computer\_name\sqlserver\_instance\_name or ip\_address\sqlserver\_instance\_name (e.g. "MYCOMPUTER\SQLEXPRESS", "127.0.0.1\SQLEXPRESS").

#### Authentication

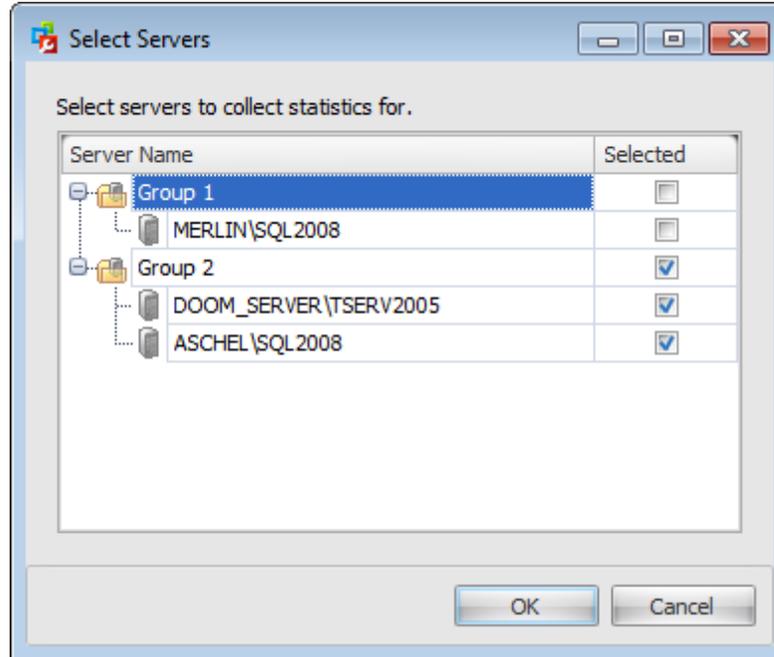
Specify the type of SQL Server authentication to be used for the connection:  *SQL Server authentication* or  *Windows authentication*. It is strongly recommended to avoid using SQL Server authentication with "sa" as the login.

If SQL Server has been selected as the authentication type, you should also provide authorization settings: **Login** and **Password**.

**Database name (new or existent)**

Type in or use the drop-down list to select the database.

If you have specified a new repository database then the following window appears.



It allows you to specify servers which system information will be stored in the repository database and therefore which statistics will be collected via the [Statistics Collector](#). For your convenience the *Select All* and *Unselect All* functions are implemented in the **context menu** of the servers list.

To apply new [statistics collector service](#) parameters you need to reinstall the service. In what follows you can define whether to collect or not statistics for the specific server by checking or unchecking the

**Enable Statistics collection** option in the [Setting statistics collection parameters](#) section of the [Server Registration Properties](#)

It is strongly recommended not to store a repository on a production server to avoid problems with a productivity.

Along with a repository creation the following database maintenance tasks are also created:

- avg\_stats - averages current statistics to reduce the place occupied with statistics of server productivity;
- cleanup\_stats - drops current statistics in compliance with storage time of the current and average statistics;
- cleanup\_alarm\_log - drops old alarm units in compliance with storage time of the alarm summary.

These tasks have schedules to be run automatically. You have to check the execution state of these tasks if they are not executed due to some reasons.

The execution state of these tasks must be checked because otherwise, if these tasks

are not executed due to some reasons, repository database may greatly increase in size, and some problems may occur in SQL Administrator when accessing to repository data.

---

**See also:**

[Register server/Registration properties](#)

[Alarm summary](#)

[WMI Connection](#)

[Statistics Collector](#)

[Server properties](#)

## 3.5 WMI Connection

Besides an SQL connection SQL Administrator uses WMI connection to servers as well. This connection is used to:

- get a system information about an SQL server, e.g. installed operating system or CPU type;
- start and stop SQL Server services;
- view Windows event logs;
- get system performance counter values to display in the product, in some [alarms](#) and for [statistics collection](#).

The authentication type used is Windows authentication. When a user connects through a Windows user account, SQL Server validates the account name and password using the Windows principal token in the operating system. Log on account grants are used to connect to the [Statistics collector services](#). Indicated above functions will be inaccessible if the WMI connection cannot be set up.

Remote connections in WMI are affected by the Windows Firewall and DCOM settings. In Windows Vista and later operating systems, User Account Control (UAC) may also require changes to some settings. Users access to the 'root\CIMV2' WMI namespace (the program uses only this namespace) should be also allowed. By default, the permission is enabled only for administrators. It is possible to enable remote access to specific WMI namespaces for a nonadministrator user. Please see Connecting to WMI on a Remote Computer topic in the MSDN library ([http://msdn.microsoft.com/en-us/library/aa389290\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa389290(v=vs.85).aspx)).

To set up remote WMI connection you should do the following:

- Create an exception in Windows firewall, to enable information sending via WMI:
  - *Windows XP and Windows Server 2003.* In Group Policy editor (to launch it click [Start] [Run] and type 'gpedit.msc' in the data input line) in Local Computer Policy\Computer Configuration\Administrative Templates\Network\Network Connections\Windows Firewall\Domain Profile or Standard Profile key enable 'Windows Firewall: Allow remote administration exception' parameter.
  - *Windows Vista and later.* In the Control Panel | Security | Windows Firewall click Change Settings and on Exceptions tab in the Exceptions window, select the check box for WMI.

For nonadministrator remote user do the following as well:

- Grant DCOM remote launch and activation permissions for a user or group: in DCOMCnfg.exe right-click on Component Services\Computers\My Computer, select Properties and on COM Security tab under Launch and Activation Permissions, click Edit Limits, then add a user or group and allow Remote Launch and Remote Activation.
- Grant DCOM remote access permissions: in the same dialog on the same tab under Access Permissions, click Edit Limits, select ANONYMOUS LOGON in the Group or user names box, then allow Remote Access.
- Allow users access to the 'root\CIMV2' WMI namespace: In Control Panel | Administrative Tools | Computer Management right-click Services and Applications\WMI Control and select Properties, then in Security tab select Root\CIMV2 and click Security button, and allow Remote Enable for a user or group.

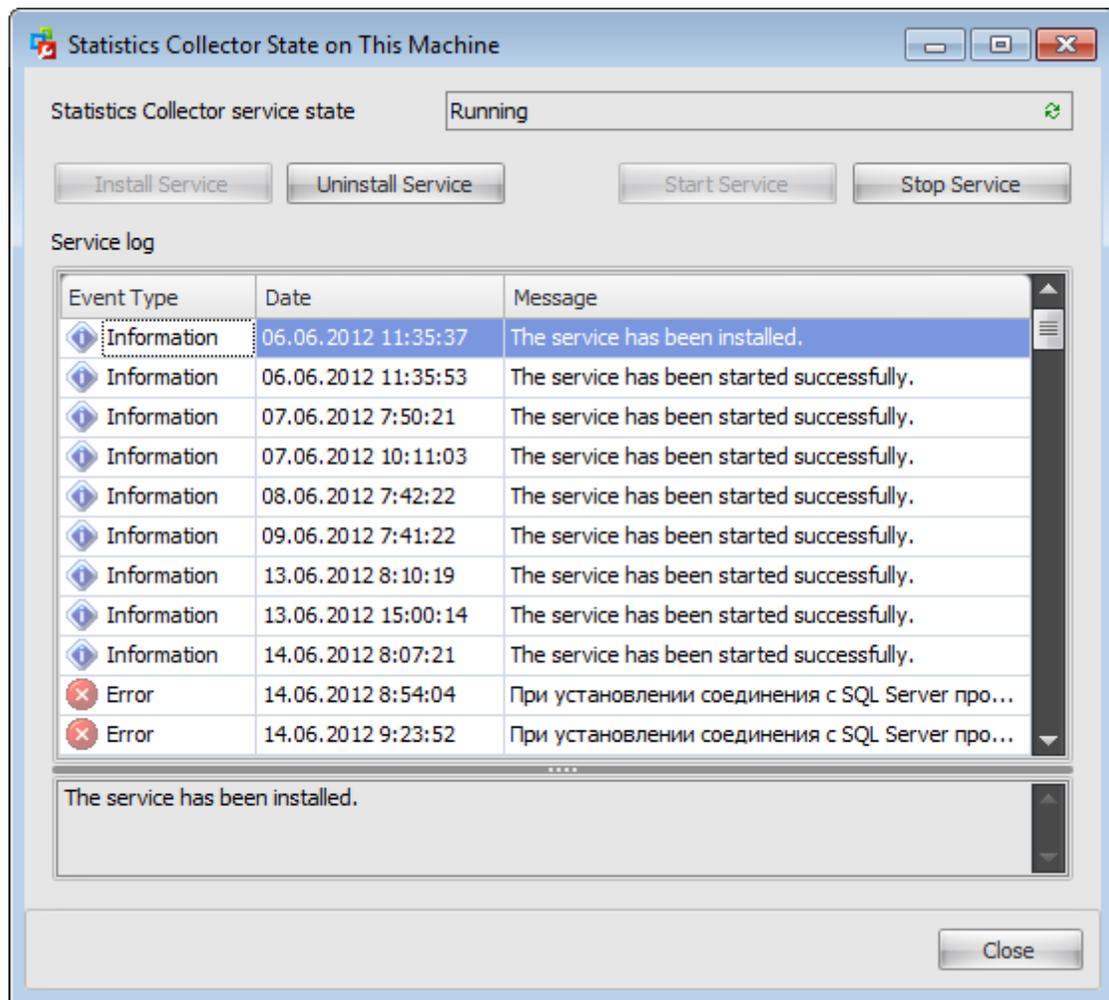
**See also:**[Register server/Registration properties](#)[Alarm summary](#)[SQL Administrator Repository](#)[Statistics Collector](#)[Server properties](#)

### 3.6 Statistics Collector

**Statistics Collector** service is used to collect statistics of the system and SQL Server productivity.

To open the Statistics Collector State dialog choose **Tools | Statistics Collector State...** main menu item or double-click the Statistics Collector State label situated in the right-bottom corner of the program window.

This dialog allows you to **Install/Uninstall** or to **Start/Stop** the service.



#### Statistics Collector service state

This field displays the information about the current statistics collector state. To refresh the status click the  **Refresh** button.

To install the Statistics Collector service click the **Install Service** button. In the window appeared specify the service options.

The screenshot shows a dialog box titled "Statistics Collector Service Options". It is divided into two main sections. The top section, "SQL Administrator repository connection properties", contains fields for "Repository server" (doom\_server\TSERV2005) and "Repository database" (SQLAdministratorRepository). Below these are radio buttons for "Windows authentication" (unselected) and "SQL server authentication" (selected). There are also fields for "Login" (screens) and "Password" (masked with asterisks). The bottom section, "Service login", has radio buttons for "Local system" (selected) and "This account" (unselected). Below "This account" are fields for "User name", "Password", and "Confirm password". At the bottom of the dialog, there is a warning icon and text: "When using windows authentication connection to the repository server or to the collected servers, this account should have permissions to connect to server." The dialog has "OK" and "Cancel" buttons at the bottom right.

### SQL Administrator repository connection properties

**Repository server** and **Repository database** fields display the information set when configuring [SQL Administrator Repository](#).

Specify the type of Microsoft® SQL Server™ authentication to be used for the repository connection:  *Windows authentication* or  *SQL Server authentication*.

#### **Windows Authentication**

SQL Administrator identifies user by his domain name and password which are defined automatically according to his Windows domain user account.

**Note:** When using windows authentication connection to the repository server or to the collected servers, this account should have permissions to connect to server.

#### **SQL Server authentication**

SQL Administrator identifies user by his server login and password. Login and password management is accomplished by Login Manager.

If  *SQL Server authentication* is selected as the *authentication type*, you should provide *authorization* settings: **Login** and **Password**. Analyzing them a system

determines information and therefore object access permissions.

### Service login

#### Local system

The Local system account is a predefined local account used by the service control manager. This type of the service login can be selected only if the type of connection to the repository server was set to *SQL Server authentication*.

#### This account

If you select this option then the service will be run under the specified domain user account.

### User name

Type in the name of the account under which the service should run. Use an account name in the form *DomainName\UserName*. The service process will be logged on as this user.

### Password

Enter the password for the user name.

Availability:

**Full** version                      **Yes**

**Free** version                      **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

---

### See also:

[Register server/Registration properties](#)

[Alarm summary](#)

[SQL Administrator Repository](#)

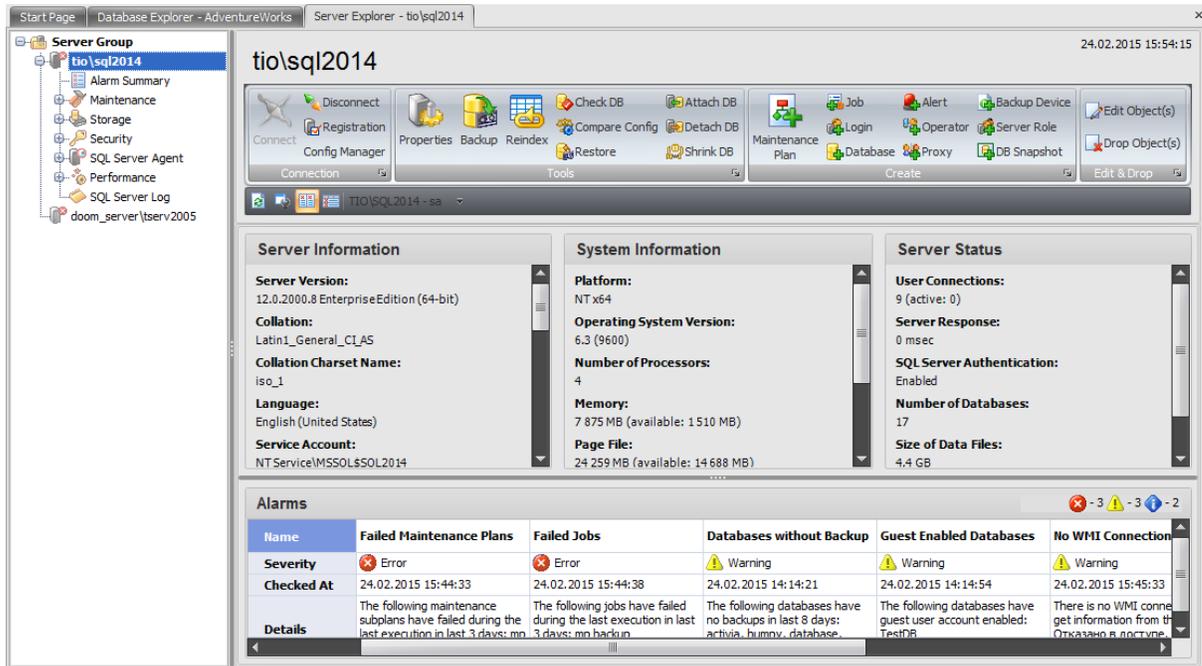
[WMI Connection](#)

[Server properties](#)

## 3.7 Server properties

The **Server Properties** dialog allows you to view/edit a number of server options which can be changed to optimize Microsoft® SQL Server performance.

To open the dialog click  **Properties** in the **Tools** ribbon. It is also available within the context menu of the **Server Security Options** area in the [Security](#) section.



All server properties are grouped in the following clusters:

- [Memory](#)
- [Processor](#)
- [Parallelism](#)
- [Security](#)
- [Surface area configuration](#)
- [Connections](#)
- [Database options](#)
- [Full-text search](#)
- [Miscellaneous](#)
- [Network](#)

On the dialog toolbar you can select what values should be displayed:

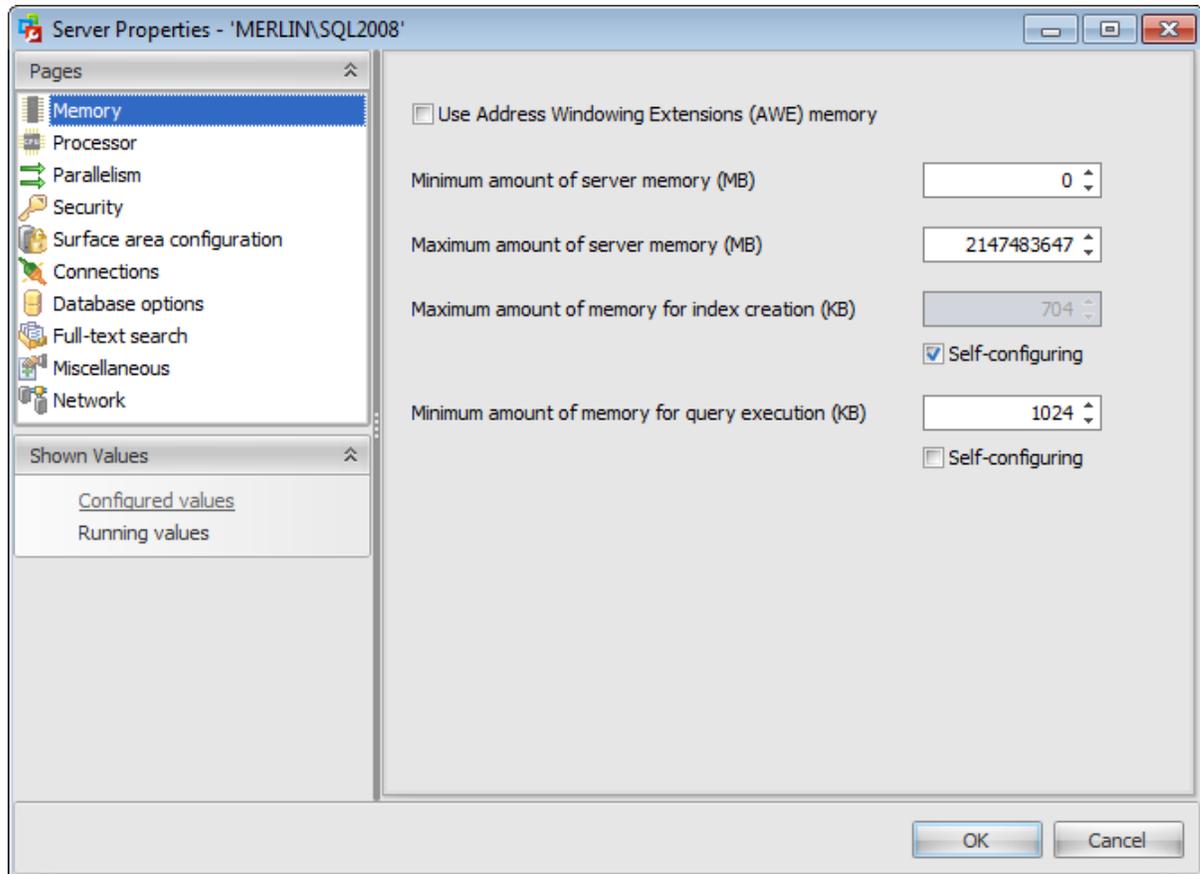
### Shown values

Configured values determines that the configured values for options are displayed on this page. If you change these values, select **Running Values** to see whether the changes have taken effect. If they have not, the instance of SQL Server must be restarted first.

**See also:**[Register server/Registration properties](#)[Alarm summary](#)[SQL Administrator Repository](#)[WMI Connection](#)[Statistics Collector](#)

### 3.7.1 Memory

The **Memory** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to memory usage.



#### Use Address Windowing Extensions (AWE) memory

Under Windows Server 2003, SQL Server 2005 (and higher) can use Address Windowing Extensions (AWE) memory to further assist in balancing its own memory requirements with those of the operating system. This balancing between SQL Server and the operating system is subject to the constraints of the *Minimum server memory* and *Maximum server memory* options.

Use the two server memory options, **Minimum amount of server memory** and **Maximum amount of server memory**, to reconfigure the amount of memory (in megabytes) in the buffer pool used by the instance of SQL Server.

By default, SQL Server can change its memory requirements dynamically on the basis of available system resources. The default setting for *Minimum server memory* is 0, and the default setting for *Maximum server memory* is 2147483647. The minimum amount of memory that can be specified for **Maximum amount of server memory** is 16 MB.

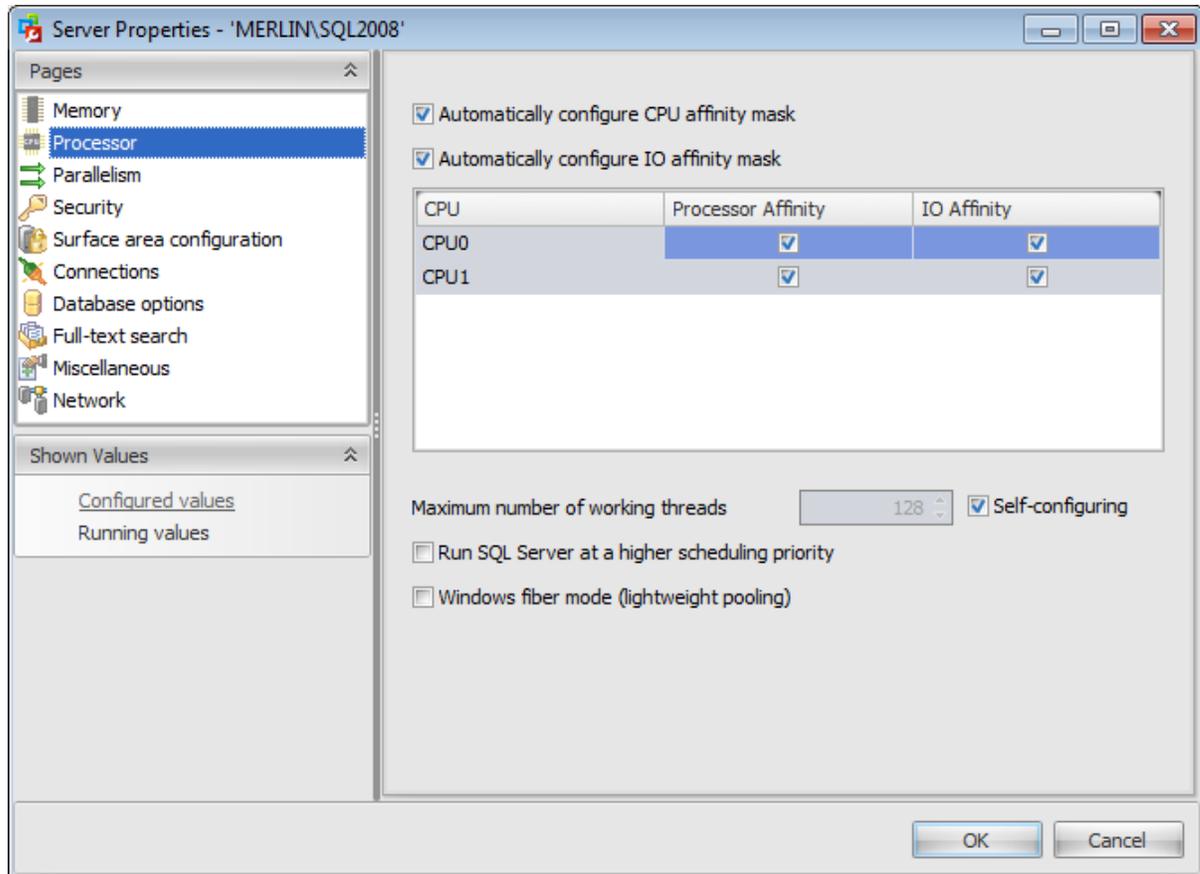
In SQL Server 2000/2005/2008 the **Index creation memory** option controls the maximum initial amount of memory allocated for index creation. If additional amount of memory is required for index creation operation, and it is available, the server will use additional memory thus exceeding the setting of this option. If additional memory is not available,

the memory that has been already allocated will be used for index creation.

Use the **Minimum amount of memory for query execution** option to specify the minimum amount of memory (in kilobytes) that will be allocated for execution of a query. For example, if **Minimum memory per query** is set to *2,048 KB*, the query is guaranteed to get at least that much total memory. You can set **Minimum memory per query** to any value from *512* up to *2,147,483,647 KB (appr. 2 GB)*. The default value is *1,024 KB*. To enable editing this field uncheck  **Self-configuring** option

### 3.7.2 Processor

The **Processor** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to processors usage.



To carry out multitasking, it is possible to distribute process threads among different processors. Although efficient from an operating system point of view, this activity can reduce SQL Server performance under heavy system loads, as each processor cache is repeatedly reloaded with data. Assigning processors to specific threads can improve performance under these conditions by eliminating processor reloads and reducing thread migration across processors (thereby reducing context switching). Such an association between a thread and a processor is called *processor affinity*.

You can enable any or both two affinity mask options supported by SQL Server 2005 (and higher):  **Automatically configure CPU affinity mask** and  **Automatically configure IO affinity mask**.

Use the **Maximum number of working threads** option to configure the number of working threads available to Microsoft® SQL Server processes. SQL Server uses the native thread services of the Microsoft® Windows 2000 and Windows Server 2003 operating systems so that one or more threads support each network that SQL Server supports simultaneously, another thread handles database checkpoints, and a pool of threads handles all users.

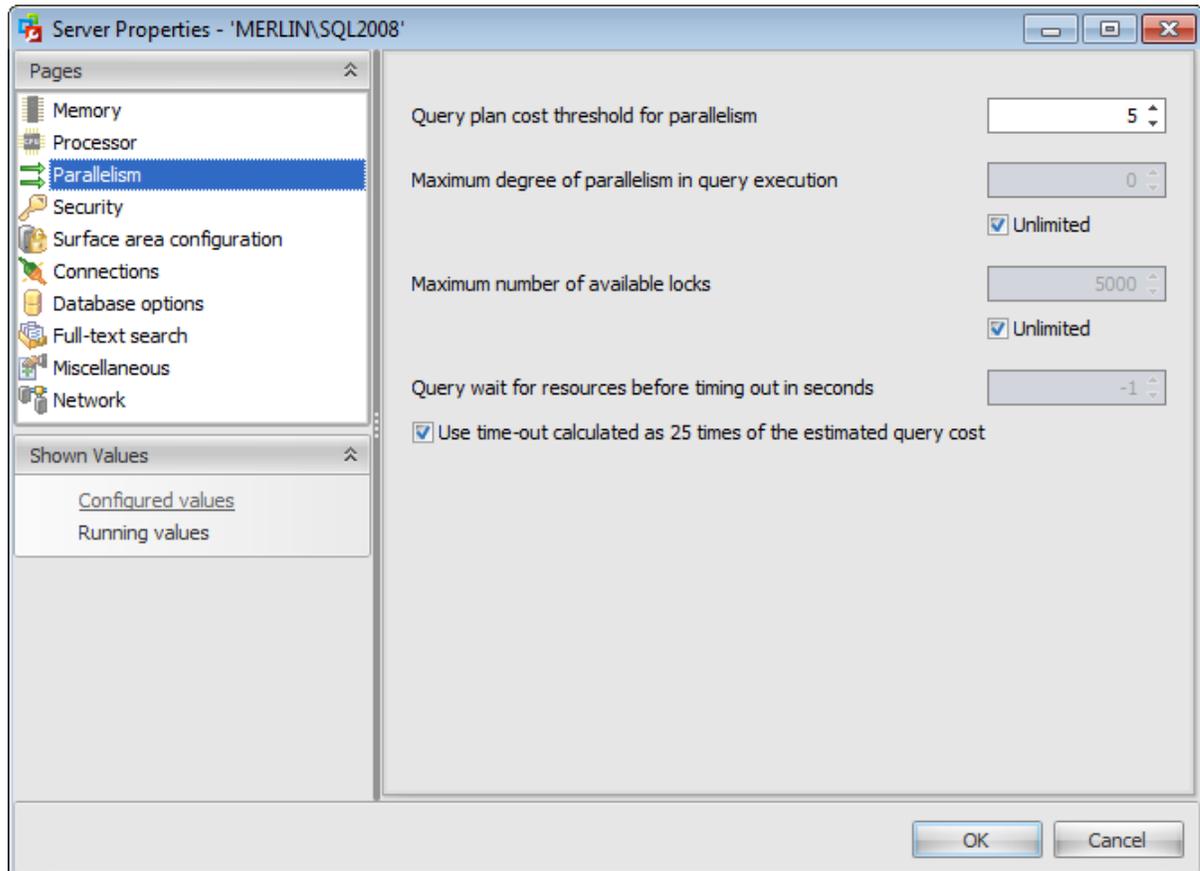
To enable editing this field uncheck  **Self-configuring** option.

Use the  **Run SQL Server at a higher scheduling priority** option to specify whether Microsoft SQL Server should run at a higher Microsoft Windows 2000, Windows Server 2003, Windows 2008, or Windows 2008 R2 scheduling priority than other processes on the same computer. If you set this option to 1, SQL Server runs at a priority base of 13 in the Windows 2000, Windows Server 2003, Windows 2008, or Windows 2008 R2 scheduler. The default is 0, which is a priority base of 7.

Select the  **Windows fiber mode (lightweight pooling)** option to provide a means of reducing the system overhead associated with the excessive context switching sometimes seen in symmetric multiprocessing (SMP) environments. When excessive context switching is present, lightweight pooling can provide better throughput by performing the context switching inline, thus helping to reduce user/kernel ring transitions.

### 3.7.3 Parallelism

The **Parallelism** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to parallel processor usage.



Use the **Query plan threshold for parallelism** option to specify the threshold at which SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in **Query plan threshold for parallelism**. The cost refers to an estimated elapsed time in seconds required to run the serial plan on a specific hardware configuration. Only set **Query plan threshold for parallelism** on symmetric multiprocessors.

When SQL Server 2005 (and higher) runs on a computer with more than one CPU, it detects the best degree of parallelism for each instance of a parallel query execution. You can use the **Maximum degree of parallelism in query execution** option to limit the number of processors to use in parallel plan execution. The default value of 0 indicates that all available processors are used. Set **Maximum degree of parallelism in query execution** to 1 to suppress parallel plan generation. Set the value to a number greater than 1 (up to a maximum of 64) to restrict the maximum number of processors used by a single query execution. If a value greater than the number of available processors is specified, or in case the  **Unlimited** option is selected, the actual number of available processors is used. If the computer has only one processor, the **Maximum degree of**

**parallelism in query execution** value is ignored.

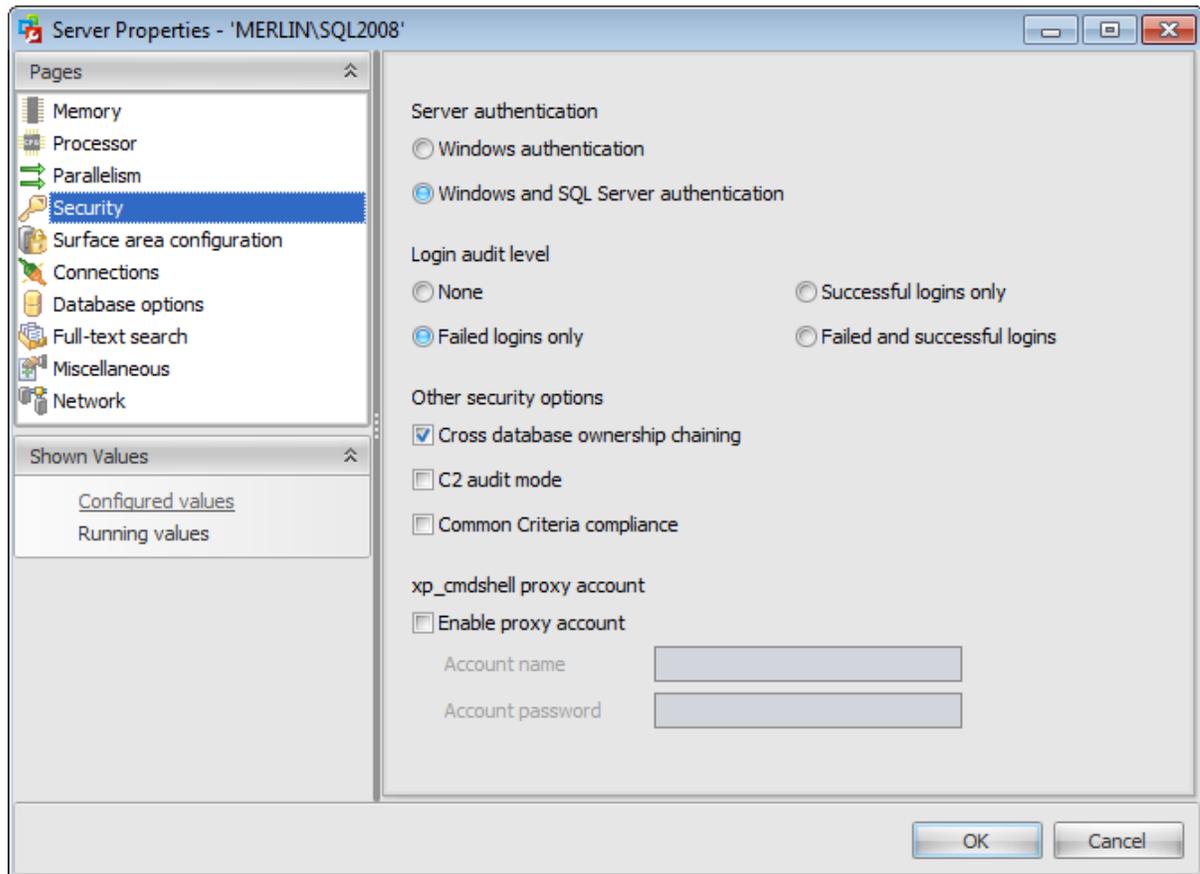
Use the **Maximum number of available locks** option to set the maximum number of available locks, thereby limiting the amount of memory the Database Engine uses for them. The default setting is 0, which allows the Database Engine to allocate and deallocate lock structures dynamically, based on changing system requirements. When the server is started with locks set to 0, the lock manager acquires sufficient memory from the Database Engine for an initial pool of 2,500 lock structures. As the lock pool is exhausted, additional memory is acquired for the pool. Allowing SQL Server to use locks dynamically by checking the  **Unlimited** option is the recommended configuration.

Use the **Query wait for resources before timing out in seconds** option to specify the time in seconds (from 0 through 2147483647) that a query waits for resources before timing out. If the default value of -1 is used, or if -1 is specified, then the time-out is calculated as 25 times of the estimated query cost. The  **Use time-out calculated as 25 times of the estimated query cost** option has the same effect.

In Microsoft SQL Server, memory-intensive queries (such as those involving sorting and hashing) are queued when there is not enough memory available to run the query. The query times out after a set time calculated by SQL Server (25 times the estimated cost of the query) or the time specified by the nonnegative value of the query wait.

### 3.7.4 Security

The **Security** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to server security in groups: *Server Authentication*, *Login audit level*, *Other security options*, *xp\_cmdshell Proxy Account*.



#### Server authentication

##### • Windows Authentication

When a user connects through a Microsoft® Windows user account, SQL Server validates the account name and password using information in the Windows operating system. This is the default authentication mode, and is much more secure than the mixed mode. Windows Authentication utilizes Kerberos security protocol, provides password policy enforcement in terms of complexity validation for strong passwords, provides support for account lockout, and supports password expiration.

##### • Windows and SQL Server Authentication

This option enables the mixed mode and allows users to connect using either Windows Authentication or SQL Server Authentication. Users who connect through a Windows user account can make use of trusted connections that are validated by Windows.

#### Login audit level

- None (does not log authentication attempts)
- Failed logins only (logs failed authentication)

- *Successful logins only* (logs successful authentication)
- *Both failed and successful logins* (logs all authentication attempts regardless of success or failure)

### Other security options

#### **Cross database ownership chaining**

This server option allows you to control cross-database ownership chaining at the database level or to allow cross-database ownership chaining for all databases. If this option is checked then cross-database ownership chaining is on for all databases. Otherwise, cross-database ownership chaining is disabled for all databases.

#### **C2 audit mode**

Selecting this option will configure the server to record both failed and successful attempts to access statements and objects. This information can help you profile system activity and track possible security policy violations.

#### **Common Criteria compliance**

The common criteria compliance option enables the following elements that are required for the *Common Criteria*:

- *Residual Information Protection (RIP)*

RIP requires a memory allocation to be overwritten with a known pattern of bits before memory is reallocated to a new resource. Meeting the RIP standard can contribute to improved security; however, overwriting the memory allocation can slow performance. After the common criteria compliance enabled option is enabled, the overwriting occurs.

- *The ability to view login statistics*

After the common criteria compliance option is enabled, login auditing is enabled. Each time a user successfully logs in to SQL Server, information about the last successful login time, the last unsuccessful login time, and the number of attempts between the last successful and current login times is made available.

- *Column GRANT should not override table DENY*

After the common criteria compliance option is enabled, a table-level DENY takes precedence over a column-level GRANT. When the option is not enabled, a column-level GRANT takes precedence over a table-level DENY.

**Note:** The common criteria compliance enabled option is an advanced option and is not available in every edition of Microsoft SQL Server.

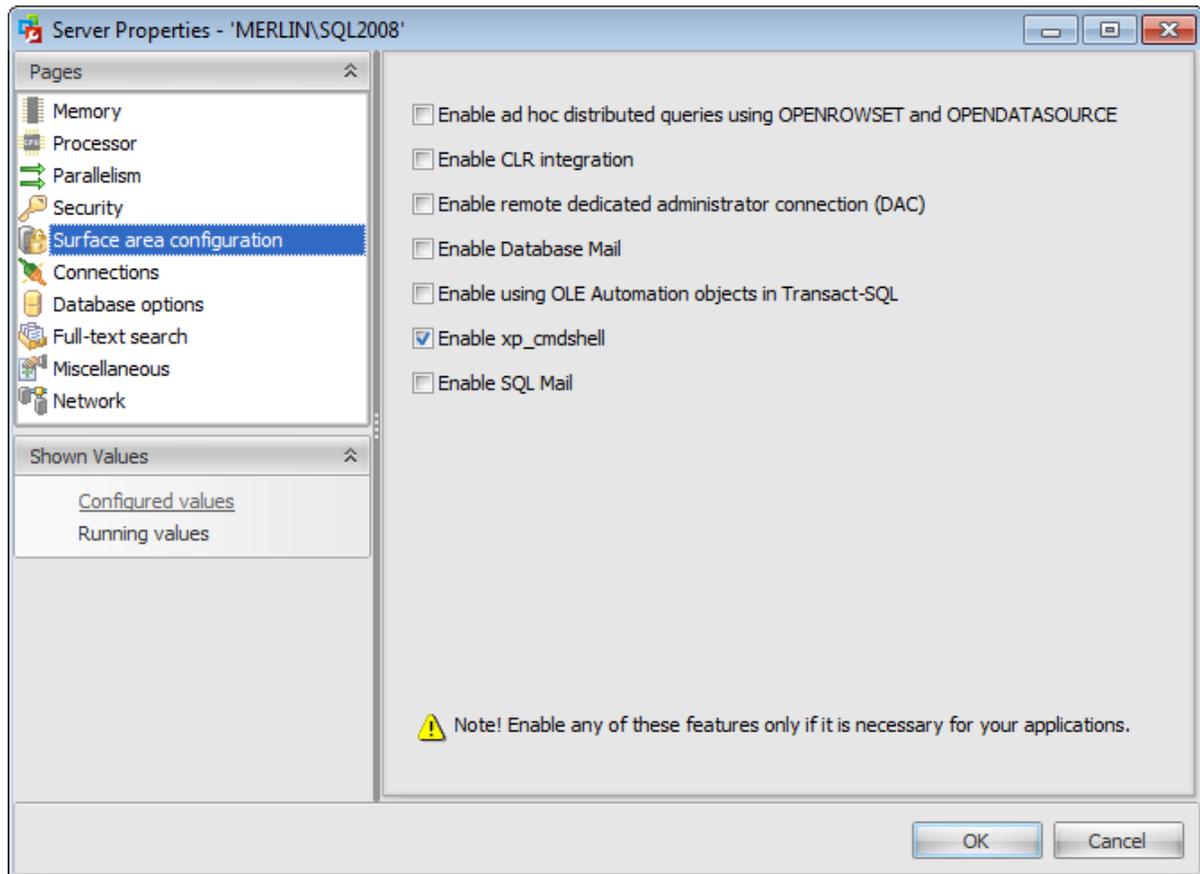
#### **xp\_cmdshell Proxy Account**

This option allows SQL Server users that do not belong to the *sysadmin* fixed server role to execute *xp\_cmdshell* commands and to own SQL Server jobs. To get access to this feature, you must enter an **Account name** and an **Account Password**.

Note that this option is only available when you use SQL Server 2005 (and higher). The login used by the server proxy account should have the least privileges required to perform the intended work. Excessive privileges for the proxy account could be used by a malicious user to compromise your system security.

### 3.7.5 Surface area configuration

The **Surface area configuration** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by managing the software's surface area. Surface area managing is a security measure that involves stopping or disabling unused components.



**Enable ad hoc distributed queries using OPENROWSET and OPENDATASOURCE**

By default, SQL Server does not allow ad hoc distributed queries using OPENROWSET and OPENDATASOURCE. When this option is checked, SQL Server allows ad hoc access. Ad hoc distributed queries use the OPENROWSET and OPENDATASOURCE functions to connect to remote data sources that use OLE DB. OPENROWSET and OPENDATASOURCE should be used only to reference OLE DB data sources that are accessed infrequently. Enabling the use of ad hoc names means that any authenticated login to SQL Server can access the provider. SQL Server administrators should enable this feature for providers that are safe to be accessed by any local login.

**Enable CLR integration**

Use the CLR enabled option to specify whether user assemblies can be run by SQL Server. Check this option in order to allow assembly execution on SQL Server.

**Enable remote dedicated administrator connection (DAC)**

SQL Server provides a special diagnostic connection for administrators when standard connections to the server are not possible. This diagnostic connection allows an

administrator to access SQL Server to execute diagnostic queries and troubleshoot problems even when SQL Server is not responding to standard connection requests.

**Enable Database Mail**

Database Mail is an enterprise solution for sending e-mail messages from the SQL Server Database Engine. Enable this option so that your database applications could send e-mail messages to users. The messages can contain query results, and can also include files from any resource on your network.

**Enable using OLE Automation objects in Transact-SQL**

Transact-SQL includes several system stored procedures that allow *OLE Automation objects* to be referenced in Transact-SQL batches, stored procedures, and triggers. These system stored procedures run as extended stored procedures, and the OLE Automation objects that are executed through the stored procedures run in the address space of an instance of the SQL Server Database Engine in the same way that an extended stored procedure runs.

**Enable xp\_cmdshell**

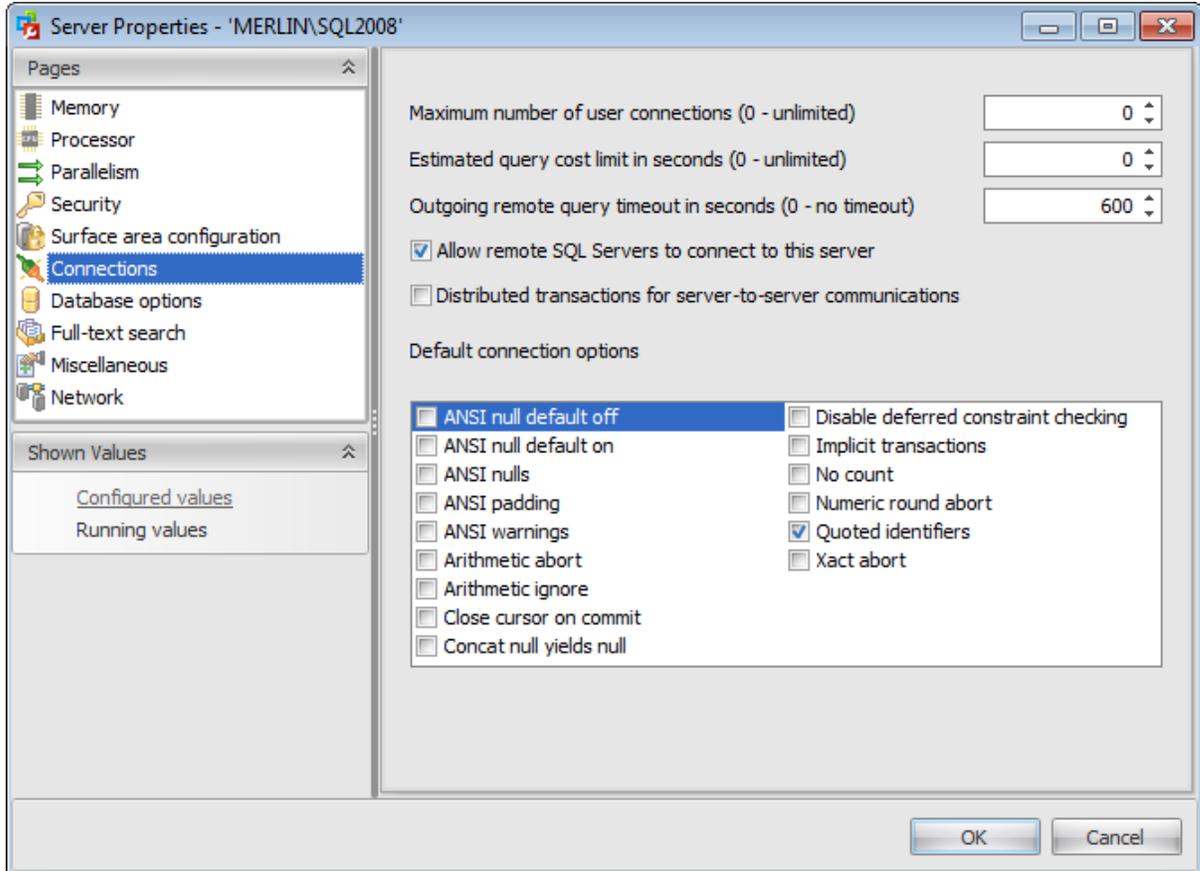
The *xp\_cmdshell* option is a server configuration option that enables system administrators to control whether the *xp\_cmdshell* extended stored procedure can be executed on a system.

**Enable SQL Mail**

SQL Mail is provided in Microsoft SQL Server for backward compatibility. SQL Mail provides a mechanism to send, receive, delete, and process e-mail messages. Messages can be triggered to provide a recipient with a particular result set from a specified SQL Server query. SQL Mail also provides mechanisms to read or delete e-mail messages. Establishing a client connection with a mail server that implements Extended MAPI allows you to perform all of these operations.

### 3.7.6 Connections

The **Connections** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to connection.



#### Maximum number of user connections

If this option is set to a value other than zero, the number of connections that SQL Server will allow is limited. The default value is 0 (unlimited).

#### Estimated query cost limit in seconds

Use this field to specify an upper limit on the time period in which a query can run. Queries that have estimated run times greater than this limit, return an error and are not executed. Query cost refers to the estimated elapsed time, in seconds, required to complete a query on a specific hardware configuration.

#### Outgoing remote query timeout in seconds

Use this field to specify how long, in seconds, a remote operation can take before Microsoft SQL Server times out. The default is 600, which allows a 10-minute wait. This value applies to an outgoing connection initiated by the Database Engine as a remote query. This value has no effect on queries received by the Database Engine.

#### Allow remote SQL Servers to connect to this server

Controls the execution of stored procedures from remote servers running instances of SQL Server. Selecting this check box has the same effect as setting the *sp\_configureremote*

access option to 1. Clearing it prevents execution of stored procedures from a remote server.

**Distributed transactions for server-to-server communication**

Protects the actions of a server-to-server procedure through a Microsoft® Distributed Transaction Coordinator (MS DTC) transaction.

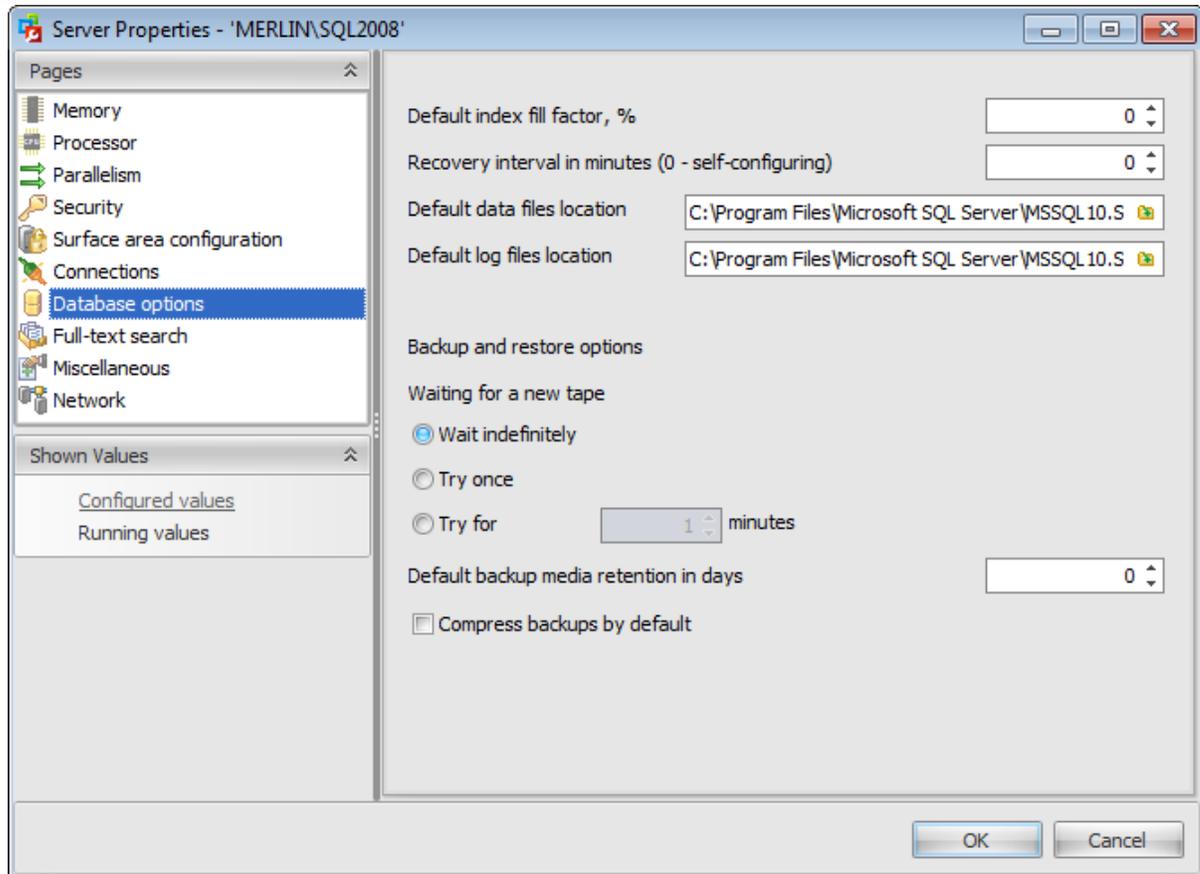
**Default connection options**

Specifies the default connection options as described in the table below:

<b>Configuration option</b>	<b>Description</b>
<input checked="" type="checkbox"/> <i>ANSI NULL default off</i>	Alters the session's behavior not to use ANSI compatibility for nullability. New columns defined without explicit nullability are defined not to allow nulls.
<input checked="" type="checkbox"/> <i>ANSI NULL default on</i>	Alters the session's behavior to use ANSI compatibility for nullability. New columns defined without explicit nullability are defined to allow nulls.
<input checked="" type="checkbox"/> <i>ANSI NULLs</i>	Controls NULL handling when using equality operators.
<input checked="" type="checkbox"/> <i>ANSI padding</i>	Controls padding of fixed-length variables.
<input checked="" type="checkbox"/> <i>ANSI warnings</i>	Controls truncation and NULL in aggregate warnings.
<input checked="" type="checkbox"/> <i>Arithmetic abort</i>	Terminates a query when an overflow or divide-by-zero error occurs during query execution.
<input checked="" type="checkbox"/> <i>Arithmetic ignore</i>	Returns NULL when an overflow or divide-by-zero error occurs during a query.
<input checked="" type="checkbox"/> <i>Close cursor on COMMIT</i>	Controls behavior of cursors after a commit operation has been performed.
<input checked="" type="checkbox"/> <i>Concat NULL yields NULL</i>	Returns NULL when concatenating a NULL value with a string.
<input checked="" type="checkbox"/> <i>Disable deferred constraint checking</i>	Controls interim or deferred constraint checking.
<input checked="" type="checkbox"/> <i>Implicit transactions</i>	Controls whether a transaction is started implicitly when a statement is run.
<input checked="" type="checkbox"/> <i>No count</i>	Turns off the message returned at the end of each statement that states how many rows were affected.
<input checked="" type="checkbox"/> <i>Numeric round abort</i>	Generates an error when a loss of precision occurs in an expression.
<input checked="" type="checkbox"/> <i>Quoted identifier</i>	Differentiates between single and double quotation marks when evaluating an expression.
<input checked="" type="checkbox"/> <i>Xact abort</i>	Rolls back a transaction if a Transact-SQL statement raises a run-time error.

### 3.7.7 Database options

The **Database options** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options pertaining to database.



#### Default index fill factor in %

This value specifies how full SQL Server should make each page when it creates a new index using existing data. The fill factor affects performance because SQL Server takes time to split pages when they fill up.

The default value is 0; valid values range from 0 to 100. A fill factor of 0 or 100 creates clustered indexes with full data pages and nonclustered indexes with full leaf pages, but it leaves some space within the upper level of the index tree. Fill factor values 0 and 100 are identical in all respects.

With low fill factor values specified, SQL Server creates indexes with pages that are not full. Each index takes more storage space, but there is more room for subsequent insertions without requiring page splits.

#### Recovery interval in minutes

Sets the maximum period of time to recover a database. The default is 0, it stands for automatic configuration by SQL Server. In practice, this means a recovery time of less than one minute and a checkpoint approximately every one minute for active databases.

#### Default data files location

Specify the default location for data files: type in the path in the **Data** field or use the 

button to browse for directory using the **SQL Server Folders** dialog.

**Default log files location**

Specify the default location for log files: type in the path in the **Log** field or use the  button to browse for directory using the **SQL Server Folders** dialog.

**Backup and restore options****Waiting for a new tape** **Wait indefinitely**

Specifies that SQL Server will never time out while waiting for a new backup tape.

 **Try once**

Specifies that SQL Server will time out if a backup tape is not available when needed.

 **Try for ... minute(s)**

Specifies that SQL Server will time out if a backup tape is not available within the defined period of time.

**Default backup media retention in days**

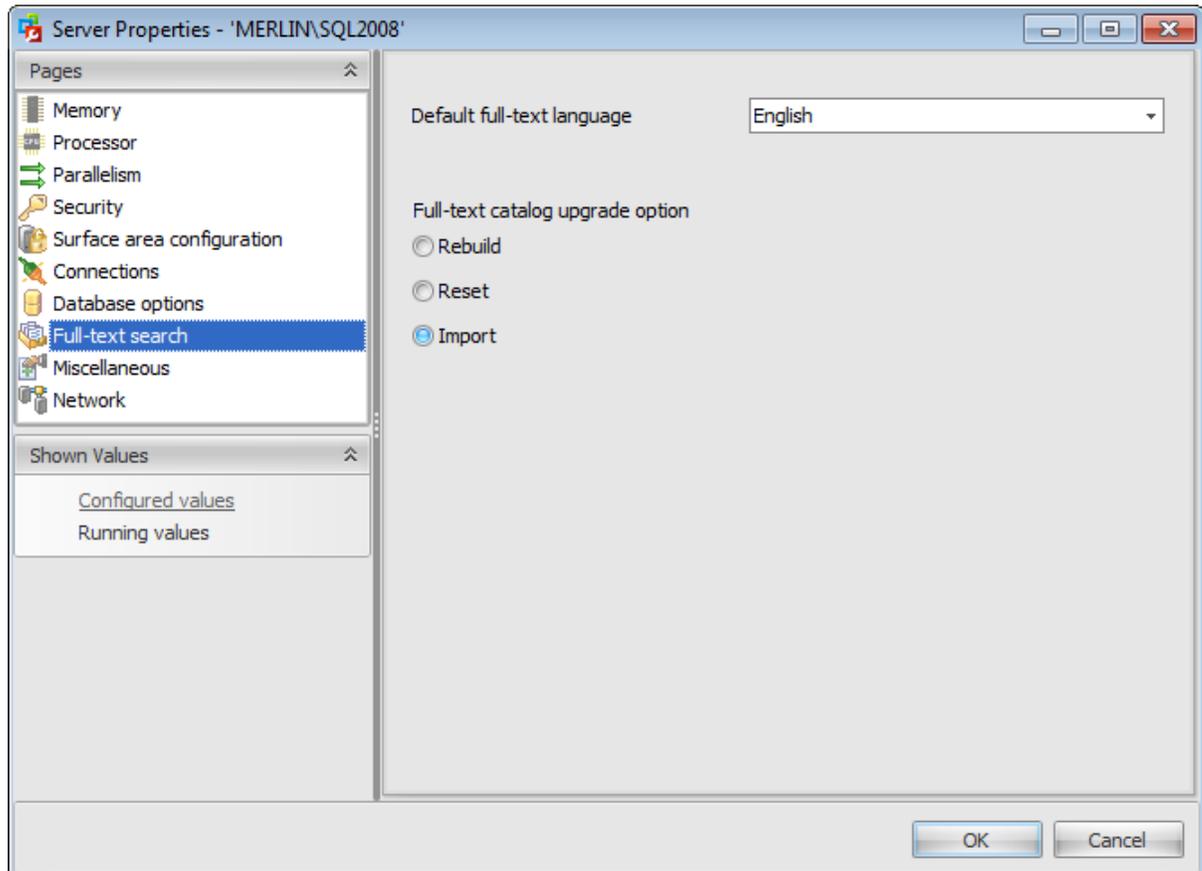
Provides a system-wide default for the length of time to retain each backup medium after it has been used for a database or transaction log backup. This option helps protect backups from being overwritten until the specified number of days has elapsed.

 **Compress backups by default**

Check this option to enable default compression of backups.

### 3.7.8 Full-text search

The **Full-text search** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by setting options of full-text queries that perform linguistic searches



#### Default full-text language

Full-text queries perform linguistic searches against text data in full-text indexes by operating on words and phrases based on rules of a particular language. Use the drop-down list to specify this language.

#### Full-text catalog upgrade option

##### Rebuild

Full-text catalogs are rebuilt using the new and enhanced word breakers. Rebuilding indexes can take awhile, and a significant amount of CPU and memory might be required after the upgrade.

##### Reset

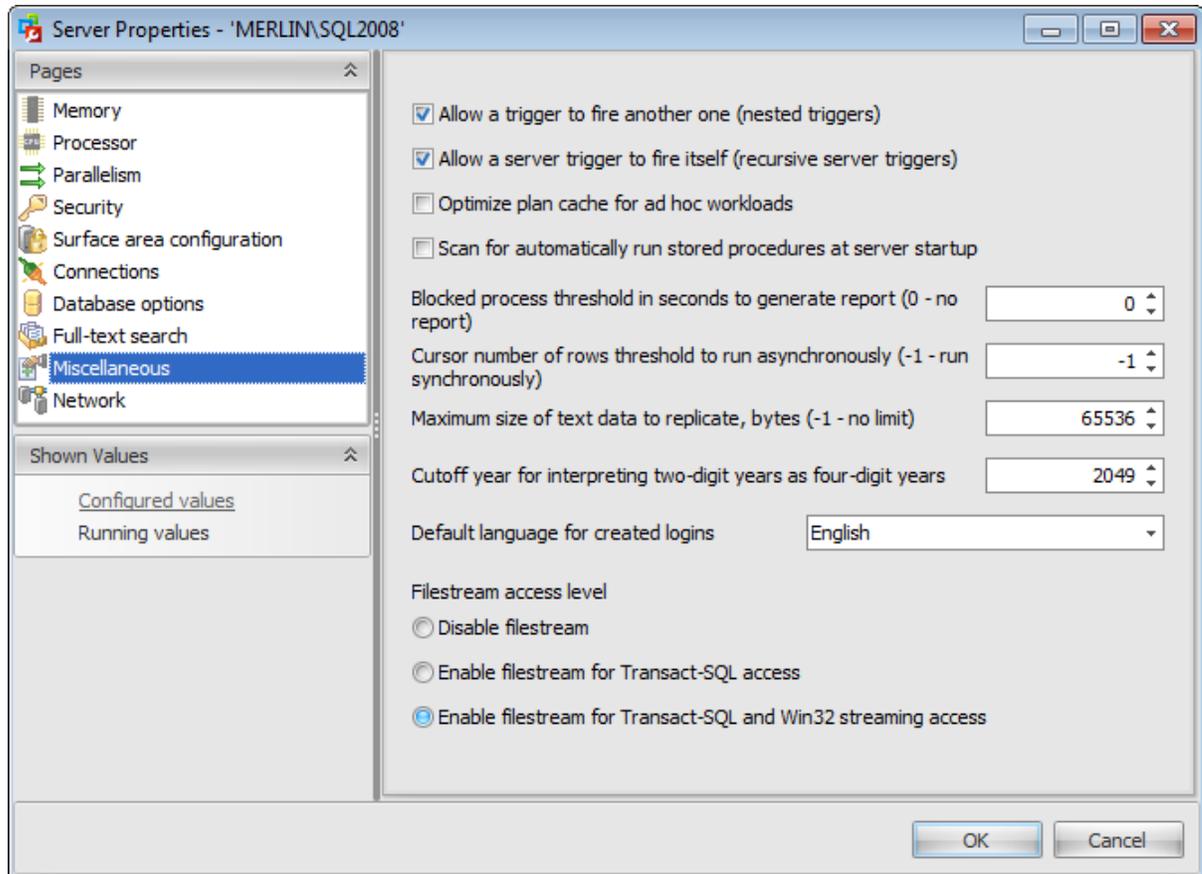
Full-text catalogs are reset. SQL Server full-text catalog files are removed, but the metadata for full-text catalogs and full-text indexes is retained. After being upgraded, all full-text indexes are disabled for change tracking and crawls are not started automatically. The catalog will remain empty until you manually issue a full population, after the upgrade completes.

**• Import**

Full-text catalogs are imported. Typically, import is significantly faster than rebuild. For example, when using only one CPU, import runs about 10 times faster than rebuild. However, an imported full-text catalog does not use the new and enhanced word breakers introduced in SQL Server 2008, so you might want to rebuild your full-text catalogs eventually.

### 3.7.9 Miscellaneous

The **Miscellaneous** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by configuring miscellaneous server settings.



**Allow a trigger to fire another one (nested triggers)**

If this option is selected then nested triggers will be enabled. Triggers are nested when a trigger performs an action that initiates another trigger. These actions can initiate other triggers, and so on. DML and DDL triggers can be nested up to 32 levels.

**Allow a server trigger to fire itself (recursive server triggers)**

Check this option to enable for an AFTER trigger to call itself recursively.

**Optimize plan cache for ad hoc workloads**

The optimize for ad hoc workloads option is used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches. If this option is checked, the Database Engine stores a small compiled plan stub in the plan cache when a batch is compiled for the first time, instead of the full compiled plan. This helps to relieve memory pressure by not allowing the plan cache to become filled with compiled plans that are not reused.

**Scan for automatically run stored procedures at server startup**

Use this option to scan for automatic execution of stored procedures at Microsoft SQL Server startup time. If this option is checked, SQL Server scans for and runs all

automatically run stored procedures defined on the server.

**Blocked process threshold in seconds to generate report**

Use this option to specify the threshold, in seconds, at which blocked process reports are generated. The threshold can be set from 0 to 86,400. By default, no blocked process reports are produced.

**Cursor number of rows threshold to run asynchronously**

Use this field to specify the number of rows in the cursor set at which cursor keysets (the set of keys that identify the rows in the cursor) are generated asynchronously. When cursors generate a keyset for a result set, the query optimizer estimates the number of rows that will be returned for that result set. If the query optimizer estimates that the number of returned rows is greater than this threshold, the cursor is generated asynchronously, allowing the user to fetch rows from the cursor while the cursor continues to be populated. Otherwise, the cursor is generated synchronously, and the query waits until all rows are returned.

If set to *-1*, all keysets are generated synchronously; this benefits small cursor sets. If set to *0*, all cursor keysets are generated asynchronously. With other values, the query optimizer compares the number of expected rows in the cursor set and builds the keyset asynchronously if it exceeds the number set.

**Maximum size of text data to replicate in bytes**

Use this option to specify the maximum size (in bytes) of *text*, *ntext*, *varchar(max)*, *nvarchar(max)*, *varbinary(max)*, *xml*, and *image* data that can be added to a replicated column or captured column in a single INSERT, UPDATE, WRITETEXT, or UPDATETEXT statement. The default value is 65536. A value of *-1* indicates no limit, other than the limit imposed by the data type. This option applies to transactional replication and Change Data Capture.

**Cutoff year for interpreting two-digit years as four-digit years**

Use this option to specify an integer less than 9999 that represents the cutoff year for interpreting two-digit years as four-digit years.

A two-digit year that is less than or equal to the last two digits of the cutoff year is in the same century as the cutoff year. A two-digit year that is greater than the last two digits of the cutoff year is in the century that precedes the cutoff year. For example, if two digit year cutoff is 2049 (the default), the two-digit year 49 is interpreted as 2049 and the two-digit year 50 is interpreted as 1950.

Use the **Default language for created logins** drop-down list to select the default language for all newly created user logins.

**Filestream access level**

FILESTREAM enables SQL Server-based applications to store unstructured data, such as documents and images, on the file system. Applications can leverage the rich streaming APIs and performance of the file system and at the same time maintain transactional consistency between the unstructured data and corresponding structured data.

 **Disable FILESTREAM support for this instance**

Check this option to disable FILESTREAM feature for handling unstructured data.

 **Enable FILESTREAM for Transact-SQL access**

Enable this option to manage FILESTREAM data with Transact-SQL statements (*insert*,

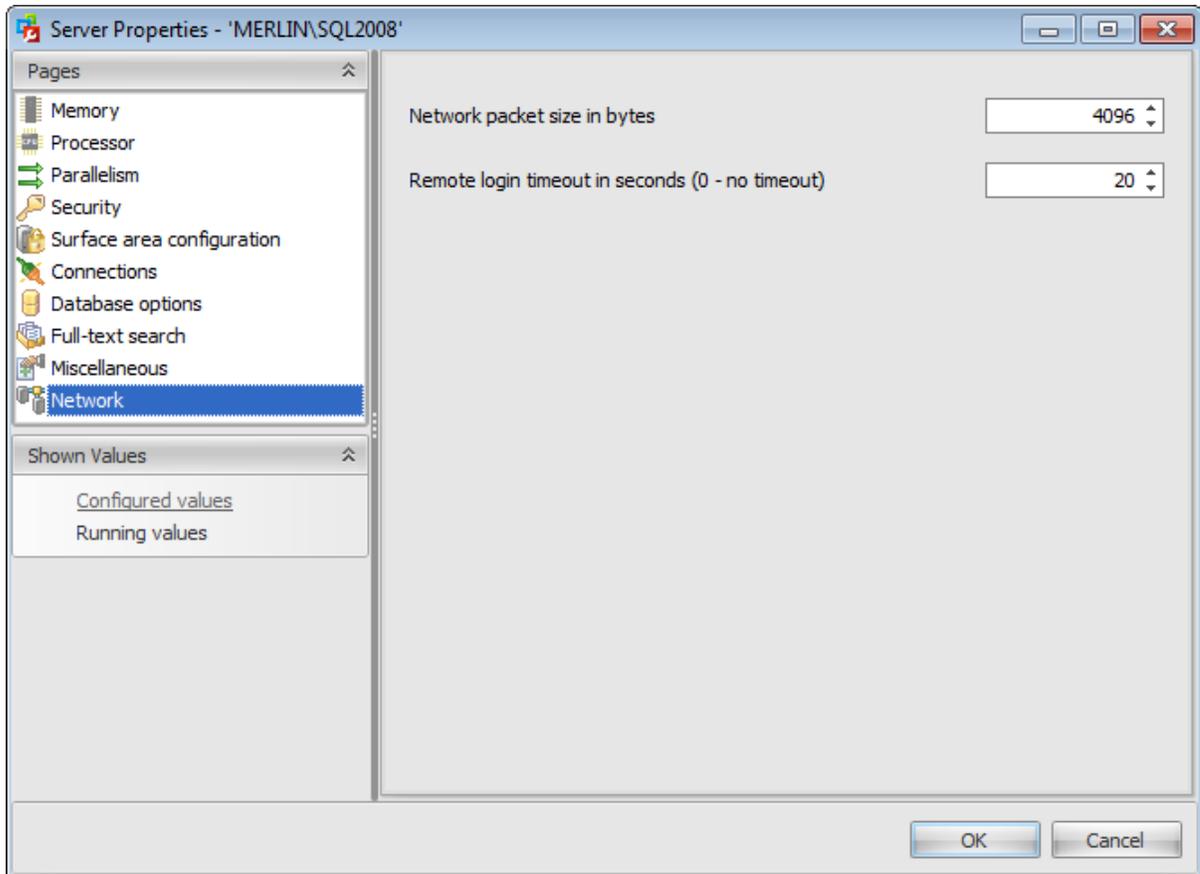
*update, query, search, and back up).*

**• Enable FILESTREAM for Transact-SQL and Win32 streaming access**

Win32 file system interfaces provide I/O streaming access to the data. You can use Win32 to read and write data to a FILESTREAM BLOB.

### 3.7.10 Network

The **Network** section of the **Server Properties** dialog allows you to configure the instance of Microsoft® SQL Server by configuring Network settings.



Use the **Network packet size** option to set the packet size (in bytes) used across the entire network.

Use the **Remote login timeout** option to specify the time interval (in seconds) to wait before returning from a failed attempt to log in to a remote server.

## 3.8 Compare server configurations

**Compare Server Configurations Wizard** allows you to save current server configuration which includes server information, server properties, global configuration settings, server principals, databases, databases principals and permissions. .Using this wizard you can also:

- compare saved and current configurations of a single server and then update the server on the basis of the saved configuration,
- compare current configurations of two different servers and then update one server on the basis of another server,
- compare saved configurations of two different servers,
- compare two saved configurations of a single server.

If a saved configuration is no longer needed you can delete it.

All saved configurations are stored in the database that is set as [SQL Administrator Repository](#).

To run the wizard, select the **Tools |  Compare Configurations...** [main menu](#) item or click the ** Compare Configurations...** button on the [toolbar](#).

The wizard consists of the following steps:

[Selecting operation](#)

### **Saving configuration**

[Setting saving configuration options](#)

[Viewing save configuration information](#)

[Executing saving configuration operation](#)

### **Comparing configurations**

[Specifying configurations for comparing](#)

[Setting comparing configurations options](#)

[Executing comparing operation](#)

[Viewing comparison results](#)

[Selecting server to update](#)

[Specifying destination file name](#)

[Viewing update information](#)

[Executing updating operation](#)

### **Deleting configuration**

[Selecting server for deleting configuration](#)

[Selecting configurations to delete](#)

[Viewing delete configuration information](#)

[Executing delete configuration operation](#)

Availability:

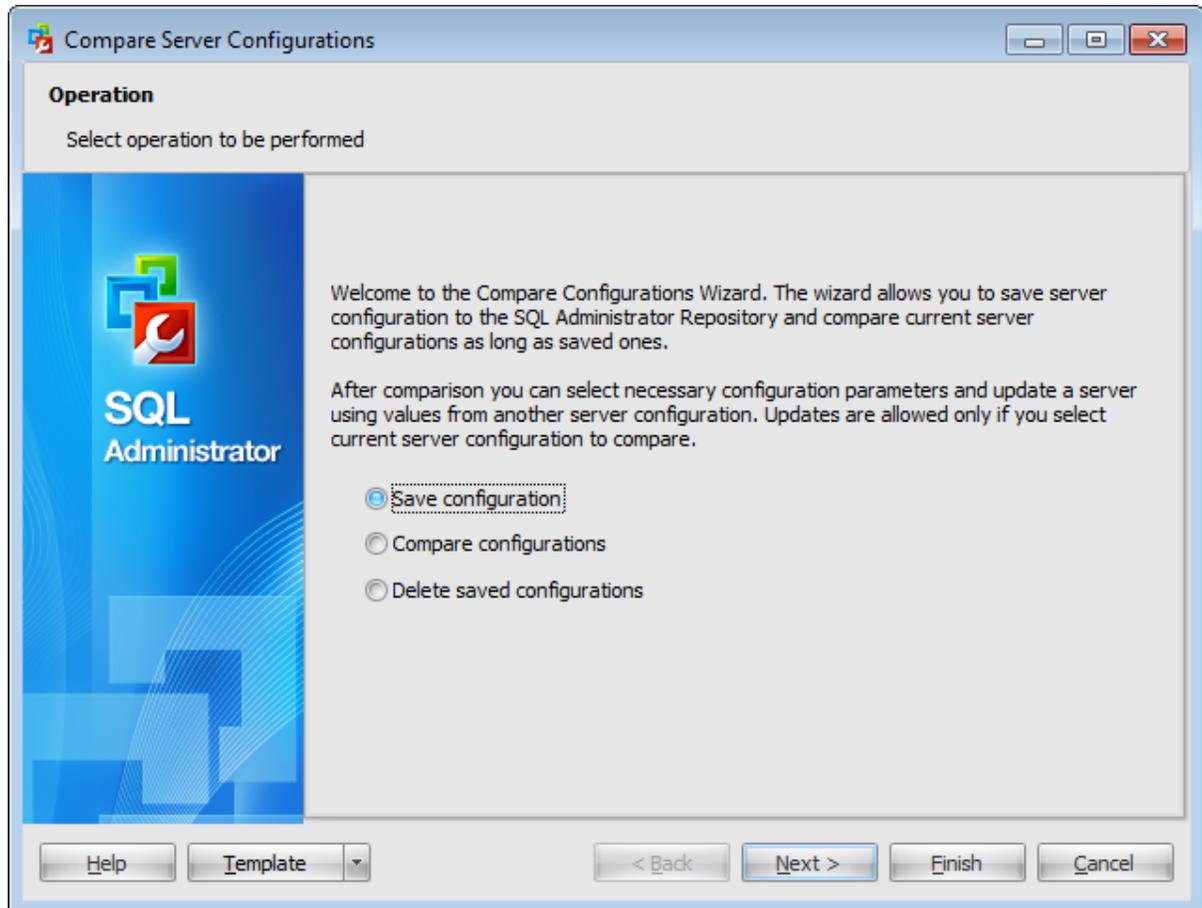
**Full** version **Yes**

**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

### 3.8.1 Selecting operation

On the first step of the wizard you need to select the operation that you want to perform by means of this wizard.



#### **Save configuration**

Save current server configuration parameters to the [SQL Administrator Repository](#). The saved configuration includes the following: server information, server properties, server configuration options, database options, database files and filegroups, server and database principals, server and database object permissions.

#### **Compare configurations**

Select this item if you want to compare current servers configurations or saved ones. You can compare configurations of a single server or of two different servers. After comparison you can update a server on the basis of another configuration. Update is allowed only if you select current server configuration as one of configurations for comparing.

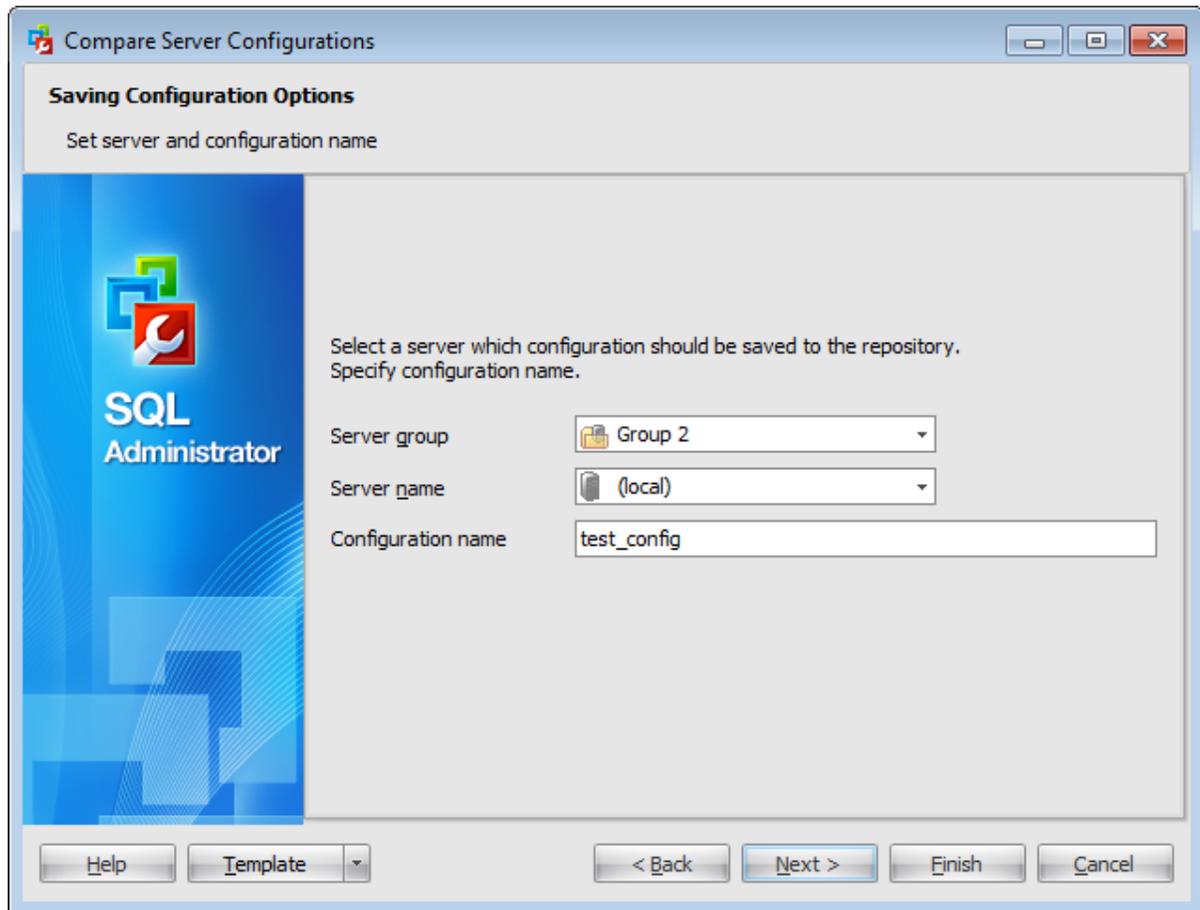
#### **Delete saved configurations**

Select this item if you want to delete configurations are no longer needed.

## 3.8.2 Saving configuration

### 3.8.2.1 Setting saving configuration options

On this step you need to specify options for saving configuration operation.



The screenshot shows a Windows-style dialog box titled "Compare Server Configurations". The main heading is "Saving Configuration Options" with the instruction "Set server and configuration name". On the left is a blue sidebar with the "SQL Administrator" logo. The main area contains the instruction "Select a server which configuration should be saved to the repository. Specify configuration name." and three input fields: "Server group" (dropdown menu showing "Group 2"), "Server name" (dropdown menu showing "(local)"), and "Configuration name" (text box containing "test\_config"). At the bottom are buttons for "Help", "Template" (with a dropdown arrow), "< Back", "Next >", "Finish", and "Cancel".

#### **Server group**

Use the drop-down list to select a server group where the necessary server is stored.

#### **Server name**

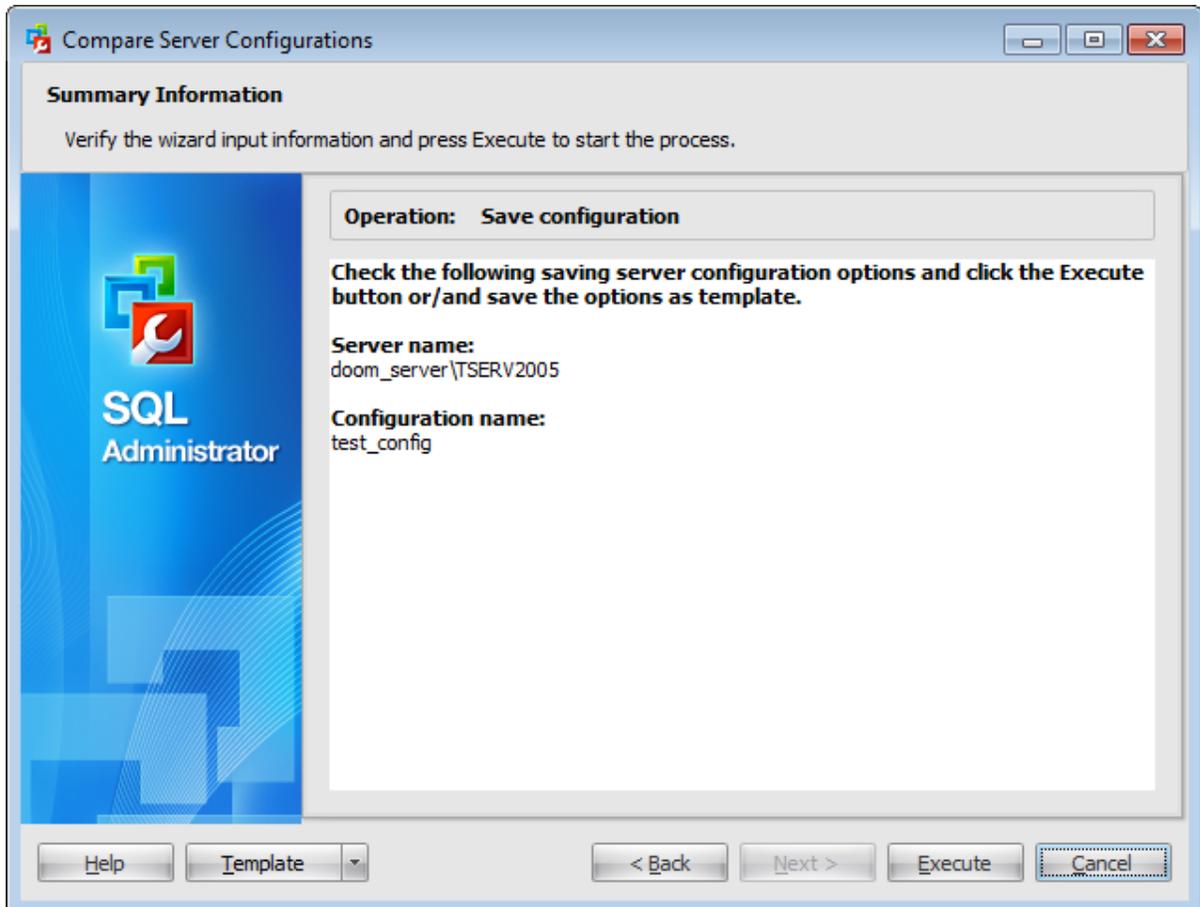
Use the drop-down list to select the server which configuration you want to save.

#### **Configuration name**

Specify the name for the configuration being saved. The allowed maximum name length is 128 symbols.

### 3.8.2.2 Viewing saving configuration information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the saving configuration process.

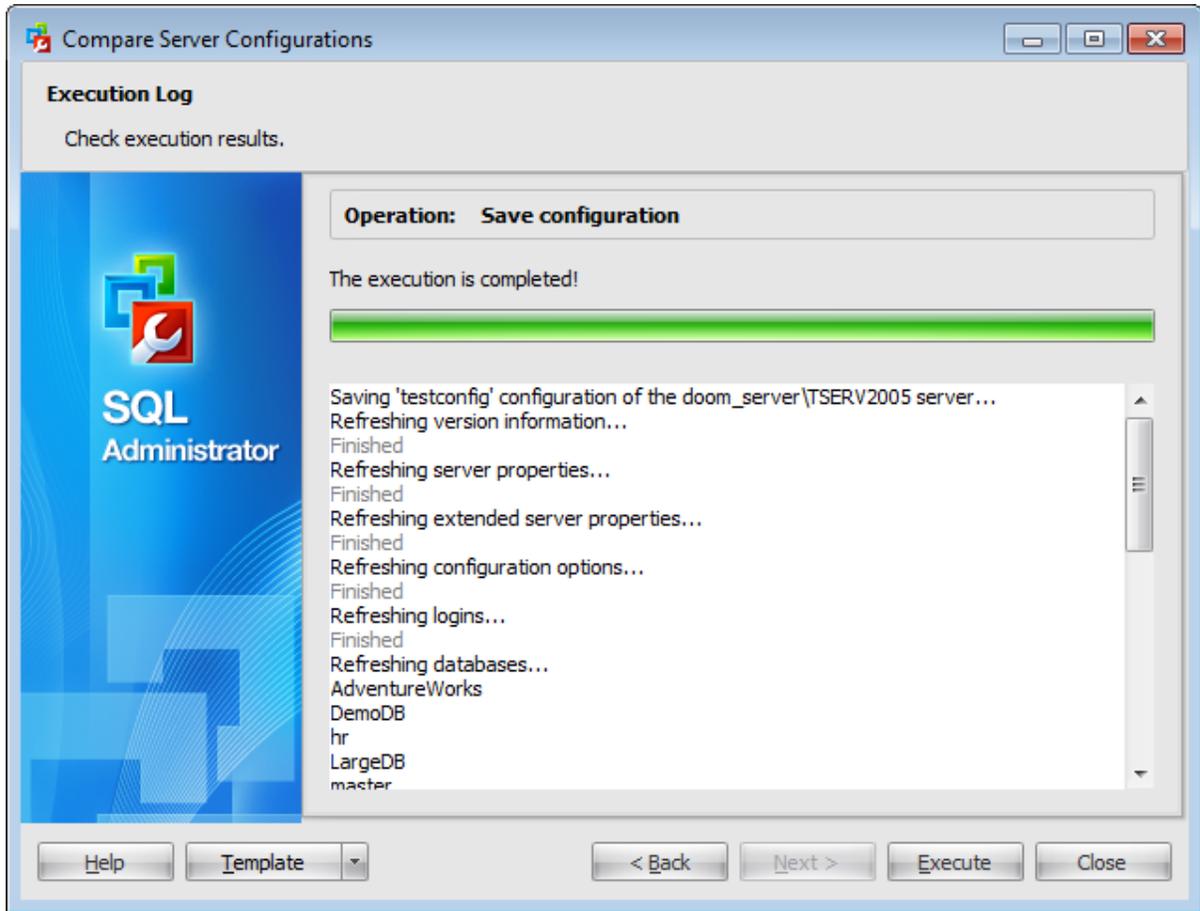


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing saving configuration operation](#) step.

### 3.8.2.3 Executing saving configuration operation

The log area allows you to view the log of operation and errors (if any).

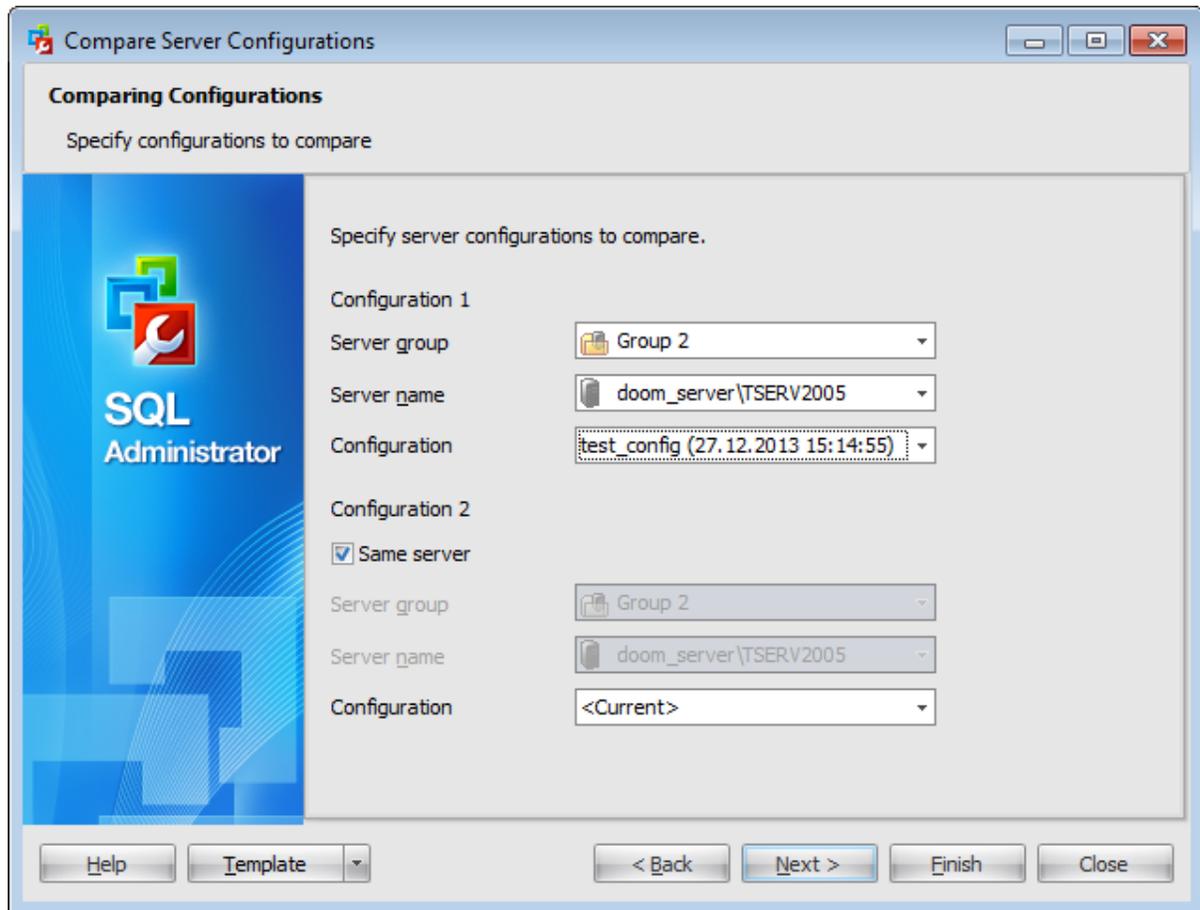


Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

### 3.8.3 Comparing configurations

#### 3.8.3.1 Specifying configurations for comparing

On this step you need to specify configurations which you want to compare.



#### **Server group**

Use the drop-down list to select a server group where the necessary server is stored.

#### **Server name**

Use the drop-down list to select the server which configuration you want to be compared.

#### **Same server**

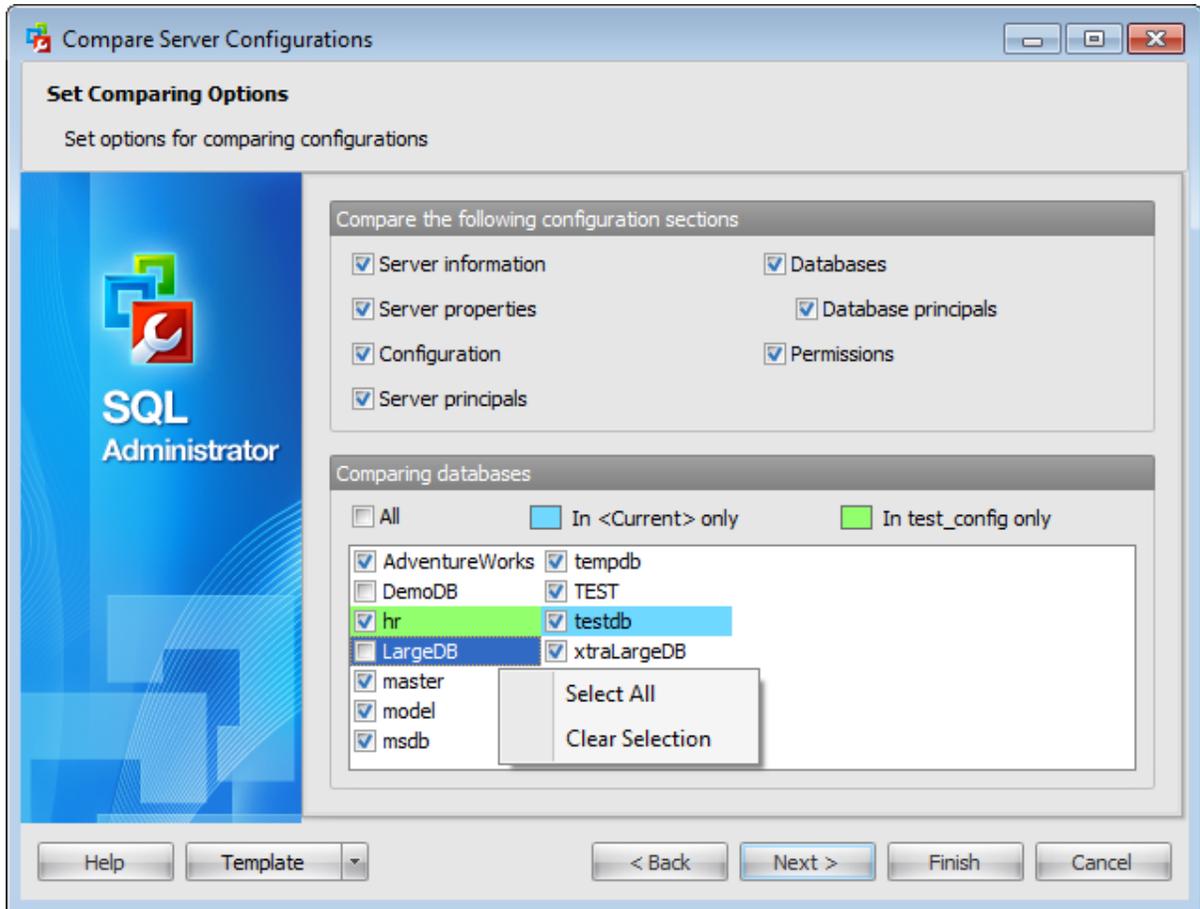
Check this option if you want to compare two configurations of a single server.

#### **Configuration**

Use the drop-down list to select the configuration for comparison. This list contains names of all saved configurations with time they were created. If you want to compare and update the current server configuration select the *<Current>* item.

### 3.8.3.2 Setting comparing configurations options

On this step you need to specify options for comparing configuration operation.



In the upper area select the configuration sections which you want to be compared:

- *server information,*
- *server properties,*
- *global configuration settings,*
- *server principals,*
- *databases,*
- *databases principals,*
- *permissions.*

In the lower section select databases which you want to compare. The following databases parameters (and principals) are compared:

- *database properties,*
- *filegroups,*
- *files,*
- *application roles,*
- *roles,*
- *users.*

Databases that are presented in the first specified configuration, but are not presented in

the second one are painted blue.

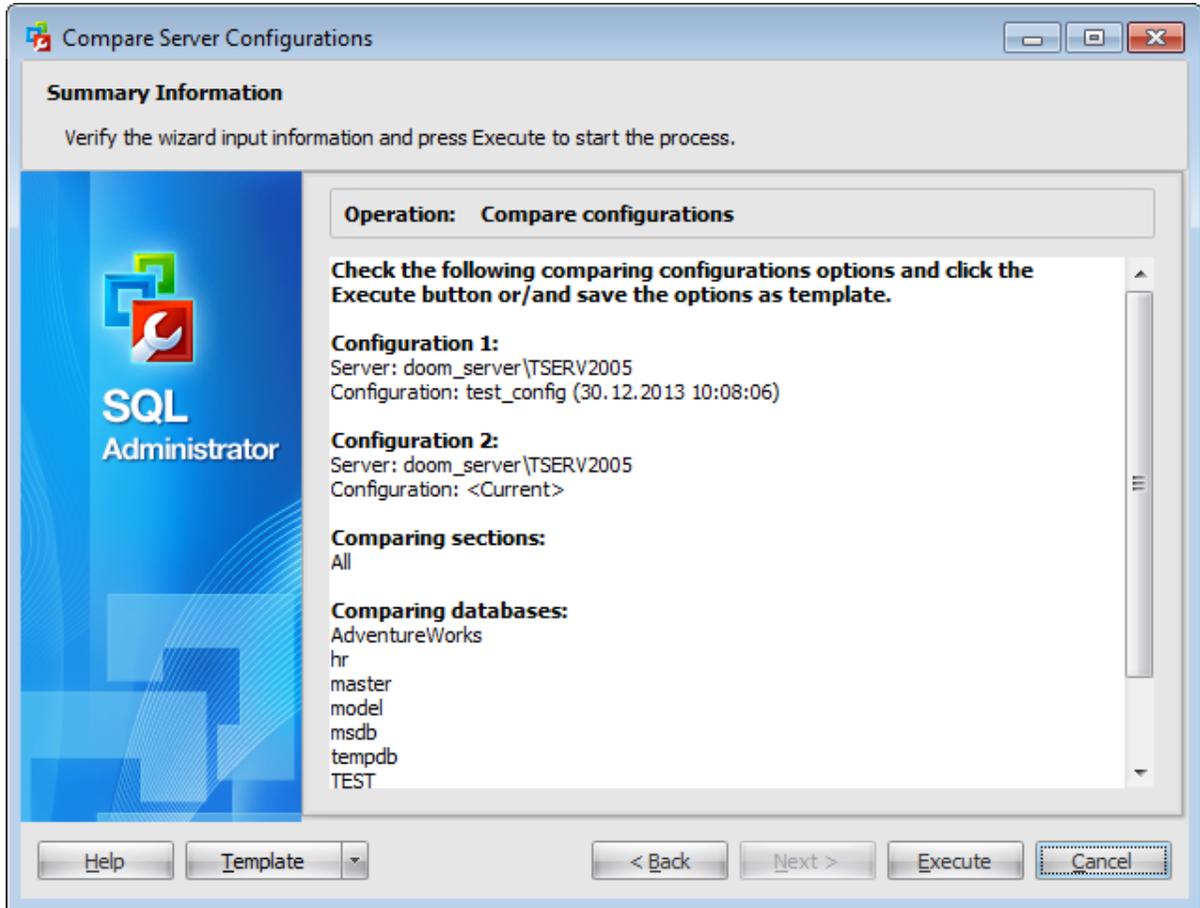
Databases that are presented in the second specified configuration, but are not presented in the first one are painted green.

For your convenience the *Select All* and *Clear Selection* functions are implemented in the **context menu** of the databases list.

Check the  **All** checkbox to perform a comparison of all databases in the list.

### 3.8.3.3 Viewing comparing configurations information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the comparing configurations process.

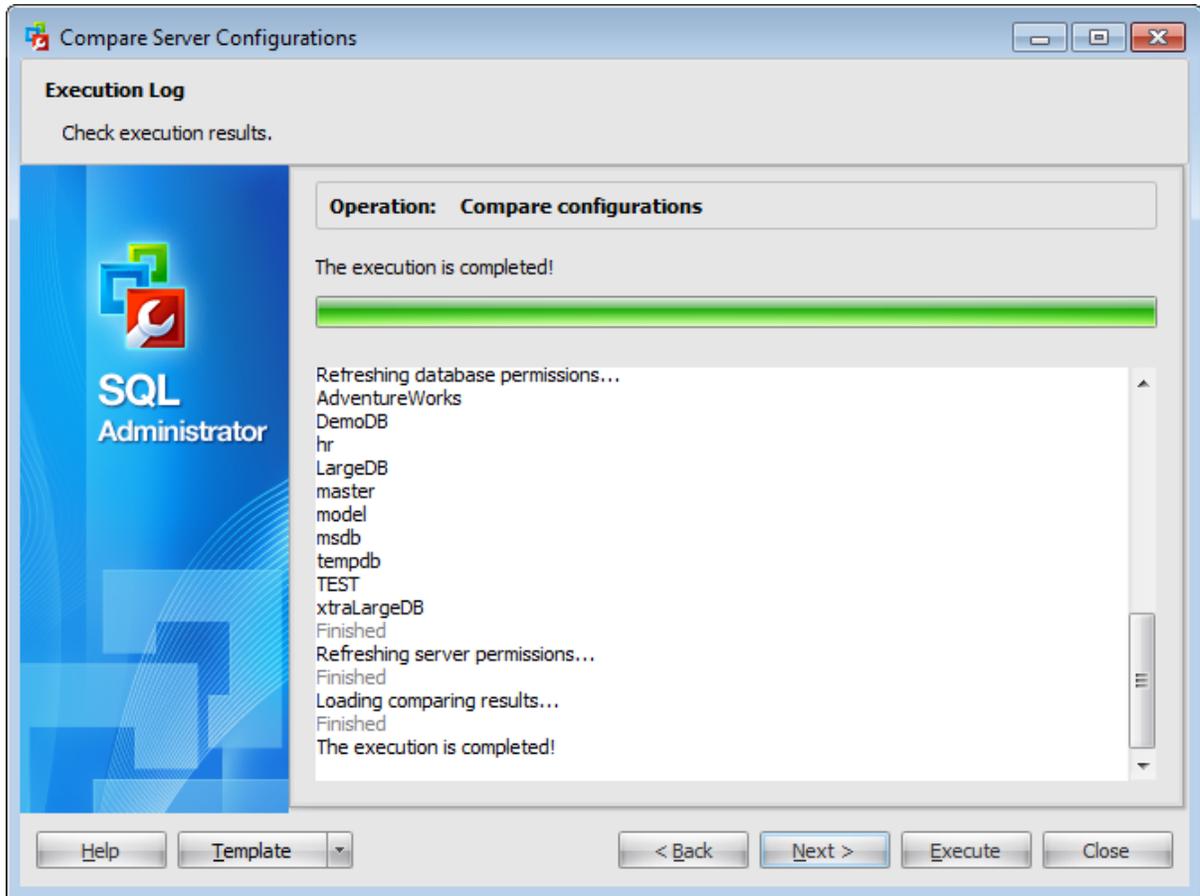


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing comparing configuration operation](#) step.

### 3.8.3.4 Executing comparing operation

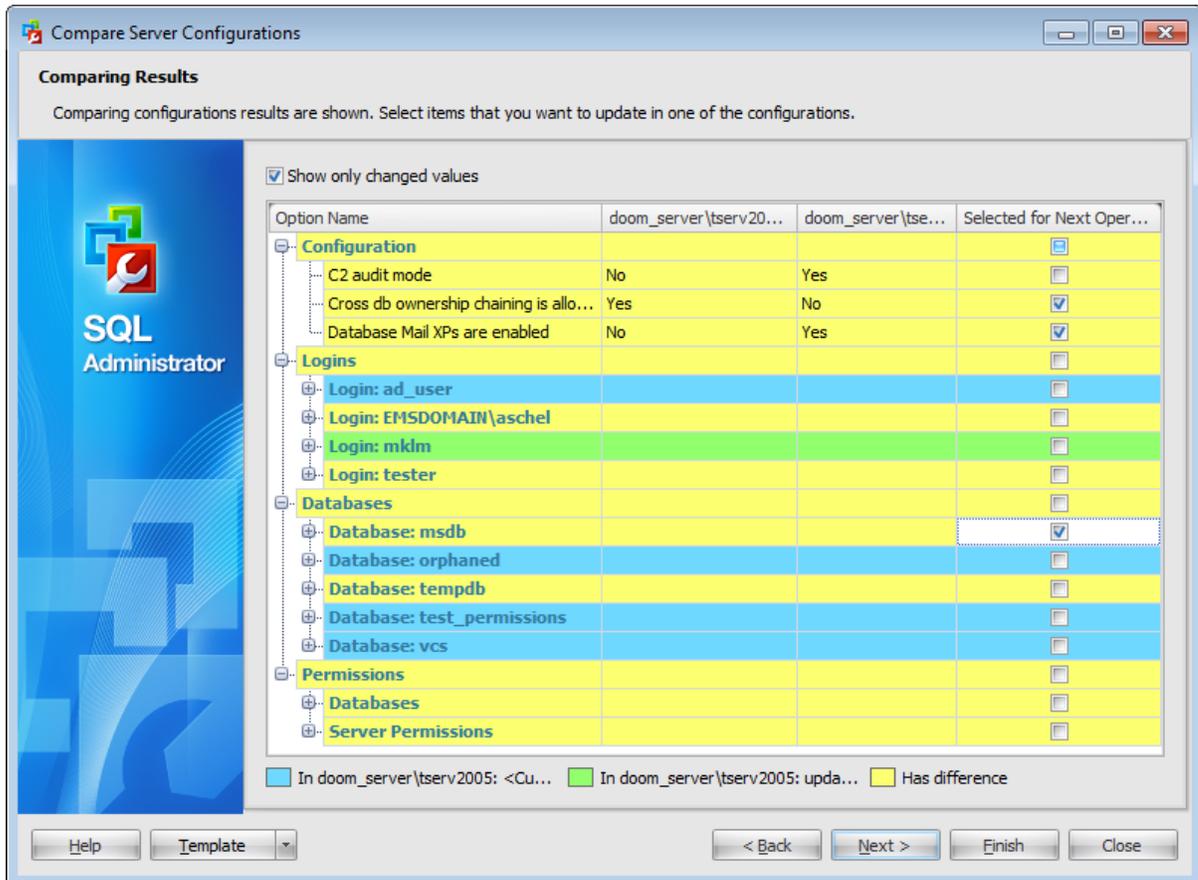
The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

### 3.8.3.5 Viewing comparison results

After comparing is performed the following grid appears. Here you can view similarities and differences between specified configurations. If a current server configuration has been compared you can select changed parameters that you want to restore.



#### Show only changed values

Check this option to display only those parameters that have different values in compared configurations.

Parameters with different values in two configurations are painted yellow.

Objects that exist only in the first configuration are painted blue.

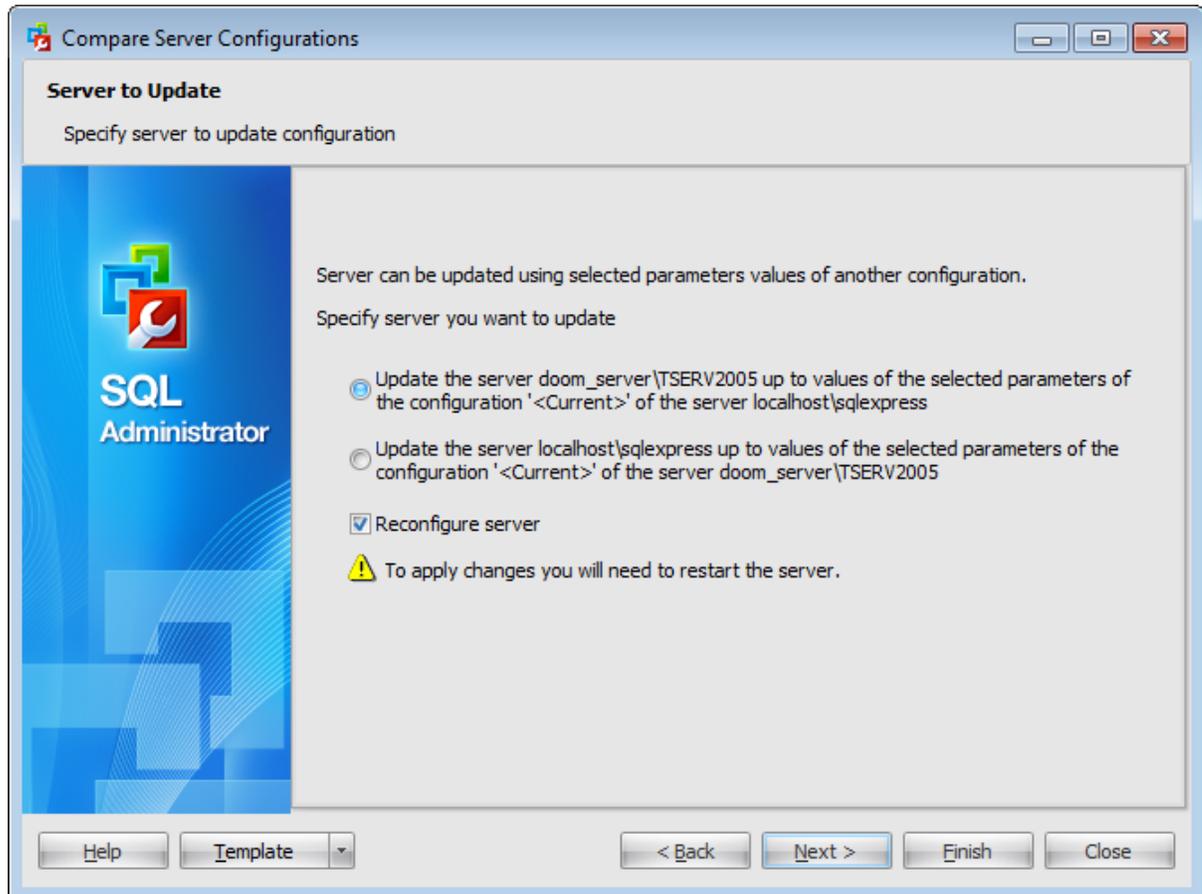
Objects that exist only in the second configuration are painted green.

To proceed to the next step you need to check parameters that you want to update or create in the current server configuration.

**Note:** The updating operation is available only if the current server configuration is compared. If you are comparing two saved configurations of two servers then the last column "Selected for next operation" is hidden.

### 3.8.3.6 Selecting server to update

On this step you need to select which server should be updated. If you are comparing a saved and a current configuration of a single server then the current configuration will be updated on the basis of the saved configuration.



The [selected parameters](#) of the selected server will be updated on the basis of another server configuration; objects will be created or dropped depending on the configuration.

#### **Reconfigure server**

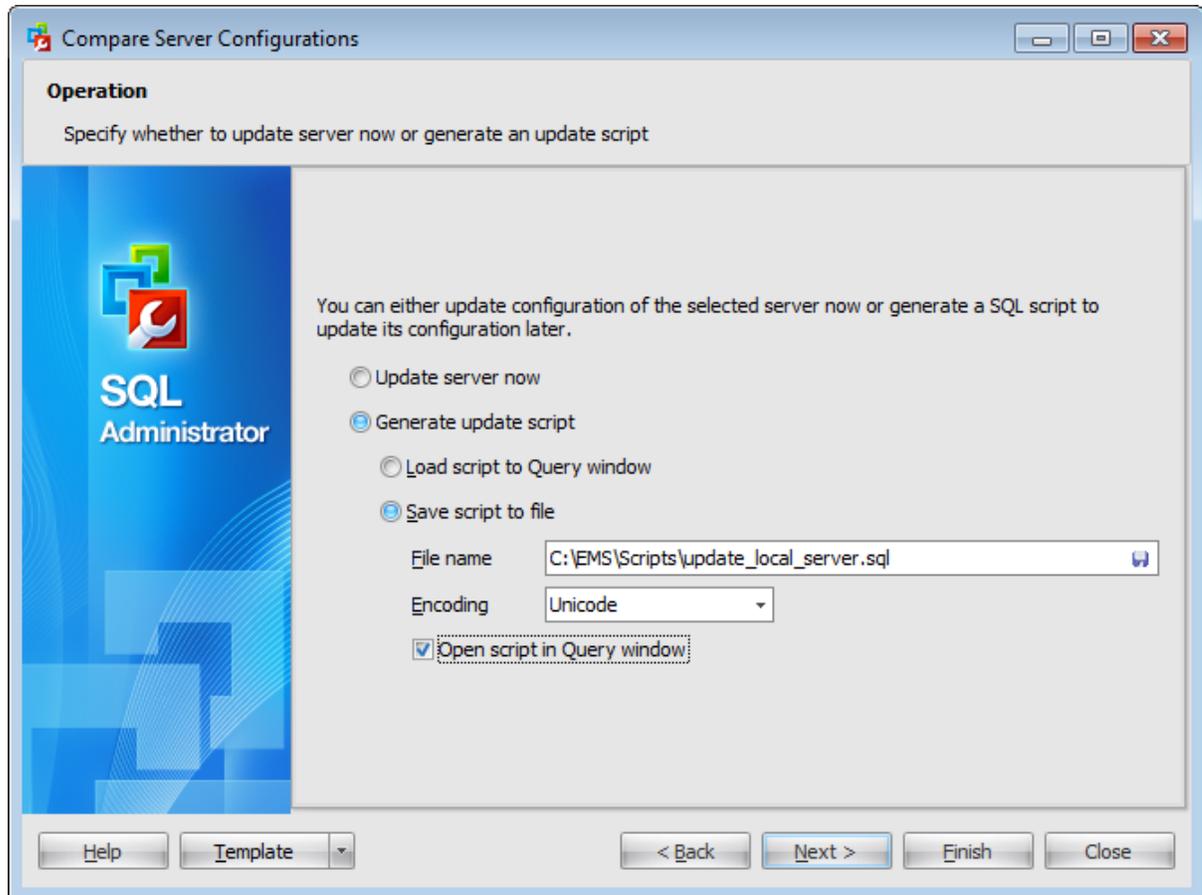
This option is available if you have selected to update parameters from the Configuration group (global configuration settings). If this option is checked then the 'RECONFIGURE WITH OVERRIDE' statement is included in the result update script, so that all server configuration parameters will be changed right after the script execution.

If this option is not checked then changes will be applied after server restart or on the next server reconfiguring.

**Note:** Applying changes of some options may require restart of the server.

### 3.8.3.7 Specifying destination file name

You can select whether the update script should be executed now, loaded to Query window or saved to file.



#### Update server now

Select this option to update selected server parameters now without viewing the execution script.

#### Generate update script

Select this option if you want to view the update script before execution. You can  Load script to [Query window](#) or  Save script to file.

#### File name

Set a name for the result \*.sql file and type in or use the Save as... button to specify the path to this file on your local machine or on a machine in the LAN.

#### Encoding

If necessary, use the drop-down list to select the encoding to be applied to the output file.

#### Open script in Query window

If this option is checked the script will be automatically loaded to the [Query editor](#) after it

is saved.

### 3.8.3.8 Setting passwords

On this step of the wizard you need to specify passwords for logins and application roles being created. This step is available if you have selected to create new logins or application roles on the server.

Specify password for creating SQL Server logins.

Login	Password	Confirm Password	Must Change on Next Logon
ad_user	.....	.....	<input checked="" type="checkbox"/>
login	.....	.....	<input checked="" type="checkbox"/>
screens	.....	.....	<input checked="" type="checkbox"/>

Specify password for creating application roles.

Database	Application Role	Password	Confirm Password
AdventureWorks	app_role	.....	.....
AdventureWorks	test_application_role		

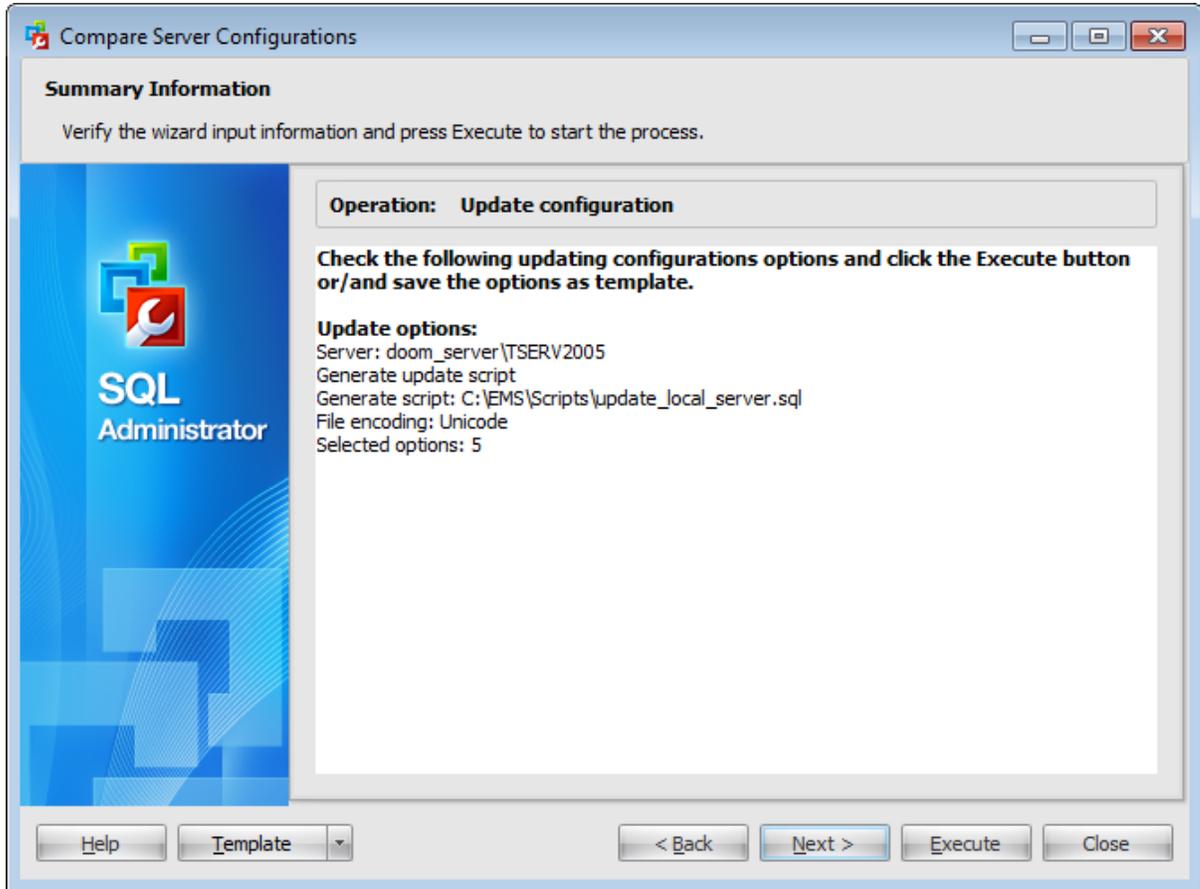
In the upper grid provide new password for every login in the list. You need to enter the password in the respective *Password* column and then enter it once more in the *Confirm Password* field.

Check the *Must Change on Next Logon* field if you want this password to be obligatory changed next time the login is used for log on.

In the lower grid you need to provide *Password* for each application role and then confirm them in the *Confirm Password* column.

### 3.8.3.9 Viewing update information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the updating configuration process.

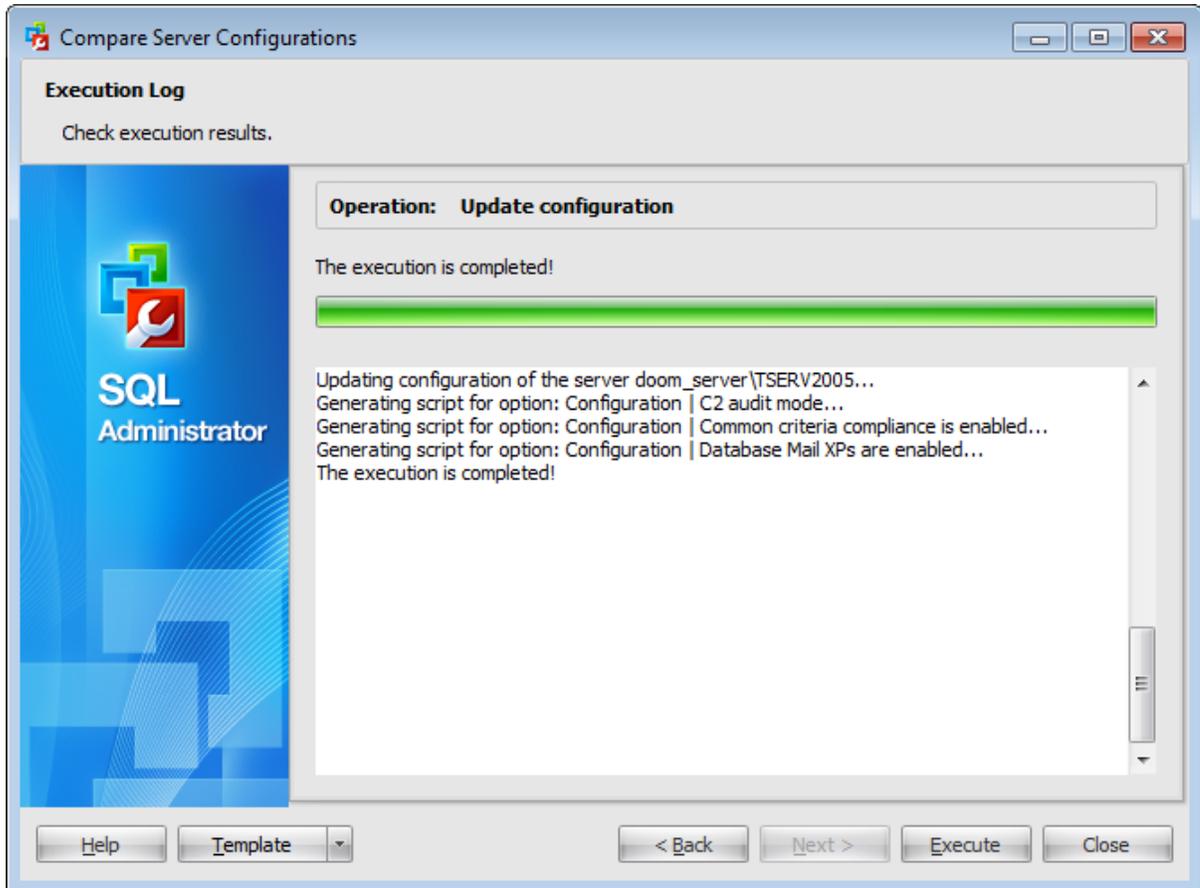


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing updating operation](#) step.

### 3.8.3.10 Executing updating operation

The log area allows you to view the log of operation and errors (if any).

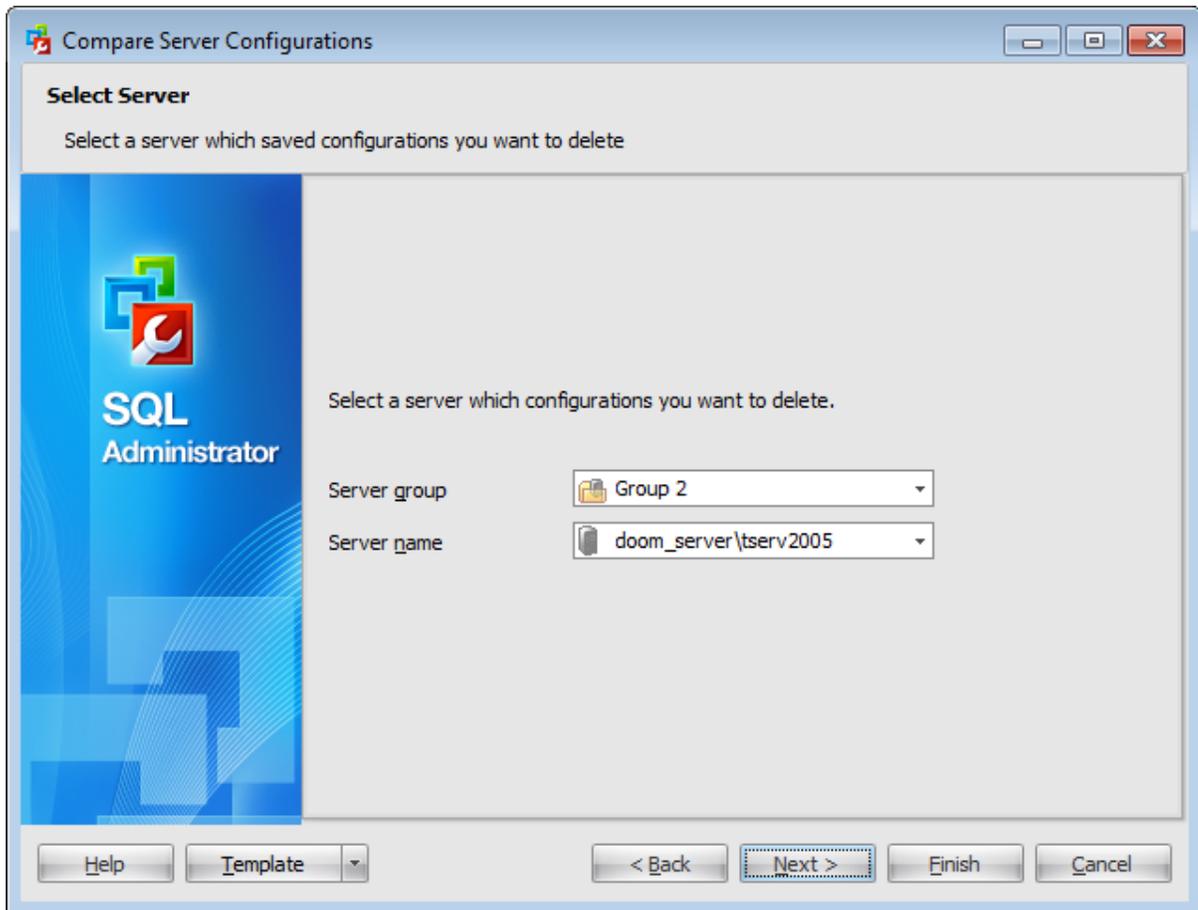


Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

### 3.8.4 Deleting configuration

#### 3.8.4.1 Selecting server for deleting configuration

On this step you need to specify a server which configurations you want to delete.

**Server group**

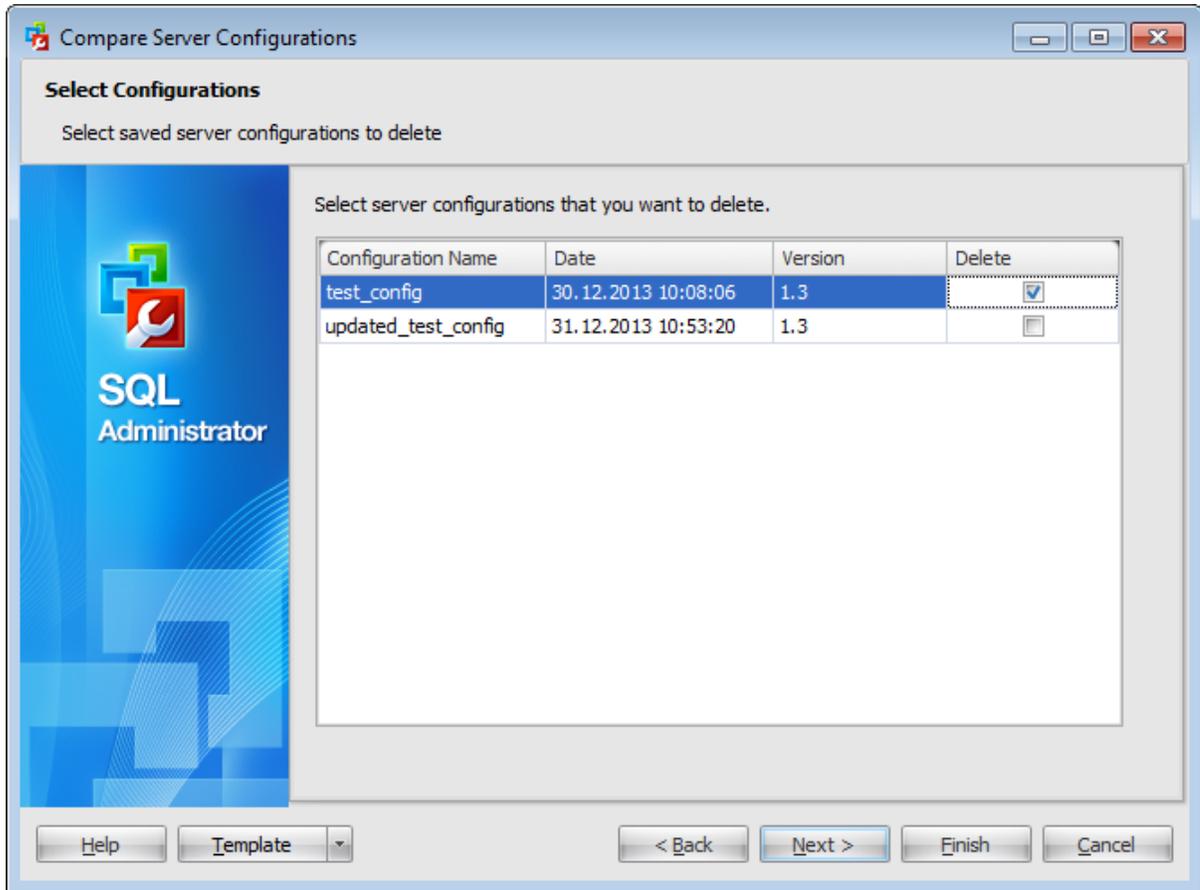
Use the drop-down list to select a server group where the necessary server is stored.

**Server name**

Use the drop-down list to select the server which configurations you want to delete.

### 3.8.4.2 Selecting configurations to delete

Use this step to select configurations that you want to delete.

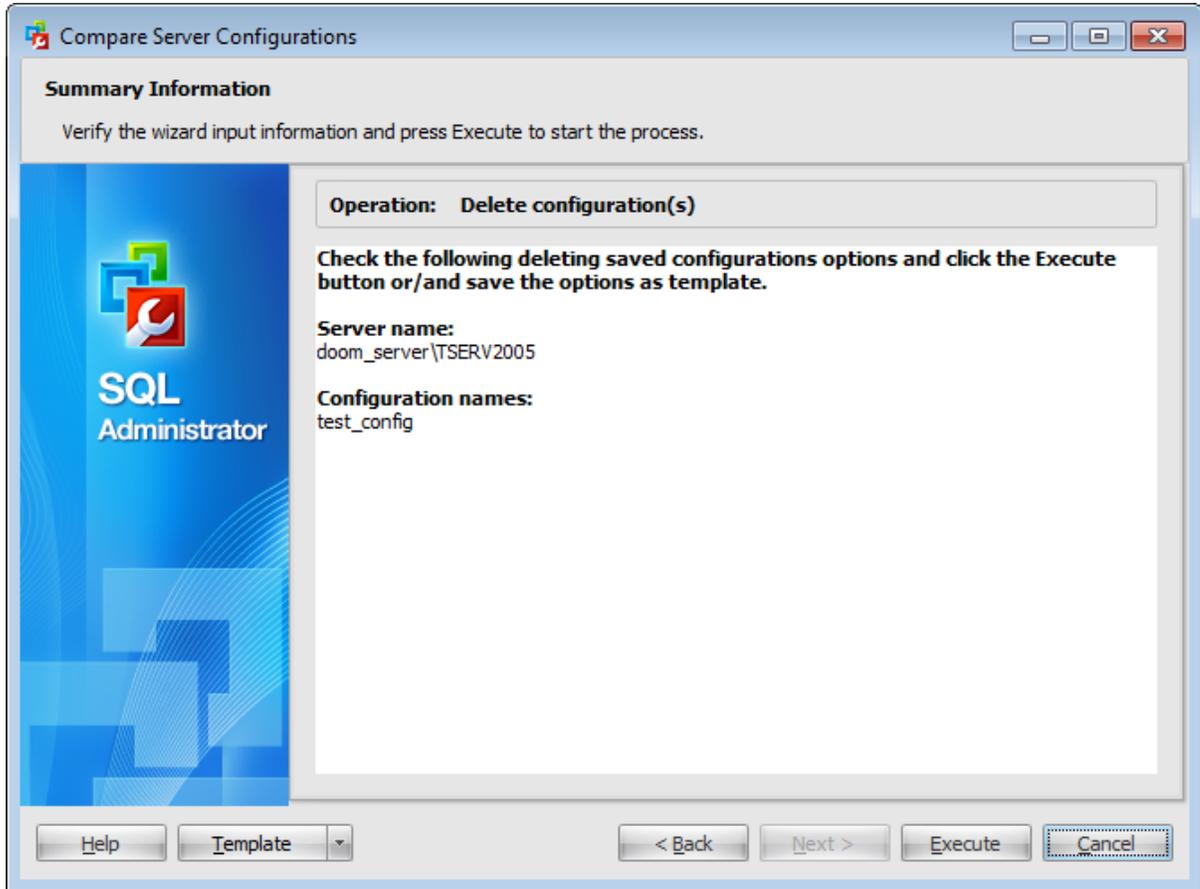


The grid lists all saved configurations of the selected server. In the grid you can view the configuration name, the date it was saved and the SQL Administrator version in which it was saved.

To select a configuration for deleting check the corresponding checkbox in the *Delete* column.

### 3.8.4.3 Viewing delete configuration information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the delete configurations process.

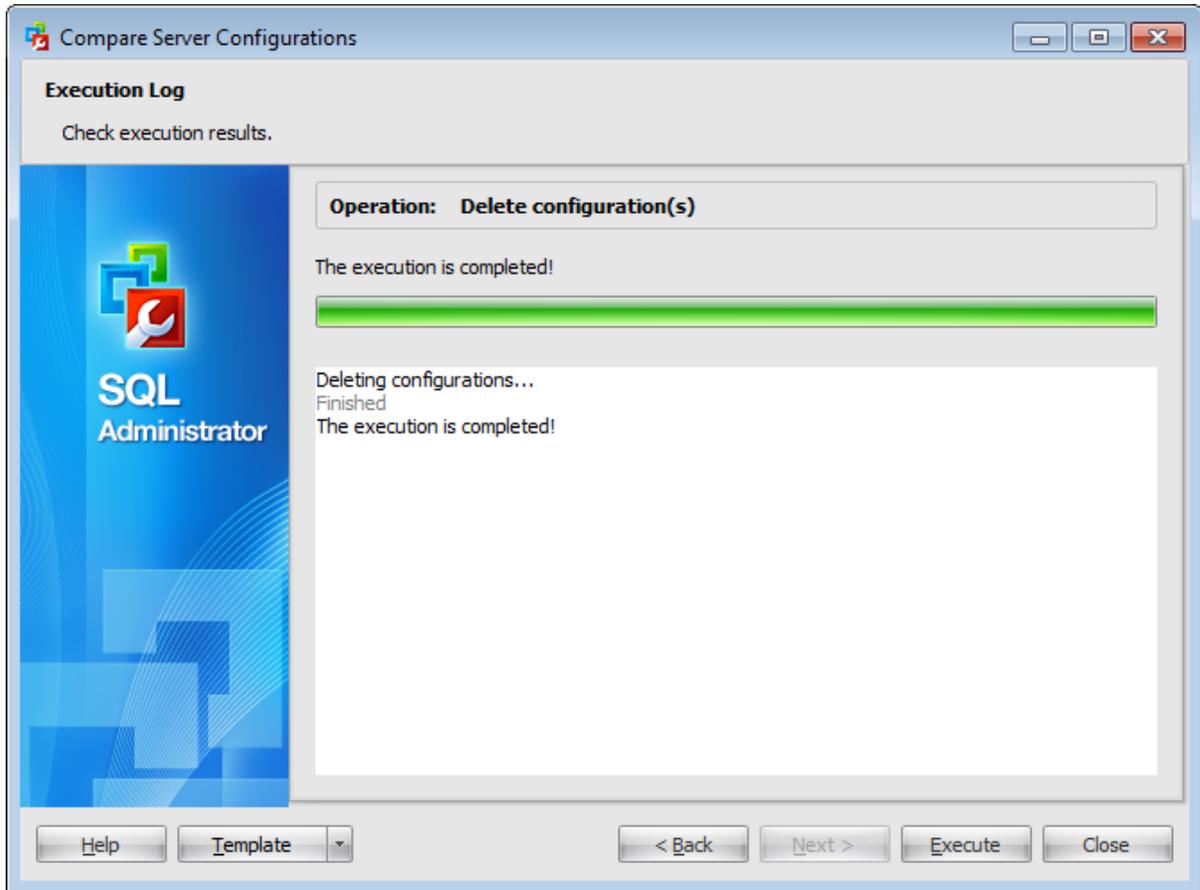


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing delete configuration operation](#) step.

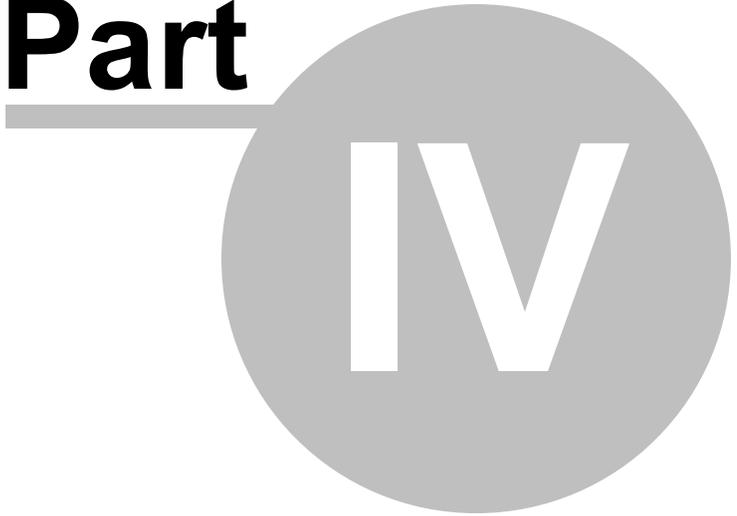
#### 3.8.4.4 Executing delete configuration operation

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

**Part**



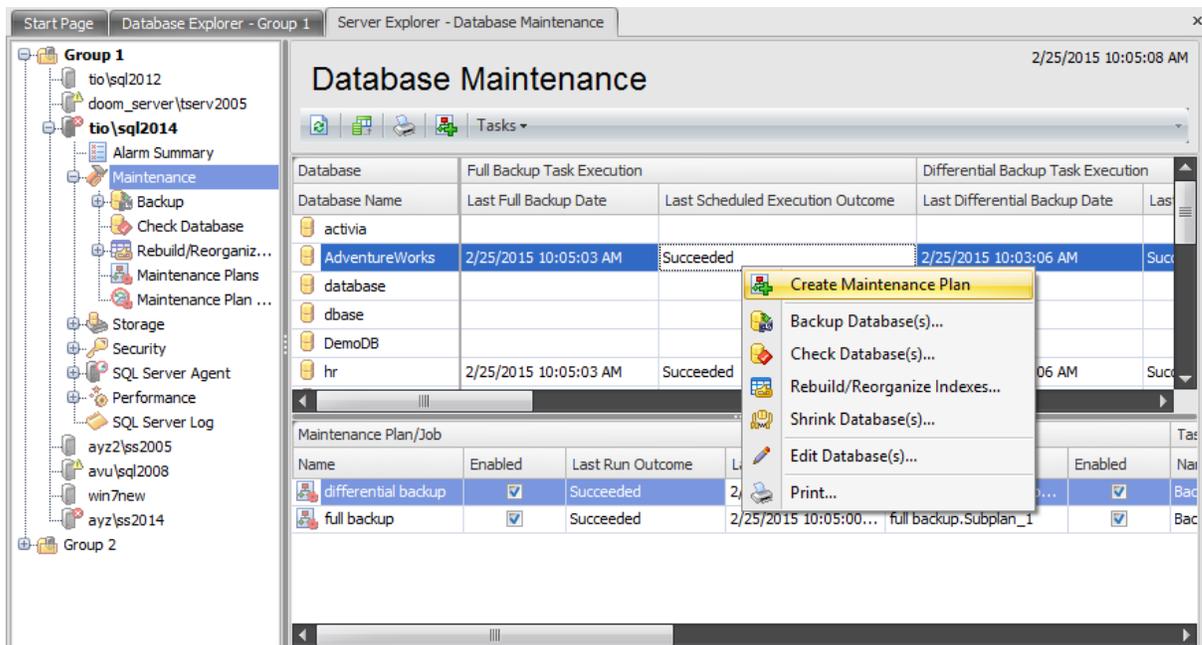
## 4 Database Maintenance

SQL Administrator provides a number of powerful tools that allow you to perform various operations over your servers and databases:

- [Manage maintenance plans](#)
- [Backup database](#)
- [Restore database](#)
- [Attach database](#)
- [Detach database](#)
- [Check database](#)
- [Rebuild/Reorganize](#)
- [Shrink database](#)

In the **Maintenance** section of the [Server Explorer](#) you can view the whole list of actions performed with every database on the selected server as well as their maintenance plans.

**Note:** This section (except for [Backup](#) page) is not available for SQL Server Express Edition as managing maintenance plans requires SQL Server Agent service.



To start managing a database you are to select a server in the [Server Explorer](#) and then select the **Maintenance** tab among the sever nodes.

The upper work area shows the list of databases vertically and maintenance actions horizontally. For each maintenance action you can see the *Last Run Date* and *Last Scheduled Execution Outcome*.

The list of actions to perform with the selected database contains the following tasks:

 **Full Backup** (a full database backup which backs up the entire database including the transaction log);

-  [Differential Backup](#) (a differential backup which records only the changes made to the data in the database after the last full database backup);
-  [Transaction Log Backup](#) (a sequence of log backups provided for a continuous chain of transaction information to support recovery forward from database, differential, or file backups);
-  [Checking Database](#) (a check of the entire database including allocation checks in addition to checks of index structure and data integrity);
-  [Rebuilding/Reorganizing Indexes of Database](#) (an operation is performed on indexes of all tables and views);
-  [Shrinking Database](#) (shrink the entire database).

The toolbar and the context menu of the upper work area allow you to perform the following actions:

-  [create maintenance plan](#)
-  [backup database](#)
-  [check database](#)
-  [rebuild/reorganize indexes](#)
-  [shrink database](#)
-  [edit database](#)
-  create and [print](#) the report

To get access to the maintenance actions via toolbar you can click the **Tasks** button and select the necessary task.

The toolbar and the context menu of the lower work area allow you to perform the following actions with the maintenance plans:

-  edit maintenance plan/job
-  drop maintenance plan/job
-  view maintenance plan/plan history
- enable/disable maintenance plan/job
-  execute maintenance plan/job
- stop job
- edit schedules
- enable/disable schedules
-  create and [print](#) the report

The lower work area shows information on the maintenance plan/job (*Name, Enabled, Last Run Outcome, Last Run Date*), schedules (*Name, Enabled, Description*) and task/job steps (*Name, Last Run Outcome, Last Run Duration, Last Run Message*) for the selected task.

---

**See also:**

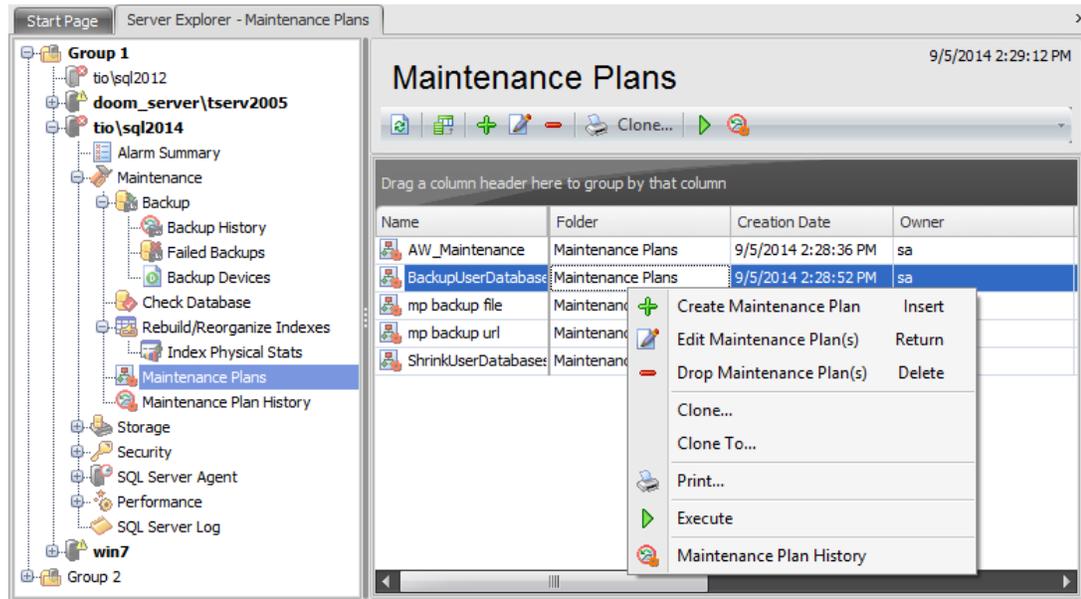
[Getting Started](#)  
[Servers](#)  
[Storage](#)  
[Security](#)  
[SQL Server Agent](#)  
[Performance](#)  
[SQL Server Log](#)

[Query Editor](#)  
[Options](#)

## 4.1 Maintenance plans

Maintenance plans create a workflow of the tasks required to make sure that your database is optimized, regularly backed up, and free of inconsistencies.

**Note:** To create and edit maintenance plans a login must be a System Administrator role member.



SQL Server maintenance plans provide the following features:

- Workflow creation using a variety of typical maintenance tasks. You can also create your own custom Transact-SQL scripts.
- Conceptual hierarchies. Each plan lets you create or edit task workflows. Tasks in each plan can be grouped into subplans, which can be scheduled to run at different times.
- Support for multiserver plans that can be used in master server/target server environments.
- Support for logging plan history to remote servers.
- Support for Windows Authentication and SQL Server Authentication. When possible, use Windows Authentication.

Use the context menu or buttons on the work area toolbar to manage the list of maintenance plans:

- [create](#) a new maintenance plan
- [edit](#) selected maintenance plan
- drop selected maintenance plan
- create the object copy on the same server
- clone the object copy to another server
- execute selected maintenance plan
- view [maintenance plan history](#)
- create and [print](#) the report

The list displays the existing maintenance plans as a grid with the following columns: *Name, Folder, Application, Creation Date, Owner, Reader Role, Writer Role, Encrypted,*

*Description.* If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Maintenance Plan in the [Maintenance Plan Editor](#), double-click it in the grid. Alternatively, you can right-click the Maintenance Plan alias and select the **Edit Login(s)** context menu item.

---

**See also:**

[Backup database](#)

[Restore database](#)

[Attach database](#)

[Detach database](#)

[Check Database](#)

[Rebuild/Reorganize Indexes](#)

[Shrink database](#)

### 4.1.1 Maintenance Plan Editor

**Maintenance Plan Editor** allows you to specify the properties of a maintenance plan and its subplans.

To open a maintenance plan in the **Maintenance Plan Editor**, double-click it in the [maintenance plan list](#) or in the [maintenance plan history](#), or right-click the maintenance plan alias and select the **Edit Maintenance Plan** context menu item.

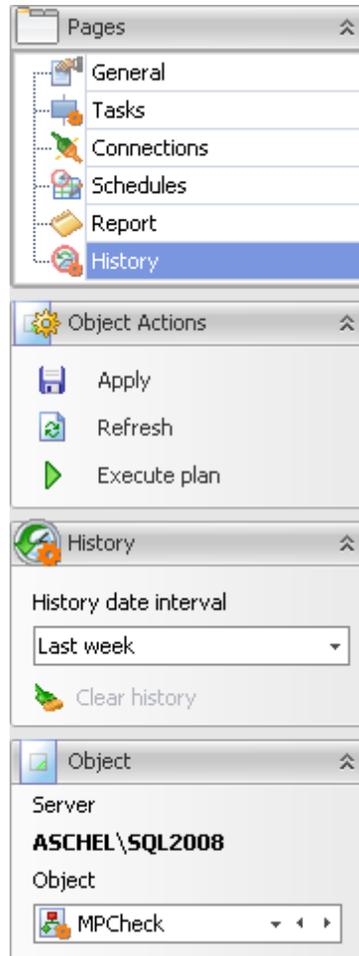
**Note:** To create a maintenance plan you need to run SQL Server Agent service on the server. Also make sure that the SQL Server Integration Services is installed on your computer - maintenance plans cannot be created or edited without it. In this case maintenance plans are opened in the Editor in the read-only mode.

The list of **Maintenance Plan Editor** tabs is displayed on the [navigation bar](#):

- [Setting general options](#)
- [Managing tasks](#)
- [Adding connections](#)
- [Managing schedules](#)
- [Setting reporting options](#)
- [Viewing maintenance plan history](#)

#### 4.1.1.1 Using Navigation bar

The **Maintenance Plan Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Maintenance Plan Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

-  apply changes
-  refresh the content of the active tab
-  execute maintenance plan

#### History

##### History date interval

Use the drop-down menu to define the period at which to display the maintenance plan history.

-  clear maintenance plan history

**Object  
Server**

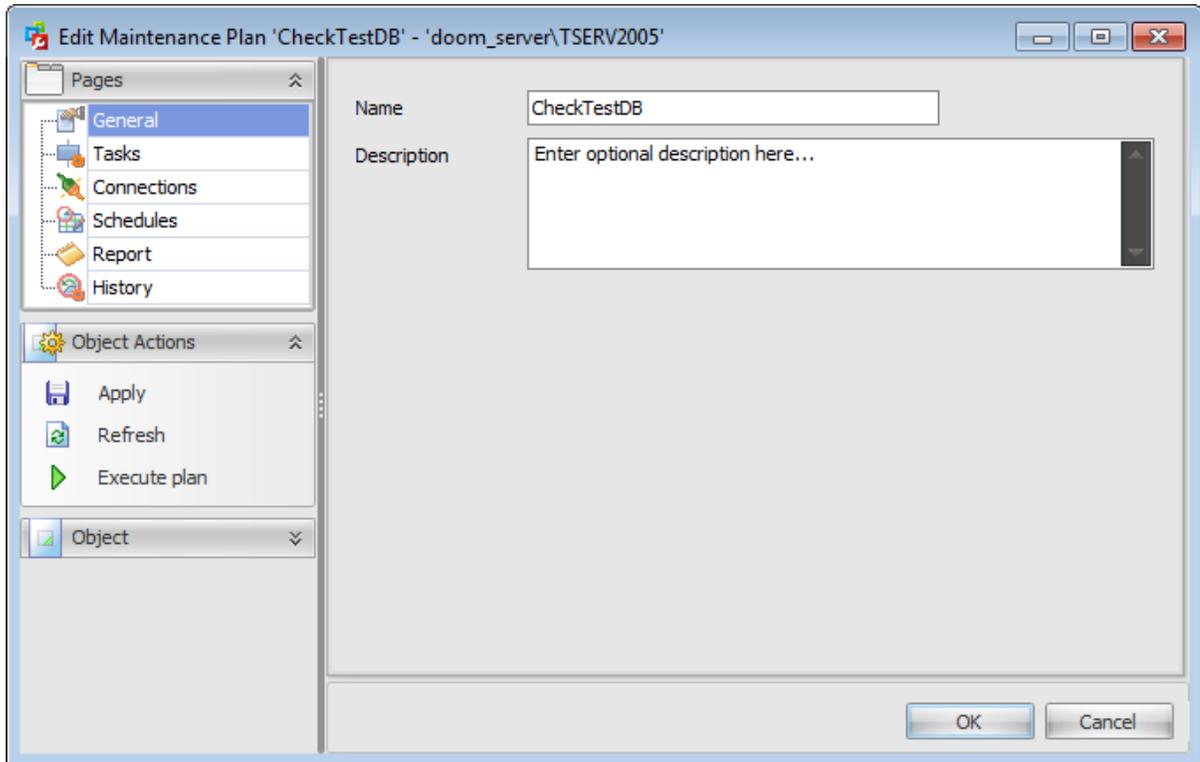
This field displays the name of the server where the edited object resides.

**Object**

This field displays the maintenance plan that is currently being edited. However, you can change the edited maintenance plan by using the drop-down list of maintenance plans or by clicking the arrow-left or arrow-right buttons.

#### 4.1.1.2 Setting general options

Use the **General** tab to set common properties.

**Name**

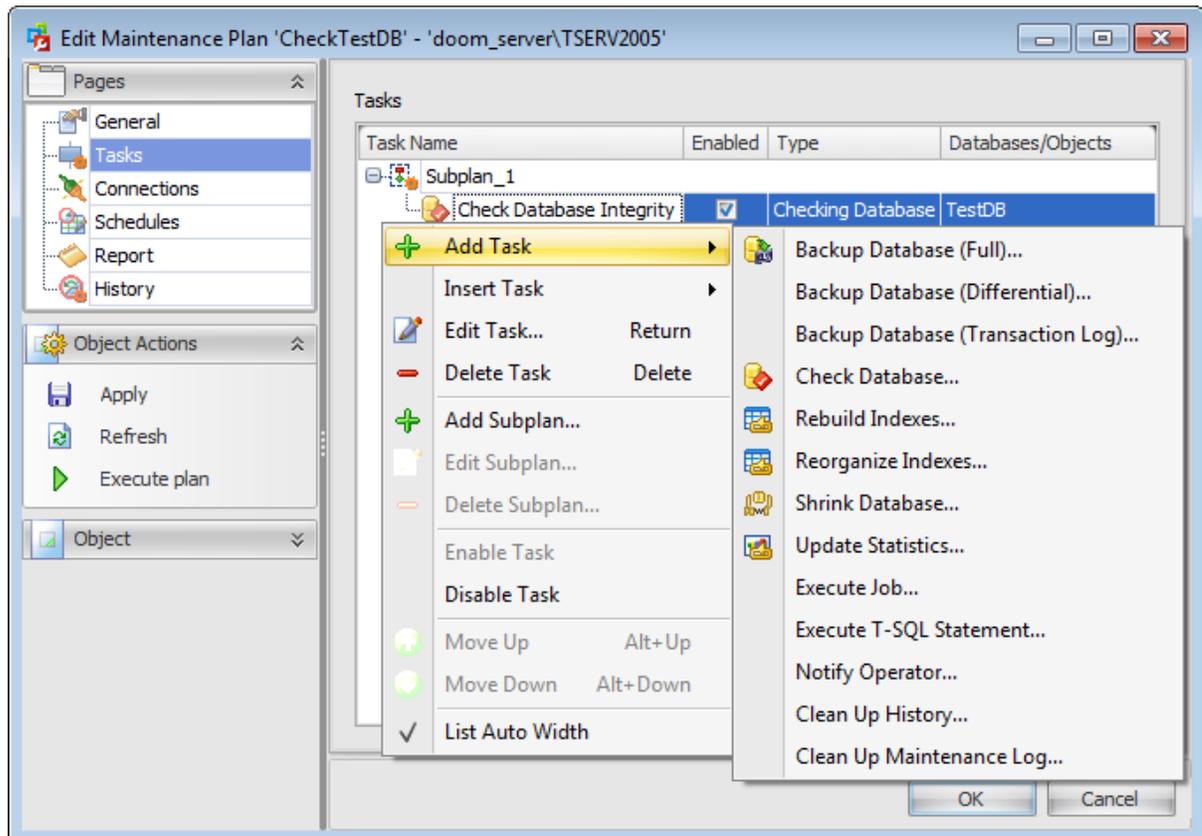
Type in a name for the new maintenance plan or edit an existing name.

**Description**

Enter the description for the new maintenance plan or edit an existing description.

#### 4.1.1.3 Managing tasks

Use this tab to create tasks grouped into subplans. Maintenance plans can contain multiple subplans, which are used to hold collections of tasks. Each subplan can be [scheduled](#) to run the subplan tasks at different times. Each maintenance plan must contain at least one subplan.

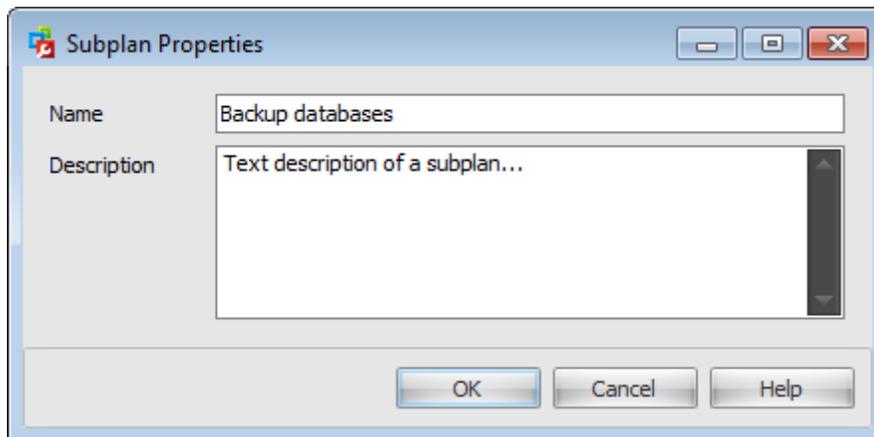


The grid in the window displays the following information about existing subplanned tasks: *Task Name, Enabled, Type, Databases/Objects*.

Using the context menu you can:

- Add a new task to the existing subplan
- Insert a new task before an existing one
- Edit selected task
- Delete selected task
- Add a new subplan
- Edit selected subplan
- Delete selected subplan
- Enable Task
- Disable Task
- Move the task up
- Move the task down

Editing subplan means setting its **Name** and **Description**.



The following tasks can be added to subplans:

- [Backup database](#)
- [Check database](#)
- [Rebuild indexes](#)
- [Reorganize indexes](#)
- [Shrink database](#)
- [Update statistics](#)
- [Execute job](#)
- [Execute T-SQL statement](#)
- [Notify operator](#)
- [Clean up history](#)
- [Clean up maintenance log](#)

## 4.1.1.3.1 Backup database

Use the **Back Up Database Task** dialog to add a backup task to the maintenance plan. This task allows you to perform full, differential, files and filegroups, and transaction log backups.

Use the **General** tab to set common properties for the backup task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the backup task name.

 **Enabled**

Check this option to enable task execution.

**Connection**

Specify which [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify which databases will be backed up during the task execution.

- *All databases*

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

- *System databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

If you have chosen only one database to backup then define whether to  **Back up entire database** or  **Back up files or filegroups**. In the latter case choose **Files or filegroups** from the drop-down list.

- Back up to the same backup device**

If you select this option database backups will be stored on the same media after performing the operation.

Then specify the **devices** to backup the database to. Note that all selected backup devices must be of the same type.

To learn more about managing list of devices see [Selecting backup devices](#).

#### **Append to device / Overwrite device**

Select whether the media should be appended or overwritten during the backup operation. By appending another backup set to existing media, the previous contents of the media remain intact, and the new backup is written after the end of the last backup on the media.

- Back up each database to a separate backup device**

If you select this option database backups will be stored on separate media sets after performing operation.

- To file**

Type in or use the ellipsis  button to specify a **Backup folder**.

- Create subfolder for each database**

Select to place each database in a subfolder.

- To Windows Azure blob**

This option is available for SQL Server 2014 only.

#### **Credential for Azure**

Use the drop-down list to select the [credential](#) for accessing the Windows Azure storage.

#### **Storage container**

Container name used for storing the Blob.

#### **URL prefix**

After defining all the Azure URL properties the template URL prefix used for accessing the backup file `https://{storage_account}.blob.core.windows.net/`

{*storage\_container*} will be formatted with the information provided. The database name and timestamp are added automatically as a blob name on uploading.

**Backup file extension**

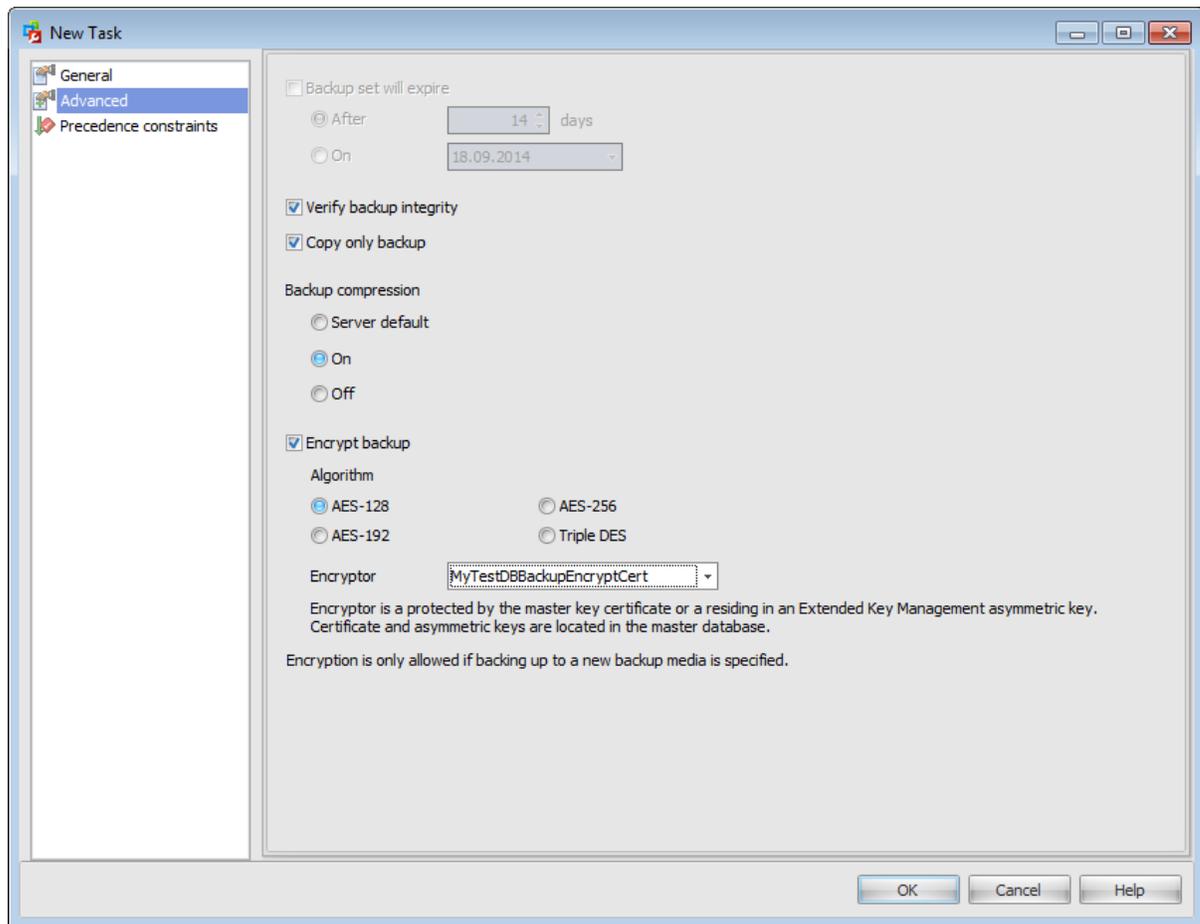
Enter the backup file extension. The default is *\*.bak*.

---

**See also:**[Check database](#)[Rebuild indexes](#)[Reorganize indexes](#)[Shrink database](#)[Update statistics](#)[Execute job](#)[Execute T-SQL statement](#)[Notify operator](#)[Clean up history](#)[Clean up maintenance log](#)[Precedence constraints](#)

## 4.1.1.3.1.1 Advanced

This section offers some additional options that may be useful for database backup operation.



**Backup set will expire**

You can set the expiration date by defining the number of days after that a backup will expire or by specifying an expiration date.

**Verify backup integrity**

Check this option to enable verifying created backup. Verifying a backup checks that the backup is intact physically, to ensure that all the files in the backup are readable and can be restored.

**Copy only backup**

This option specifies that the backup does not affect the normal sequence of backups. A copy-only backup does not affect the overall backup and restore procedures for the database. You can create a copy-only backup for any type of backup.

**Backup compression** (for SQL Server 2008 Enterprise and later only)

**Server default**

The backup compression is defined in compliance with the [SQL Server property](#)

**Compress backups by default.**

**On**

Check this option to compress the backup, regardless of the server-level default.

**Off**

Click to create an uncompressed backup, regardless of the server-level default.

 **Encrypt backup**

Check this option to enable the data encryption while creating a backup. This option is available for SQL Server 2014 only.

To encrypt during backup, you must specify an encryption algorithm, and an encryptor to secure the encryption key.

**Algorithm**

Select one of the encryption algorithms for a backup:

- AES-128
- AES-192
- AES-256
- Triple DES

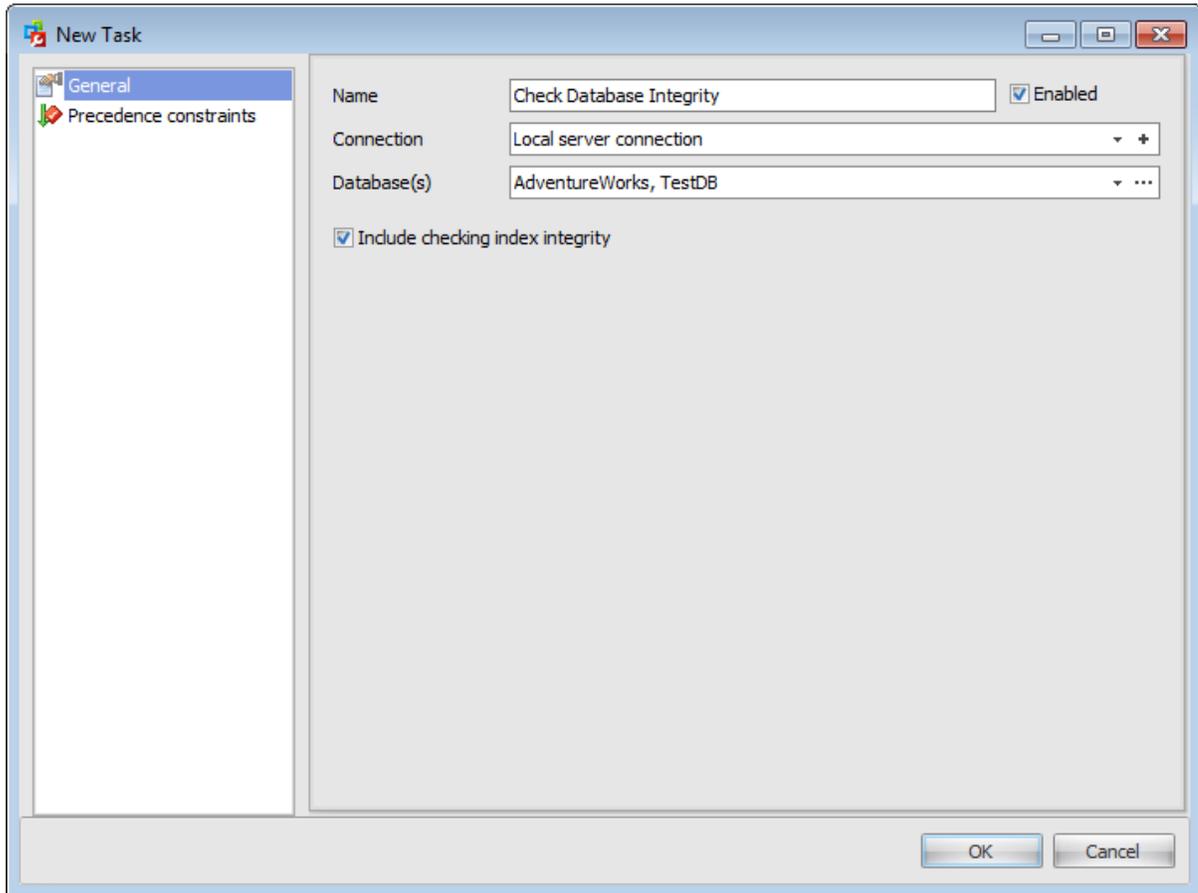
**Encryptor**

If you choose to encrypt you also have to specify the encryptor. Encryptors can be a master key certificate or an Extended Key Management asymmetric key from the master database. All available encryptors are listed in the combobox.

## 4.1.1.3.2 Check database

Use the **Check Database Integrity Task** dialog to check the allocation and structural integrity of user and system tables, and indexes in the database.

Use the **General** tab to set common properties for the check database task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the check database task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify which databases will be checked during the task execution.

- *All databases*

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

- *All system databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

**Include checking index integrity**

Check the integrity of all index pages as well as table data pages.

---

**See also:**

[Backup database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up history](#)

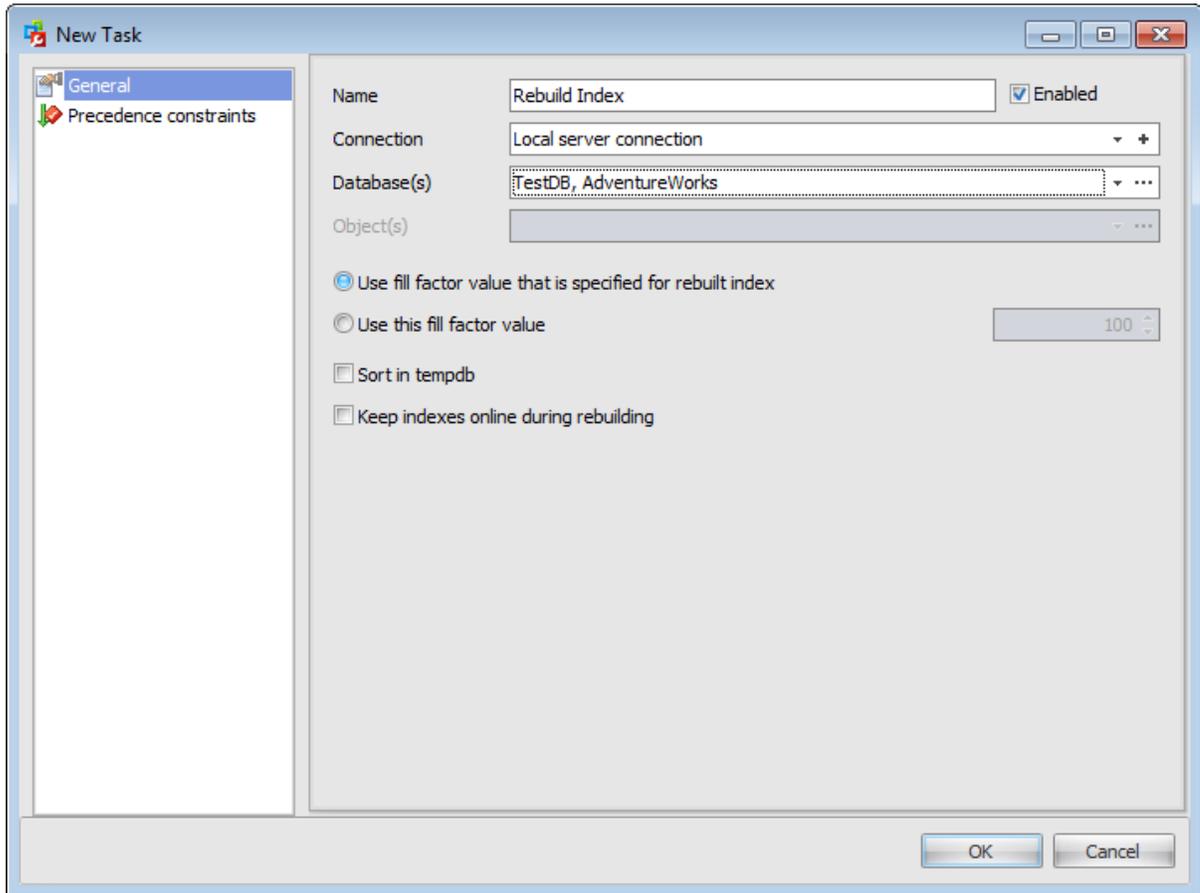
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.3 Rebuild indexes

Use the **Rebuild Index Task** dialog to re-create the indexes on the tables in the database with a new fill factor.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the rebuild indexes task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify which databases indexes will be rebuilt during the task execution.

- *All databases*

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

- *All system databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

### Object(s)

If you have chosen a single database in the **Database(s)** field then you are to define the objects (tables or views) which indexes you want to be rebuilt. The drop-down menu contains the following items: *All tables and views, All tables, All views, Select...* .

If you choose the latter item or click the ellipsis  button then define tables and views using the [Select Tables/Views dialog](#).

- Use fill factor value that is specified for rebuilt index**

Drop the indexes on the tables in the database and re-create them with the fill factor that was specified when the indexes were created.

- Use this fill factor value**

Drop the indexes on the tables in the database and re-create them with a new, automatically calculated fill factor, thereby reserving the specified amount of free space on the index pages. The higher the percentage, the more free space is reserved on the index pages, and the larger the index grows. Valid values are from 0 to 100.

- Sort in tempdb**

This option determines where the intermediate sort results, generated during index creation, are temporarily stored.

- Keep indexes online during rebuilding**

This option allows users to access the underlying table or clustered index data and any associated nonclustered indexes during index operations.

---

### See also:

[Backup database](#)

[Check database](#)

[Reorganize indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up history](#)

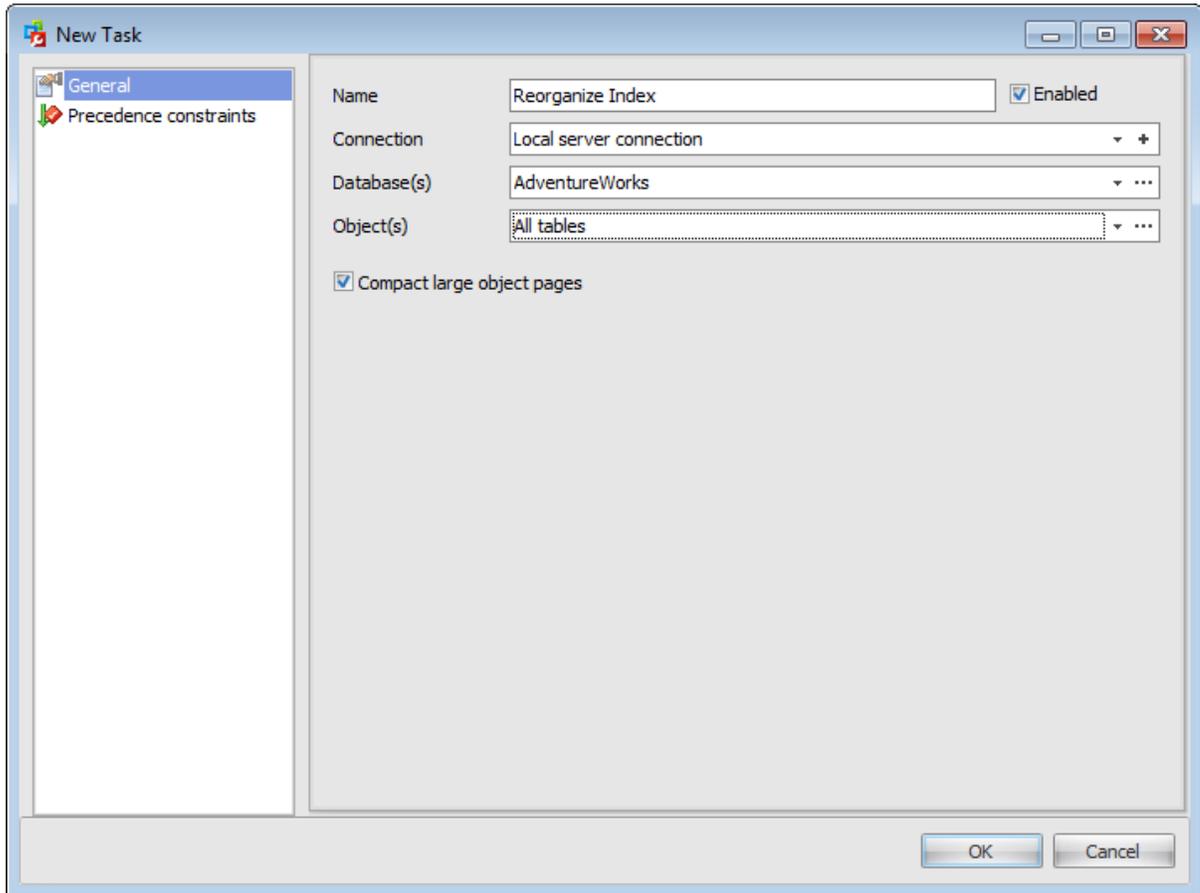
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.4 Reorganize indexes

Use the **Reorganize Index Task** dialog to move index pages into a more efficient search order.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the reorganize indexes task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify which databases indexes will be rebuilt during the task execution.

- *All databases*

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

- *All system databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

### Object(s)

If you have chosen a single database in the **Database(s)** field then you are to define the objects (tables or views) which indexes you want to be rebuilt. The drop-down menu contains the following items: *All tables and views*, *All tables*, *All views*, *Select...* .

If you choose the latter item or click the ellipsis  button then define tables and views using the [Select Tables/Views dialog](#).

**Compact large object pages**

Check this option to deallocate space for tables and views when possible.

---

### See also:

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up history](#)

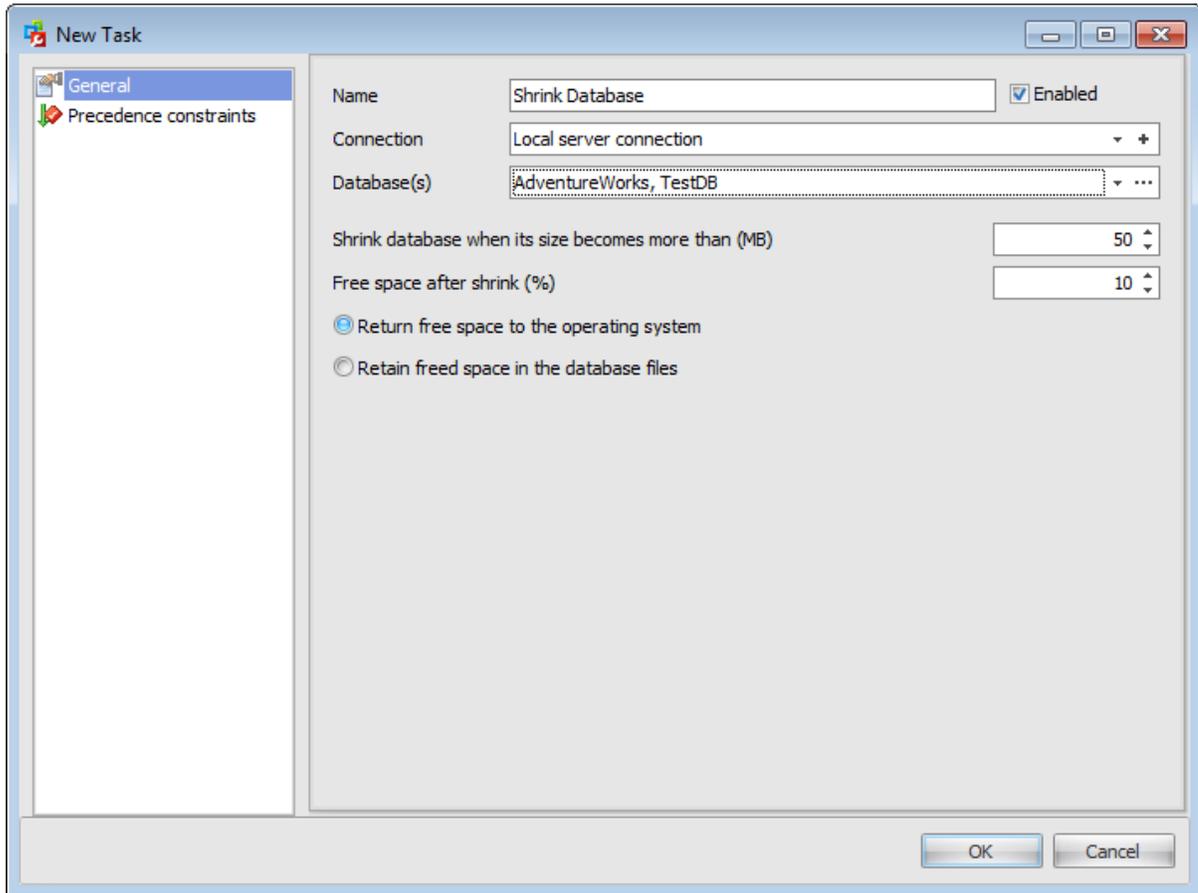
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.5 Shrink database

Use the **Shrink Database Task** dialog to create a task that attempts to reduce the size of the selected databases.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the shrink database task name.

 **Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify which databases will be shrunk during the task execution.

- *All databases*

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

- *All system databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

**Shrink database when its size becomes more than (MB)**

Specify the size in megabytes that causes the task to execute.

**Free space after shrink (%)**

Stop shrinking when free space in database files reaches this size.

Specify whether to  **Return free space to the operating system** or to  **Retain freed space in the database files** after the execution of the task is completed.

---

**See also:**

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up history](#)

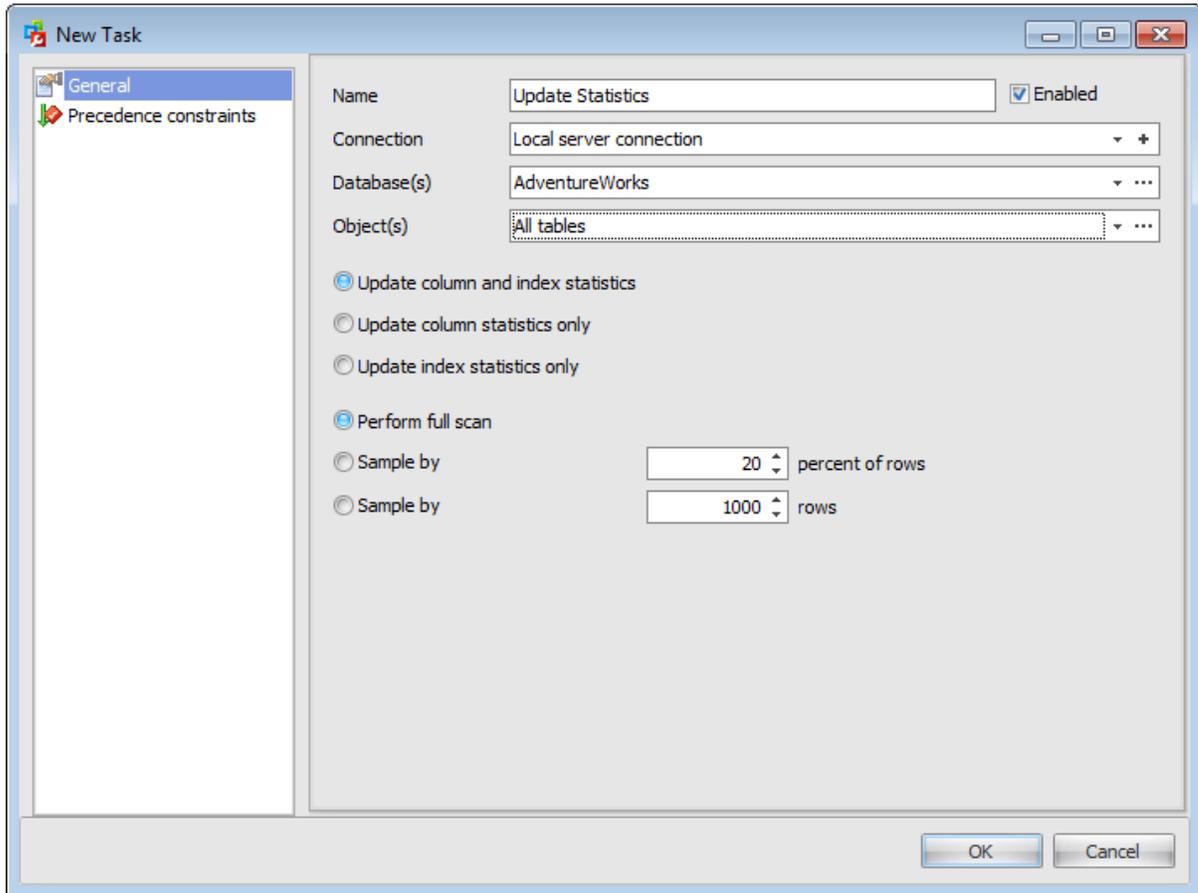
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.6 Update statistics

Use the **Update Statistics Task** dialog to update Microsoft SQL Server information about the data in the tables and indexes.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the update statistics task name.

 **Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Database(s)**

Use the drop-down list to specify the databases which will be affected by the task.

- *All databases*  
Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.
- *All system databases*

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

- *All user databases*

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

If you choose *Select databases...* item or click the ellipsis  button then define databases using the [Select databases dialog](#).

### Object(s)

If you have chosen a single database in the **Database(s)** field then you are to define the objects (tables or views) which statistics you want to be update. The drop-down menu contains the following items: *All tables and views, All tables, All views, Select...*

If you choose the latter item or click the ellipsis  button then define tables and views using the [Select Tables/Views dialog](#).

Specify whether you want to  **Update column and index statistics**,  **Update column statistics only** or  **Update index statistics only**.

Select the type of scan used to gather updated statistics.

- Perform full scan**

Read all rows in the table or view to gather the statistics.

### Sample by n percent of rows

Specify the percentage of the table or indexed view to sample when collecting statistics for larger tables or views.

### Sample by n rows

Specify the number of rows to sample when collecting statistics for larger tables or views.

---

#### See also:

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Shrink database](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up history](#)

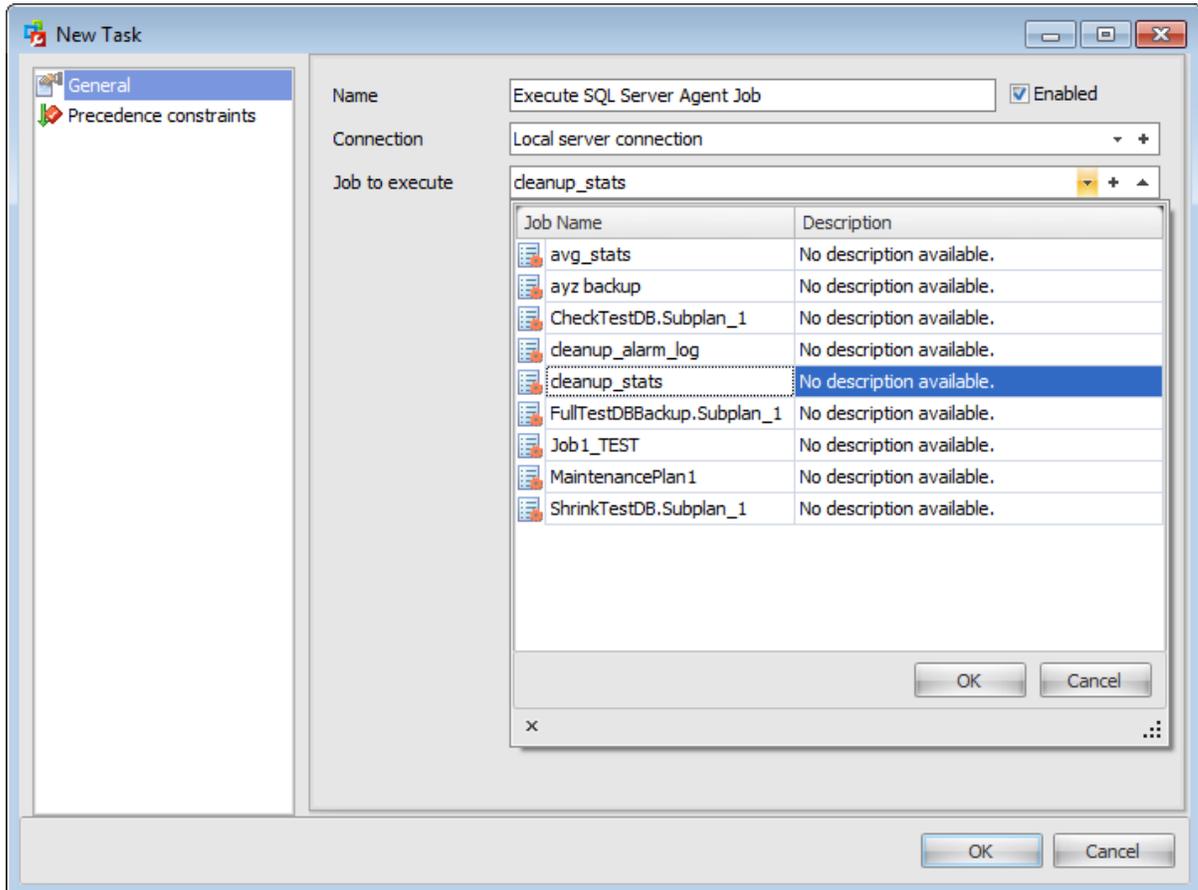
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.7 Execute job

Use the **Execute Job Task** dialog to execute Microsoft SQL Server Agent [jobs](#) within a maintenance plan. This option is not available if you have no SQL Server Agent jobs on the selected connection.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Job to execute**

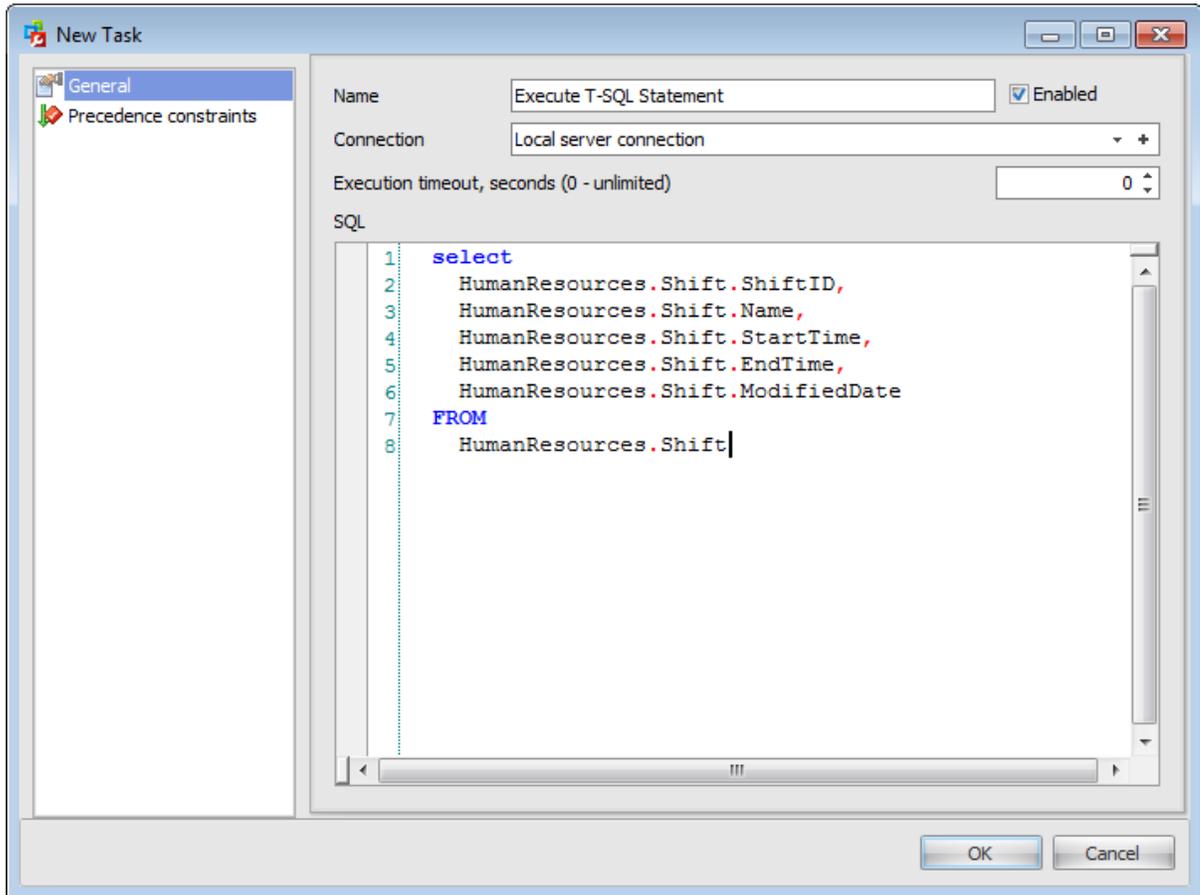
Use the drop-down list to specify the SQL Agent job to execute during the task. The list provides the Job name and Description to identify the jobs. To edit the selected job properties press the  $\wedge$  button.

You can also create a new job in the [Job Editor](#) by clicking the + button.

**See also:**[Backup database](#)[Check database](#)[Rebuild indexes](#)[Reorganize indexes](#)[Shrink database](#)[Update statistics](#)[Execute T-SQL statement](#)[Notify operator](#)[Clean up history](#)[Clean up maintenance log](#)[Precedence constraints](#)

## 4.1.1.3.8 Execute T-SQL statement

Use the **Execute T-SQL Statement Task** dialog to customize your maintenance plan by adding Transact-SQL statements of your choice to this maintenance plan. The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Execution timeout, seconds**

Define the time (in seconds) to wait for task completion before timing out (terminating task).

**SQL**

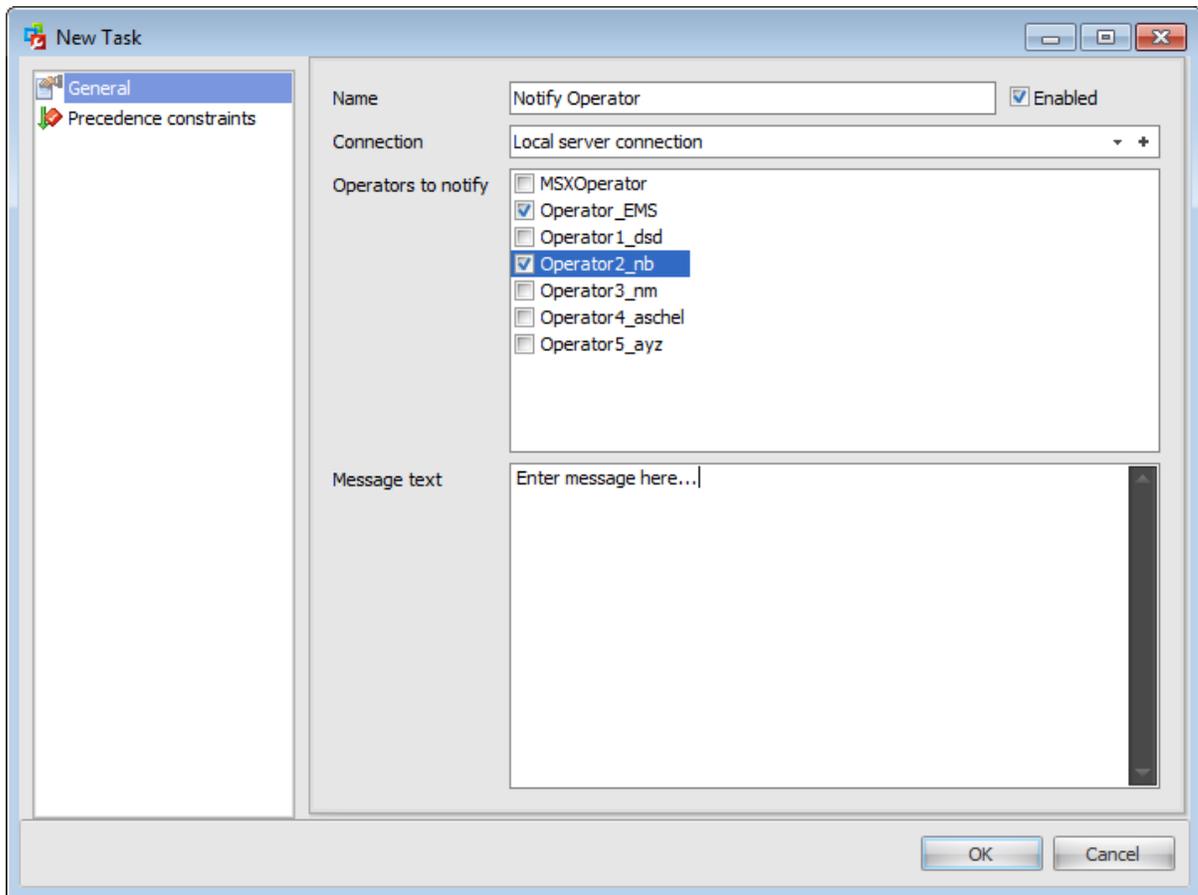
Specify in the Transact-SQL statement to execute.

**See also:**[Backup database](#)[Check database](#)[Rebuild indexes](#)[Reorganize indexes](#)[Shrink database](#)[Update statistics](#)[Execute job](#)[Notify operator](#)[Clean up history](#)[Clean up maintenance log](#)[Precedence constraints](#)

## 4.1.1.3.9 Notify operator

Use the **Notify Operators Task** dialog to add an automatic notification to this maintenance plan. To use this task you must have Database Mail enabled and properly configured with *msdb* as a Mail Host Database, and have a Microsoft SQL Server Agent [operator](#) with a valid e-mail address.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**Operators to notify**

Specify the recipient of the e-mail. Check the operator from the list of available [operators](#)

**Message subject**

Specify the text to include in the notification message subject line.

**Message text**

Enter the text to include in the notification message body.

---

**See also:**

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Clean up history](#)

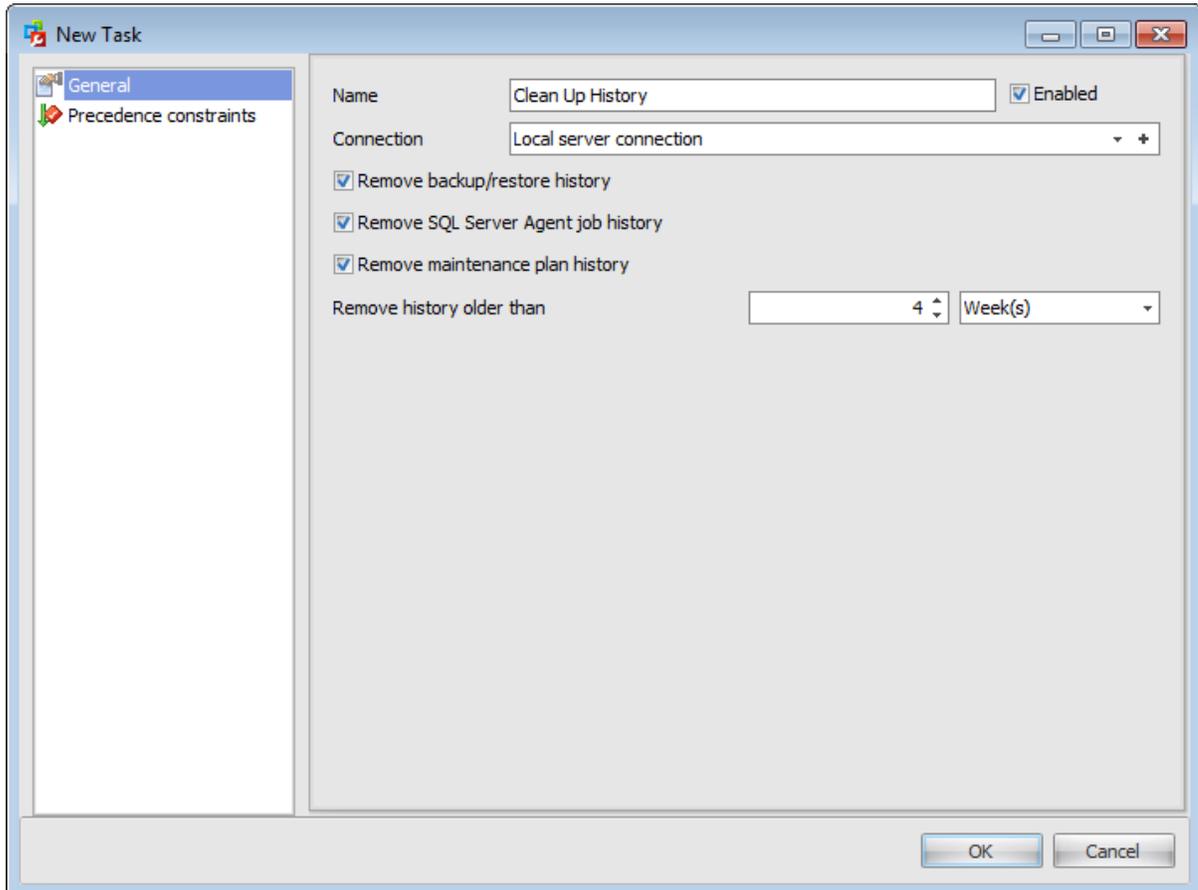
[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.10 Clean up history

Use the **Clean Up History Task** dialog to discard old historical information from tables in the *msdb* database. This task supports deleting backup and restore history, SQL Server Agent Job history, and maintenance plan history.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

**Name**

Enter the task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

 **Remove backup/restore history**

Checking this option causes the backup and restore history for the subject database to be deleted from the *msdb* database.

 **Remove SQL Server Agent job history**

Check this option for SQL Server Agent to remove entries that have been in the log longer than a specified length of time.

**Remove maintenance plan history**

If this option is checked then the [maintenance plan history](#) will be deleted from the *msdb* database.

**Remove history that is older than**

Specify age of items that you want to delete.

---

**See also:**

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

[Notify operator](#)

[Clean up maintenance log](#)

[Precedence constraints](#)

## 4.1.1.3.11 Clean up maintenance log

Use the **Clean Up Maintenance Log Task** dialog to remove old files related to maintenance plans, including text reports created by maintenance plans and database backup files.

The **General** tab allows you to set common properties for this task. To set relationships between tasks use the [Precedence constraints](#) tab.

The screenshot shows the 'New Task' dialog box with the following configuration:

- Name:** Maintenance Cleanup Task
- Enabled:**
- Connection:** Local server connection
- File types to remove:**
  - Backup files
  - Maintenance plan reports
- File location:**
  - Delete specific file
  - Delete files in specific folder by extension
    - Folder name:** C:\backup
    - File extension:** bak
    - Include first level subfolders
- File age:**
  - Delete only files older than
    - Value:** 4
    - Unit:** Week(s)

**Name**

Enter the task name.

**Enabled**

Check this option to enable task execution.

**Connection**

Specify which server [connection](#) should be used for task execution. You can choose among existing connections or create a new one.

**File types to remove** **Backup files**

Delete SQL Server backup files.

 **Maintenance plan reports**

Delete text reports of previously run maintenance plans.

#### File location

##### Delete specific file

Delete the specific file defined in the **File name** field.

##### File name

Type in or use the  button to specify the path to the file to be deleted.

##### Delete files in specific folder by extension

Delete all files with the specified extension in the specified folder. Use this to delete multiple files at once.

##### Folder name

Type in or use the  button to specify the path to the folder containing the files to be deleted.

##### File extension

Provide the file extension of the files to be deleted. If you have chosen  **Backup files** in the **File types to remove** group then the default is *\*.bak*. If you have chosen  **Maintenance plan reports** then default file extension is *\*.txt*.

##### Include first level subfolders

Delete files with the extension specified for File extension from first-level subfolders under Folder.

#### File age

##### Delete only files that are older than

Specify the minimum age of the files that you want to delete by providing a number, and unit of time (Hour, Day, Week, Month, or Year). Files older than the time frame specified will be deleted.

---

#### See also:

[Backup database](#)

[Check database](#)

[Rebuild indexes](#)

[Reorganize indexes](#)

[Shrink database](#)

[Update statistics](#)

[Execute job](#)

[Execute T-SQL statement](#)

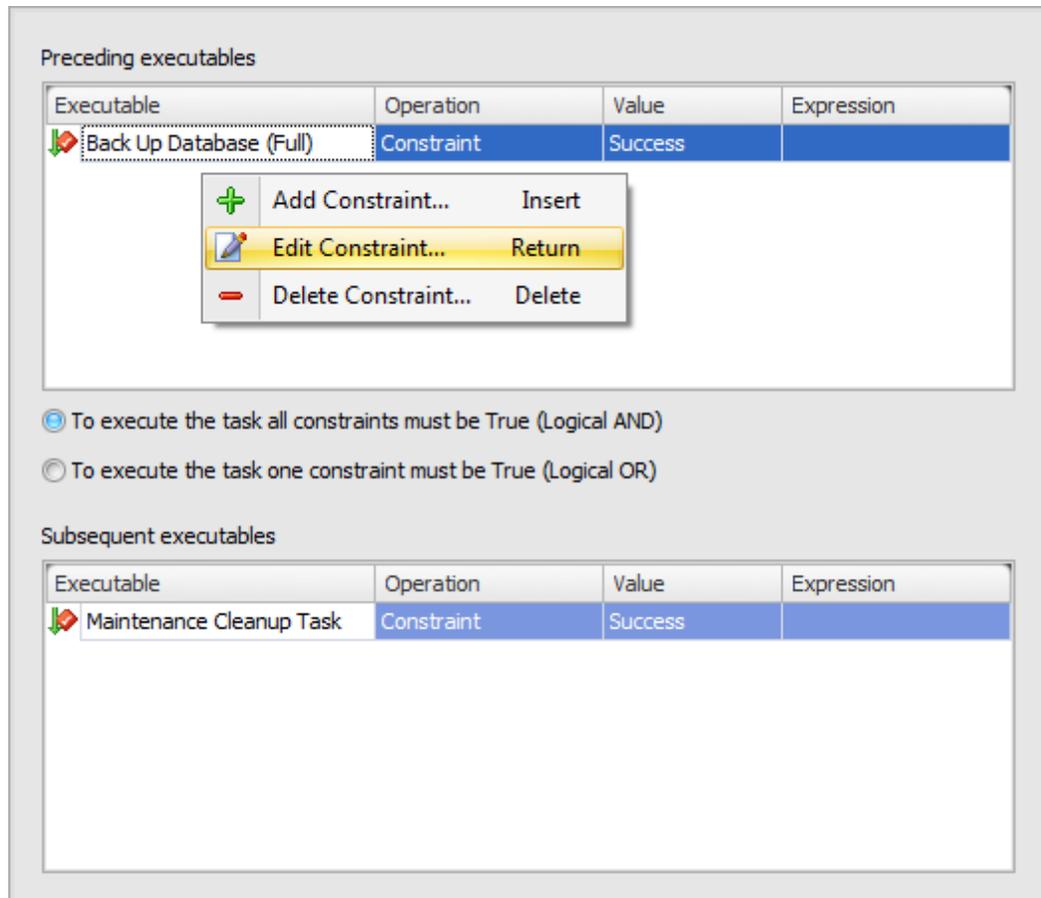
[Notify operator](#)

[Clean up history](#)

[Precedence constraints](#)

## 4.1.1.3.12 Precedence constraints

Use the **Precedence constraints** section to establish relationships between the tasks.

**Preceding executables**

In this area define tasks after which the configured maintenance task should run. After that choose whether  **To execute the task all constraints must be True (Logical AND)** or  **To execute the task one constraint must be True (Logical OR)**.

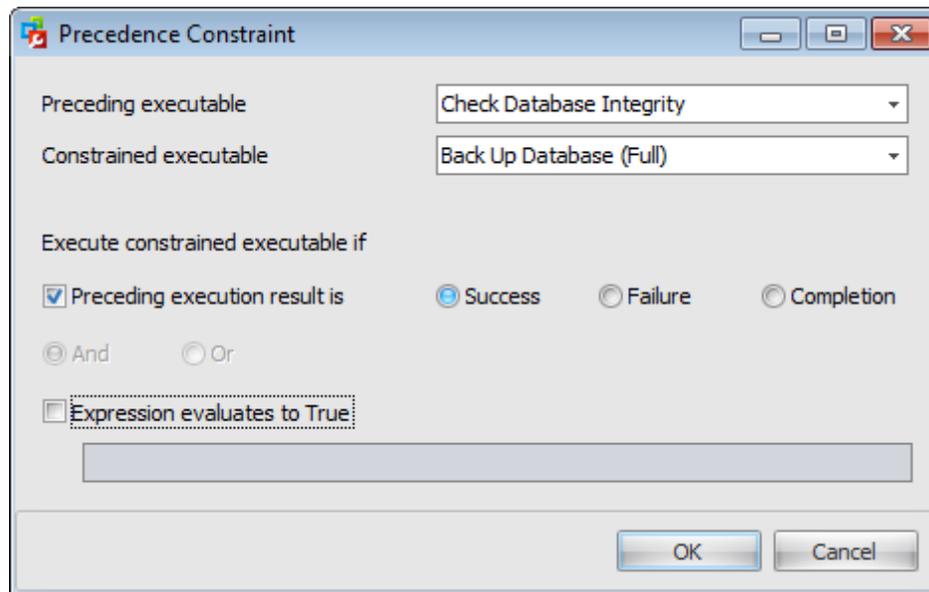
**Subsequent executables**

Here define tasks that should run after the configured maintenance task.

Using the context menu of both areas you can:

- + Add a new constraint
- Edit an existing constraint
- Delete constraint

Alternatively, you can use the 'Insert' button to create a constraint, 'Enter' button to edit constraint and 'Delete' button to drop a constraint.



The second task (the **Constrained executable**) executes only if the execution result of the first task (the **Preceding executable**) matches specified criteria. You can set this criteria by checking the  **Preceding execution result is** and/or  **Expression evaluates to True**.

The execution result may be specified as  **Success**,  **Failure** or  **Completion**. **Success** requires that the precedence executable succeed, **Failure** requires that the precedence executable fail, and **Completion** indicates that the constrained executable should run regardless of whether the precedence task succeeds or fails.

**Expression evaluates to True**

An expression that is evaluated to determine whether the constrained executable runs. If the expression evaluates to true, the constrained executable runs.

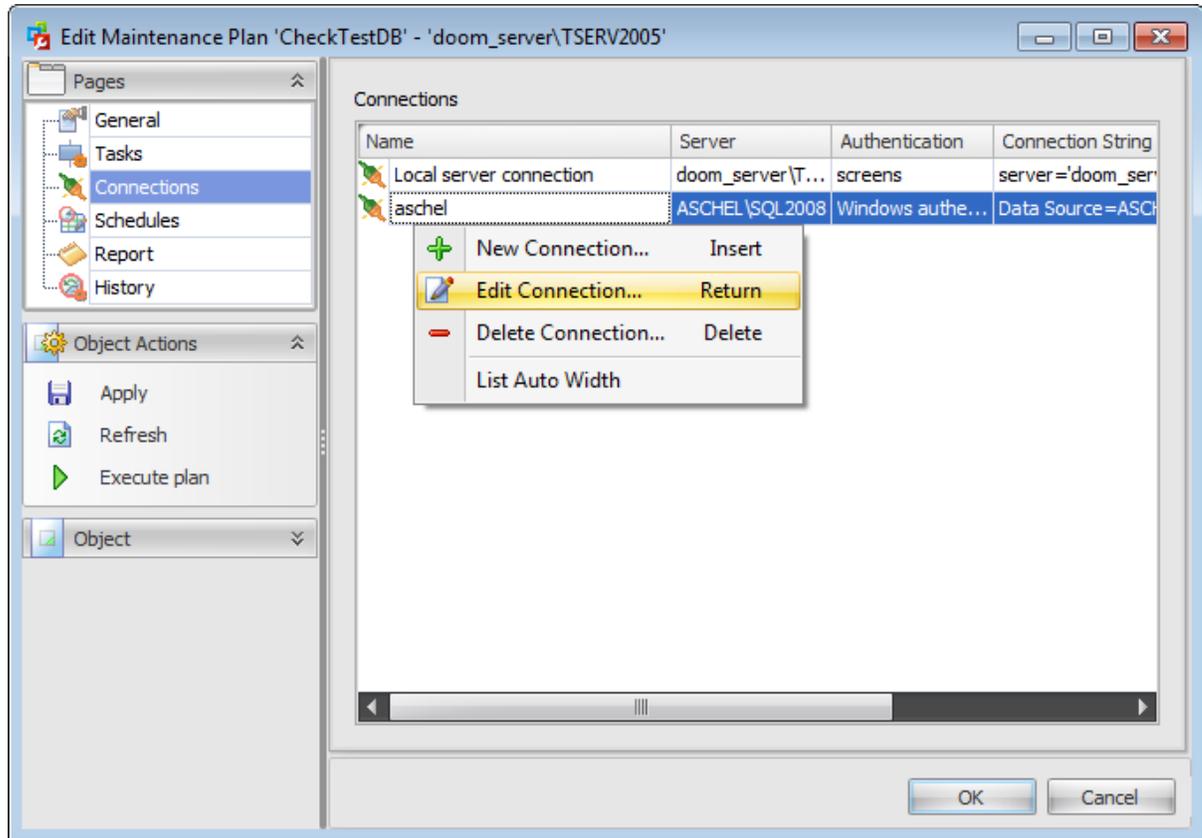
---

**See also:**

[Backup database](#)  
[Check database](#)  
[Rebuild indexes](#)  
[Reorganize indexes](#)  
[Shrink database](#)  
[Update statistics](#)  
[Execute job](#)  
[Execute T-SQL statement](#)  
[Notify operator](#)  
[Clean up history](#)  
[Clean up maintenance log](#)

#### 4.1.1.4 Adding connections

In this section you can view the list of available server connections.



The grid displays the following connection parameters: *Name*, *Server*, *Authentication*, *Connection String*.

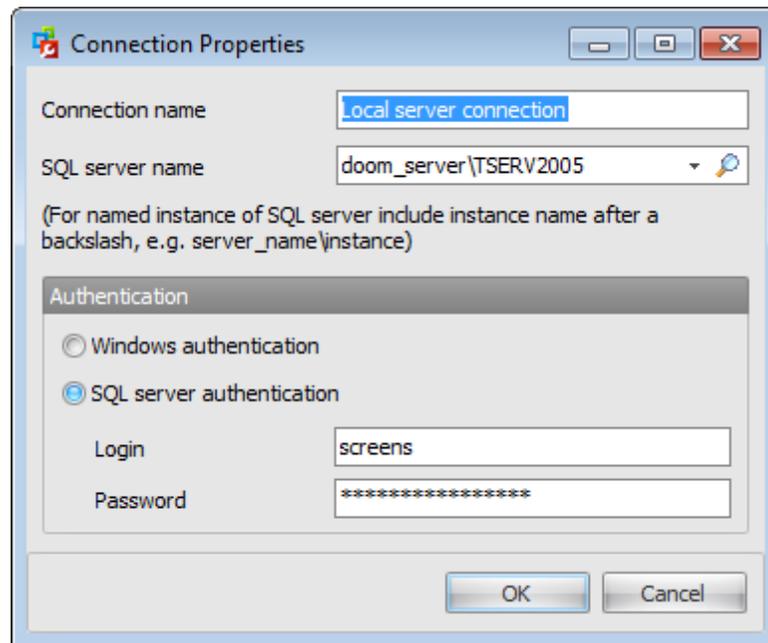
It always contains the *Local Server connection* item that cannot be dropped. By default *Local Server connection* is the connection to the server on which the maintenance plan is being created.

Using the context menu you can:

- + add a new connection;
- edit an existing connection;
- drop an existing connection.

Alternatively, you can use the '*Insert*' button to create a connection, '*Enter*' button to edit connection and '*Delete*' button to drop connection.

When you are adding or editing a connection the following window appears:



Here you can set common connection properties.

#### **Connection name**

Enter a name for the new connection.

#### **SQL Server name**

Select server to connect from the drop-down list. Click the  **Search** button to search for servers from a local network. In the list appeared choose the necessary server and press the 'OK' button.

**Note:** If server name is not displayed in this list it can be entered manually.

#### **Authentication**

Specify the type of Microsoft® SQL Server™ authentication to be used for the connection:  *Windows authentication* or  *SQL Server authentication*.

##### **Windows Authentication**

SQL Administrator identifies user by his domain name and password which are defined automatically according to his Windows domain user account.

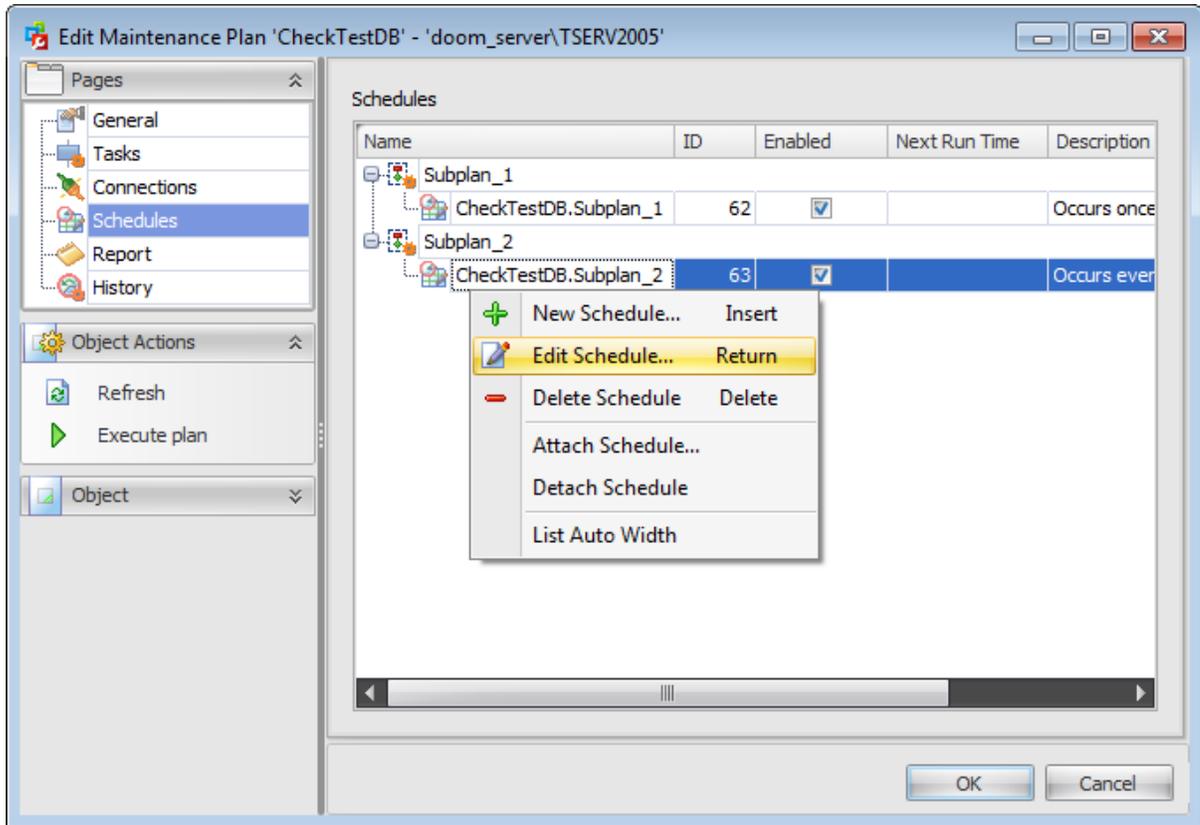
##### **SQL Server authentication**

SQL Administrator identifies user by his server login and password. Login and password management is accomplished by Login Manager.

If  *Server authentication* is selected as the authentication type, you should provide *authorization* settings: **Login** and **Password**. Analyzing them a system determines information and therefore object access permissions.

#### 4.1.1.5 Managing schedules

Use this section to schedule subplans defined in the [Tasks](#) section.



The grid in the window displays the following information about existing subplans schedules: *Name, ID, Enabled, Next Run Time, Description*.

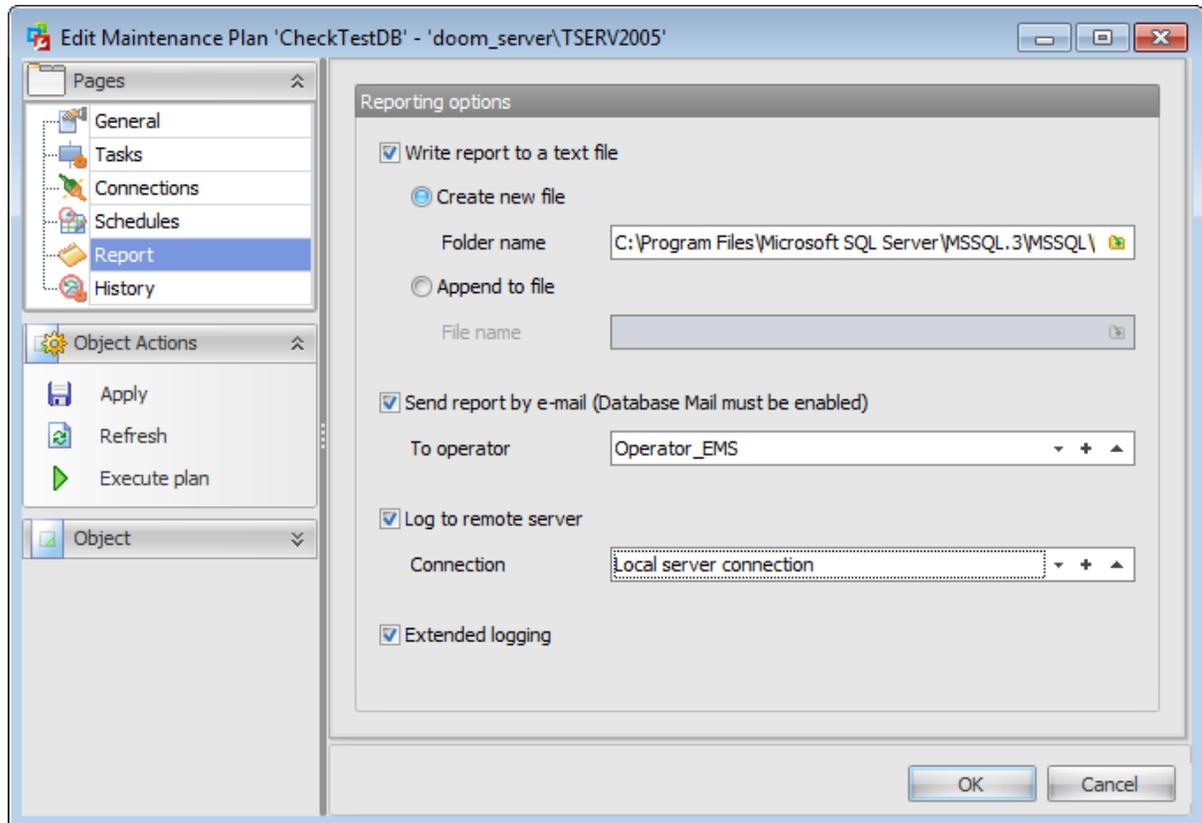
Using the context menu you can:

- [create](#) a new schedule and add it to the existing subplan
- edit selected schedule in the [Schedule Editor](#)
- delete selected schedule
- attach an available schedule to the subplan
- detach a schedule from the subplan

Alternatively, you can use the 'Insert' button to create a schedule, 'Enter' button to edit schedule(s) and 'Delete' button to drop schedule(s).

#### 4.1.1.6 Setting reporting options

Use this section to configure reporting options. Reports are logs that are generated when maintenance plans are executed.



##### **Write report to a text file**

Check this option if you want SQL Server to write a text file report.

##### **Create new file**

Create a new report file for each execution of the maintenance plan. By default, the report files are written to the computer hosting the instance of SQL Server that contains this maintenance plan, in the folder established as the default log folder during SQL Server setup.

To specify a different folder, enter the full path of the folder in the **Folder name** field, or click the **Open file** button and navigate to the desired folder.

##### **Append to file**

Append the report from each plan execution to the file specified in the **File name** field. You may also specify a file by clicking the **Open file** button and selecting a file from the dialog box.

##### **Send report by e-mail**

Select this option to transmit the outcome of a maintenance plan execution via e-mail. This option is only available if Database Mail is enabled and properly configured.

**To operator**

Select an operator from the available [operators](#) list who will be the recipient of the e-mail. This option is only available if mail is enabled and properly configured.

 **Log to remote server**

Check this option to log maintenance plan history to a remote server.

**Connection**

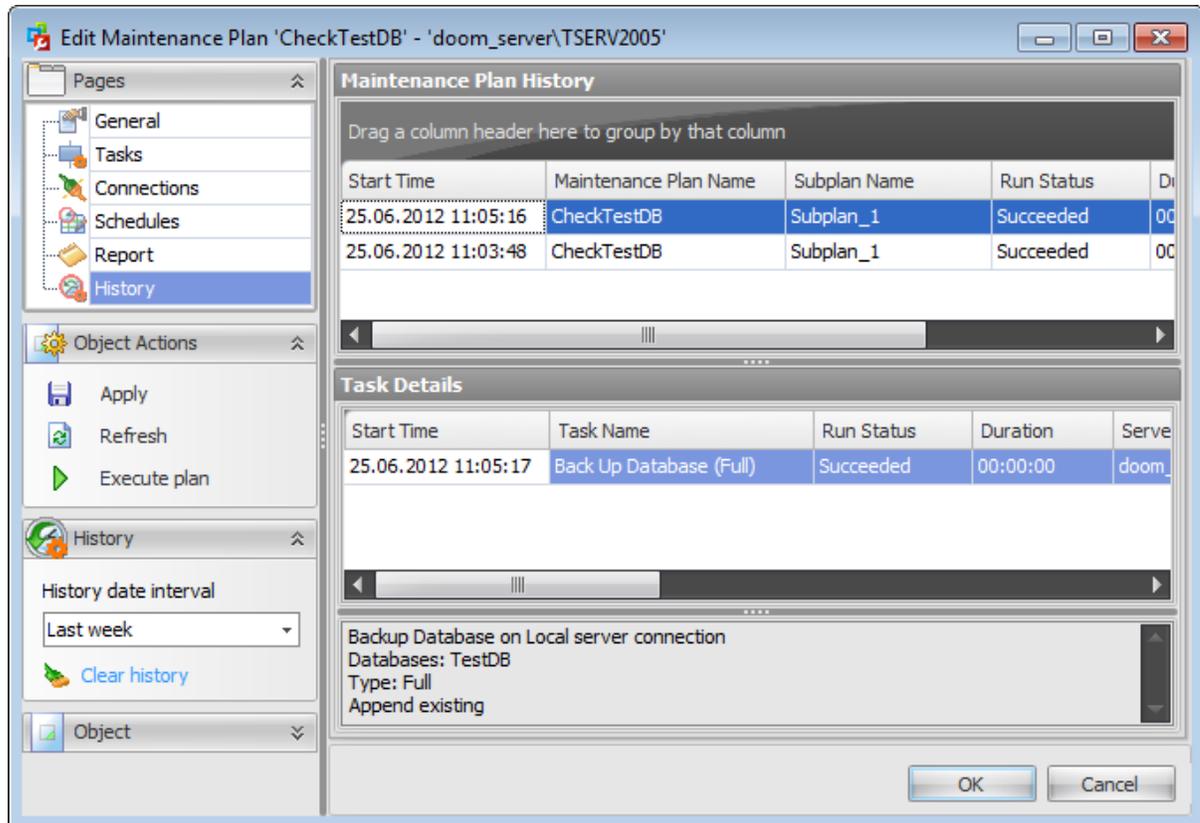
Specify the [connection](#) information to use when logging to a remote server.

 **Extended logging**

Select this option to include more information in the log. Including this option will increase the size of the stored maintenance plan history.

#### 4.1.1.7 Viewing maintenance plan history

In this section you can view maintenance plan [history](#). It is available only on editing a maintenance plan.



The list in the **Maintenance Plan History** area displays the history of maintenance plans as a grid with the following columns: *Maintenance Plan Name*, *Start Time*, *End Time*, *Subplan Name*, *Run Status*, *Logged Remotely*, *Plan ID*, *Duration*, *Source Server*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

When you choose a particular maintenance plan the **Task Details** area displays the following detailed information about the subplanned tasks: *Task name*, *Start Time*, *Run Status*, *Duration*, *Server*, *Error Number*, *Error Message*, *Command*, *Task Details*. If you select the step with the *Run Status* value "Failed" then in the lower area you will see the message informing about the reason of the occurred mistake. Otherwise you will see more information about the particular task.

Use the History panel on the navigation bar to specify the **History date interval**.

## 4.1.2 Maintenance plan History

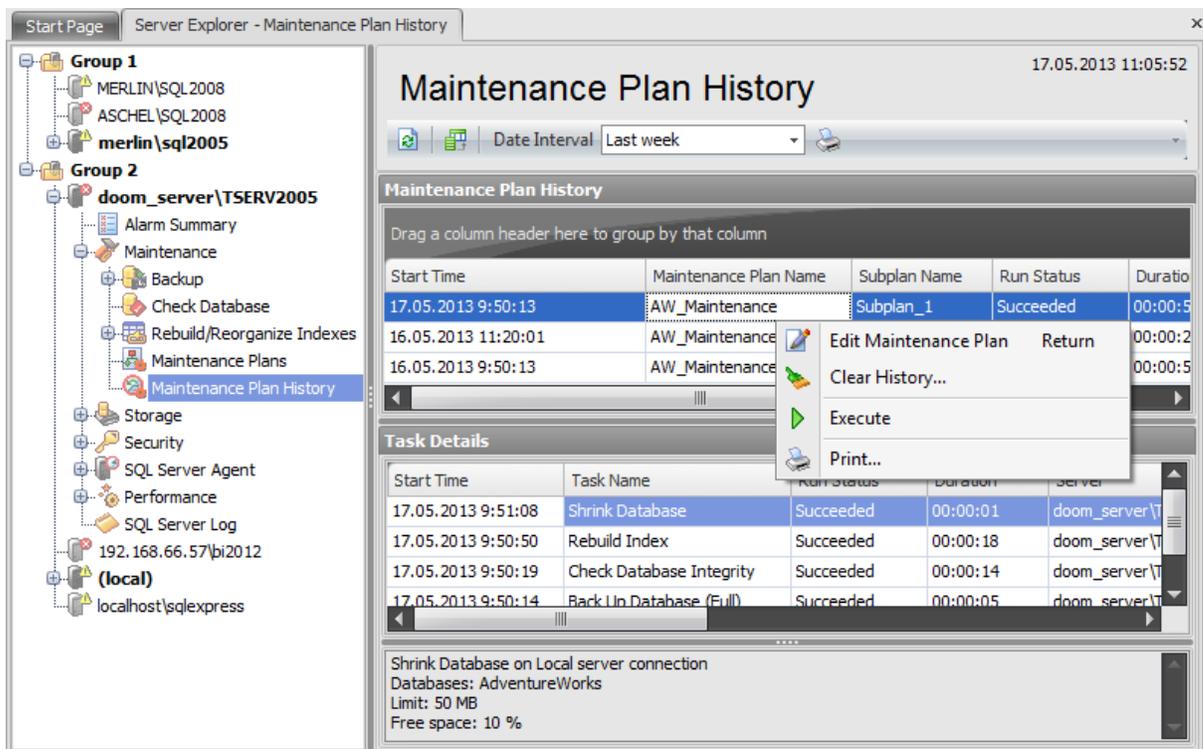
**Maintenance Plan History** section of the [Server Explorer](#) is used to view the history of maintenance plans executions.

Use the context menu or buttons on the work area toolbar to manage the list:

-  [edit](#) maintenance plan
-  clear maintenance plan history
-  execute maintenance plan
-  create and [print](#) the report

**Date Interval** - select an interval to display maintenance plans within

**Job** - select a maintenance plan from the drop-down list to display information about



The screenshot shows the 'Maintenance Plan History' window in SQL Server Enterprise Manager. The window title is 'Maintenance Plan History' and the date/time is '17.05.2013 11:05:52'. The 'Date Interval' is set to 'Last week'. The main area displays a table of maintenance plan history with the following data:

Start Time	Maintenance Plan Name	Subplan Name	Run Status	Duration
17.05.2013 9:50:13	AW_Maintenance	Subplan_1	Succeeded	00:00:5
16.05.2013 11:20:01	AW_Maintenance			00:00:2
16.05.2013 9:50:13	AW_Maintenance			00:00:5

A context menu is open over the first row, showing options: Edit Maintenance Plan, Clear History..., Execute, and Print... Below the table is the 'Task Details' section, which shows details for the selected task:

Start Time	Task Name	Run Status	Duration	Server
17.05.2013 9:51:08	Shrink Database	Succeeded	00:00:01	doom_server\T
17.05.2013 9:50:50	Rebuild Index	Succeeded	00:00:18	doom_server\T
17.05.2013 9:50:19	Check Database Integrity	Succeeded	00:00:14	doom_server\T
17.05.2013 9:50:14	Back Up Database (Full)	Succeeded	00:00:05	doom_server\T

At the bottom of the window, there is a summary for the 'Shrink Database on Local server connection' task:

Shrink Database on Local server connection  
 Databases: AdventureWorks  
 Limit: 50 MB  
 Free space: 10 %

The list in the **Maintenance Plan History** area displays the history of maintenance plans as a grid with the following columns: *Maintenance Plan Name*, *Start Time*, *End Time*, *Subplan Name*, *Run Status*, *Logged Remotely*, *Plan ID*, *Duration*, *Source Server*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

When you choose a particular maintenance plan the **Task Details** area displays the following detailed information about the subplanned tasks: *Task name*, *Start Time*, *Run Status*, *Duration*, *Server*, *Error Number*, *Error Message*, *Command*, *Task Details*. If you select the step with the *Run Status* value "Failed" then in the lower area you will see the message informing about the reason of the occurred mistake. Otherwise you will see more information about the particular task.



## 4.2 Backup database

**Backup** node is used for managing backups of the server.

To open a backup tab choose the **Maintenance | Backup** node of the [Server Explorer](#) tree.

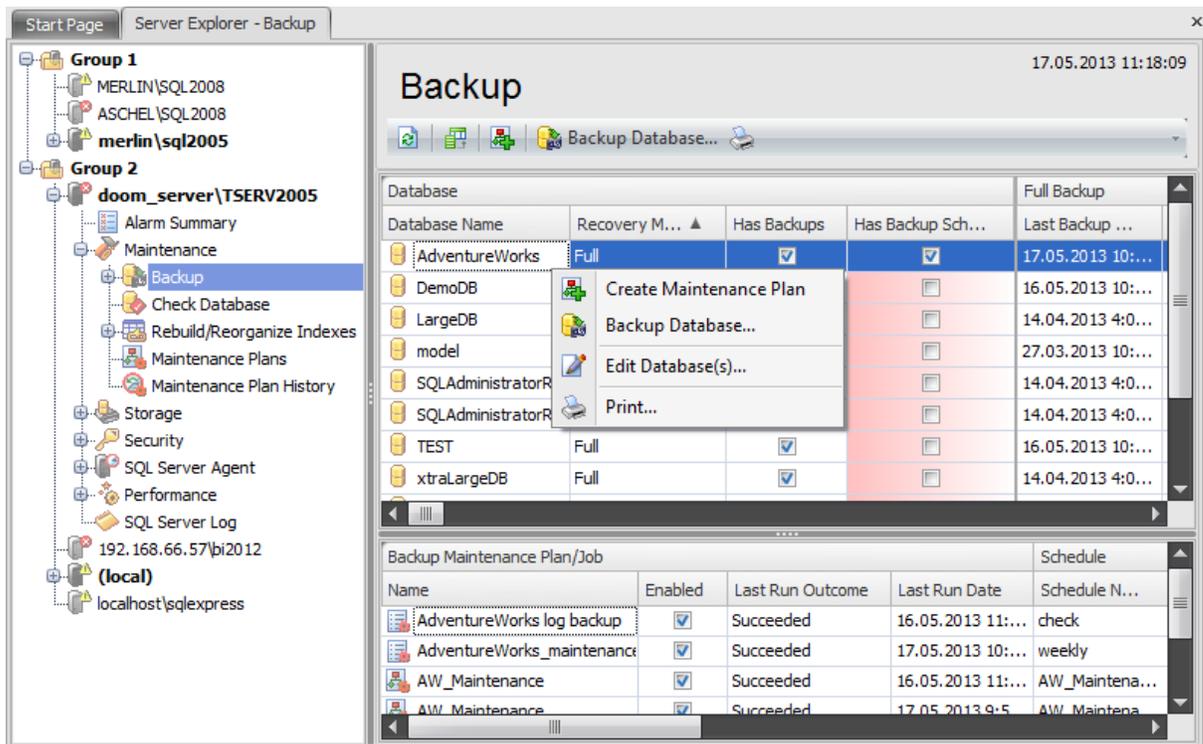
Use the context menu or buttons on the work area toolbar to manage backups:

 [create](#) a new maintenance plan

 [create](#) a database backup

 [edit database\(s\)](#)

 create and [print](#) the report



The upper area is the grid with the following columns: *Database Name*, *Recovery Model*, *Has Backups*, *Has Backup Schedules*, *Last Backup Date*, *Last Execution Outcome*, *Schedule Period during a Month* (It represents a grey axis with points which correspond to 6-hours intervals, days and weeks. Green lines show the planned time of the backup.). If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

When you choose a particular database the lower area displays the following detailed information about the backup database [maintenance plans](#): **Backup Maintenance Plan/Job**: *Name*, *Enabled*, *Last Run Outcome*, *Last Run Date*; **Schedule**: *Schedule Name*, *Enabled*, *Description*; **Task/Job Step**: *Name*, *Last Run Outcome*, *Last Run Duration*, *Last Run Message*.

To open a maintenance plan in the [Maintenance Plan Editor](#), double-click it in the grid or select it and press the 'Enter' button. Alternatively, you can right-click the maintenance

plan alias and select the **Edit Plans/Jobs** context menu item.

The Backup section has the following nodes:

- [Backup Database Wizard](#)
- [Backup History](#)
- [Failed backups](#)
- [Backup devices](#)

---

**See also:**

[Maintenance plans](#)

[Restore database](#)

[Attach database](#)

[Detach database](#)

[Check Database](#)

[Rebuild/Reorganize Indexes](#)

[Shrink database](#)

## 4.2.1 Backup Database Wizard

**Backup Database Wizard** allows you to perform the database backup operation on your SQL Server system (the *BACKUP DATABASE* Transact-SQL statement is issued).

This operation is used to create a backup copy of the entire database, transaction log or files/filegroups.

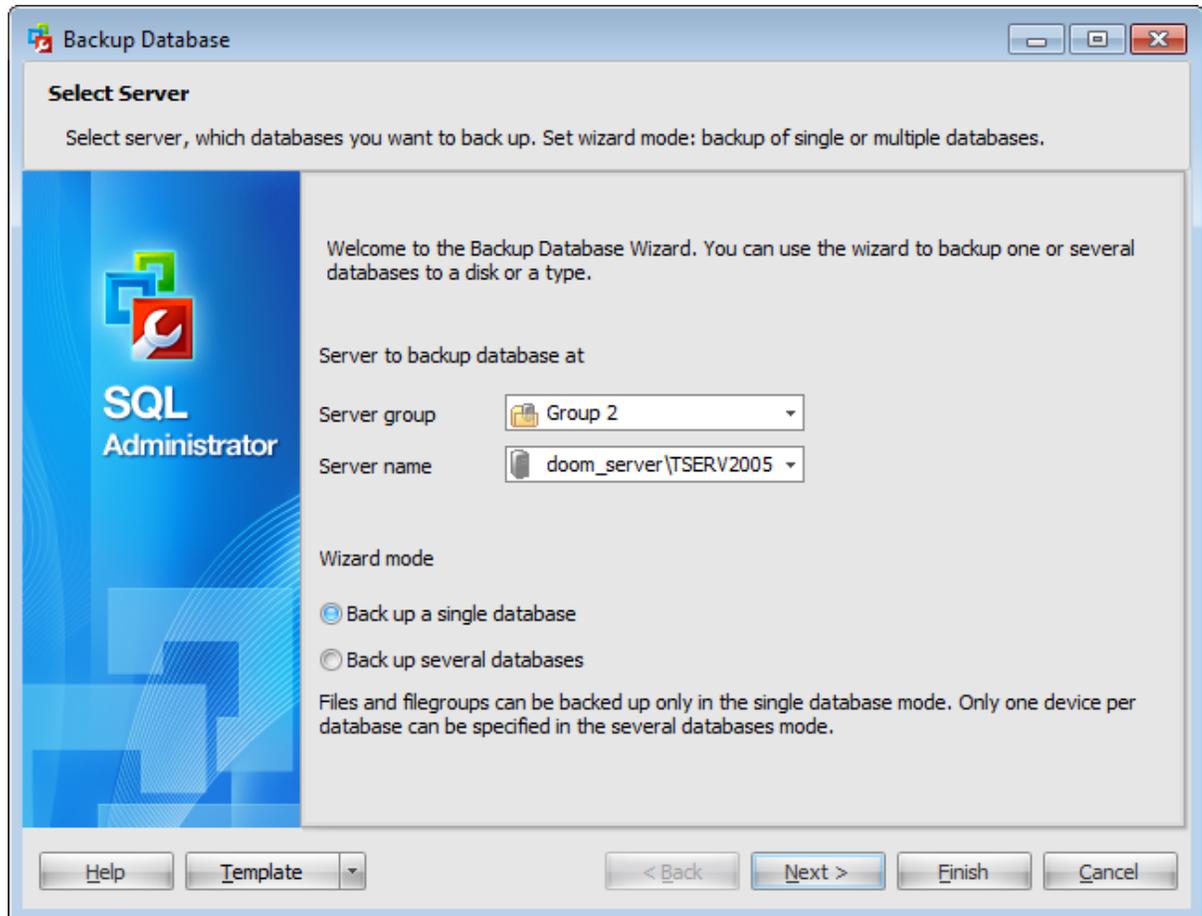
To run the wizard select the **Tools** |  **Backup Database...** [main menu](#) item or click  **Backup Database...** button on the [toolbar](#). Alternatively, you can right-click the database alias in the list of databases and select the **Backup Database...** item of the context menu.

The wizard consists of the following steps:

- [Selecting server](#)
- [Setting backup type](#)
- [Selecting database to backup](#)
- [Selecting database files to backup](#)
- [Selecting backup devices](#)
- [Setting backup media options](#)
- [Setting backup options](#)
- [Setting schedule options](#)
- [Running database backup](#)

#### 4.2.1.1 Selecting server

This step of the wizard allows you to specify the **server** containing the database for backup.



##### Server group

Use the drop-down list to select a server group where the server is stored.

##### Server name

Use the drop-down list to select the server where the database resides.

##### Wizard mode

###### Back up a single database

This mode allows you to choose only one database for backup.

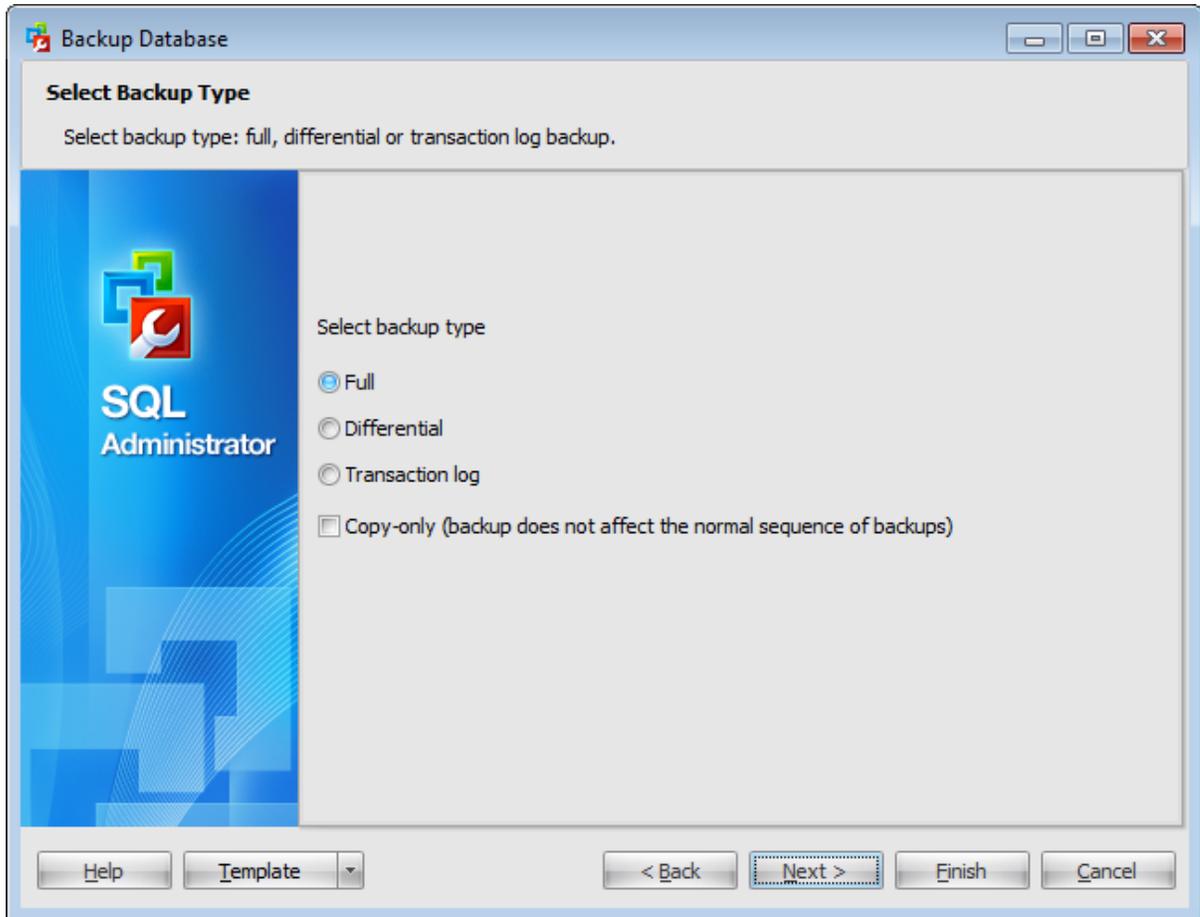
###### Back up several databases

This mode allows you to choose several databases for backup.

Click the **Next** button to proceed to the [Setting backup type](#) step of the wizard.

#### 4.2.1.2 Setting backup type

This step of the wizard allows you to select backup type.



#### Select backup type

Use this parameter to define the backup type:

- Full** (a full database backup which backs up the entire database including the transaction log)
- Differential** (a differential backup which records only the changes made to the data in the database after the last full database backup)
- Transaction log** (a sequence of log backups provided for a continuous chain of transaction information to support recovery forward from database, differential, or file backups)

**Copy only**

This option specifies that the backup does not affect the normal sequence of backups. A copy-only backup does not affect the overall backup and restore procedures for the database. You can create a copy-only backup for any type of backup.

Click the **Next** button to proceed to the [Selecting a single database to backup](#) step of the wizard (if you have selected  *Back up a single database* on the [first step](#)), or to the

[Selecting several databases to backup](#) step (if you have selected  *Back up a several databases* on the [first step](#)).

#### 4.2.1.3 Selecting a single database to backup

This step of the wizard allows you to specify the database for backup. This step appears only if  *Back up a single database* option was selected on the first step.

The screenshot shows the 'Backup Database' wizard window. The window title is 'Backup Database'. The main heading is 'Select Database' with the instruction 'Select a database to back up and set backup name and description.' On the left is a blue sidebar with the SQL Administrator logo. The main area contains four fields: 'Database name' (a drop-down menu showing 'AdventureWorks'), 'Recovery model' (a text field showing 'Full'), 'Backup name' (a text field showing 'AdventureWorks Full Backup'), and 'Backup description' (a text area showing 'AW backup'). At the bottom are buttons for 'Help', 'Template', '< Back', 'Next >', 'Finish', and 'Cancel'.

##### **Database name**

Use the drop-down list of all registered databases on the selected host to specify the database to backup.

##### **Recovery model**

This label displays the recovery model defined for the database.

##### **Backup name**

Enter a name for the current backup.

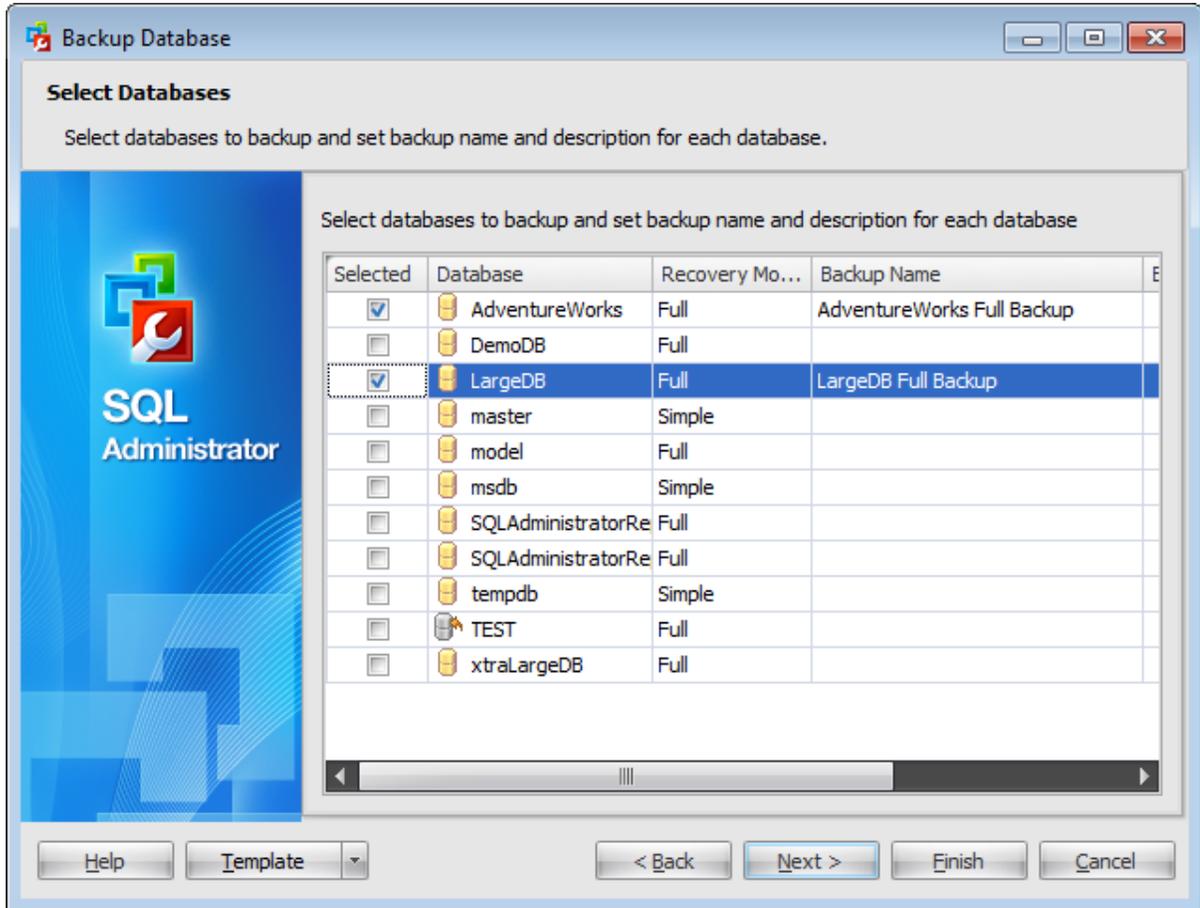
##### **Backup description**

Enter any optional text to describe the backup operation.

Click the **Next** button to proceed to the [Selecting database files to backup](#) step of the wizard.

#### 4.2.1.4 Selecting several databases to backup

This step of the wizard allows you to specify databases for backup. This step appears only if  *Back up several databases* option was selected on the [first step](#).



To choose a database to backup select it in the list and set the respective flag in the *Selected* column.

When the database is selected a *Backup name* appears in the corresponding column. You can modify or accept it.

The *Recovery model* defined for the selected database is displayed in a corresponding column.

The *Backup description* column allows you to enter any optional text to describe the backup operation.

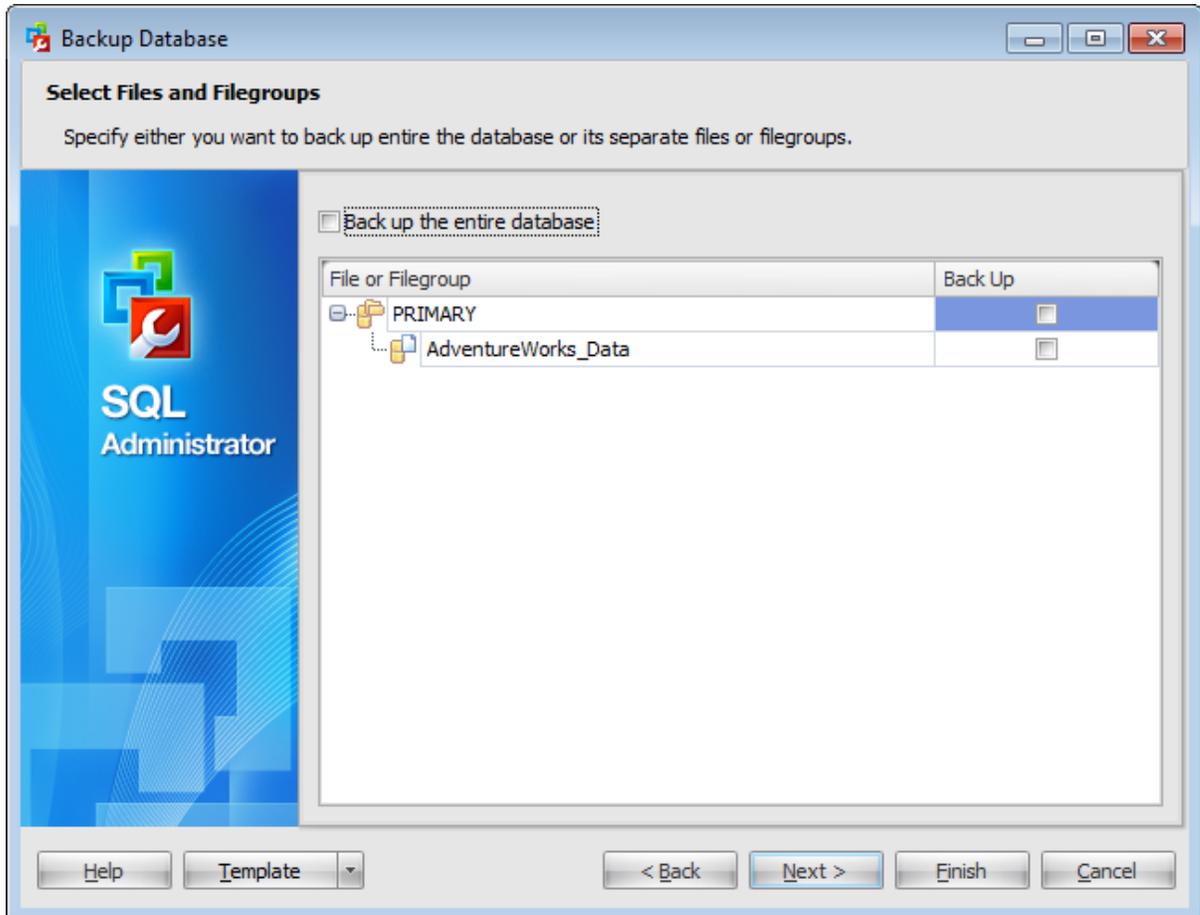
Click the **Next** button to proceed to the [Specifying media sets](#) step of the wizard.

#### 4.2.1.5 Selecting database files to backup

This step allows you to **select files and filegroups** of the database to backup. This step appears only if  *Back up a single database* option was selected on the first step.

To add a file to the backup, select it in the list and set the respective flag in the **Back Up** column.

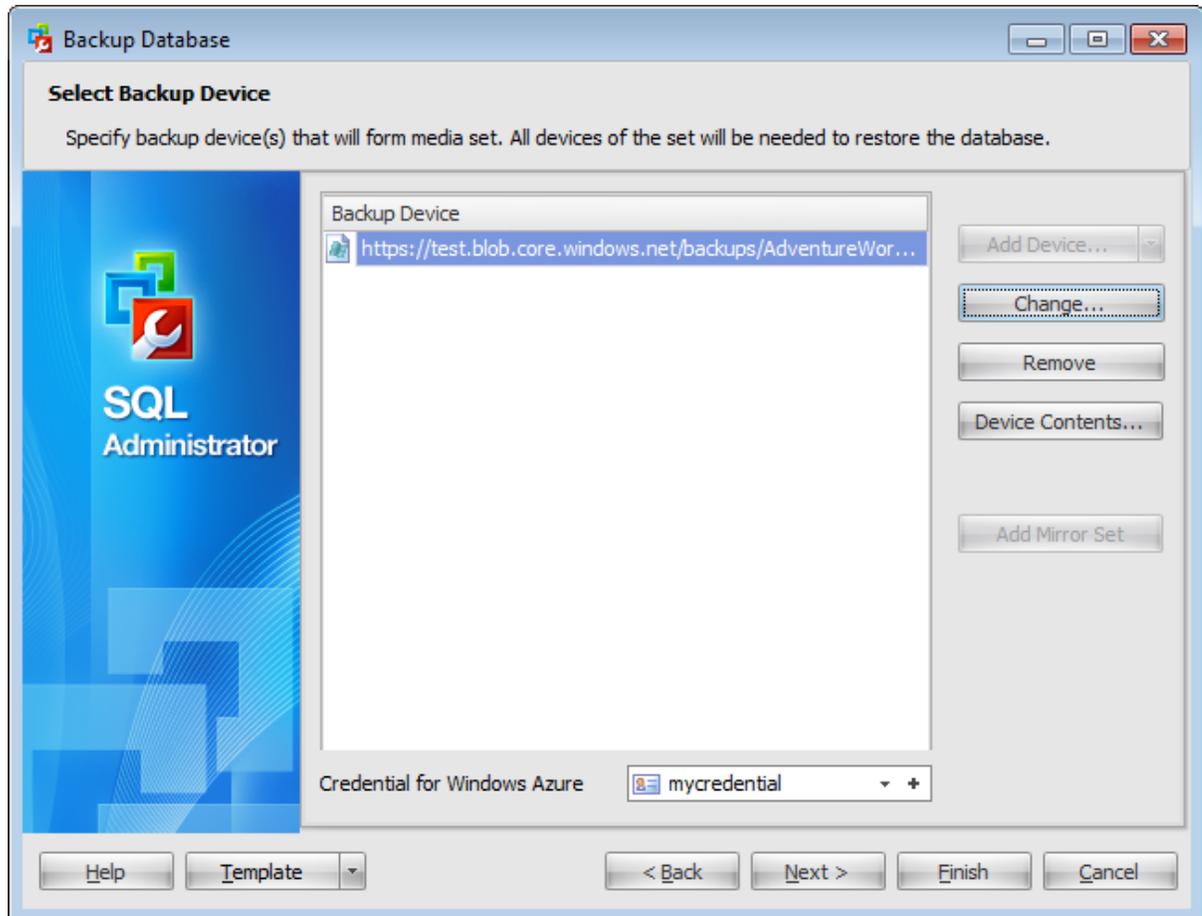
To add all files to the backup, check the  **Backup entire database** option.



Click the **Next** button to proceed to the [Selecting backup devices](#) step of the wizard.

#### 4.2.1.6 Selecting backup devices

Use this step of the wizard to specify the **devices** to backup the database to. Note that all the selected backup devices must be of the same type.



To add a device, click the **Add Device...** button and select a backup device from the menu.

To replace the selected device with another one select it in the Backup Device list and click the **Change...** button.  
select a backup device from the menu.

To remove a device from the list, select it and click the **Remove** button.

To browse the selected device, click the **Device Contents...** button.

Using the **Add Device...** button you can also create a new device and specify its properties using the [New Logical Backup Device dialog](#), add a file, a tape.

**Disk backup devices** are files on hard disks or other disk storage media the same as regular operating system files. Referring to a disk backup device is the same as referring to any other operating system file. You can define disk backup devices on a local disk of a

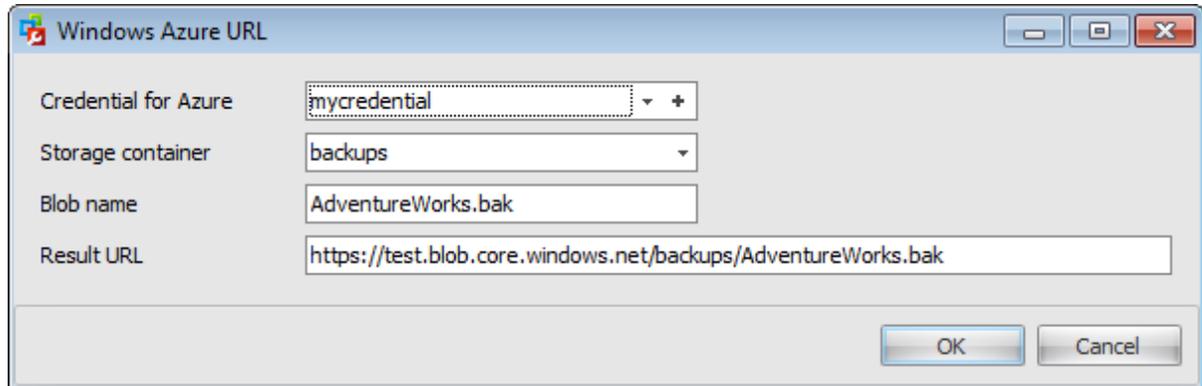
server, or on a remote disk on a shared network resource, and they can be as large or as small as needed. The maximum file size is the free disk space available on the disk.

**Tape backup devices** are used in the same way as disk devices, but the tape device must be connected physically to the computer running an instance of SQL Server, and if a tape backup device is filled during the backup operation, but more data still needs to be written, SQL Server prompts for a new tape and continues the backup operation. To back up SQL Server data to tape, use a tape backup device or tape drive supported by the Microsoft Windows platform. Additionally, use only the tapes recommended by the drive manufacturer for the specific tape drive.

### Add Mirror Set

Mirroring a media set increases backup reliability by reducing the impact of backup-device malfunctions. Media mirroring is a property of the media set. A mirrored media set consists of multiple copies (mirrors) of the media set. A mirrored media set contains one or more media families, each of which corresponds to a backup device. A mirrored media set possesses from two to four mirrors. Each mirror contains all the media families in the media set. The mirrors require the same number of devices, one per media family. Each mirror requires a separate backup device for each media family. For example, a mirrored media set that consists of four media families with three mirrors requires twelve backup devices. All of these devices must be equivalent. For example, tape drives that have the same model number from the same manufacturer.

For backups performed on the SQL Server 2014 adding *Windows Azure URL* is possible as well.



Use the Windows Azure URL dialog to form the URL where the database backup will be uploaded.

### Credential for Azure

Use the drop-down list to select the [credential](#) for accessing the Windows Azure storage.

### Storage container

Container name used for storing the Blob.

### Blob name

Backup file name.

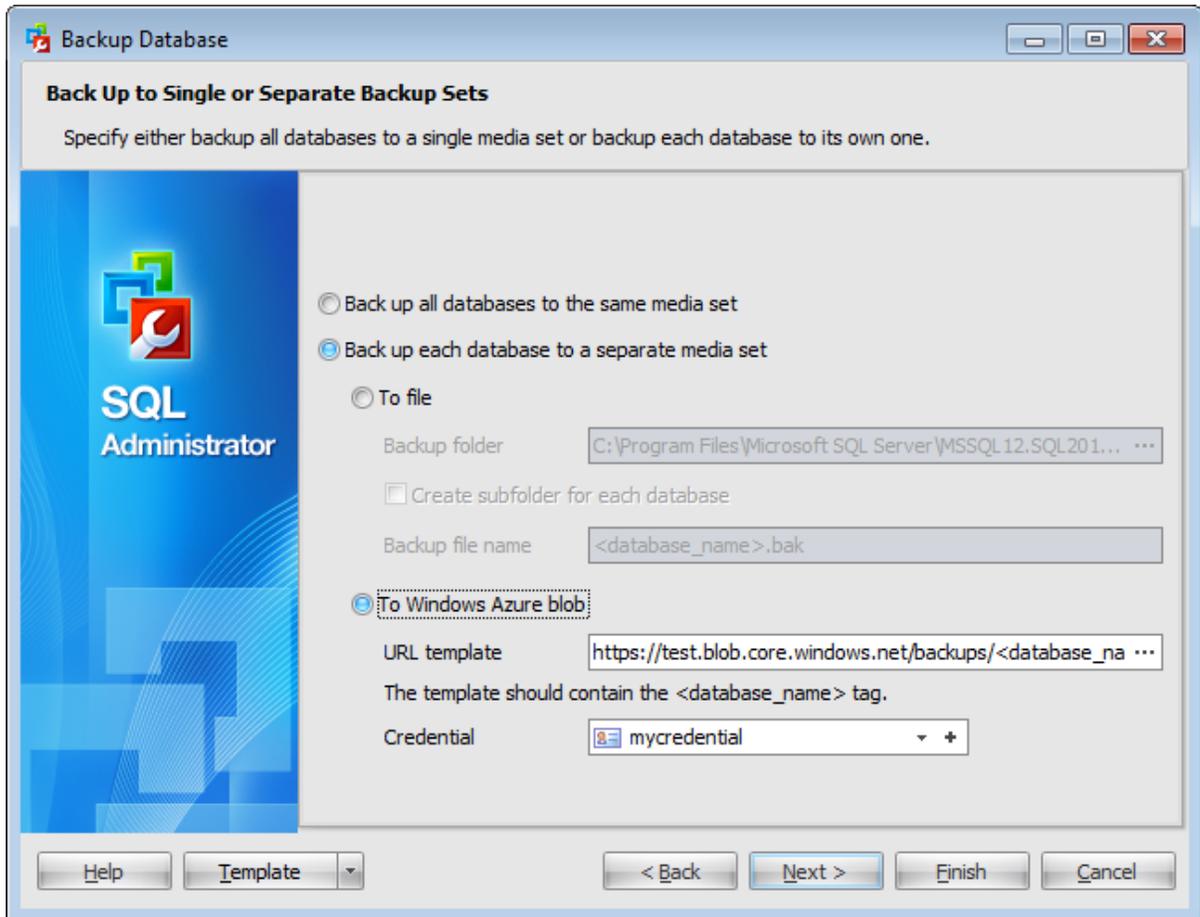
### Result URL

After defining all the Azure URL properties the template URL used for accessing the backup file `https://{storage_account}.blob.core.windows.net/{storage_container}/{blob_name}` will be formatted with the information provided.

Click the **Next** button to proceed to the [Setting backup media options](#) step of the wizard.

#### 4.2.1.7 Specifying backup sets

Use this step of the wizard to specify whether all databases will be backed up to the same media or each database will be backed up to a separate media set. This step appears only if  *Back up several databases* option was selected on the [first step](#).



**Back up all databases to the same media set**

If you select this option database backups will be stored on the same media after performing operation.

**Back up each database to a separate media set**

If you select this option database backups will be stored on separate media sets after performing operation.

**To file**

**Backup folder**

Type in or use the ellipsis  button to specify a *Backup folder*.

**Create subfolder for each database**

Select to place each database in a subfolder.

**Backup file name**

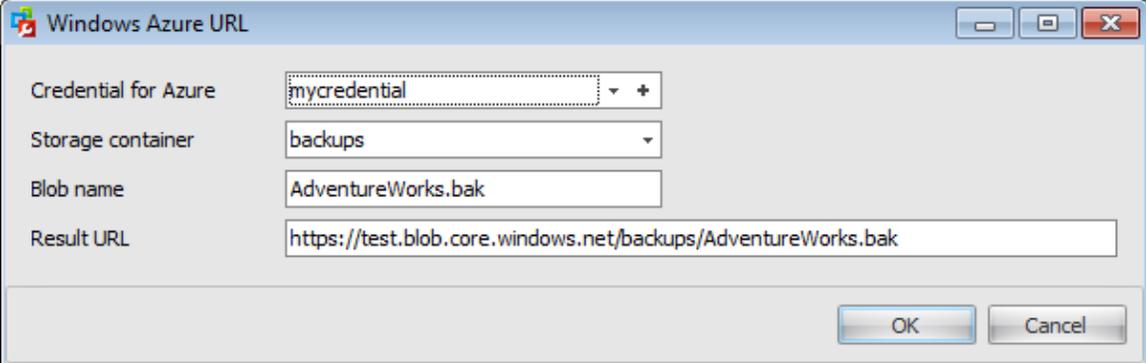
Enter a file name for the current backup. By default, the backup file named `<database_name>.bak` is created where `<database_name>` is a name of the database being backed up.

**To Windows Azure blob**

This option is available for SQL Server 2014 only.

**URL template**

Type in or use the ellipsis  button to specify a Windows Azure URL template. Note that the template must contain the `<database_name>` tag. It will be substituted with the real database name on uploading the database backup to the Azure URL.



Use the Windows Azure URL dialog to form the URL where the database backup will be uploaded.

**Credential for Azure**

This field is available only when creating a backup to the Windows Azure Blob storage service. Use the drop-down list to select the [credential](#) for accessing the Windows Azure storage.

**Storage container**

Container name used for storing the Blob.

**Blob name**

Backup file name.

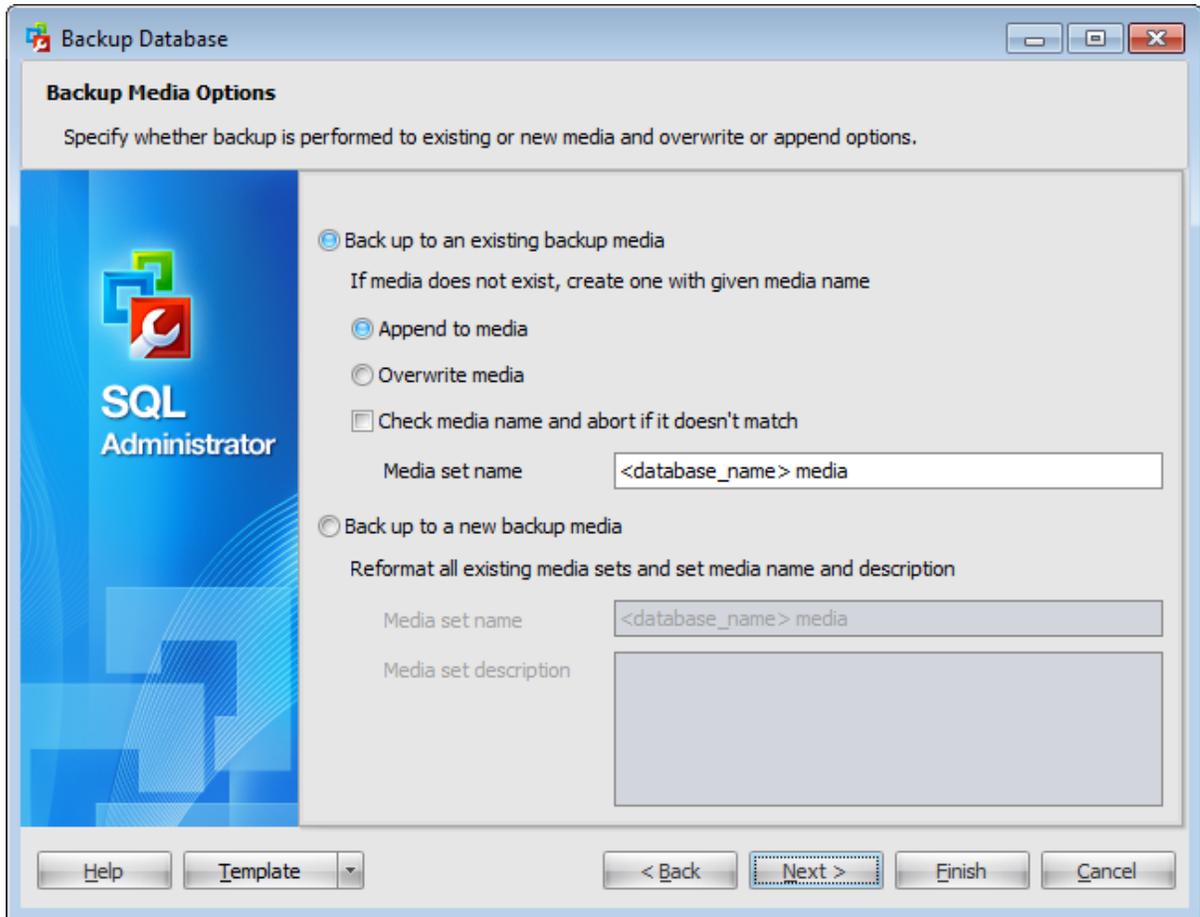
**Result URL**

After defining all the Azure URL properties the template URL used for accessing the backup file `https://{storage_account}.blob.core.windows.net/{storage_container}/{blob_name}` will be formatted with the information provided.

Click the **Next** button to proceed to the [Setting backup media options](#) step of the wizard.

#### 4.2.1.8 Setting backup media options

At this step you can set a number of options pertaining to the new or existing media.



#### Back up to an existing backup media

##### **Append to media / Overwrite media**

Select whether the media should be appended or overwritten during the backup operation. By appending another backup set to existing media, the previous contents of the media remain intact, and the new backup is written after the end of the last backup on the media.

##### Check media name and abort if it doesn't match

If the option is selected, the media name will be first checked whether it matches media set name.

##### **Media set name**

Enter the name of the media.

#### Back up to a new backup media

Check this option to create a new media set to store backups. In this case the backup operation will write a new media header on all media volumes used for the backup

operation. The existing contents of the volume become invalid, because any existing media header and backup sets are overwritten.

**Media set name**

Enter a name for a new media set.

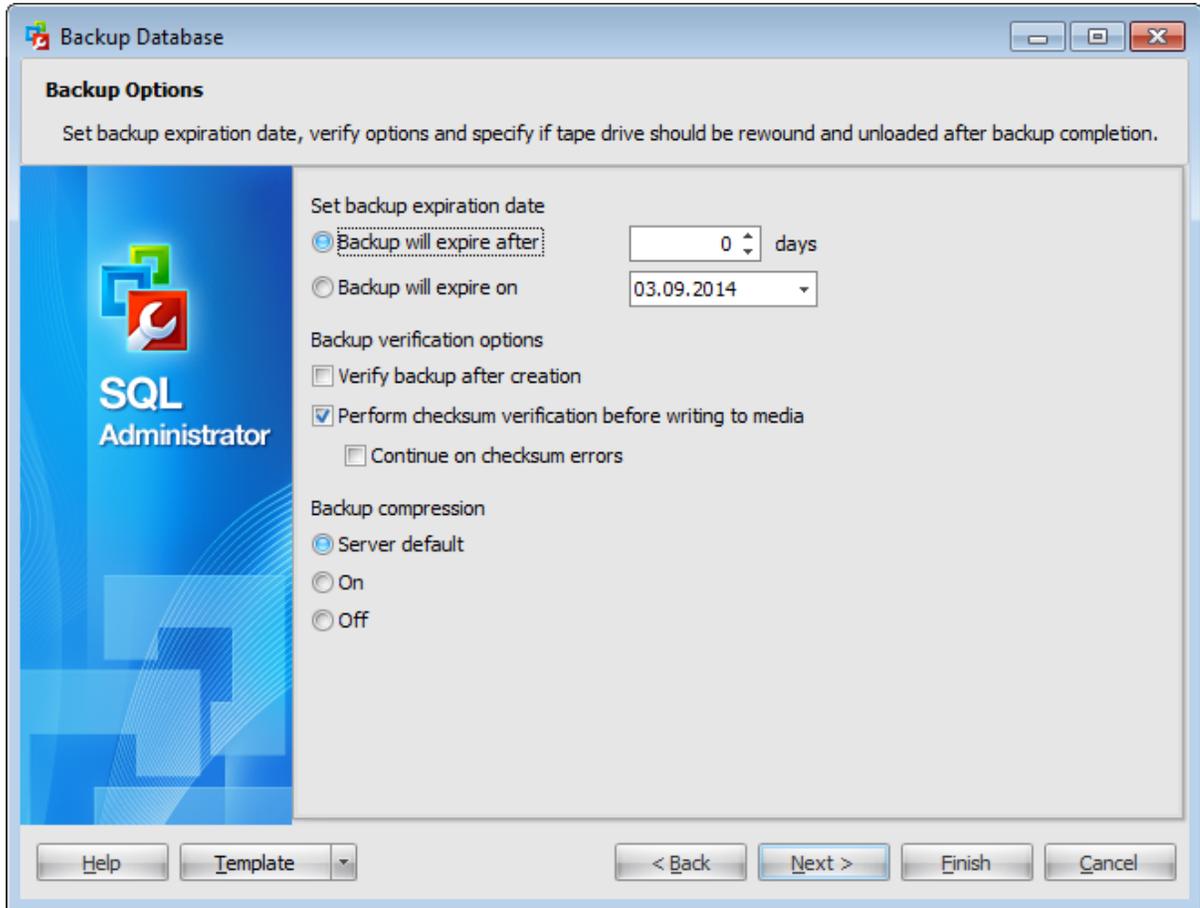
**Media set description**

Enter any optional text to describe a new media set.

Click the **Next** button to proceed to the [Setting backup options](#) step of the wizard.

#### 4.2.1.9 Setting backup options

This step offers some **additional options** that may be useful for the database backup operation.



#### Set backup expiration date

You can set the expiration date by defining the number of days after that a backup will expire or by specifying an expiration date.

#### Backup verification options

##### **Verify backup after creation**

Check this option to enable backup verification.

##### **Perform checksum verification before writing to media**

This option enables/disables checksum verification before writing to the backup media.

##### **Continue on checksum error**

This option determines whether the backup operation will be stopped or forced to continue if an error occurs.

#### Backup compression

**Server default**

The backup compression is defined in compliance with the [SQL Server property](#)

**Compress backups by default.**

**On**

Explicitly enable backup compression.

**Off**

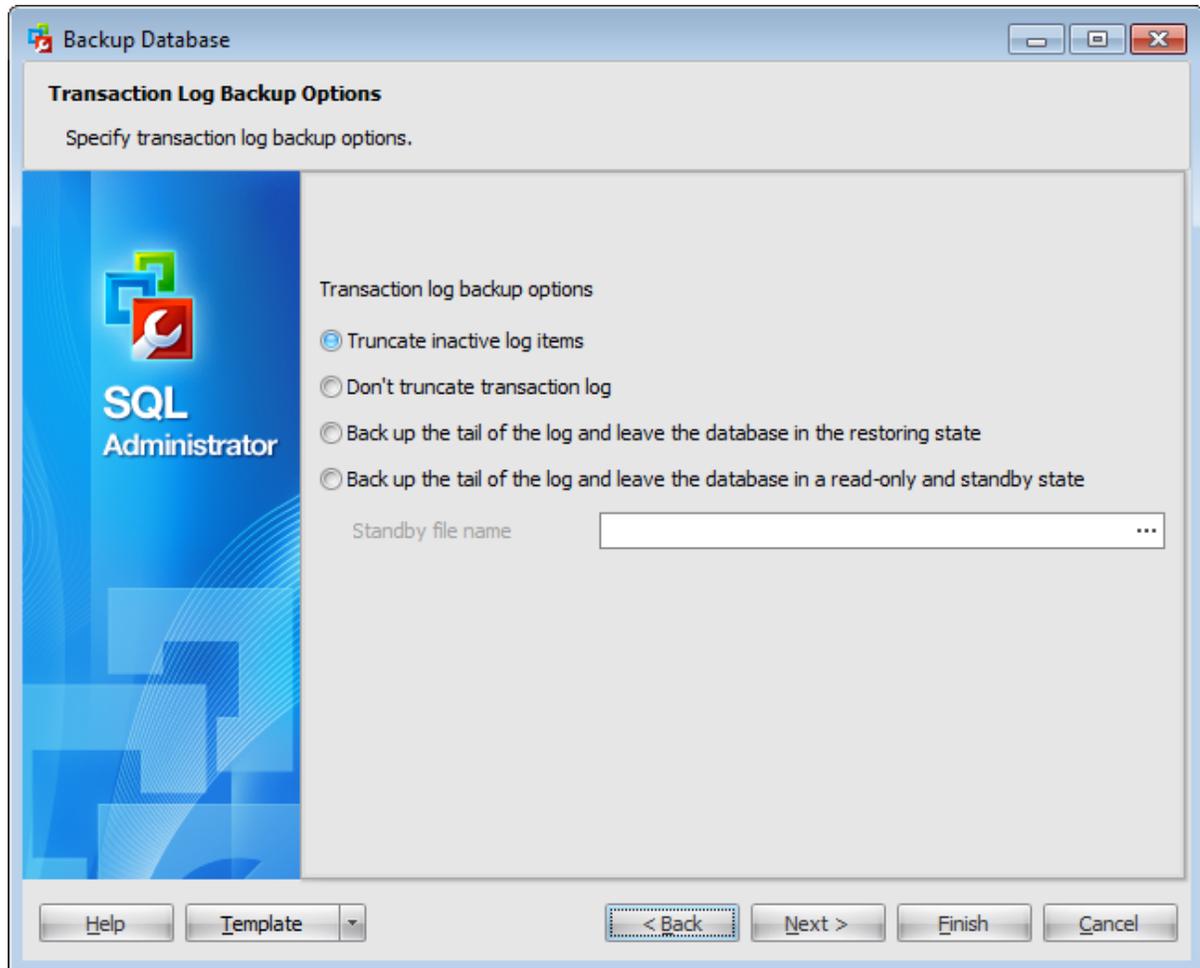
Explicitly disable backup compression.

**Note:** Creating compressed backups is supported only in SQL Server 2008 Enterprise and later, but beginning in SQL Server 2008, every edition can restore a compressed backup.

Click the **Next** button to proceed to the [Setting schedule options](#) step of the wizard.

#### 4.2.1.10 Specifying transaction log backup options

Use this step to set parameters for transaction log backup. This step appears only if the  *Transaction log* option was selected on the [second step](#).



**Truncate inactive log items**

The log is truncated after all the records within one or more virtual log files become inactive.

**Don't truncate transaction log**

This option specifies that the log is not truncated and causes the Database Engine to attempt the backup regardless of the state of the database.

**Note:** A backup taken with this option might have incomplete metadata.

**Back up the tail of the log and leave the database in the restoring state**

This option creates a tail-log backup, which backs up logs that have not yet been backed up (the active log), typically, in preparation for restoring a database. The database will be unavailable to users until it is completely restored.

**Back up the tail of the log and leave the database in a read-only and standby state**

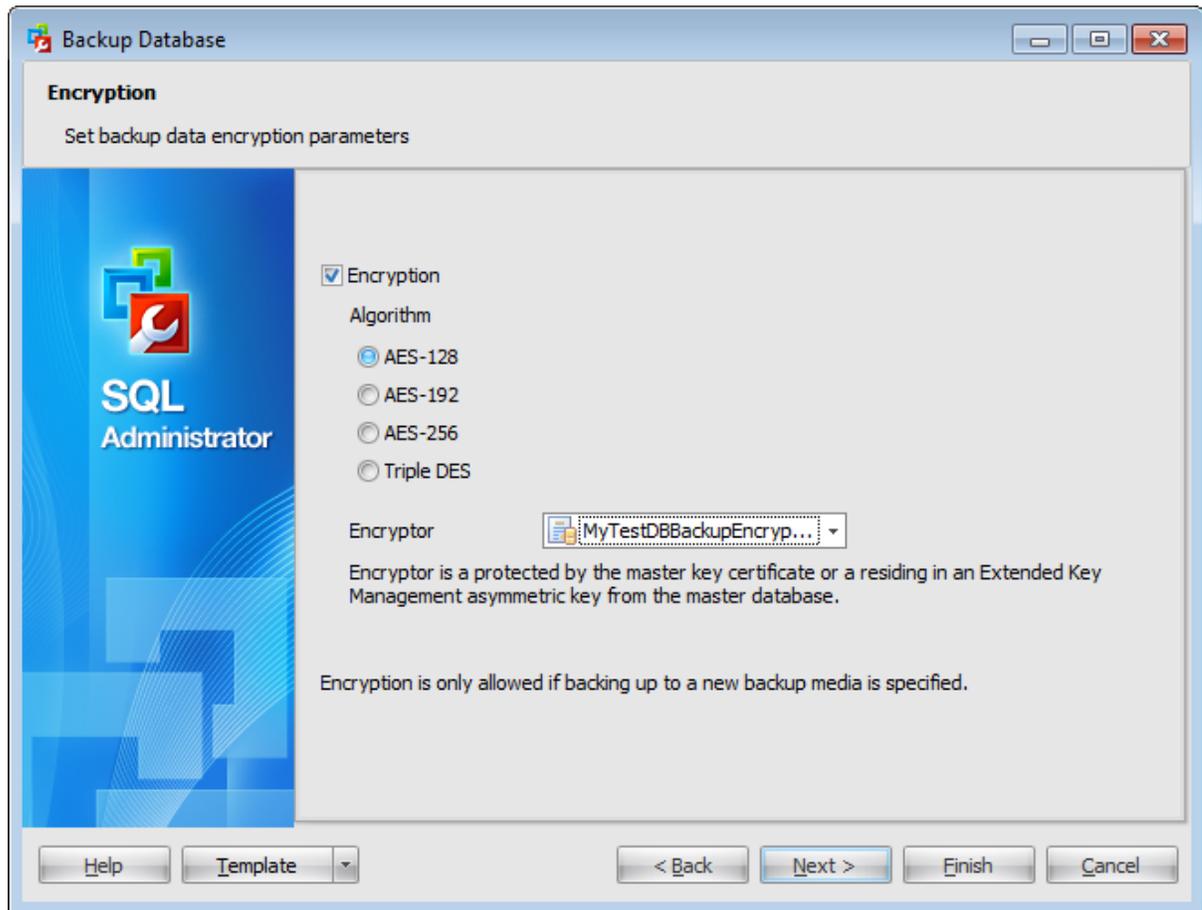
Use this option to back up the tail of the log and leave the database in a read-only and standby state. The standby clause writes standby data (performing rollback, but with the option of further restores).

Provide **Standby file name** in the corresponding field. The standby file becomes part of the database. It holds the rolled back changes, which must be reversed if *RESTORE LOG* operations are to be subsequently applied.

#### 4.2.1.11 Setting backup encryption options

Use this step to set backup data encryption parameters.

This step appears only when performing backup on SQL Server 2014 and is available for editing only if the  **Back up to a new backup media** option was selected on the [Backup Media Options](#) step.



#### Encryption

Check this option to enable the data encryption while creating a backup.

To encrypt during backup, you must specify an encryption algorithm, and an encryptor to secure the encryption key.

#### Algorithm

Select one of the encryption algorithms for a backup:

- AES-128
- AES-192
- AES-256
- Triple DES

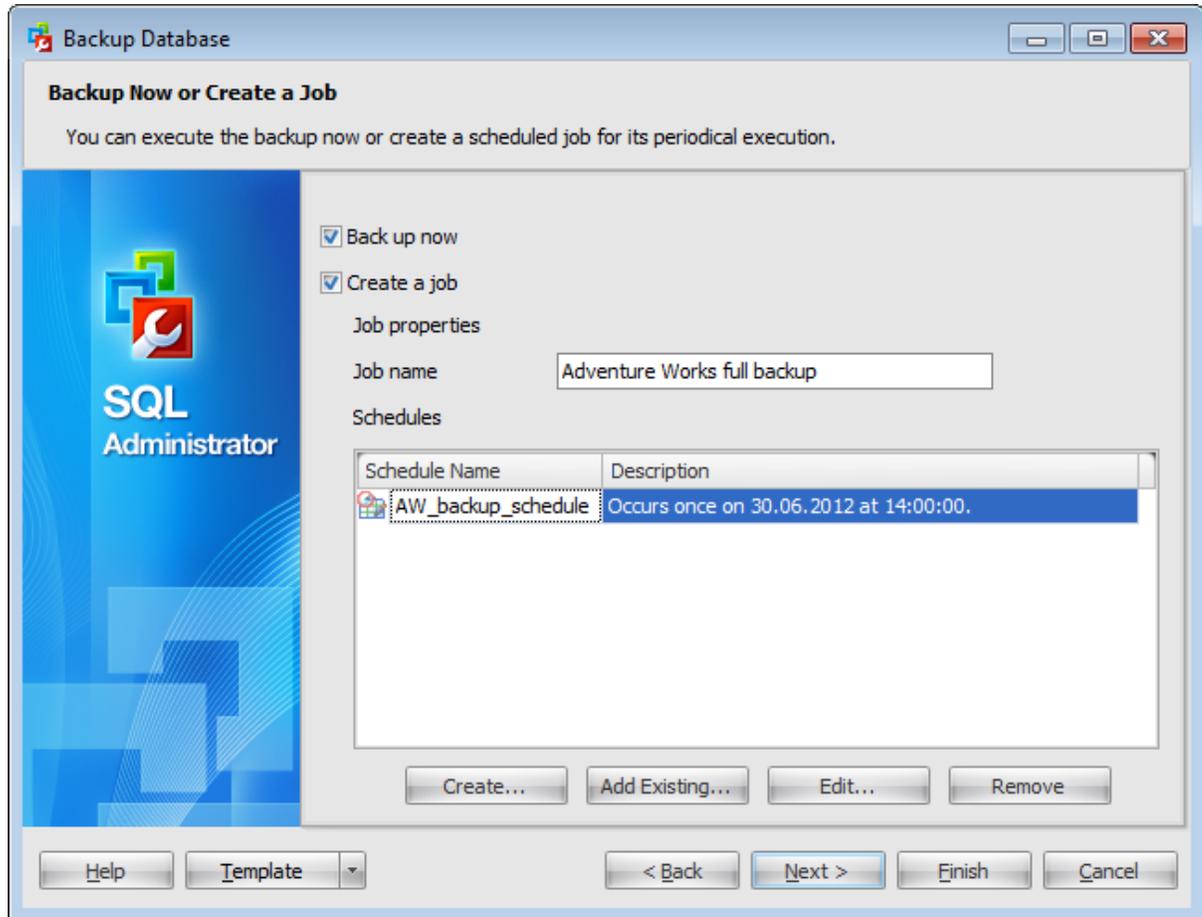
#### Encryptor

If you choose to encrypt you also have to specify the encryptor. Encryptors can be a master key certificate or an Extended Key Management asymmetric key from the master

database. All available encryptors are listed in the combobox.

#### 4.2.1.12 Setting schedule options

Using this step you can define when the backup will be executed.



##### **Back up now**

Check this option to start the backup process after all necessary options are set.

##### **Create a job**

Select this option to create a **scheduled** backup.

**Note:** To create a job you need to run [SQL Server Agent](#) service on the server.

Define **job properties**:

##### **Job name**

Enter a name of the new backup job.

##### **Schedules**

This list displays all schedules that cause the backup job to begin running. You can add existing schedule or [create](#) a new one.

Click the **Next** button to proceed to the [Running database backup](#) step of the wizard.



#### 4.2.1.13 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the backup database process.

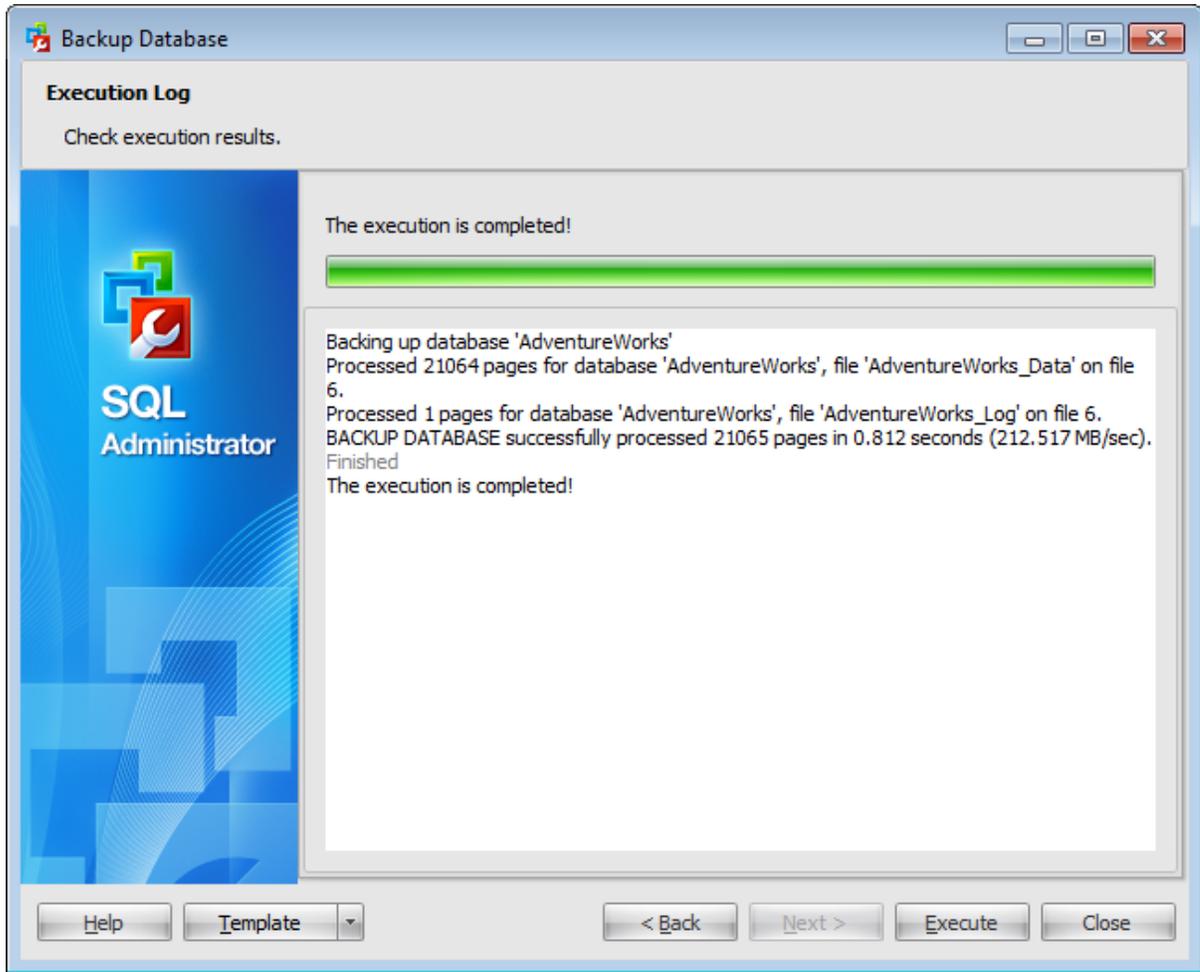


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing backup database task](#) step.

#### 4.2.1.14 Executing backup database task

The log area allows you to view the log of operation and errors (if any).



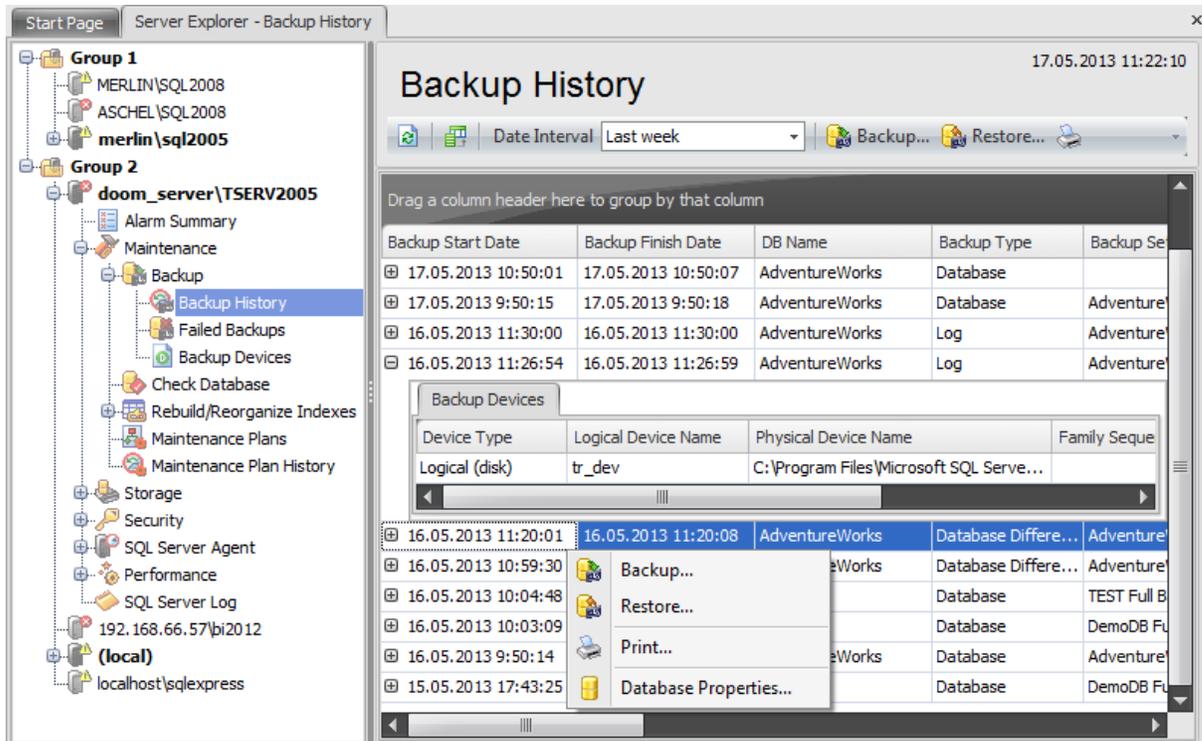
Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.2.2 Backup History

This window displays information about all performed backups during the specified **Date Interval**.

Use the context menu or buttons on the work area toolbar to manage the list of backups:

-  create a database [backup](#)
-  [restore](#) a database
-  [edit database properties](#)
-  create and [print](#) the report



The list displays the backup history as a grid with the following columns: *Backup Start Date*, *Backup Finish Date*, *DB Name*, *Backup Type*, *Backup Set Name*, *Backup Set Description*, *Position*, *Backup Size (MB)*, *Compressed Size (MB)*, *Media Set Name*, *Media Set Description*, *User Name*, *Server Name*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To show/hide information about [backup devices](#), click the "+" / "-" button at the beginning of each row.

To learn about [device contents](#), select the **View Device Contents** item from the device context menu or click the 'Enter' button.

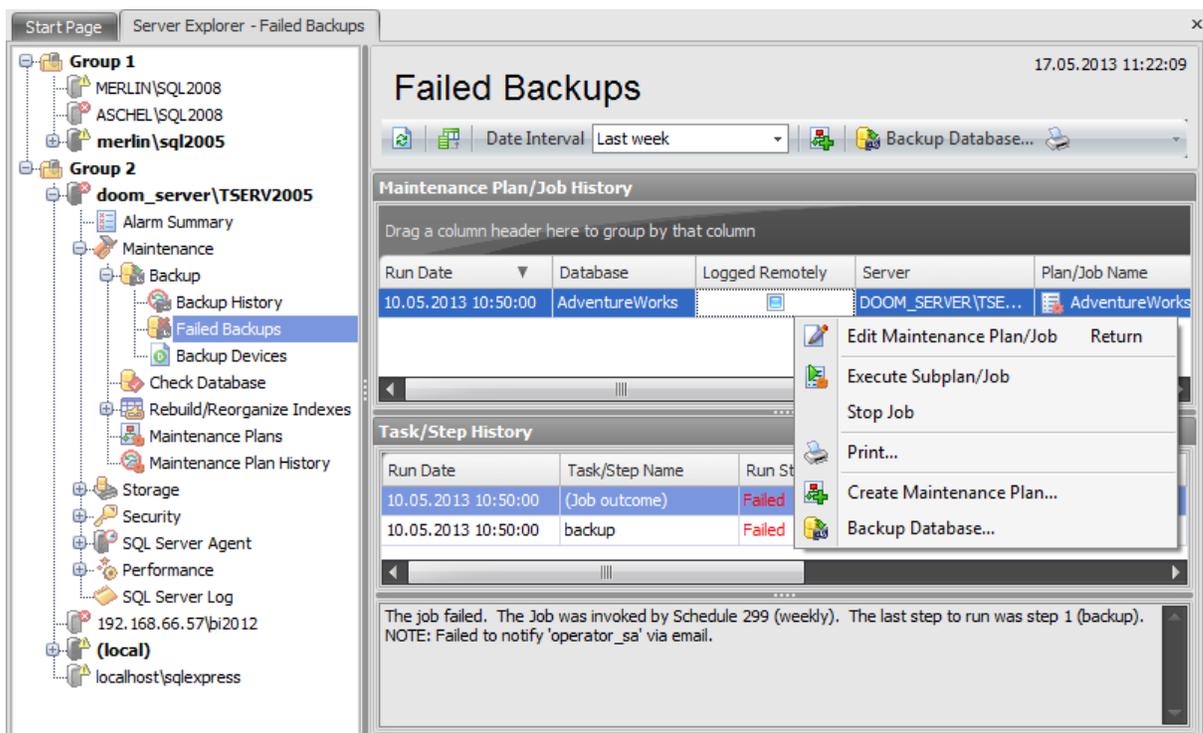
### 4.2.3 Failed backups

The upper area of this window shows backup [maintenance plans](#) that have failed during their execution.

The lower area provides information about steps of the selected task.

Use the context menu or buttons on the work area toolbar to manage the list of backups:

-  create a new [maintenance plan](#)
-  create a database [backup](#)
-  edit selected maintenance plan/job
-  execute selected subplan/job
-  create and [print](#) the report



The list in the **Maintenance Plan/Job History** area displays failed backups as a grid with the following columns: *Run Date*, *Database*, *Run Status*, *Logged Remotely*, *Server*, *Plan/Job Name*, *Subplan Name*, *Backup Task/Step*, *Duration*. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

All these maintenance plans have the *Run Status* value "Failed". In the lower grid you can explore the particular step on which the execution plan has failed. When you choose a particular maintenance plan the **Task/Step History** area displays the following detailed information about the subplanned tasks: *Run Date*, *Task/Step Name*, *Duration*, *Server*, *Command*, *Message*.

If you select the step with the *Run Status* value 'Failed' then in the lower area you will see the message informing about the reason of the occurred mistake.



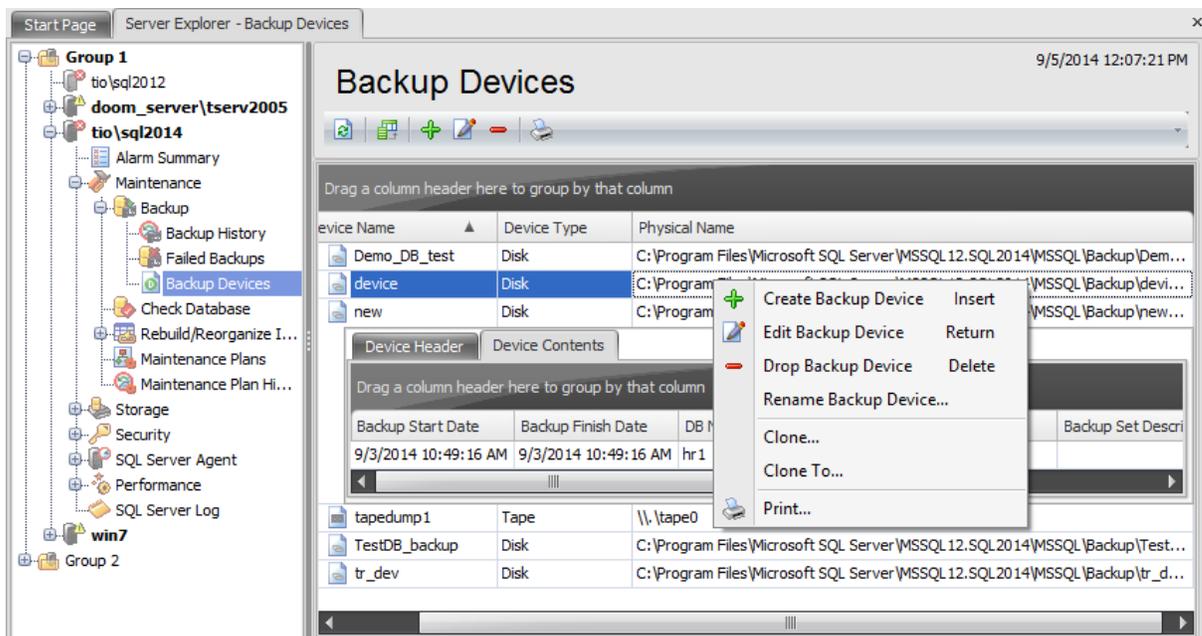
#### 4.2.4 Backup devices

A **Backup device** is a tape drive or a disk drive used in [backup](#) and [restore](#) operations. When creating a backup, you must [select a backup device](#) where the data will be written. Microsoft SQL Server 2005 (and higher) can back up databases, transaction logs, and files to disk and tape devices.

Use the context menu or buttons on the work area toolbar to manage the list of maintenance plans:

- + [create](#) a new backup device
-  [edit](#) selected backup device
- [drop](#) selected backup device
- rename selected backup device
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a backup device, 'Enter' button to edit backup device(s) and 'Delete' button to drop backup device(s).



The list displays the existing database backup devices as a grid with the following columns: *Device Name*, *Device Type*, *Physical Name*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Backup Device in the [Backup Device Editor](#), double-click it in the grid. Alternatively, you can right-click the Backup Device alias and select the **Edit Backup Device(s)** context menu item.

Use the +/- icons near the device name to expand/collapse device properties: device header and device contents.

You can launch the [Restore Database Wizard](#) when viewing device contents: right-click a backup and then select the  **Restore** item of the context menu.

If you click the  button then the **Device Header** and **Device Contents** tabs will be expanded.

#### 4.2.4.1 Backup Device Editor

**Device Editor** allows you to define device properties.

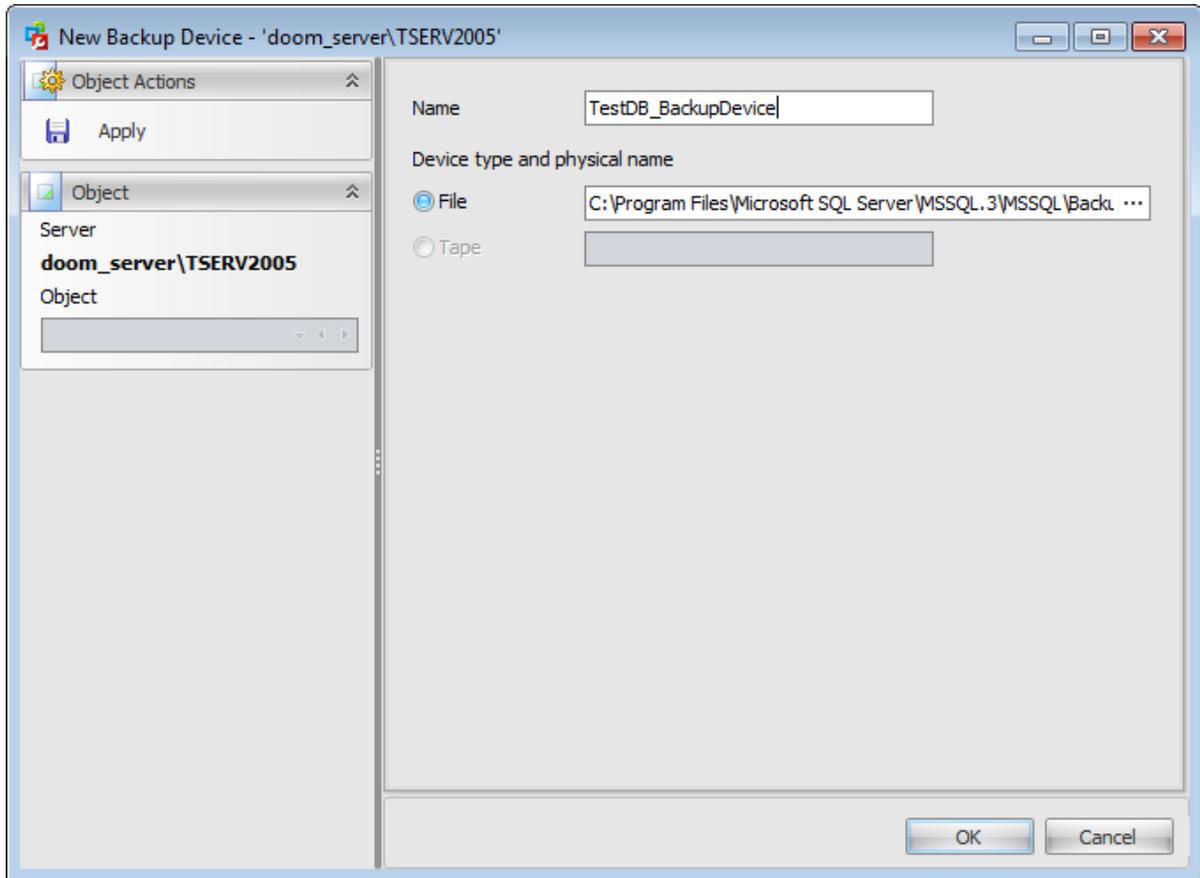
To open a device in the **Backup Device Editor**, double-click it in the [list of devices](#). Alternatively, you can right-click the device alias and select the **Edit Device** or **Create Device** context menu item.

The list of **Backup Device Editor** tabs is displayed on the navigation bar:

- [Setting general properties](#)
- [Viewing device contents](#)
- [Viewing SQL definition](#)

## 4.2.4.1.1 Setting general properties

Use this tab to set main device **Properties**.

**Name**

Type in a backup device name.

**Device type and physical name** can be defined only on creating a new backup device. A logical backup device is an optional, user-defined name that points to a specific physical backup device (a disk file or tape drive). A logical backup device lets you use indirection when referencing the corresponding physical backup device.

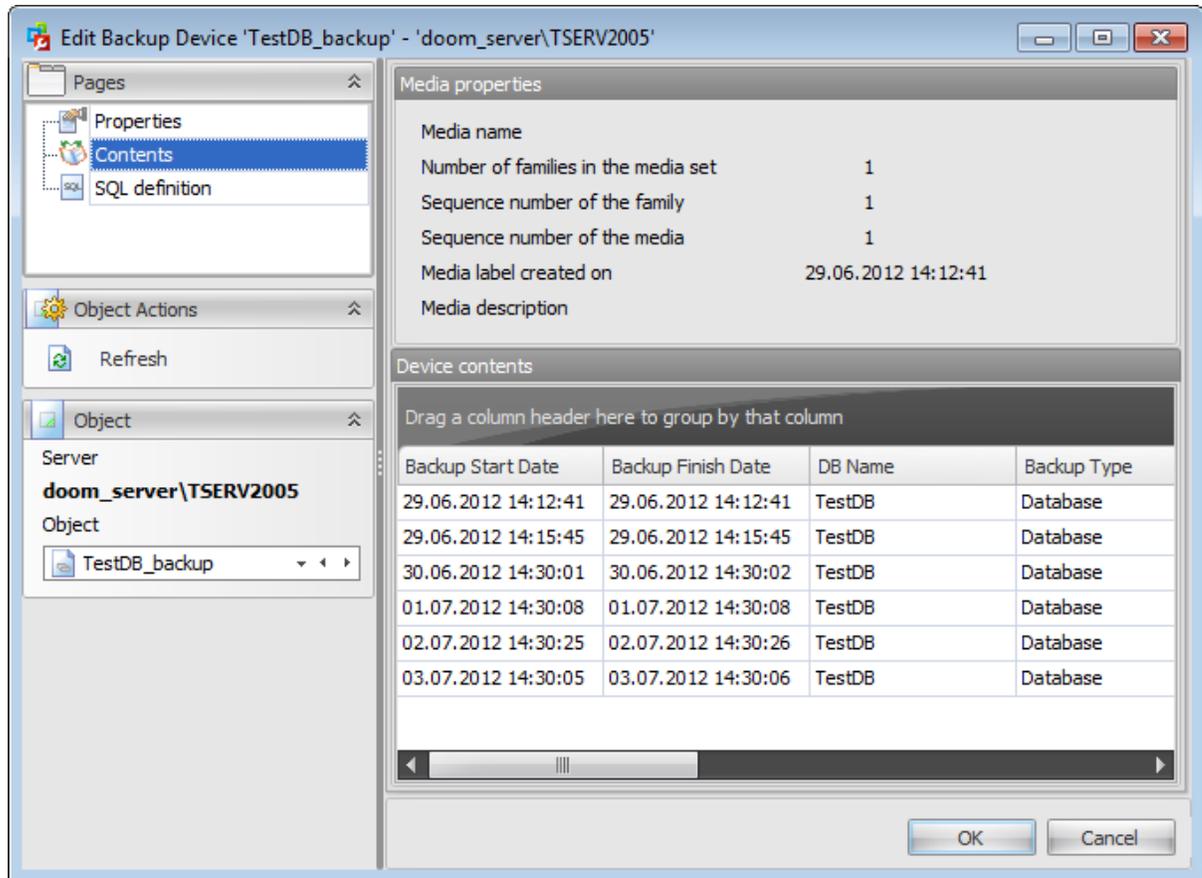
Specify the type of the logical device:  **File** or  **Tape**.

If you have selected  **File** then specify the full path of the file.

If you have selected  **Tape** then select a tape drive that is not already associated with another backup device. If no such tape drives are available, the Tape option is inactive.

## 4.2.4.1.2 Viewing device contents

The **Contents** section allows you to view **Media properties** and **Device contents**. This section is available on device editing only.

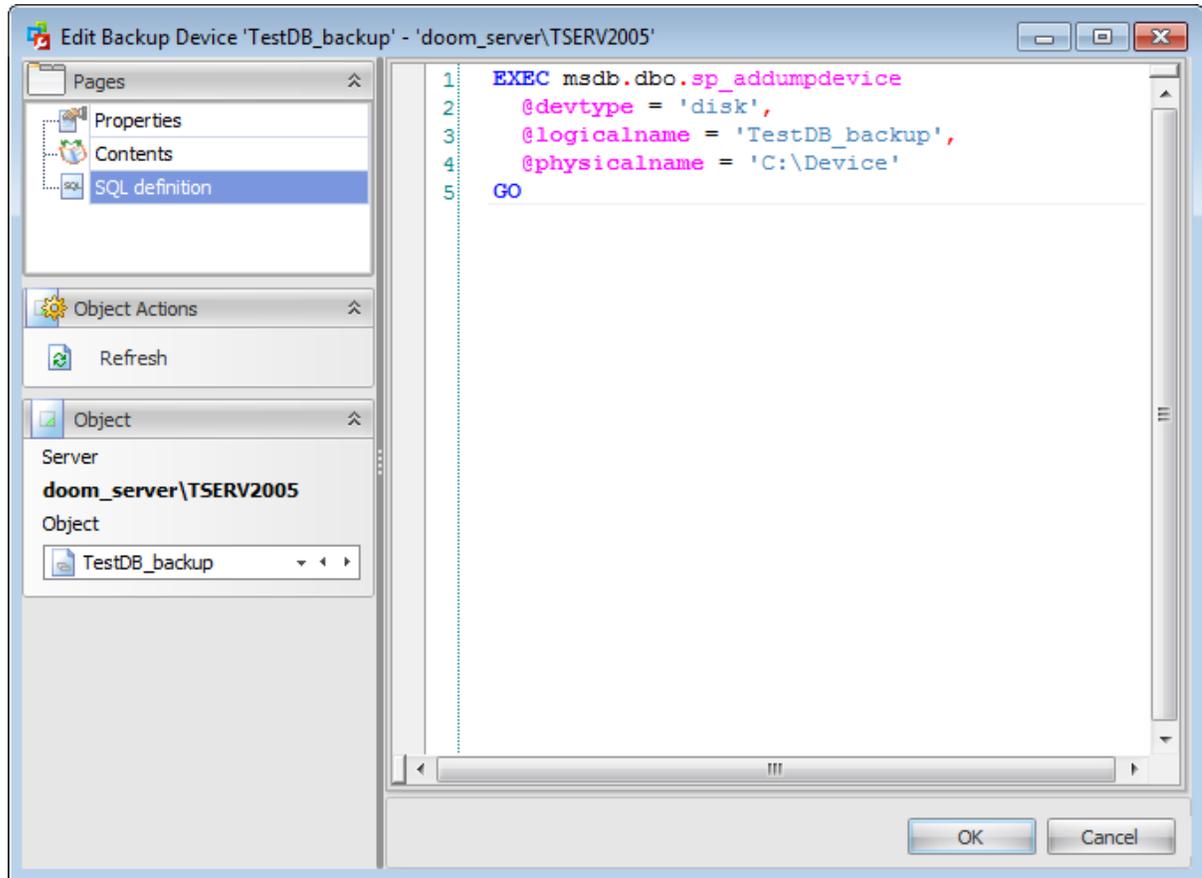


The **Media Properties** area displays following properties of the backup device: *Media name, Number of families in the media set, Sequence number of the family, Sequence number of the media, Media label created on, Media description.*

The **Device contents** area displays the [backup](#) operation(s) as a grid with the following columns: *Backup Start Date, Backup Finish Date, DB Name, Backup type, Backup Set Name, Backup Set description, Position, Backup Size (MB), User Name, Server Name.* You can launch the [Restore Database Wizard](#) when viewing device contents: right-click a backup operation and then select the  **Restore** item of the context menu.

## 4.2.4.1.3 Viewing SQL definition

This section allows you to view the object DDL. It is available on device editing only.



## 4.3 Restore database

**Restore Database Wizard** allows you to perform the database restore operation on your SQL Server system (the *RESTORE DATABASE* Transact-SQL statement is issued).

This operation is used to rebuild a damaged or corrupted database that has been backed up using [Backup Database Wizard](#). You can restore an entire database, part of a database, specific files and filegroups, transaction logs.

To run the wizard, select the **Tools** |  **Restore Database...** [main menu](#) item or click  **Restore Database...** button on the [toolbar](#).

The wizard consists of the following steps:

- [Selecting server and setting restore type](#)
- [Setting database](#)
- [Specifying point in time](#)
- [Selecting backup set to restore](#)
- [Selecting files and filegroups to restore](#)
- [Setting restore options](#)
- [Setting recovery options](#)
- [Running database restore](#)

---

### See also:

[Maintenance plans](#)

[Backup database](#)

[Attach database](#)

[Detach database](#)

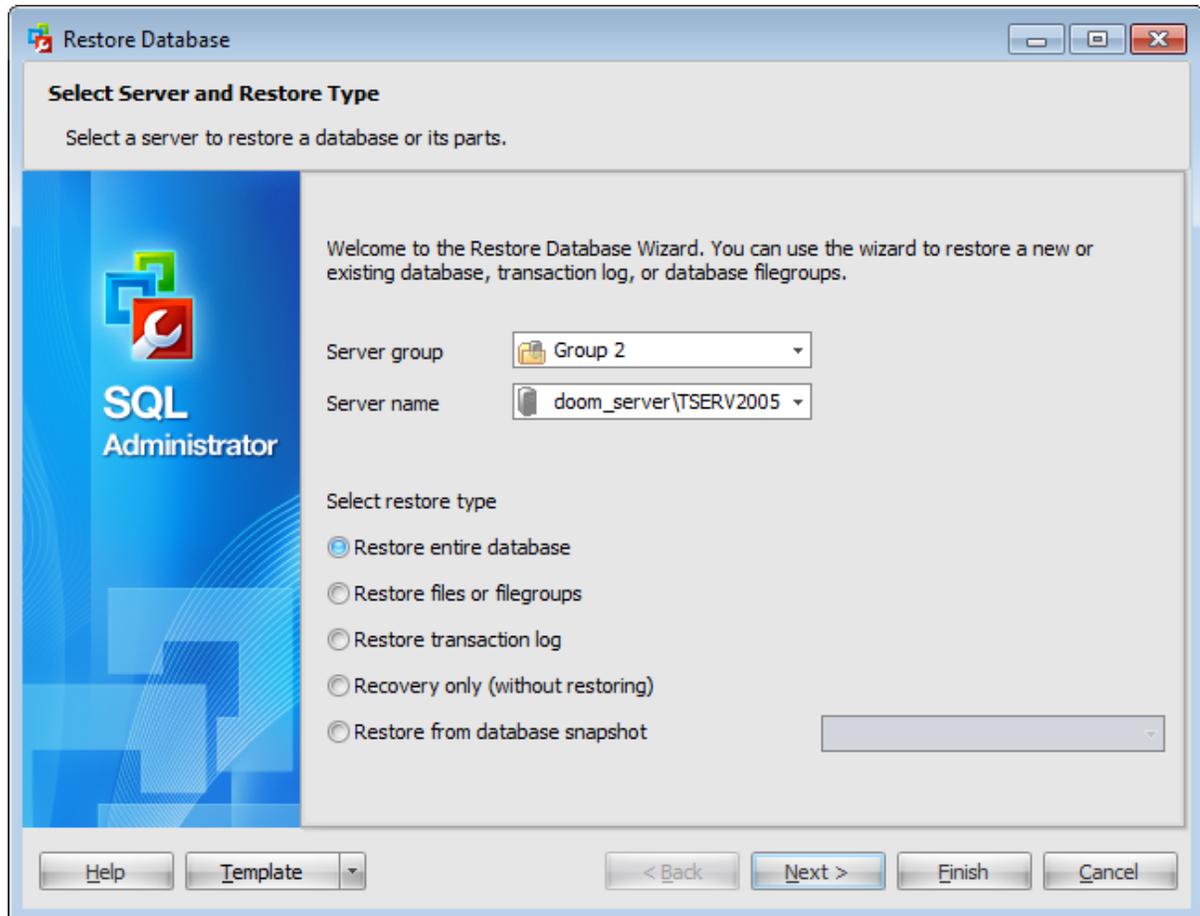
[Check Database](#)

[Rebuild/Reorganize Indexes](#)

[Shrink database](#)

### 4.3.1 Selecting server and setting restore type

This step of the wizard allows you to specify the **server** to restore database, and choose restore type.



#### Server group

Use the drop-down list to select a server group where the server is stored.

#### Server name

Use the drop-down list to select the host/instance name to restore database to.

#### Select restore type

This group of options allows you to specify the restore type to be used for the restore operation.

**Restore entire database**  
Full restore of entire database.

**Restore files or filegroups**  
Differential restore of individual files and filegroups. Files and filegroups can be restored either from a file or filegroup backup operation, or from a full database backup operation.

**Restore transaction log**

Specifies that a transaction log backup will be used to restore this database. SQL Server checks the backed up transaction log to ensure that the transactions are being loaded into the correct database and in the appropriate sequence.

**Recovery only (without restoring)**

Select this option to execute the database recovery only.

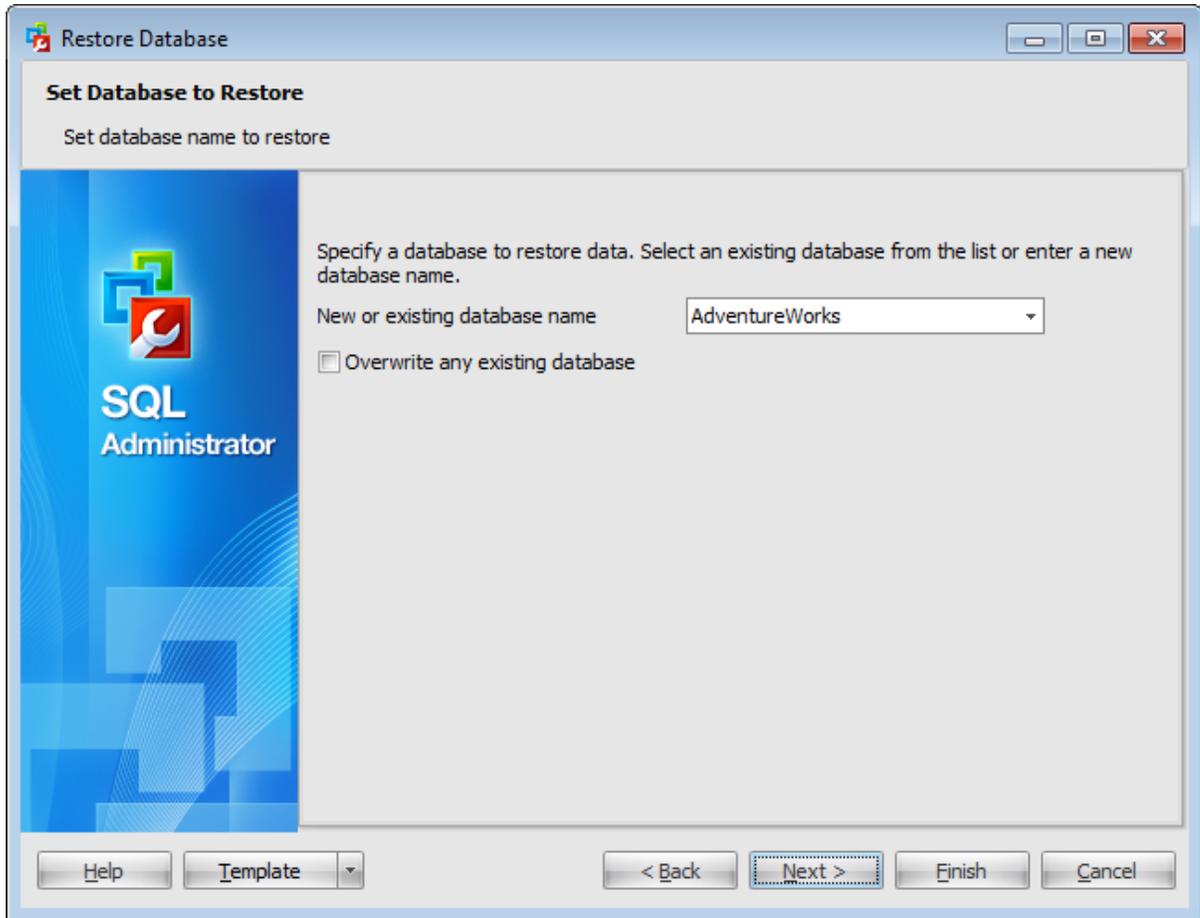
**Restore from database snapshot**

Select this option to restore the database from a previously saved snapshot and provide the snapshot name.

Click the **Next** button to proceed to the [Setting database](#) step of the wizard.

### 4.3.2 Setting database

Use this step of the wizard to define a database to restore.



#### **New or existing database name**

Enter a name for the new database or select the database to restore to from the drop-down list of all available databases on the specified host.

#### **Overwrite any existing database**

Check this option to restore to an existing database. Uncheck this option if you want to create a new database on the basis of a backup.

In case the  *Restore transaction log* option was selected on the [first step](#) then on this step you are to select an existing database which transaction log should be restored on.

In case the  *Recovery only (without restoring)* option has been selected on the [first step](#) then on this step you are to select among existing databases that are in the restoring state so as to make it ready for use by rolling back any uncommitted transactions.

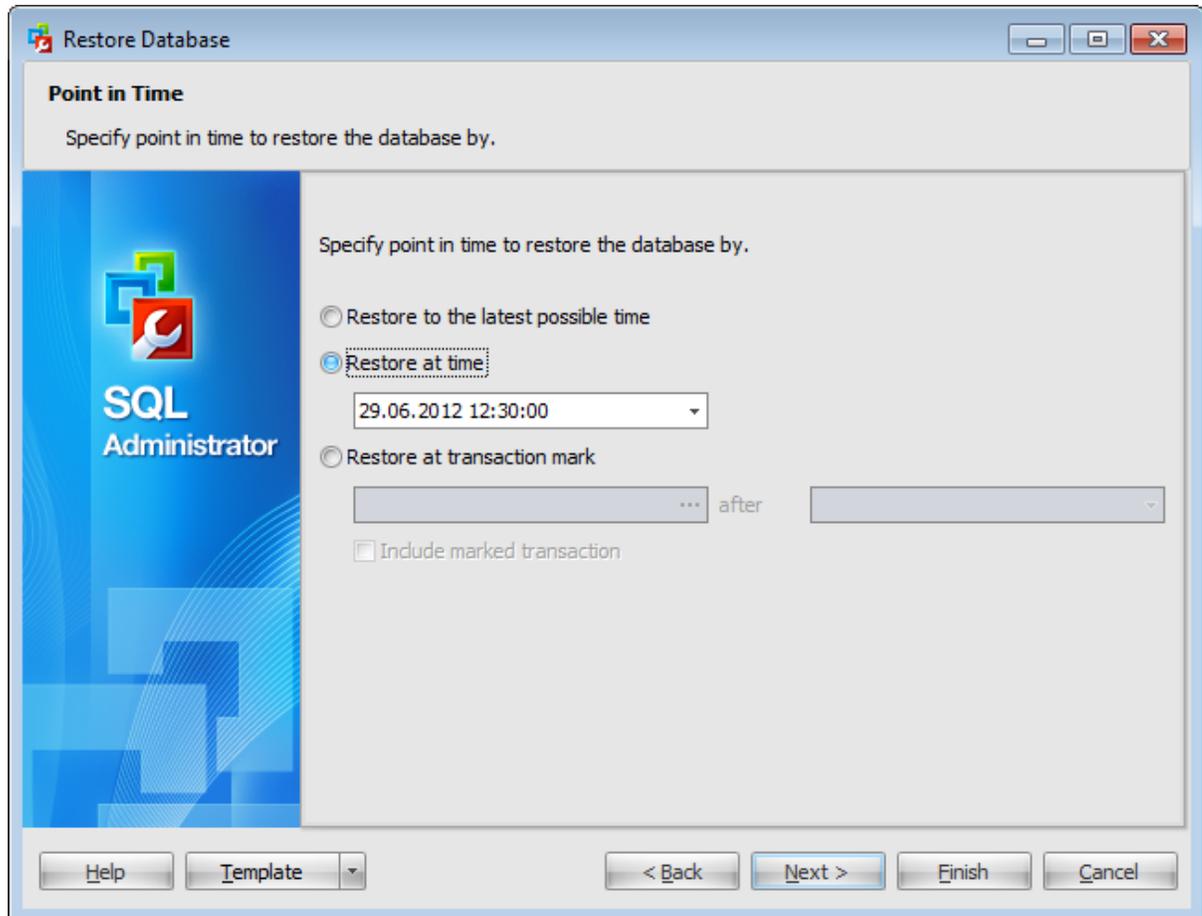
In case the  *Restore from database snapshot* option was selected on the [first step](#) then on this step you can view the database name that will be reverted from the specified

snapshot.

Click the **Next** button to proceed to the [Specifying point in time](#) step of the wizard.

### 4.3.3 Specifying point in time

This step of wizard allows you to specify a point in time to which a database should be restored.



You can  **Restore a database to the latest possible time** or pick up a date in the  **Restore at time field**.

When restoring from the transaction log you can specify additional parameters:

**Restore at transaction mark**

SQL Server supports inserting named marks into the transaction log to allow recovery to that specific mark. Log marks are transaction specific and are inserted only if their associated transaction commits. As a result, marks can be tied to specific work, and you can recover to a point that includes or excludes this work.

By default, the restore is up to, but excluding, the marked transaction. To restore the marked transaction also, check  **Include marked transaction**.

Click the **Next** button to proceed to the [Selecting backup set to restore](#) step of the wizard.

#### 4.3.4 Selecting backup set to restore

At this step of the wizard you need to select backup sets to restore the database from.

**From database**

Select a *database* as a source of backup sets using drop-down list.

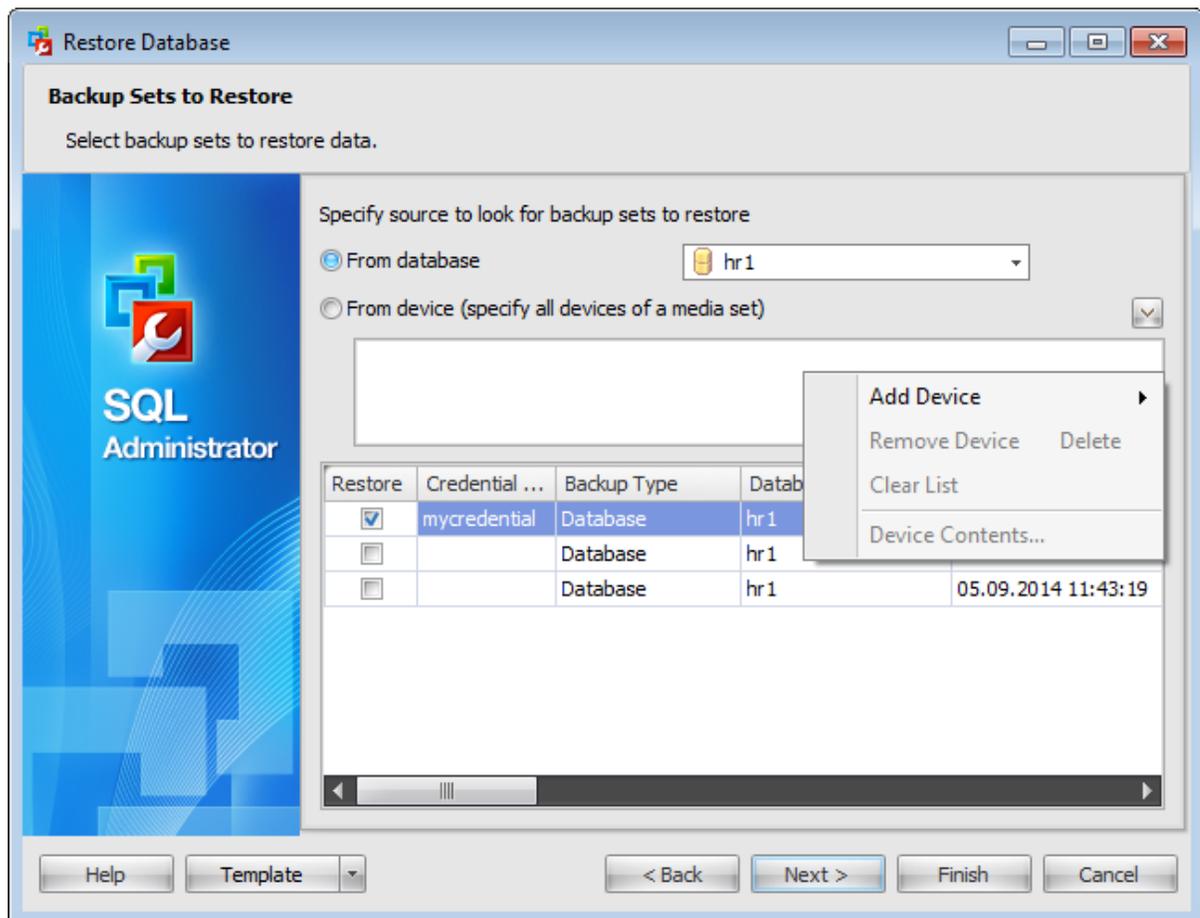
**From device**

You can specify a backup set manually by adding devices to the device list.

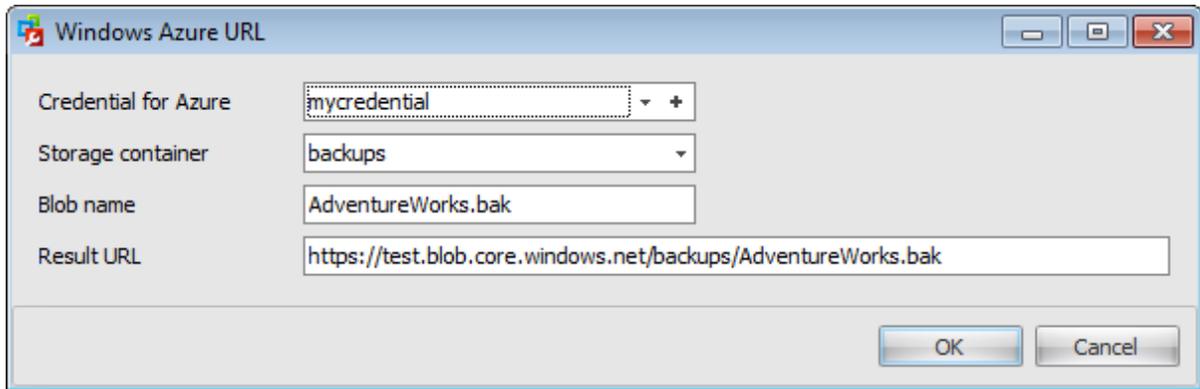
To add a device, click the  **Add Item** button and select a media type from the menu. Select the required backup device using the corresponding dialog.

To remove a device from the list, select it and click the **Delete Item** button.

Using the **Add Item** menu you can add a *logical device*, a *file*, or a *tape*.



For restoring a database on the SQL Server 2014 adding *Windows Azure URL* is possible as well.



Use the Windows Azure URL dialog to form the URL where the database backup will be uploaded.

**Credential for Azure**

Use the drop-down list to select the [credential](#) for accessing the Windows Azure storage.

**Storage container**

Container name used where the Blob is stored.

**Blob name**

Backup file name.

**Result URL**

After defining all the Azure URL properties the template URL used for accessing the backup file `https://{storage_account}.blob.core.windows.net/{storage_container}/{blob_name}` will be formatted with the information provided.

The grid displays the whole information about a backup set. To add a backup set to the restore, select it in the list and set the respective flag in the **Restore** column.

Click the **Next** button to proceed to the [Selecting files and filegroups to restore](#) step of the wizard.

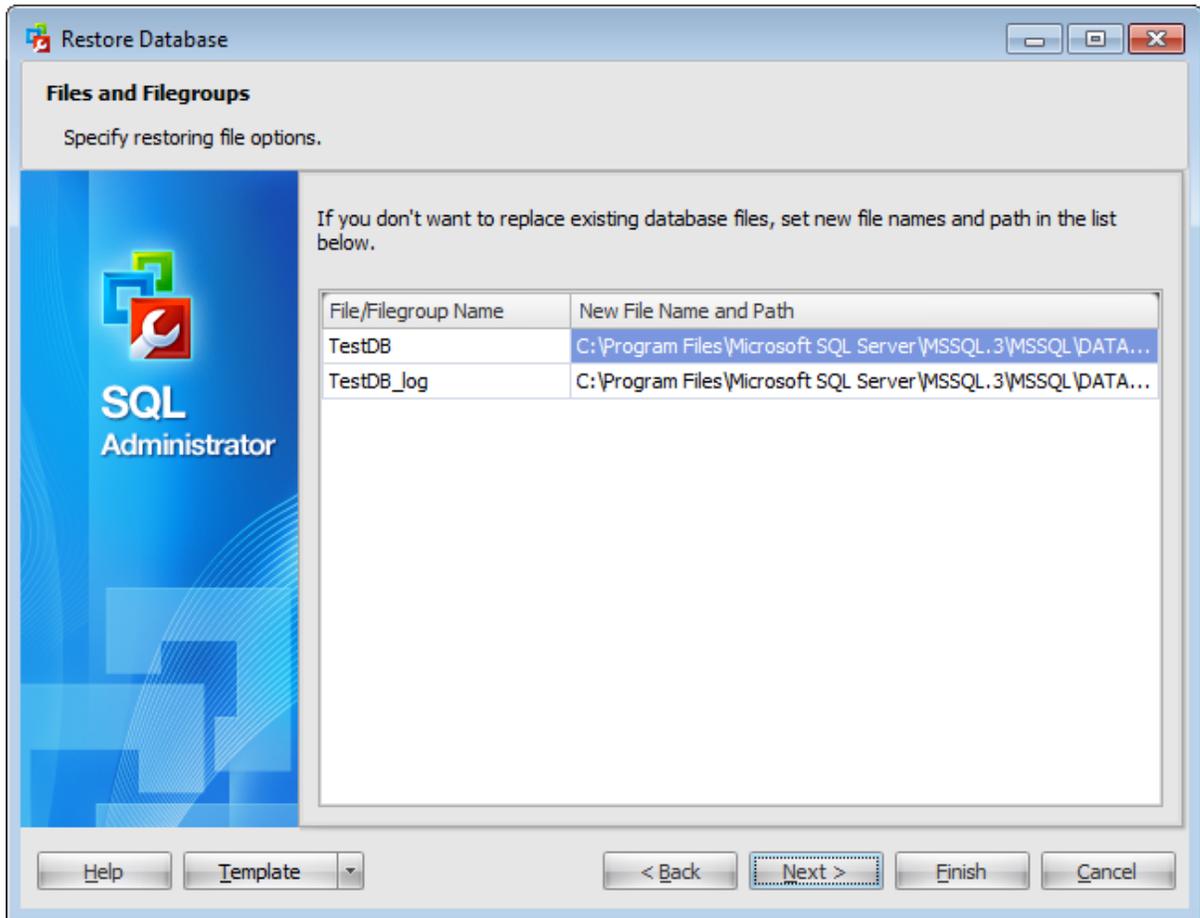
### 4.3.5 Selecting files and filegroups to restore

This step allows you to **select files** that will be used to restore the database.

By default the application restores all files.

To add a file to restore, select it in the list and set the respective flag in the **Restore** column.

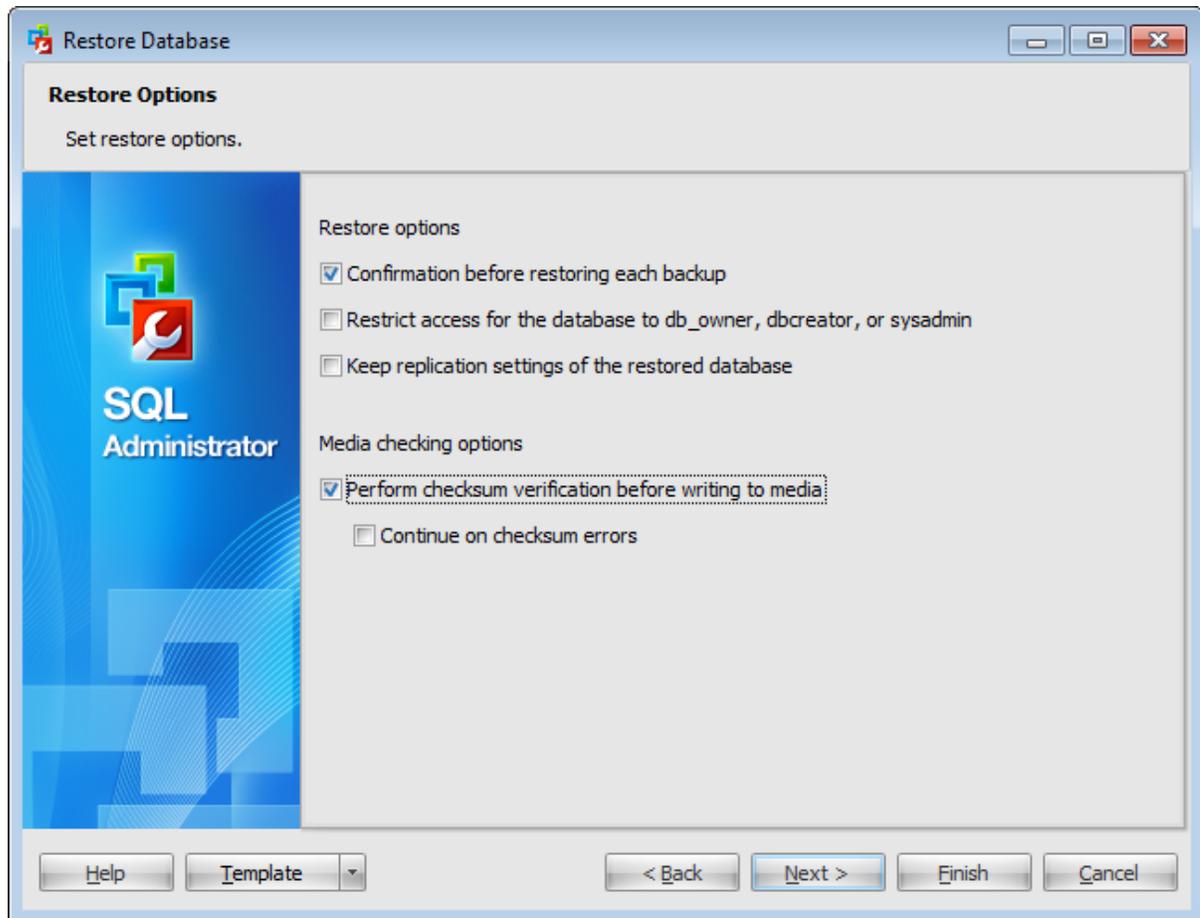
You can also edit file paths by clicking the corresponding rows of the **New File Name and Path** column.



Click the **Next** button to proceed to the [Setting restore options](#) step of the wizard.

### 4.3.6 Setting restore options

This step offers some **additional options** that are specific for each restoration mode.



#### Restore options

**Confirmation before restoring each backup**

If this option is selected, the program requires confirmation each time you run the process of restoring a database.

**Restrict access for the database to db\_owner, dbcreator or sysadmin**

This option restricts access to the newly restored database to members of the *db\_owner*, *dbcreator*, or *sysadmin* roles.

**Keep replication settings of the restored database**

This option specifies that the replication settings are preserved upon restoration of a published database to a server other than that on which it was created. It prevents replication settings from being removed when a database or log backup is restored on a warm standby server and the database is recovered.

#### Media checking options

**Perform checksum verification before writing to media**

This option enables/disables checksum verification before restoring from the backup media.

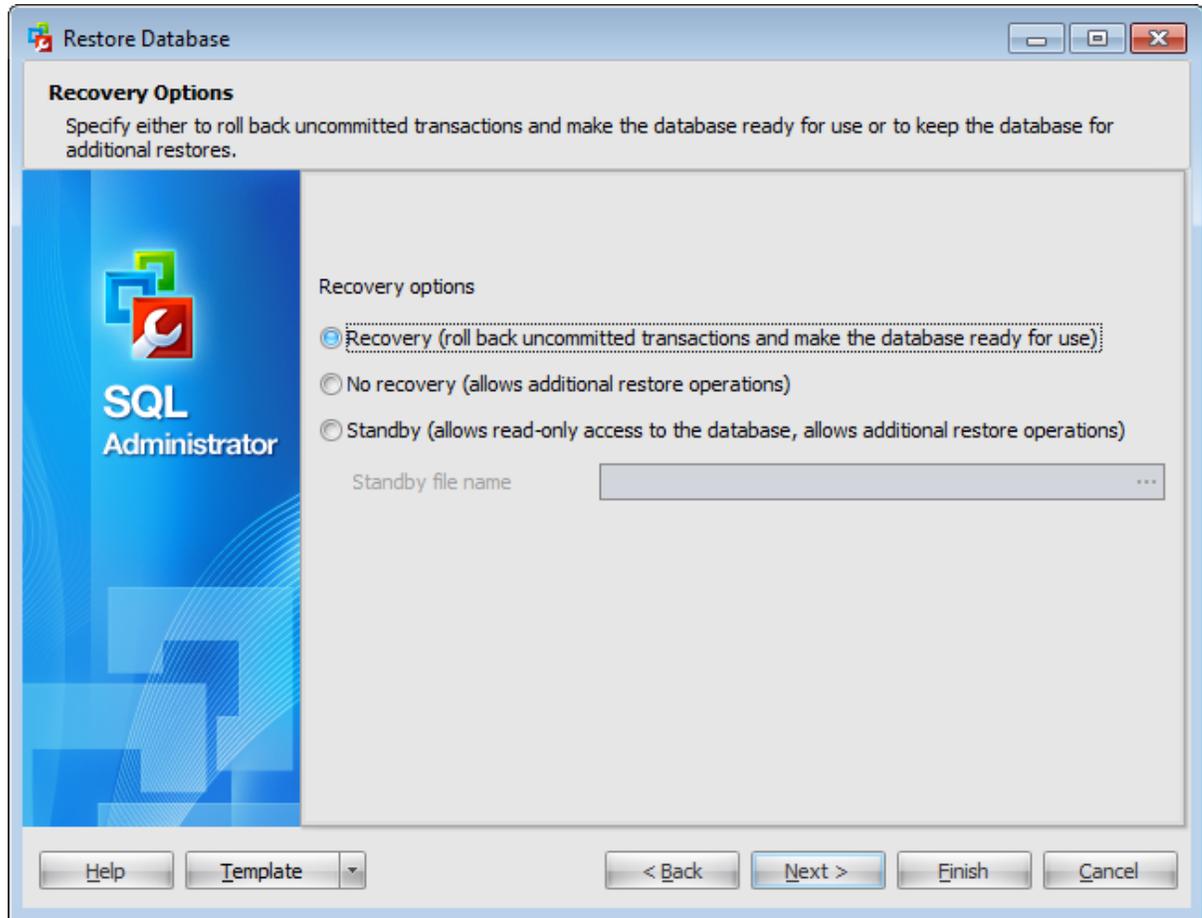
 **Continue on checksum errors**

This option determines whether the restore operation will be stopped or forced to continue if an error occurs.

Click the **Next** button to proceed to the [Setting recovery options](#) step of the wizard.

### 4.3.7 Setting recovery options

Use this step of the wizard to set recovery options.



#### Recovery options

This group of options allows you to specify the recovery options to be applied to the restore operation.

*Recovery*

This option specifies that the restore operation rolls back any uncommitted transactions. After the recovery process the database is ready for use.

*No recovery*

This option specifies that the restore operation does not roll back any uncommitted transactions.

**Note:** When this option is selected, the database is not usable in this intermediate, non-recovered state.

When used with a file or filegroup restore operation, this option forces the database to remain in the restoring state after the restore operation. This is useful in either of the following situations:

- a restore script is being run and the log is always being applied;
- a sequence of file restores is used and the database is not intended to be usable

between two of the restore operations.

*Standby*

This option allows the database to be brought up for read-only access between transaction log restores and can be used with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores.

**Standby file name**

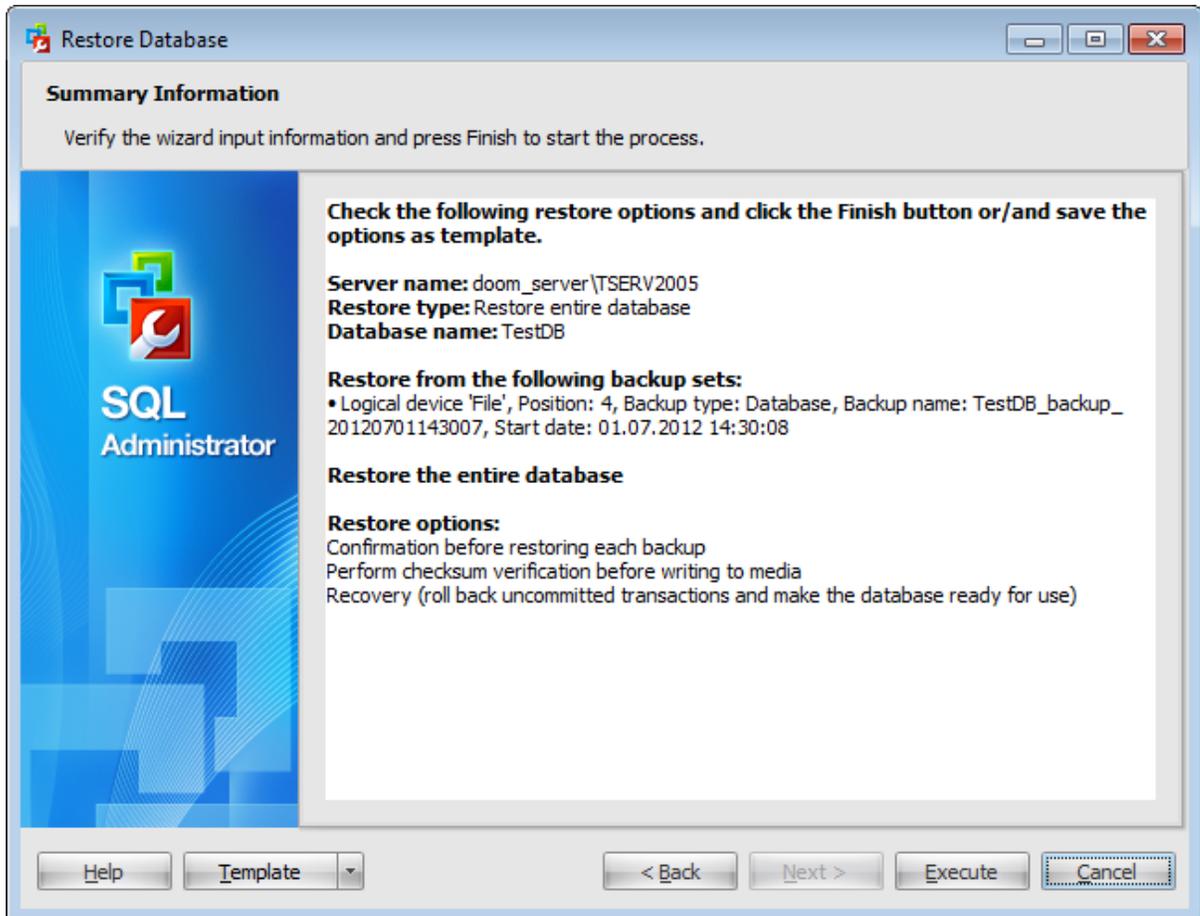
Specify the undo file name, so that the recovery effects can be undone. The size required for the undo file depends on the volume of undo actions resulting from uncommitted transactions. The same undo file can be used for consecutive restores of the same database.

**NB:** If free disk space is exhausted on the drive containing the specified undo file name, the restore operation stops.

Click the **Next** button to proceed to the [Running database restore](#) step of the wizard.

### 4.3.8 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the restore database process.

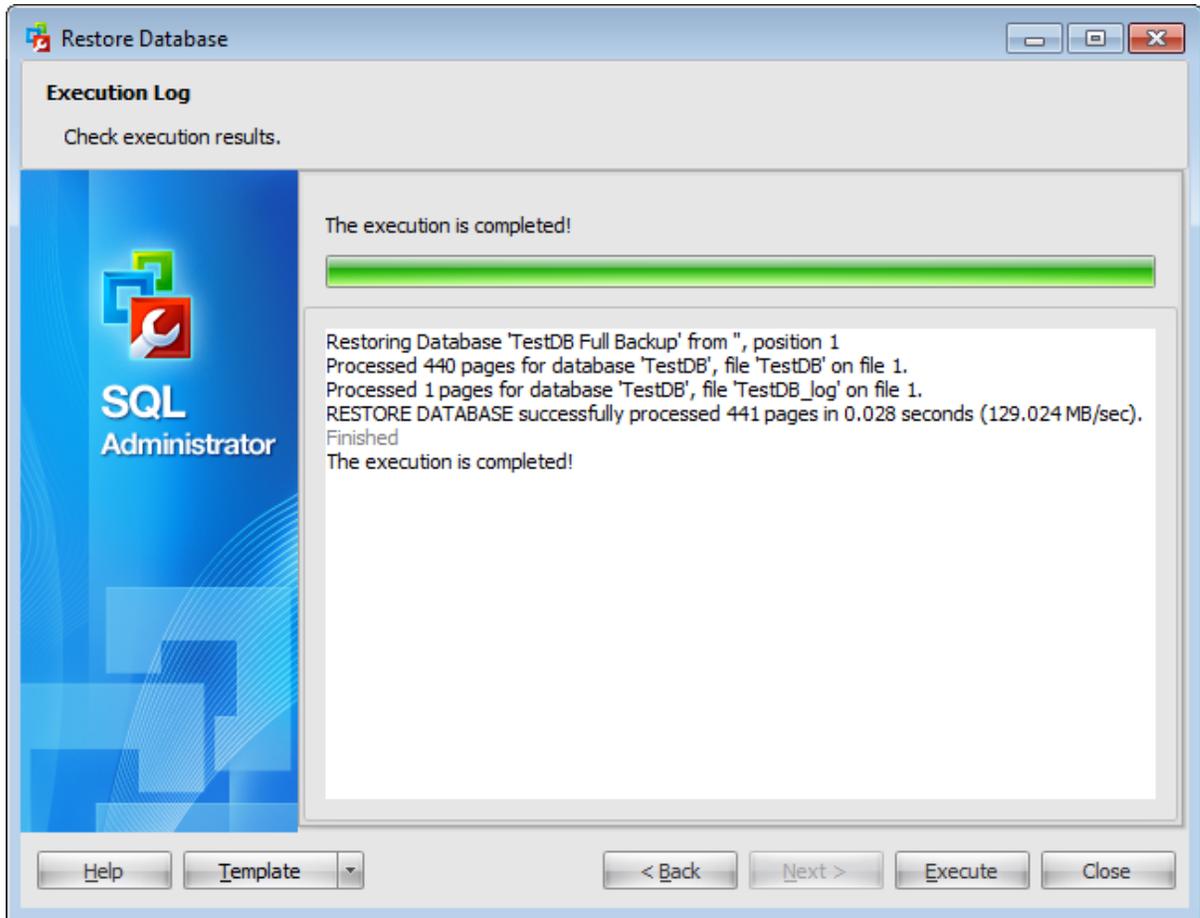


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing restore database task](#) step.

### 4.3.9 Executing restore database task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.4 Attach database

**Attach Database Wizard** allows you to attach an existing database that is made up of one or more files to the instance of Microsoft® SQL Server™ (with SQL Server routines used).

This operation is used to attach a database with the specified name and using the specified data files.

To run the wizard, select the **Tools |  Attach Database...** [main menu](#) item or click ** Attach Database...** button on the [toolbar](#).

**Note:** The database must first be detached. Attempting to attach a database that has not been detached will return an error.

The wizard consists of the following steps:

- [Selecting server](#)
- [Defining database files](#)
- [Attaching database](#)

---

**See also:**

[Maintenance plans](#)

[Backup database](#)

[Restore database](#)

[Detach database](#)

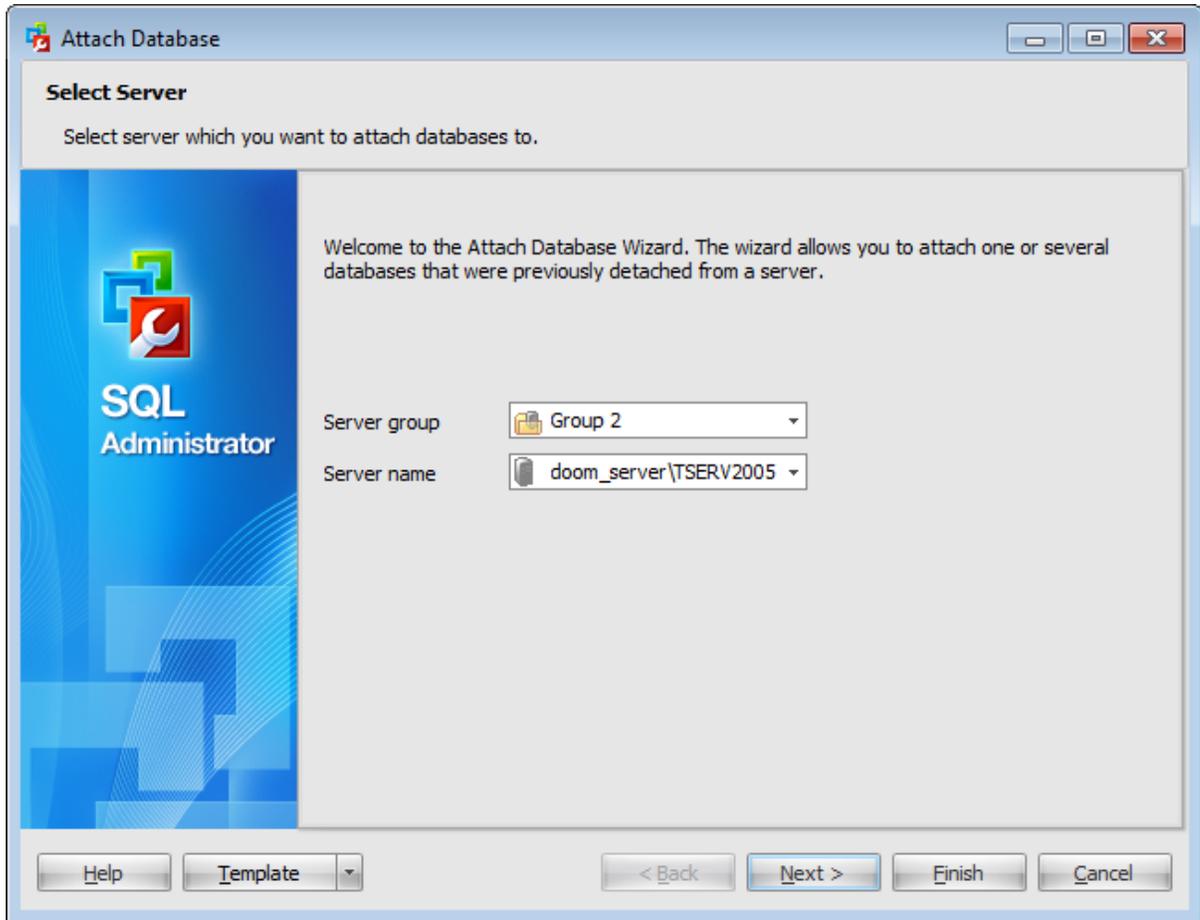
[Check Database](#)

[Rebuild/Reorganize Indexes](#)

[Shrink database](#)

#### 4.4.1 Selecting server

This step of the wizard allows you to specify the **server** to attach database to.



##### **Server group**

Use the drop-down list to select a server group where the server is stored.

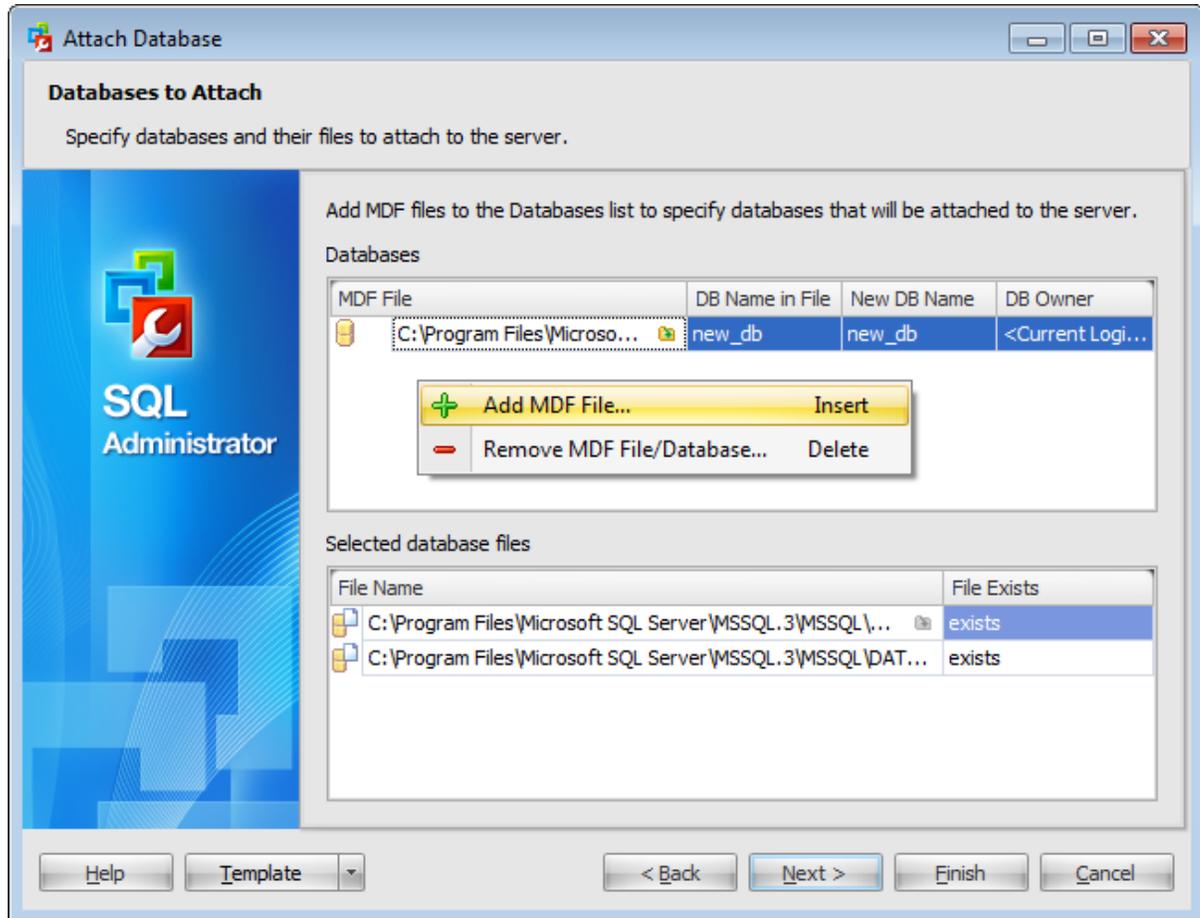
##### **Server name**

Use the drop-down list to select the host/instance name to attach database to.

Click the **Next** button to proceed to the [Defining database files](#) step of the wizard.

## 4.4.2 Defining database files

Use this step of the wizard to **select the files** to attach to the server as a database.



### Databases

Right-click this area and select **Add MDF File...** context menu item to specify the path and the name of the file to attach to the server.

To delete files from this list select **Remove MDF File/Database...** item of the context menu.

### Selected database files

The file selected in the upper grid points to the other files (*\*.ndf* and *\*.ldf*) in the database which are displayed in this list along with the *\*.mdf* file.

When you choose a *\*.mdf* file SQL Administrator checks for the rest of necessary files by paths stored in it and then these files are added automatically in the grid.

When you attach a database, all data files (MDF and NDF files) must be available. If any data file has a different path from when the database was first created or last attached, you must specify the current path of the file by pressing the  button or choosing the **Add File...** context menu item.

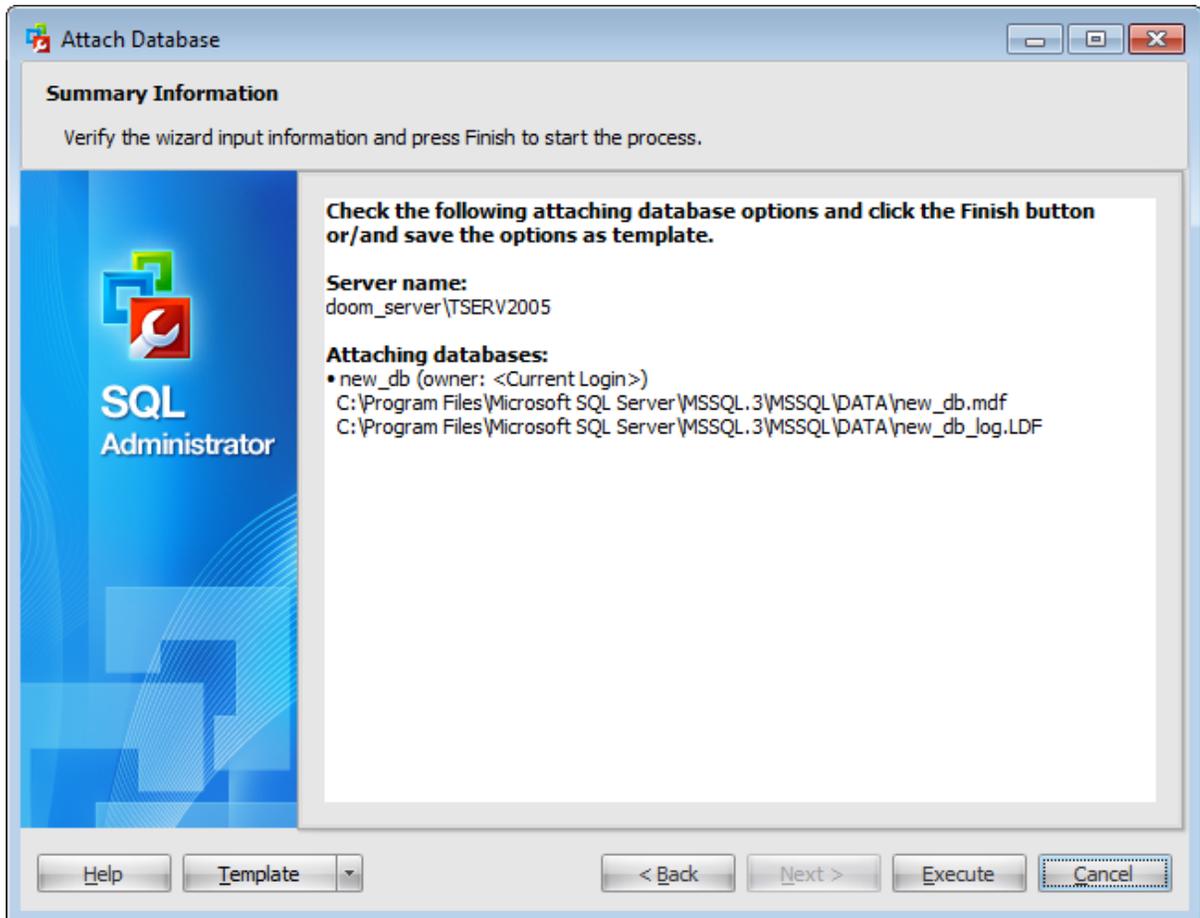
To delete a file from the grid select it and choose the **Remove File...** context menu item or use the 'Delete' button.

**Note:** File paths can be edited manually. The context menu of the file path allows you to perform common editing operations: *Undo/Redo, Cut, Copy, Paste, Delete, Select all*.

When you are done, click the **Next** button to proceed to the [Attaching database](#) step of the wizard.

### 4.4.3 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the attach database process.

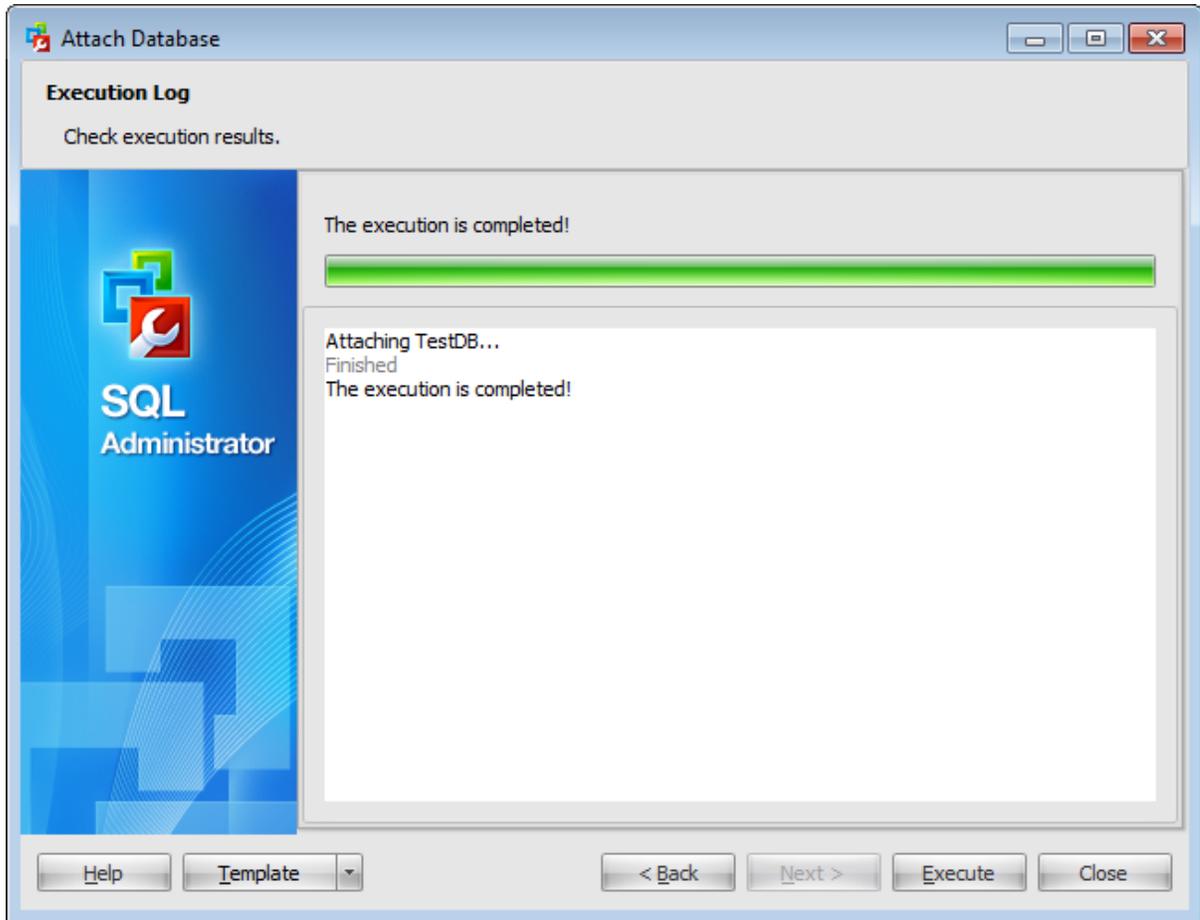


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing attach database task](#) step.

#### 4.4.4 Executing attach database task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.5 Detach database

**Detach Database Wizard** allows you to detach a database from the instance of Microsoft® SQL Server™ (with SQL Server routines used).

This operation is used to detach the specified database with or without updating statistics before the database is detached.

To run the wizard, select the **Tools** |  **Detach Database...** [main menu](#) item, or click  **Detach Database...** button on the [toolbar](#).

The wizard consists of the following steps:

- [Selecting server](#)
- [Selecting databases](#)
- [Detaching database](#)

---

**See also:**

[Maintenance plans](#)

[Backup database](#)

[Restore database](#)

[Attach database](#)

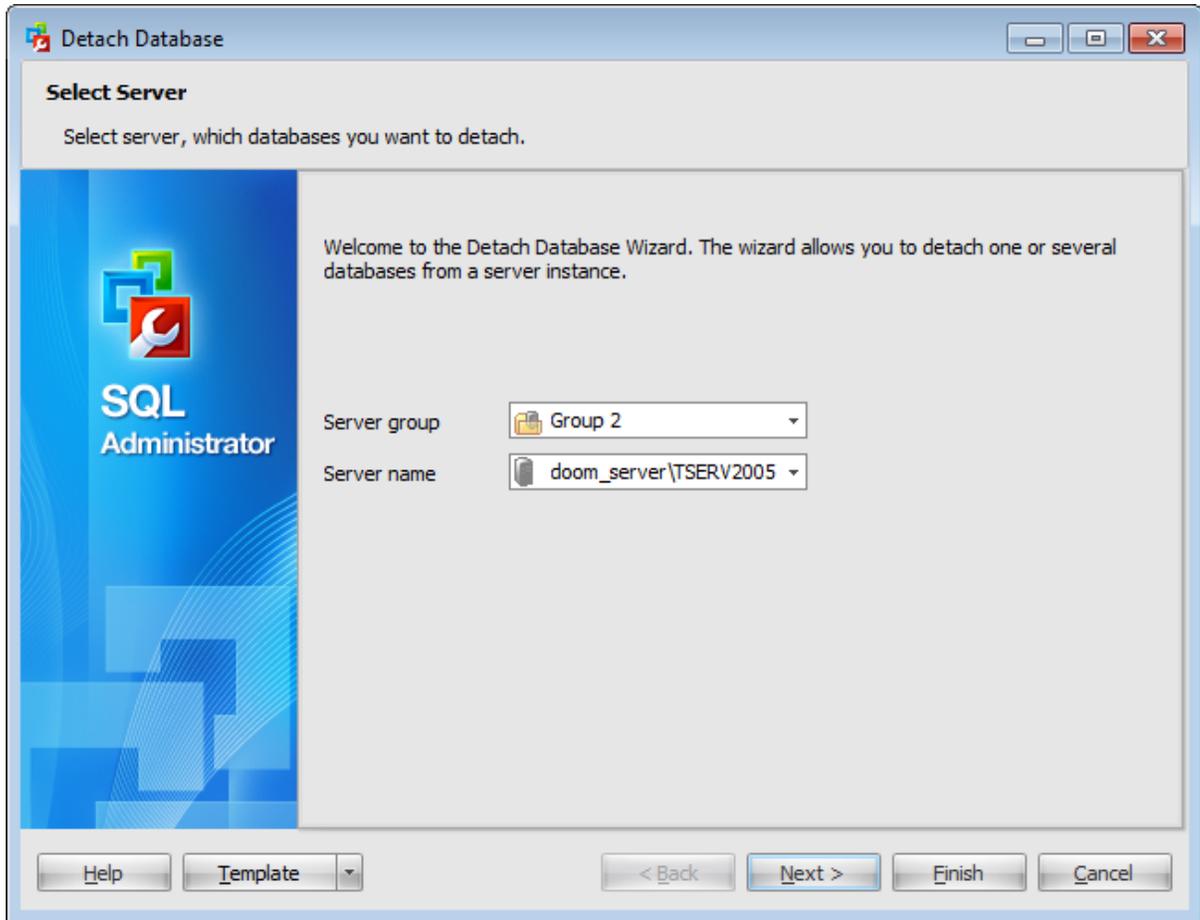
[Check Database](#)

[Rebuild/Reorganize Indexes](#)

[Shrink database](#)

### 4.5.1 Selecting server

This step of the wizard allows you to specify the **server** to detach database from.



#### **Server group**

Use the drop-down list to select a server group where the server is stored.

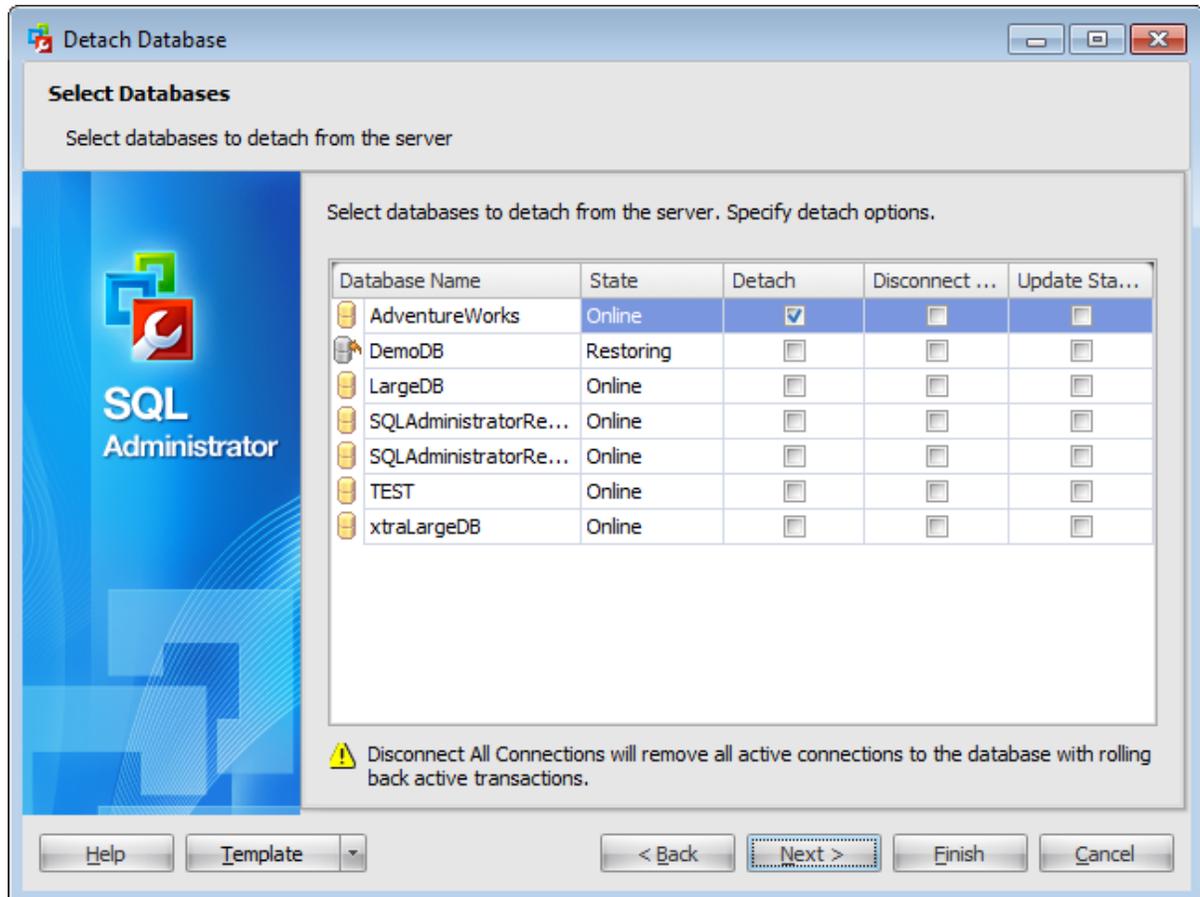
#### **Server name**

Type in or use the drop-down list to select the host/instance name to detach database from.

Click the **Next** button to proceed to the [Selecting databases](#) step of the wizard.

## 4.5.2 Selecting databases

At this step of the wizard you should **specify the databases** to be detached from the instance of SQL Server.



The grid displays all databases located on the selected server and their states. To select a database to be detached from a server select it in the list and set the respective flag in the **Detach** column.

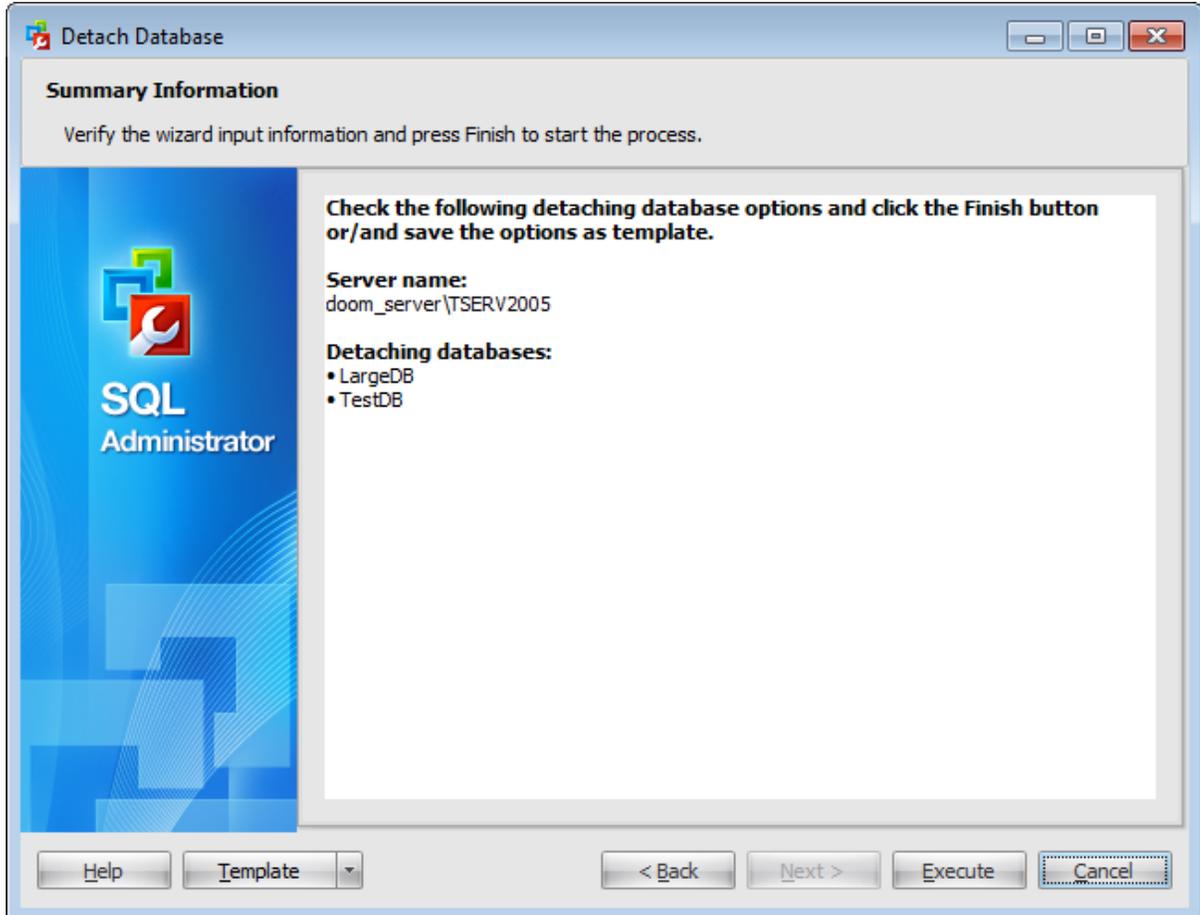
Note that you can detach a database only if it has no active connections and it is not enabled for replication. To clear the current connections to the database, select it in the list and set the respective flag in the **Disconnect all connections** column.

To update the database statistics before detaching the database, select it in the list and set the respective flag in the **Update Statistics** column.

Click the **Next** button to proceed to the [Detaching database](#) step of the wizard.

### 4.5.3 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the detach database process.

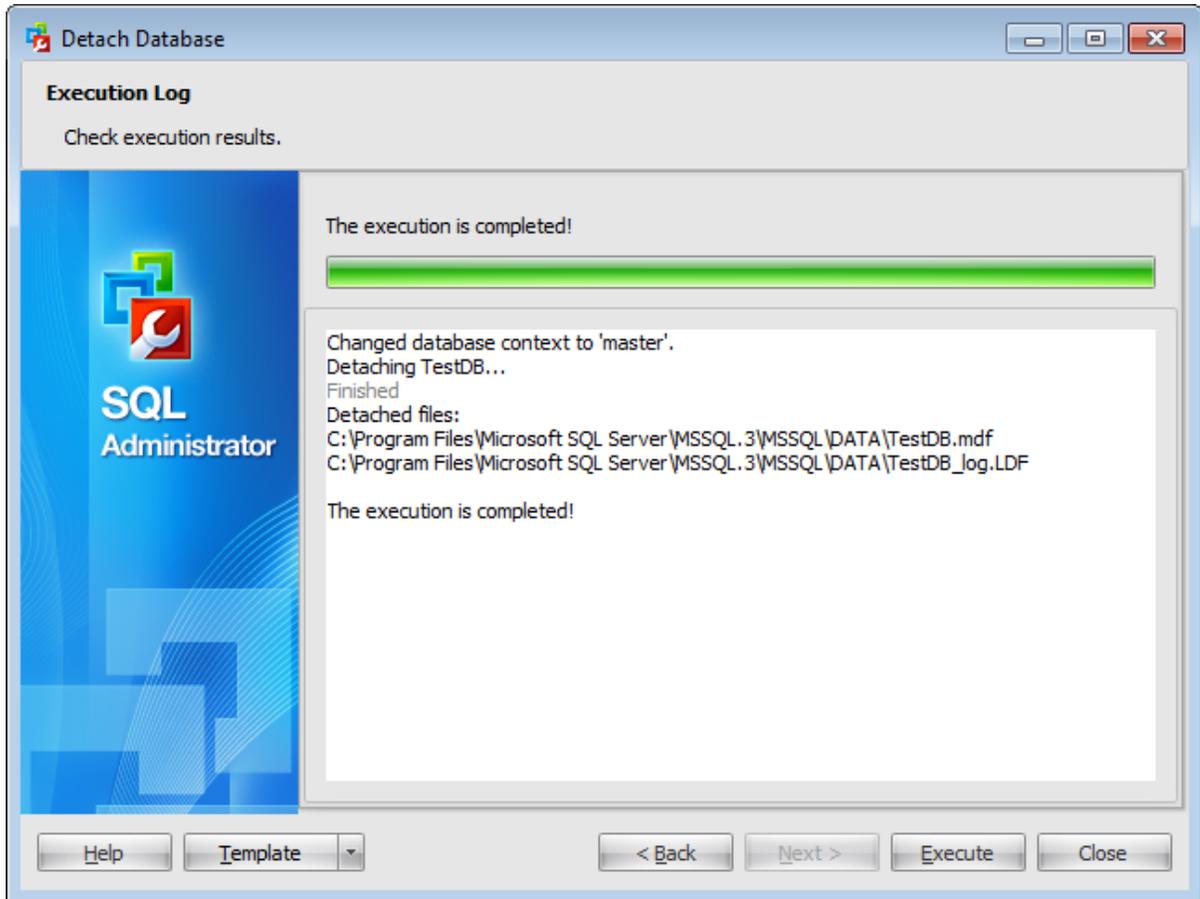


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing detach database task](#) step.

#### 4.5.4 Executing detach database task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.6 Check database

**Check Database** node is used to manage check database maintenance plans.

To open this tab choose the **Maintenance | Check Database** node of the [Server Explorer](#) tree.

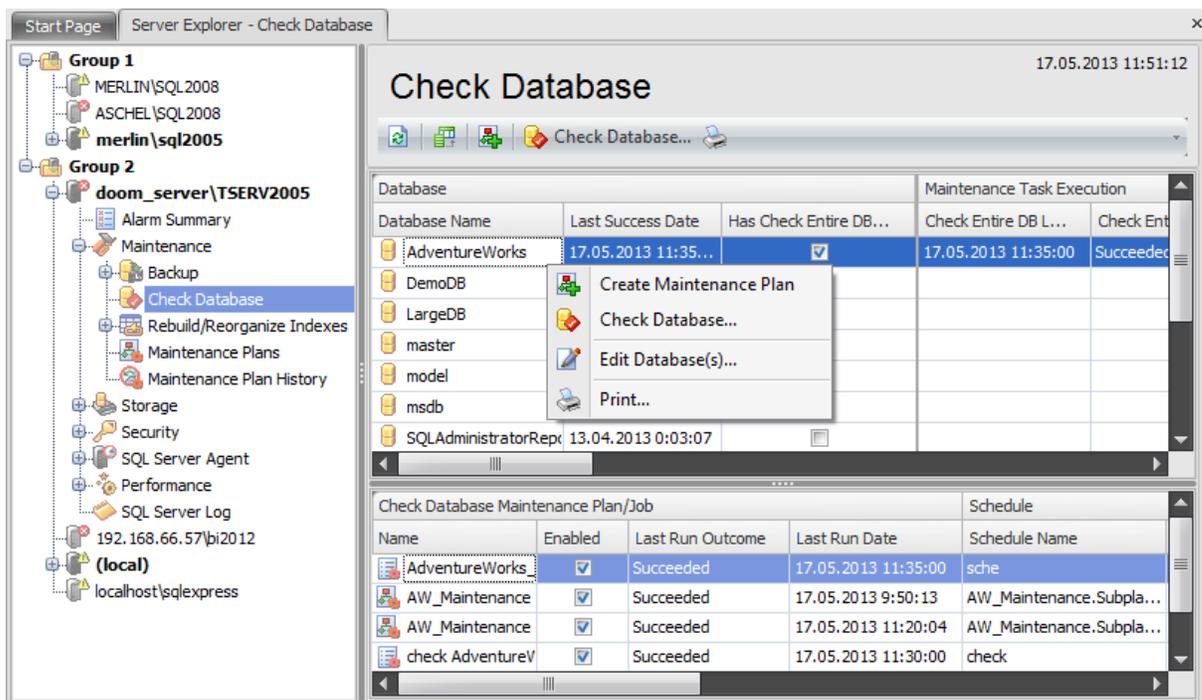
Use the context menu or buttons on the work area toolbar to manage checking:

 [create](#) a new maintenance plan

 [check](#) database

 [edit database](#)

 create and [print](#) the report



The upper area is the grid with the following columns: *Database Name*, *Last Success Date*, *Has Check Entire DB Schedule*, *Check Entire DB Last Execution*, *Check Entire DB Last Outcome*, *Schedule Period during a Month* (It represents a grey axis with points which correspond to 6-hours intervals, days and weeks. Green lines show the planned time of the checking). If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

When you choose a particular database the lower area displays the following detailed information about the check database [maintenance plans](#): **Check Database Maintenance Plan/Job**: *Name*, *Enabled*, *Last Run Outcome*, *Last Run Date*; **Schedule**: *Schedule Name*, *Enabled*, *Description*; **Task/Job Step**: *Name*, *Last Run Outcome*, *Last Run Duration*, *Last Run Message*.

To open a maintenance plan in the [Maintenance Plan Editor](#), double-click it in the grid or select it and press the 'Enter' button. Alternatively, you can right-click the maintenance plan alias and select the **Edit Plans/Jobs** context menu item.

**See also:**[Maintenance plans](#)[Backup database](#)[Restore database](#)[Attach database](#)[Detach database](#)[Rebuild/Reorganize Indexes](#)[Shrink database](#)

### 4.6.1 Check database wizard

**Check Database Wizard** allows you to perform the check database operation on your SQL Server system (the *DBCC CHECKDB* Transact-SQL statement is issued).

This operation is used to check the allocation, structural, and logical integrity of all the objects in the specified database.

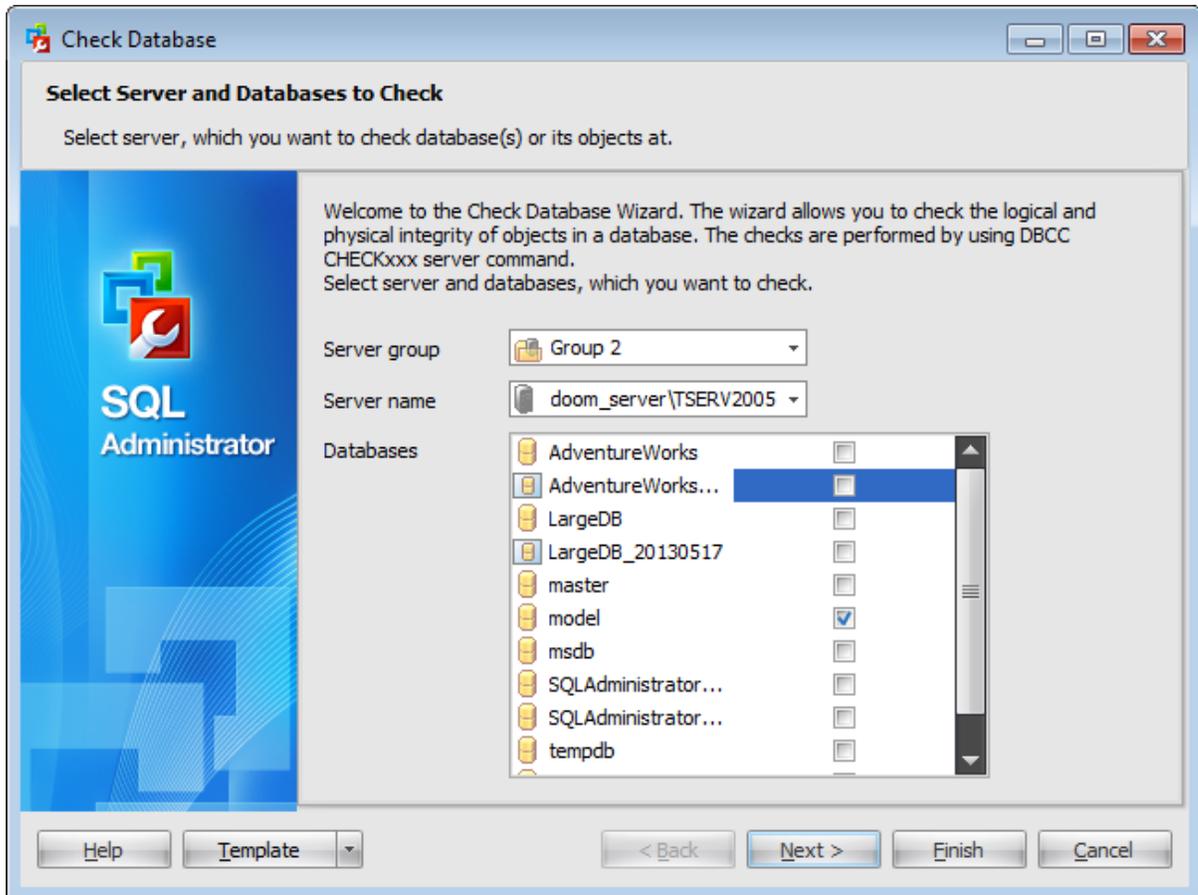
To run the wizard, select the **Tools |  Check Database...** [main menu](#) item or click  **Check Database...** button on the [toolbar](#).

The wizard consists of the following steps:

- [Selecting server and databases to check](#)
- [Setting schedule options](#)
- [Specifying checks to perform](#)
- [Setting repair options](#)
- [Setting check options](#)
- [Checking database](#)

#### 4.6.1.1 Selecting server and databases to check

This step of the wizard allows you to select the **server** containing the database for checking and databases.



##### Server group

Use the drop-down list to select a server group where the server is stored.

##### Server name

Use the drop-down list to select the server where the database resides.

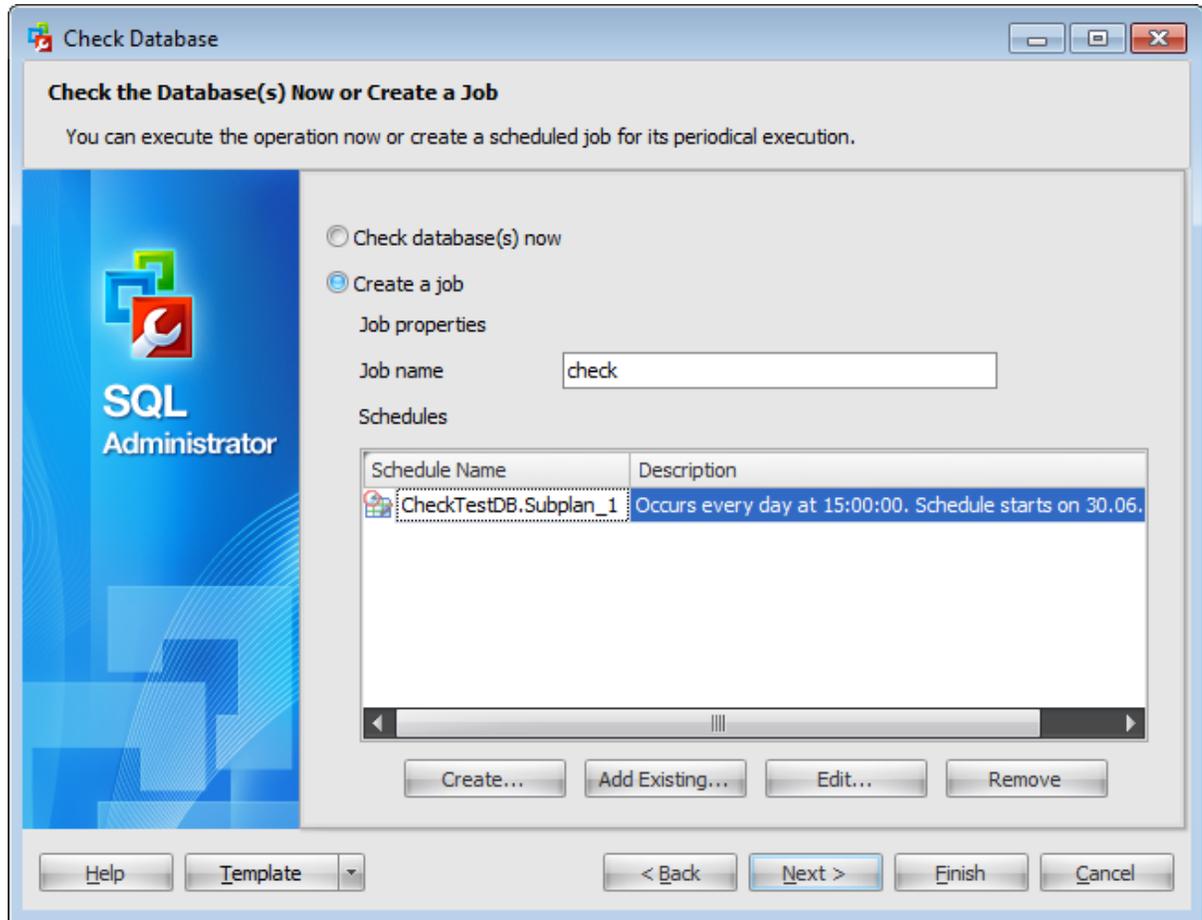
##### Databases

To choose a database to check select it in the list and set the respective flag.

Click the **Next** button to proceed to the [Setting schedule options](#) step of the wizard.

#### 4.6.1.2 Setting schedule options

Using this step you can define when the checking will be executed.



##### **Check database(s) now**

Check this option to start the checking process after all necessary options are set.

##### **Create a job**

Select this option to create a **scheduled** check.

**Note:** To create a job you need to run [SQL Server Agent](#) service on the server.

Define **job properties**:

##### **Job name**

Enter a name of the new checking job.

##### **Schedules**

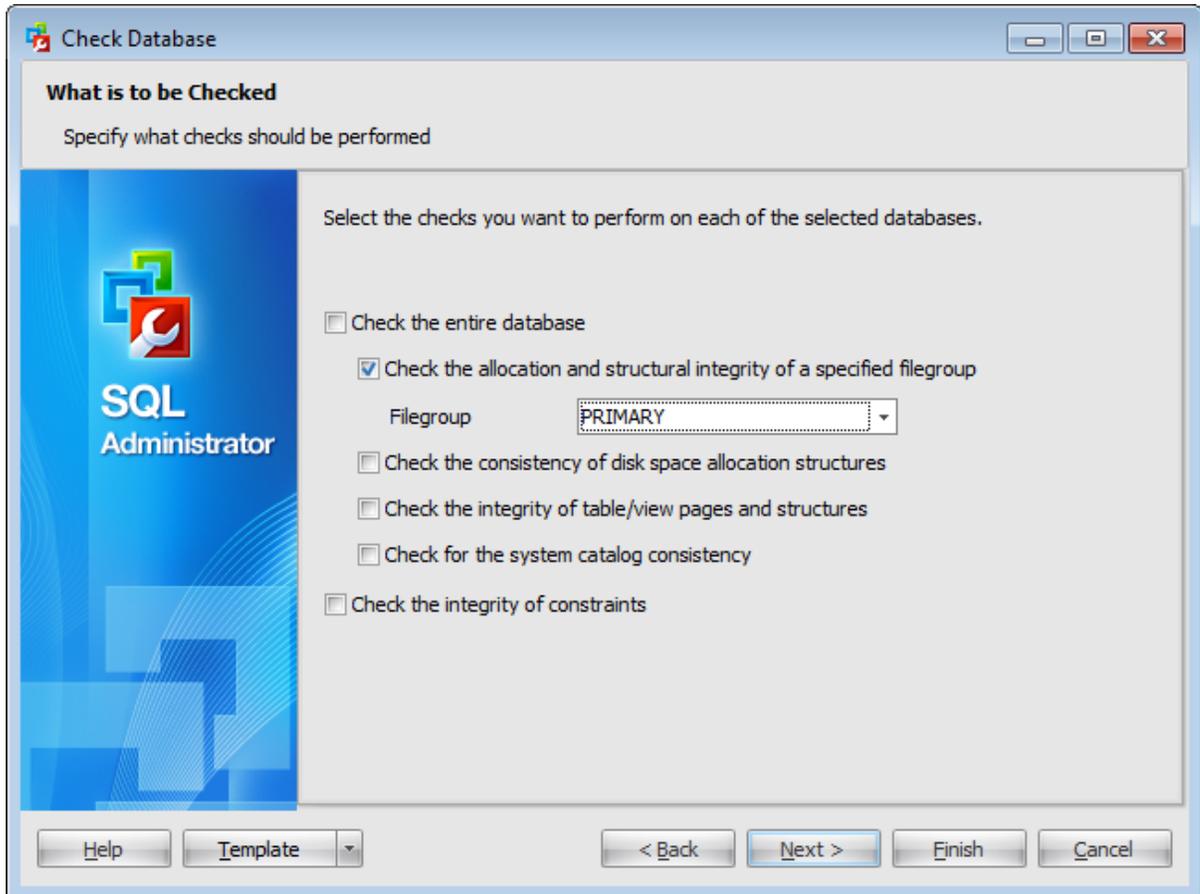
This list displays all schedules that cause the checking job to begin running. You can add existing schedule or [create](#) a new one.

Click the **Next** button to proceed to the [Specifying checks to perform](#) step of the wizard.



#### 4.6.1.3 Specifying checks to perform

Use this step of the wizard to specify the checks to perform on each selected database.



**Check the entire database** (*DBCC CHECKDB*)

Select this option to execute a check of the entire database including allocation checks in addition to checks of index structure and data integrity.

**Check the allocation and structural integrity of a specified filegroup** (*DBCC CHECKFILEGROUP*)

Select this option to check the allocation and structural integrity of all tables and indexed views in the specified filegroup of the current database.

**Filegroup**

Use the drop-down list to select a filegroup to be checked.

**Check the consistency of disk space allocation structures** (*DBCC CHECKALLOC*)

If this option is selected then checking allocation and page usage is performed.

**Check the integrity of table/view pages and structures** (*DBCC CHECKTABLE*)

Select this option to perform checking of a single table, indexed view, or even a particular index.

**Check for the system catalog consistency** (*DBCC CHECKCATALOG*)

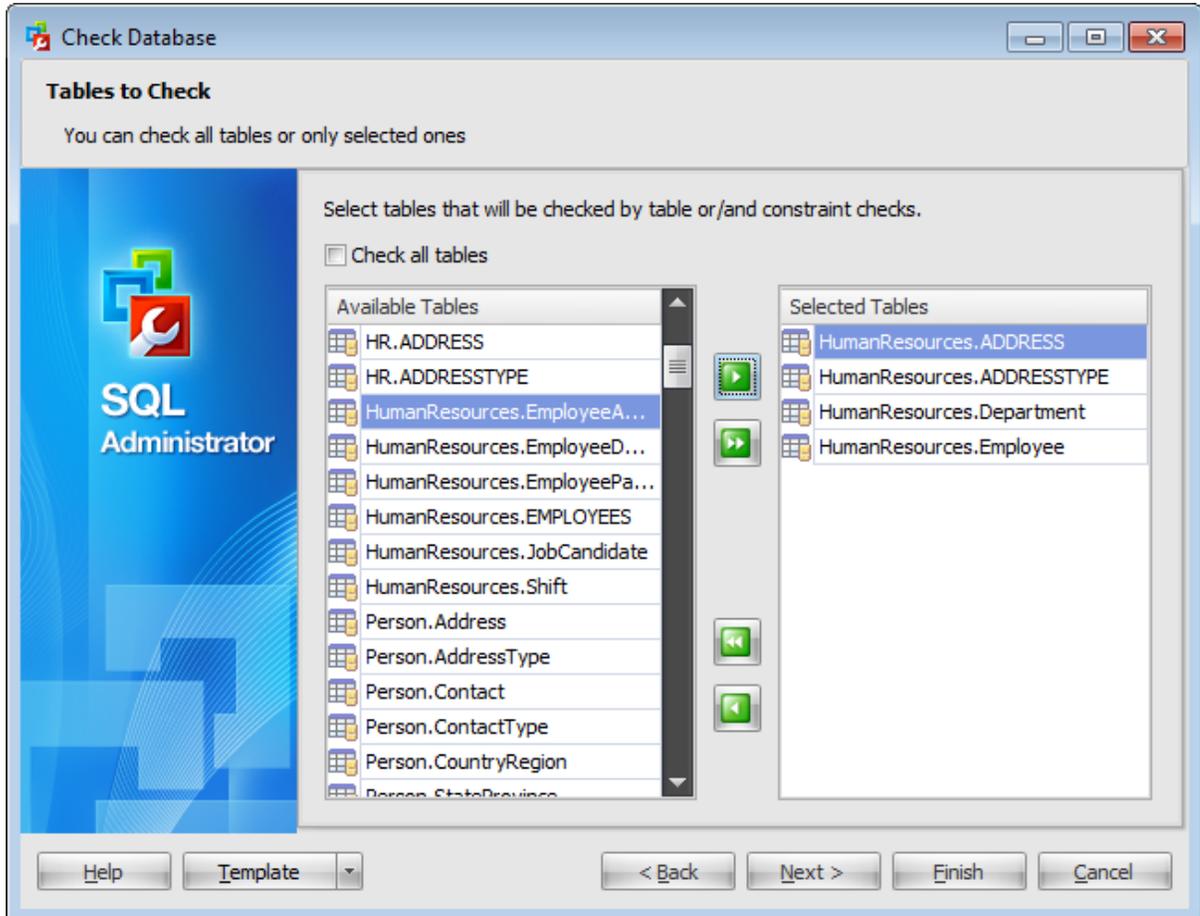
Select this option to check the consistency in and between system tables of a specified database.

**Checks the integrity of constraints**

If this option is selected the checking of the integrity of a all constraints is performed.

#### 4.6.1.4 Selecting tables to check

This step appears if the  **Check the integrity of table/view pages and structures** option was selected on the [previous step](#).



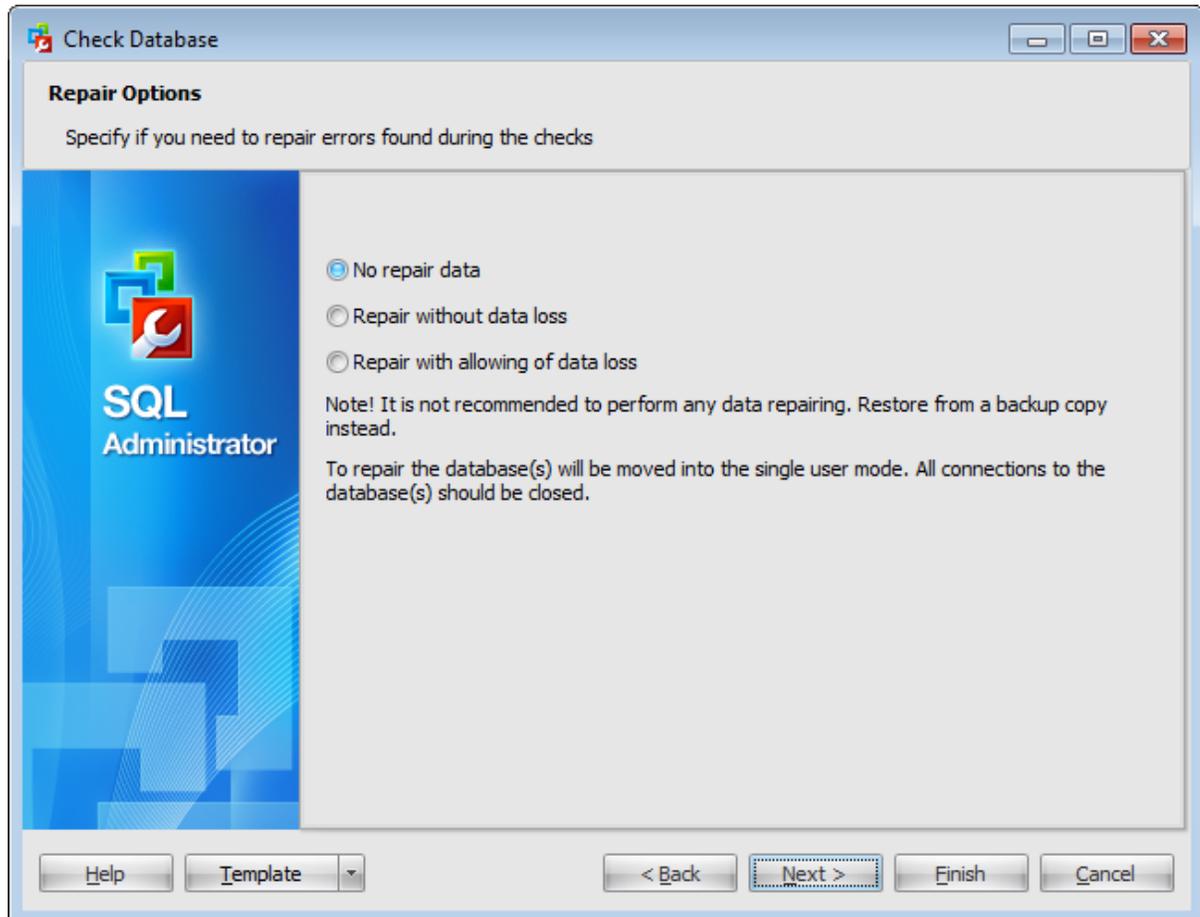
If the  **Check all tables** is set then all tables of a database will be checked.

To select a table, you need to move it from the **Available Tables** list to the **Selected Tables** list. Use the  buttons or drag-and-drop operations to move the tables from one list to another.

Click the **Next** button to proceed to the [Setting repair options](#) step of the wizard.

#### 4.6.1.5 Setting repair options

Use this step of the wizard to specify repair error options. This step appears only if  *Check database(s) now* option was selected on the [second](#) step. It depends on the checks select on the [third step](#) as well.



**No repair data**

Selecting this option equals using the *REPAIR\_FAST* argument. This argument enables fast repair of errors which is maintained for backward compatibility.

**Repair without data loss**

Selecting this option equals using the *REPAIR\_REBUILD* argument. This argument performs quick repairs, such as repairing missing rows in non-clustered indexes, and more time-consuming repairs, such as rebuilding an index. These repairs can be performed without risk of data loss.

**Repair with allowing of data loss**

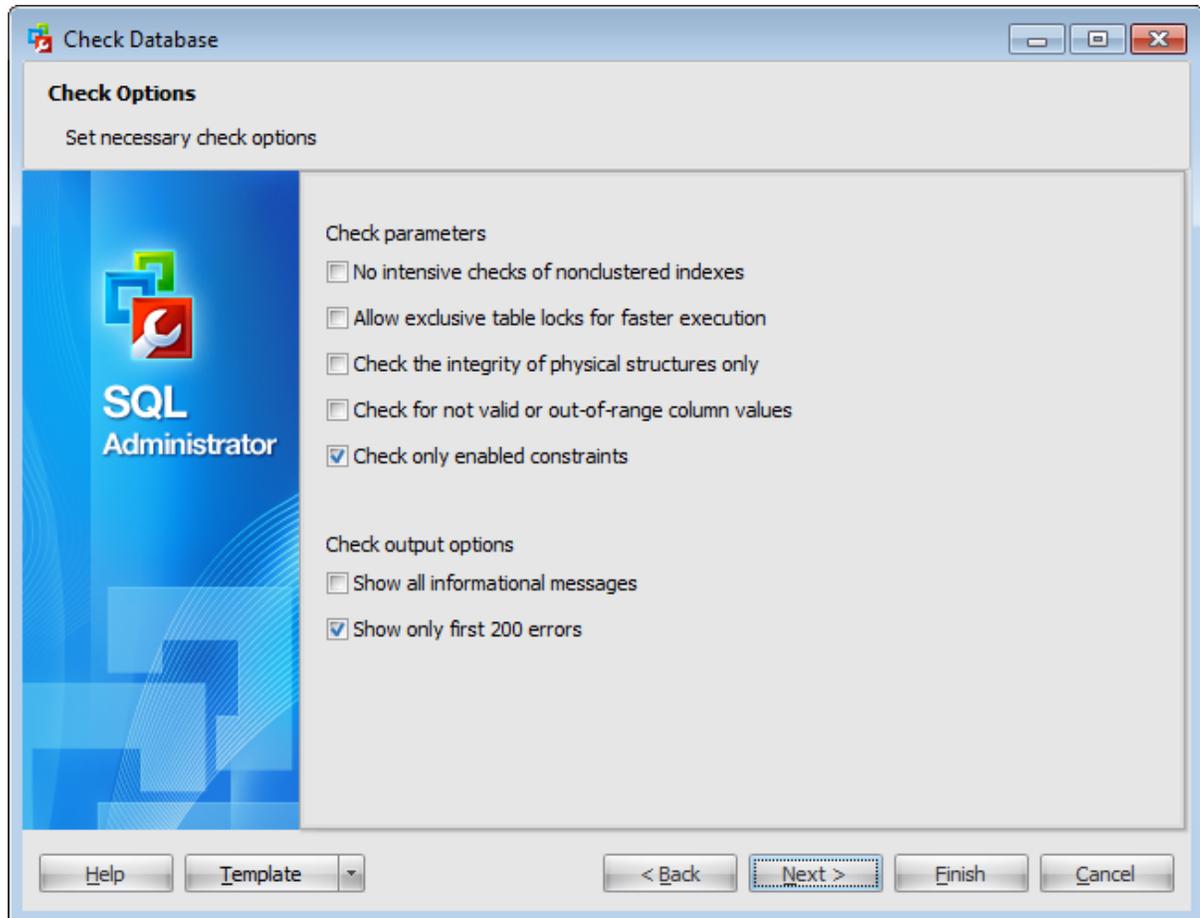
When this argument (*REPAIR\_ALLOW\_DATA\_LOSS*) is used, all reported errors are repaired. These repairs can cause data loss.

**Note:** It is not recommended to perform any data repairing. Restore from a backup copy instead.

Click the **Next** button to proceed to the [Setting check options](#) step of the wizard.

#### 4.6.1.6 Setting check options

Use this step of the wizard to **options** for the check database operation. The set options displayed depends on the selected checks to perform on the [third step](#).



#### Check parameters

**No intensive checks of nonclustered indexes**

This argument (*NOINDEX*) specifies that intensive checks of non-clustered indexes for user tables should not be performed; therefore, the overall execution time is decreased.

**Note:** This option is available only if  *No repair data* has been selected on the [previous step](#).

**Extended index logical checks (intensive operation)**

This argument (*EXTENDED\_LOGICAL\_CHECKS*) performs logical consistency checks on an indexed view, XML indexes, and spatial indexes, where present if the compatibility level is 100 (SQL Server 2008) or higher.

**Allow take exclusive table locks to execute faster**

This argument (*TABLOCK*) causes *DBCC CHECKTABLE* to obtain a shared table lock instead of using an internal database snapshot. *TABLOCK* will cause *DBCC CHECKTABLE* to run faster on a table under heavy load, but decreases the concurrency available on the table

while *DBCC CHECKTABLE* is running.

**Check the integrity of physical structures only**

This argument (*PHYSICAL\_ONLY*) limits the checking to the integrity of the physical structure of the page, record headers and the physical structure of B-trees. Designed to provide a small overhead check of the physical consistency of the table, this check can also detect torn pages, and common hardware failures that can compromise data.

**Note:** This option is available only if  *No repair data* has been selected on the [previous step](#).

**Check for valid or out-of-range column values**

This argument (*DATA\_PURITY*) causes *DBCC CHECKTABLE* to check the table for column values that are not valid or out-of-range. For example, *DBCC CHECKTABLE* detects columns with date and time values that are larger than or less than the acceptable range for the datetime data type; or decimal or approximate-numeric data type columns with scale or precision values that are not valid.

**Note:** If *PHYSICAL\_ONLY* is specified, column-integrity checks are not performed.

### Check output options

**Show all information messages**

If this option is selected then an unlimited number of information messages is displayed.

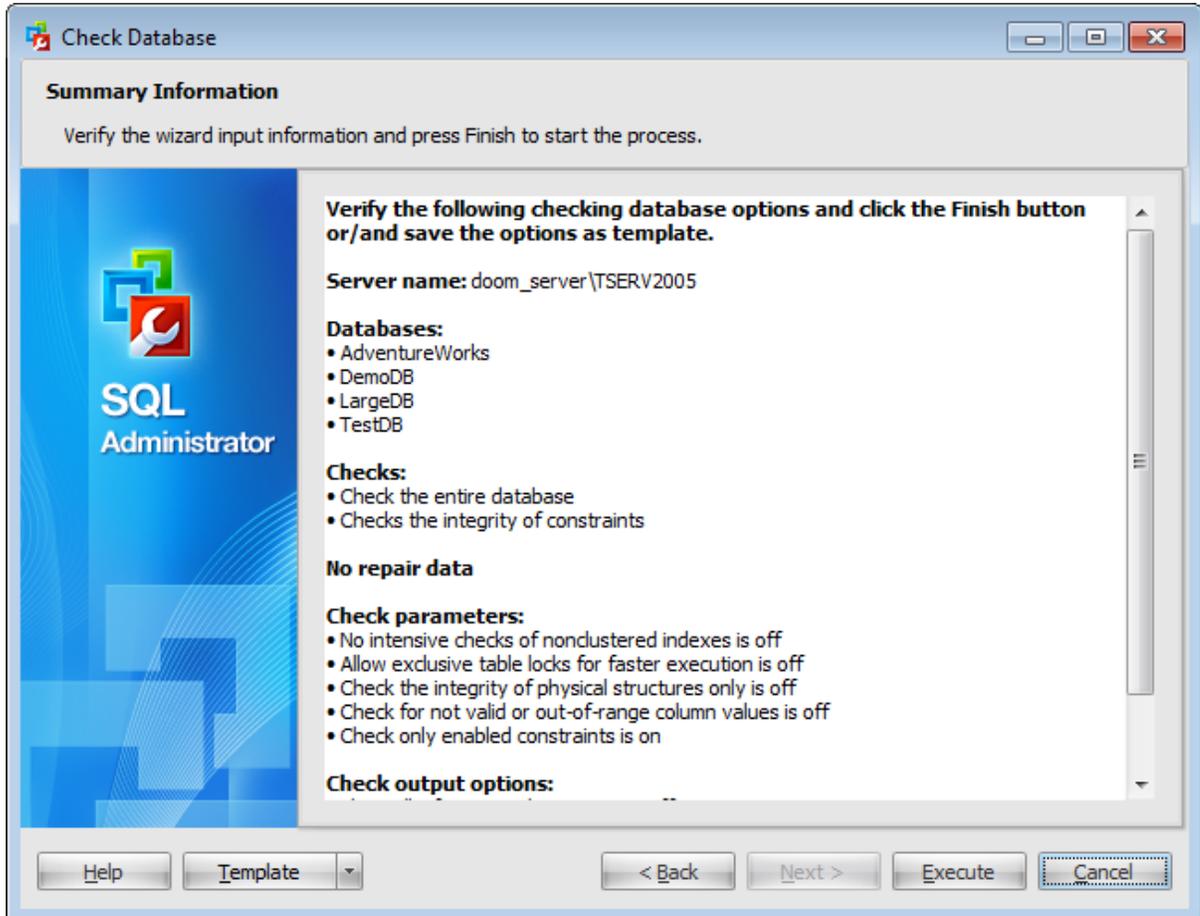
**Show only the first 200 errors**

If this option is selected then only first 200 errors will be displayed. By default all error messages are displayed.

Click the **Next** button to proceed to the [Checking database](#) step of the wizard.

#### 4.6.1.7 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the check database process.

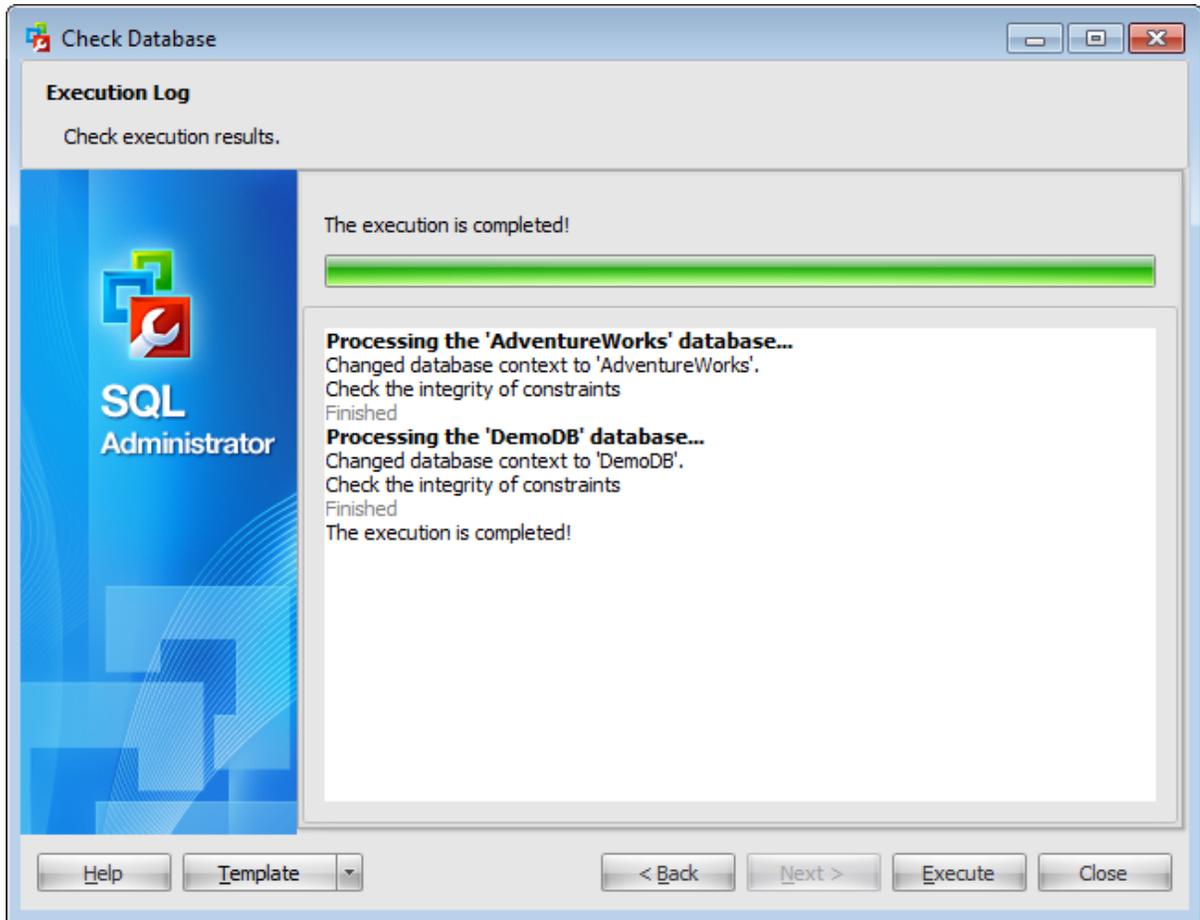


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing check database task](#) step.

#### 4.6.1.8 Executing check database task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.7 Rebuild/Reorganize Indexes

**Rebuild\Reorganise Indexes** node is used to rebuild\reorganise indexes and manage maintenance plans.

To open this tab choose the **Maintenance | Rebuild\Reorganize Indexes** node of the [Server Explorer](#) tree.

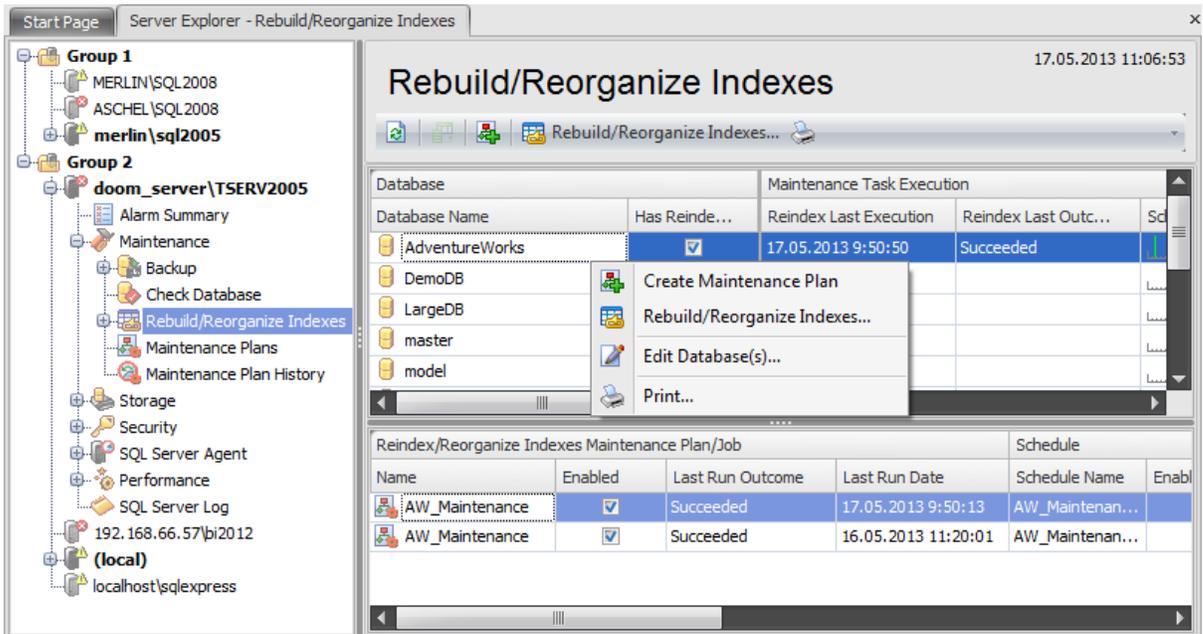
Use the context menu or buttons on the work area toolbar to manage indexes:

 [create](#) a new maintenance plan

 [rebuild/reorganize](#) indexes

 [edit database](#)

 create and [print](#) the report



Database Name	Has Reinde...	Reindex Last Execution	Reindex Last Outc...	Sc
AdventureWorks	<input checked="" type="checkbox"/>	17.05.2013 9:50:50	Succeeded	
DemoDB				
LargeDB				
master				
model				

Name	Enabled	Last Run Outcome	Last Run Date	Schedule Name	Enabl
AW_Maintenance	<input checked="" type="checkbox"/>	Succeeded	17.05.2013 9:50:13	AW_Maintenan...	
AW_Maintenance	<input checked="" type="checkbox"/>	Succeeded	16.05.2013 11:20:01	AW_Maintenan...	

The upper area is the grid with the following columns: *Database Name*, *Has Reindex Schedule*, *Reindex Last Execution*, *Reindex Last Outcome*, *Schedule Period during a Month* (It represents a grey axis with points which correspond to 6-hours intervals, days and weeks. Green lines show the planned time of the rebuilding.). If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

When you choose a particular database the lower area displays the following detailed information about the reindex [maintenance plans](#): **Reindex/Reorganize Indexes Maintenance Plan/Job**: *Name*, *Enabled*, *Last Run Outcome*, *Last Run Date*; **Schedule**: *Schedule Name*, *Enabled*, *Description*; **Task/Job Step**: *Name*, *Last Run Outcome*, *Last Run Duration*, *Last Run Message*.

To open a maintenance plan in the [Maintenance Plan Editor](#), double-click it in the grid or select it and press the 'Enter' button. Alternatively, you can right-click the maintenance plan alias and select the **Edit Plans/Jobs** context menu item.

**See also:**[Maintenance plans](#)[Backup database](#)[Restore database](#)[Attach database](#)[Detach database](#)[Check Database](#)[Shrink database](#)

### 4.7.1 Rebuild/Reorganize Indexes Wizard

**Rebuild/Reorganize Indexes Wizard** allows you to perform index management operations in your SQL Server database: *reorganizing, rebuilding, disabling* (the `ALTER INDEX REORGANIZE ... REBUILD ... DISABLE` Transact-SQL statements are issued).

When indexes have pages in which the logical ordering based on the key value, does not match the physical ordering inside the data file, these indexes are called *fragmented*. Highly fragmented indexes may considerably decrease query performance. Index fragmentation is remedied by either reorganizing or by rebuilding an index.

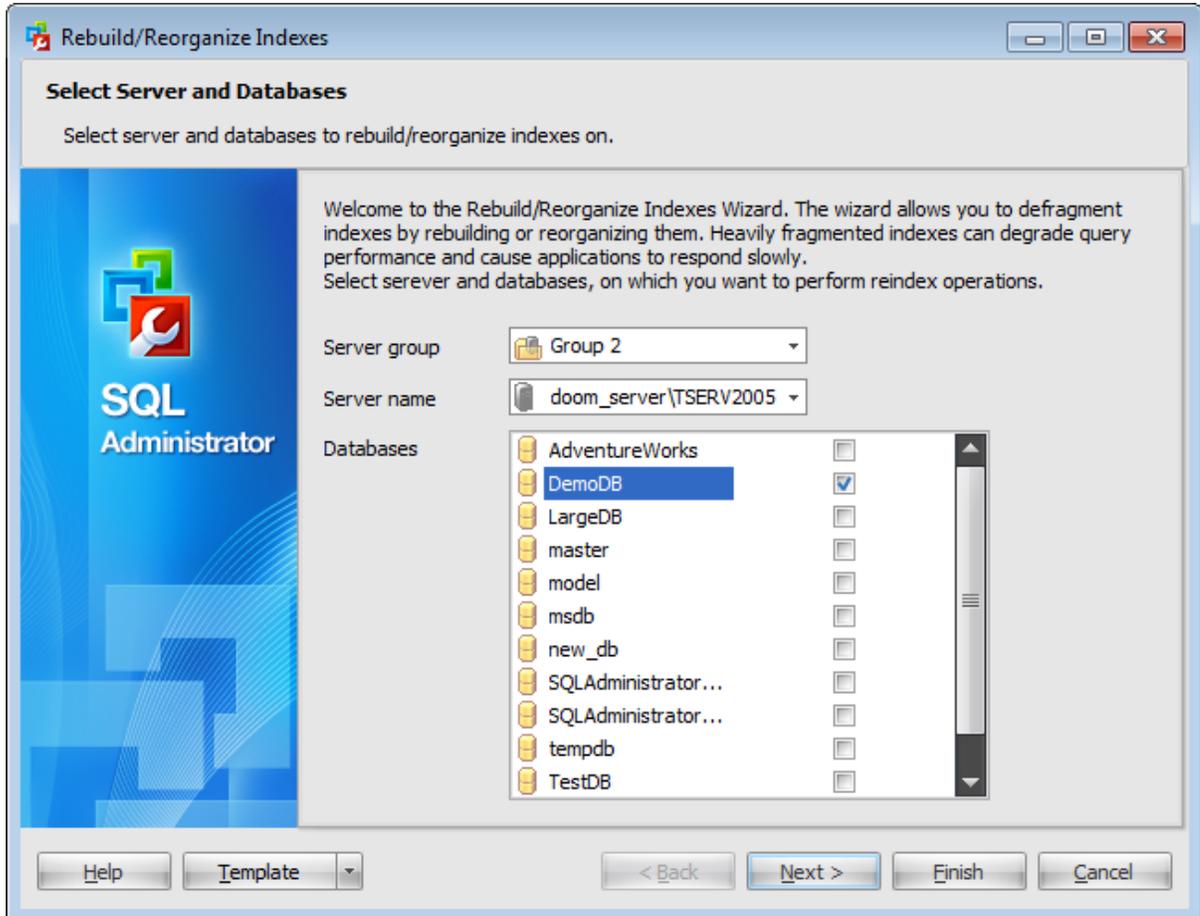
To run the wizard select the **Tools** |  **Rebuild/Reorganize Indexes...** [main menu](#) item or click  **Rebuild/Reorganize Indexes...** button on the [toolbar](#). Alternatively, you can right-click the database alias in the list of databases and select the  **Rebuild/Reorganize Indexes...** item of the context menu.

The wizard consists of the following steps:

- [Selecting server and databases](#)
- [Setting schedule options](#)
- [Setting reindex action condition](#)
- [Setting index actions and options](#)
- [Performing rebuilding/reorganizing](#)

#### 4.7.1.1 Selecting server and databases

This step of the wizard allows you to specify the **server** containing the databases for indexes management.



##### Server group

Use the drop-down list to select a server group where the server is stored.

##### Server name

Use the drop-down list to select the server where the database resides.

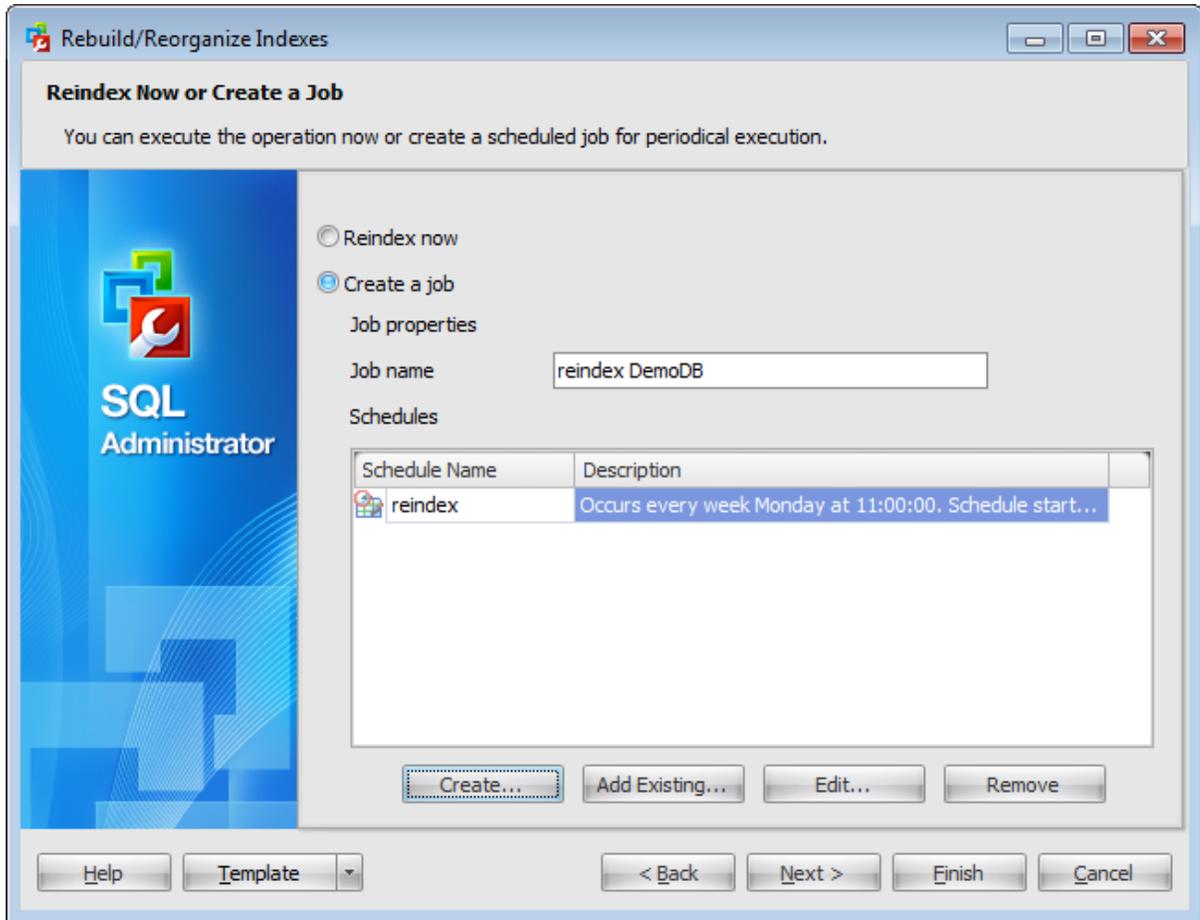
##### Databases

To choose a database for rebuilding/reorganizing indexes select it in the list and set the respective flag.

Click the **Next** button to proceed to the [Setting schedule options](#) step of the wizard

#### 4.7.1.2 Setting schedule options

Using this step you can define when the reindex process will be launched.



##### **Reindex now**

Check this option to start the reindex process after all necessary options are set.

##### **Create a job**

Select this option to create a **scheduled** reindex.

**Note:** To create a job you need to run [SQL Server Agent](#) service on the server.

Define **job properties**:

##### **Job name**

Enter a name of the new reindex job.

##### **Schedules**

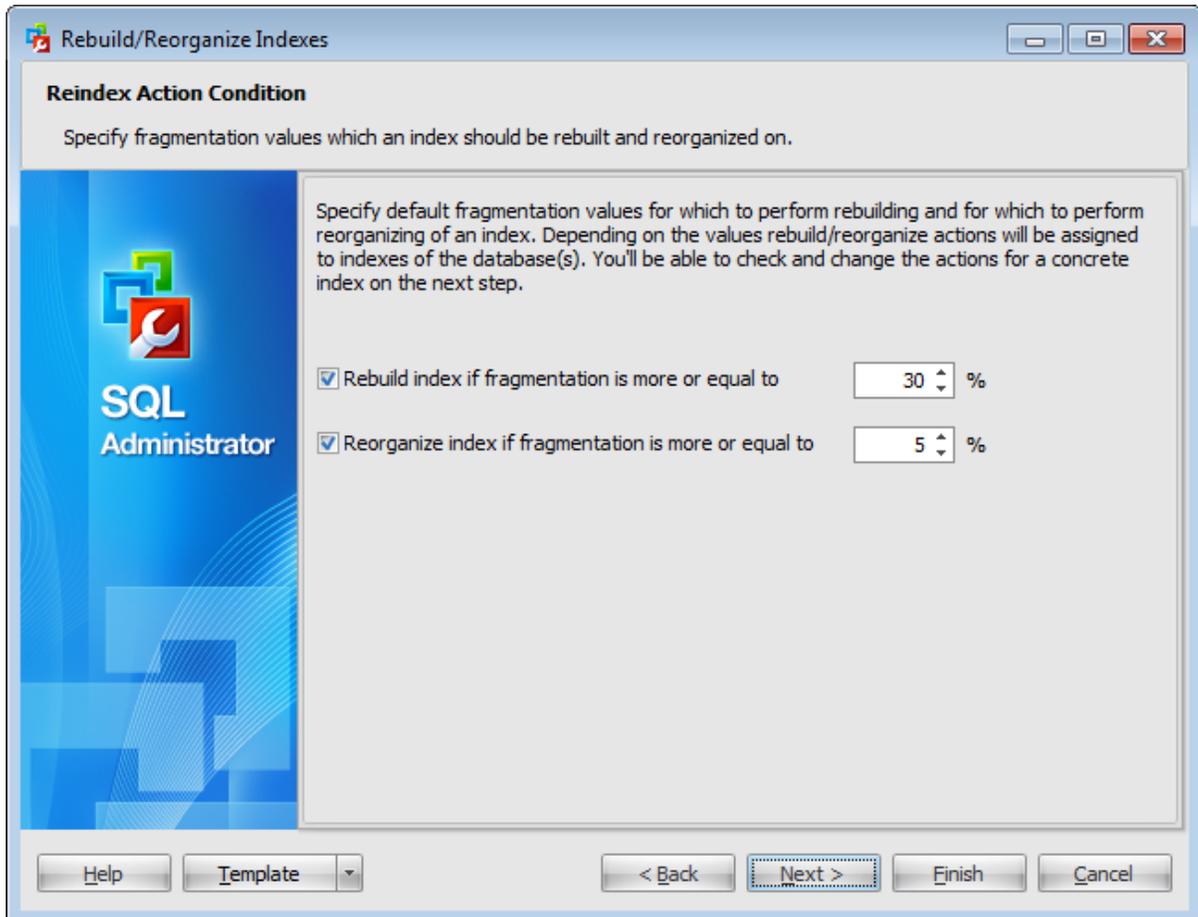
This list displays all schedules that cause the reindex job to begin running. You can add existing schedule or [create](#) a new one.

Click the **Next** button to proceed to the [Setting reindex action condition](#) step of the

wizard.

#### 4.7.1.3 Setting reindex action condition

Use this step to specify fragmentation values for rebuilding and reorganizing. If you have selected **Reindex now** option on the [previous step](#) then the following window appears:



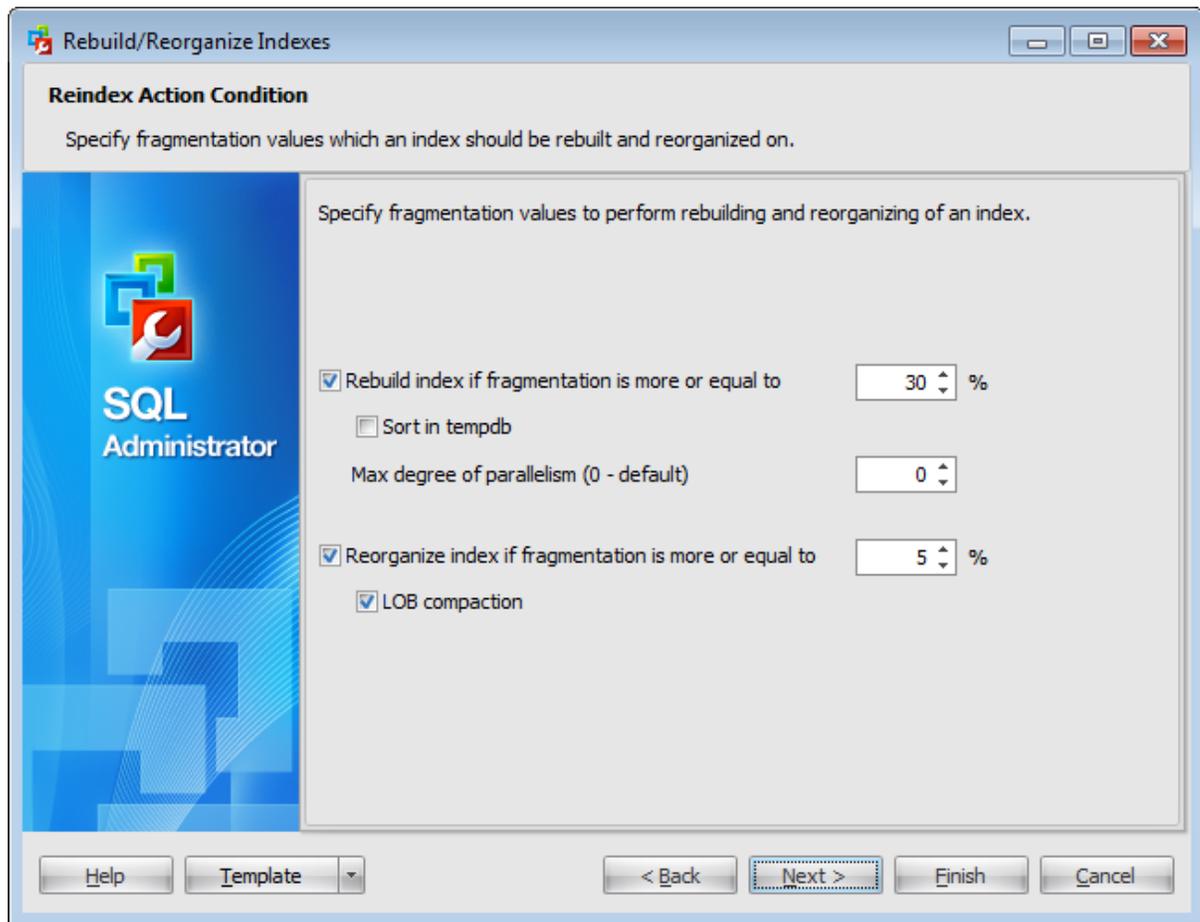
**Rebuild index if fragmentation more or equal to**

Define the fragmentation value (%) for which to perform index rebuilding.

**Reorganise index if fragmentation more or equal to**

Define the fragmentation value (%) for which to perform index reorganizing.

If you have selected **Create a job** on the [previous step](#) then the following window appears.



Define the fragmentation values for which to perform index rebuilding and reorganizing using corresponding fields.

**Sort in tempdb**

Determines where the intermediate sort results, generated during index creation, are stored. When this option is checked, the sort results are stored in tempdb. When it is not checked, the sort results are stored in the filegroup or partition scheme in which the resulting index is stored.

**Max degree of parallelism**

This parameter sets the maximum number of processors the query processor can use to execute a single index statement. Fewer processors may be used depending on the current system workload.

**LOB compaction**

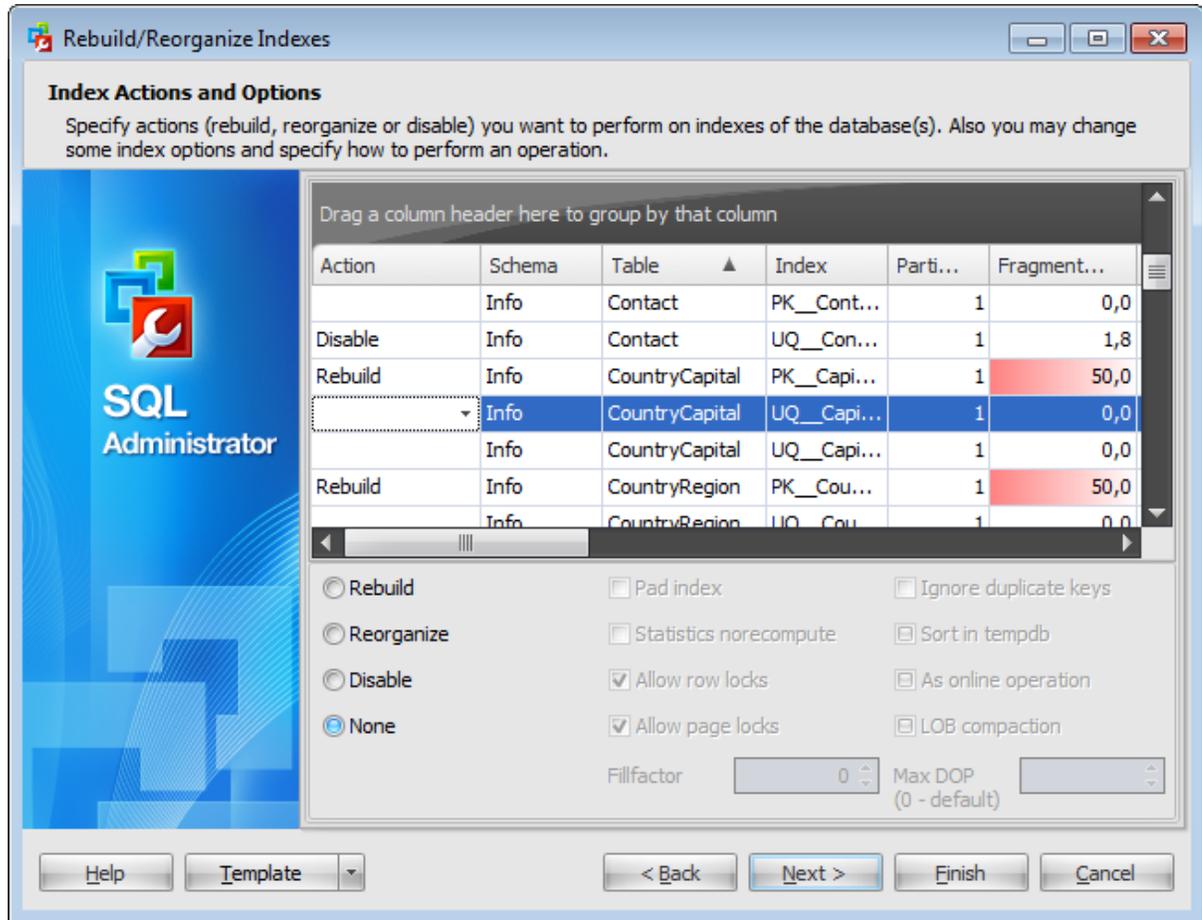
Check this option to compact large object data (LOBs). They are contained in the clustered index or underlying table and can be compacted when an index is reorganized. The data types *image*, *text*, *ntext*, *varchar(max)*, *nvarchar(max)*, *varbinary(max)*, and *xml* are large object data types. Compacting this data can cause better disk space use.

Click the **Next** button to proceed to the [Setting index actions and options](#) step of the

wizard (if  **Reindex now** option was selected on the [previous step](#)), or to the [Performing rebuilding/reorganizing](#) step (if you have selected  **Create a job** on the [previous step](#)).

#### 4.7.1.4 Setting index actions and options

Use this step to set index options and actions (rebuild, reorganize or disable).



The lower area lists the **parameters** that can be defined for the specified action (except for **Disable**).

##### **Rebuild**

##### **Pad index**

This parameter sets the percentage of free space in the intermediate level pages during index creation.

##### **Statistics norecompute**

This option specifies whether out-of-date index statistics should be automatically recomputed.

##### **Allow row locks**

This option determines whether row locks are used in accessing index data.

##### **Allow page locks**

This option determines whether page locks are used in accessing index data.

**Ignore duplicate keys**

This option applies only to insert operations after the index is created or rebuilt.

 **Sort in tempdb**

Determines where the intermediate sort results, generated during index creation, are stored. When this option is checked, the sort results are stored in tempdb. When it is not checked, the sort results are stored in the filegroup or partition scheme in which the resulting index is stored.

 **As online operation**

This option determines whether concurrent user access to the underlying table or clustered index data and any associated non-clustered indexes is allowed during index operations.

**Fillfactor**

This parameter sets the percentage of free space in the leaf level of each index page during index creation.

**Max DOP**

This parameter sets the maximum number of processors the query processor can use to execute a single index statement. Fewer processors may be used depending on the current system workload.

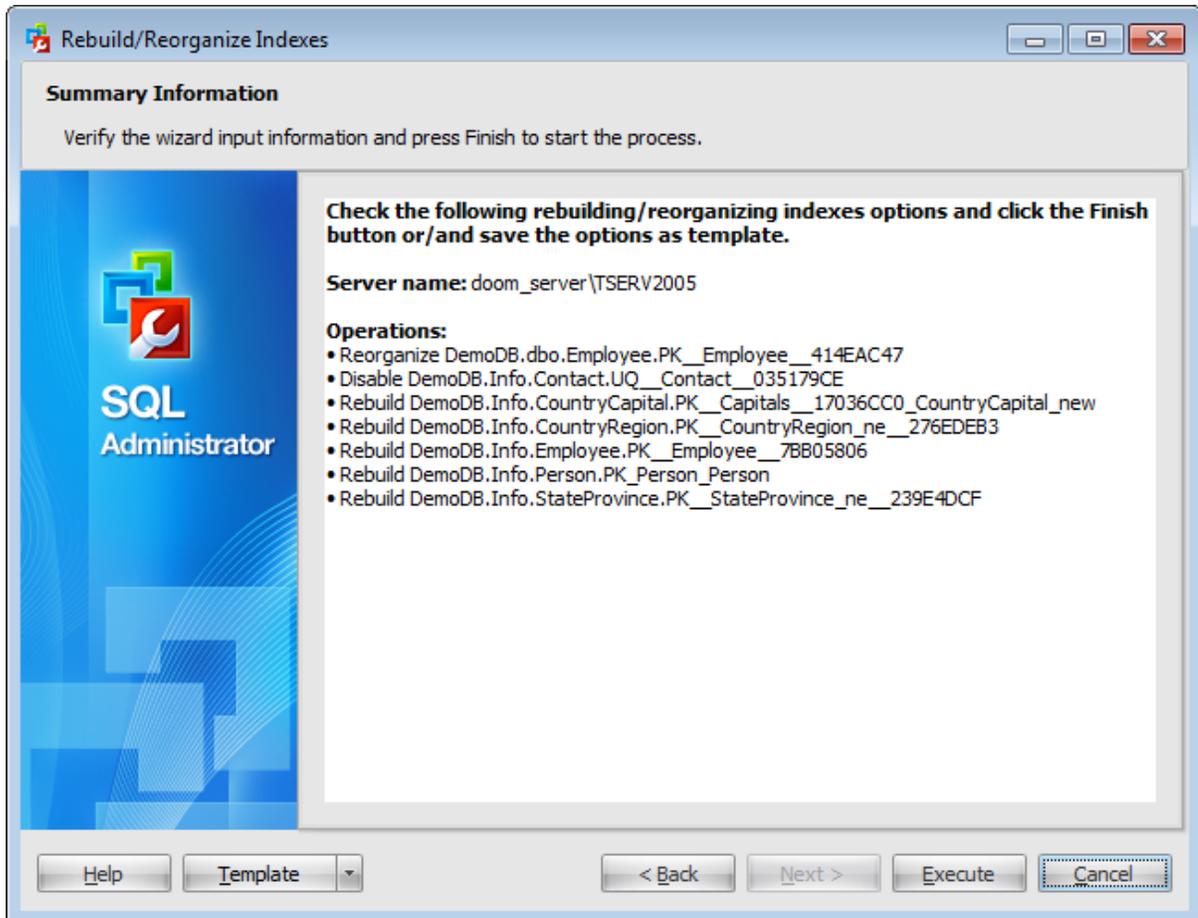
 **Reorganise** **LOB compaction**

Check this option to compact large object data (LOBs). They are contained in the clustered index or underlying table and can be compacted when an index is reorganized. The data types *image*, *text*, *ntext*, *varchar(max)*, *nvarchar(max)*, *varbinary(max)*, and *xml* are large object data types. Compacting this data can cause better disk space use.

Click the **Next** button to proceed to the [Performing rebuilding/reorganizing](#) step of the wizard.

#### 4.7.1.5 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the rebuild/reorganize indexes process.

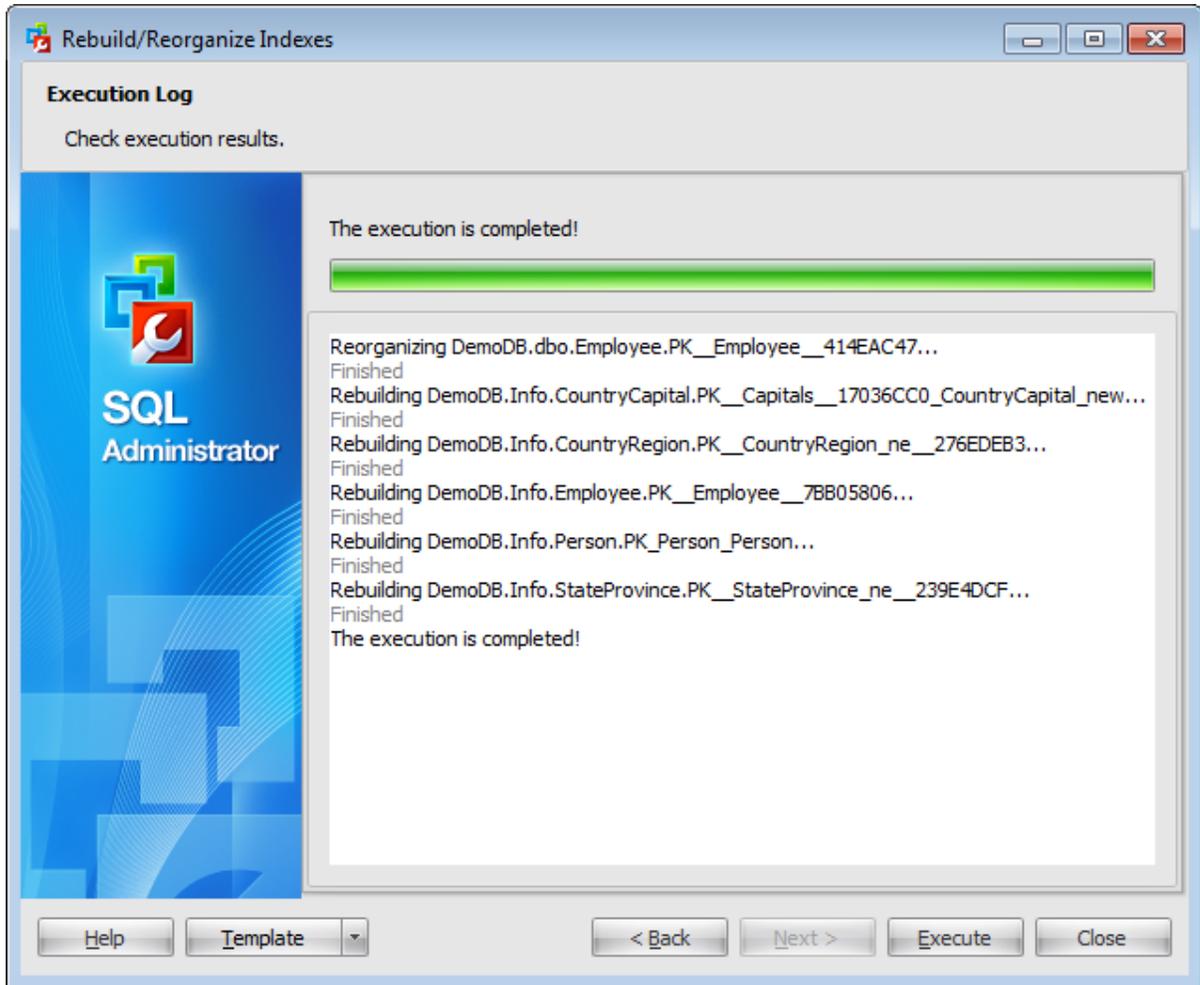


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing rebuild/reorganize indexes task](#) step.

#### 4.7.1.6 Executing rebuild/reorganize indexes task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

## 4.8 Shrink database

**Shrink Database Wizard** allows you to perform database shrinking on your SQL Server system (the *DBCC SHRINKDATABASE* and *SHRINKFILE* Transact-SQL statements are issued).

This operation is used to reduce any file within the database to remove unused pages. Both data and transaction log files can be reduced, or shrunk.

To run the wizard, select the **Tools** |  **Shrink Database...** [main menu](#) item or click  **Shrink Database...** button on the [toolbar](#).

The wizard consists of the following steps:

- [Selecting server and databases](#)
- [Setting shrink type](#)
- [Specifying shrink database options](#)
- [Specifying shrink file options](#)
- [Setting schedule options](#)
- [Shrinking database](#)

---

### See also:

[Maintenance plans](#)

[Backup database](#)

[Restore database](#)

[Attach database](#)

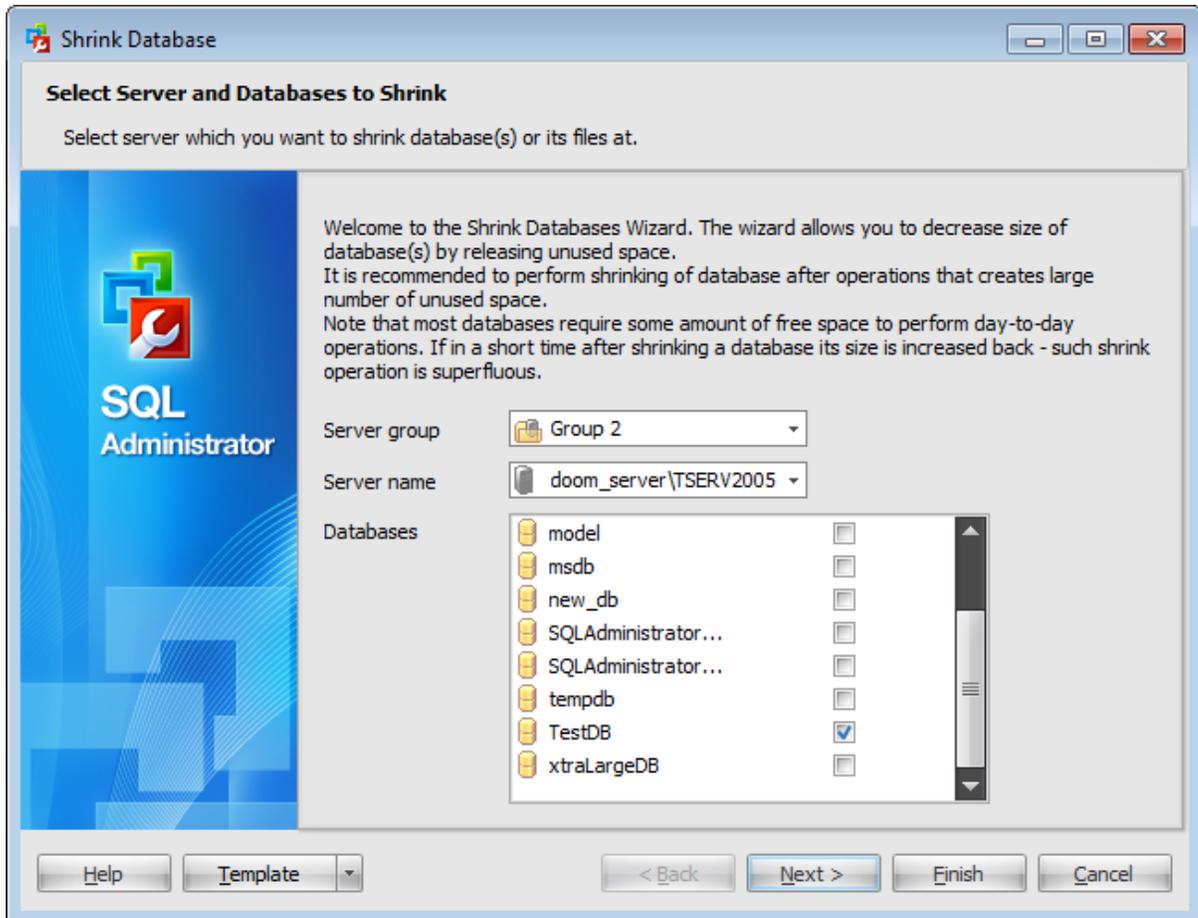
[Detach database](#)

[Check Database](#)

[Rebuild/Reorganize Indexes](#)

### 4.8.1 Selecting server and databases

This step of the wizard allows you to specify the **host/instance** containing the database for shrinking, and to select the **shrink mode**.



#### Server group

Use the drop-down list to select a server group where the server is stored.

#### Server name

Use the drop-down list to select the server where the database resides.

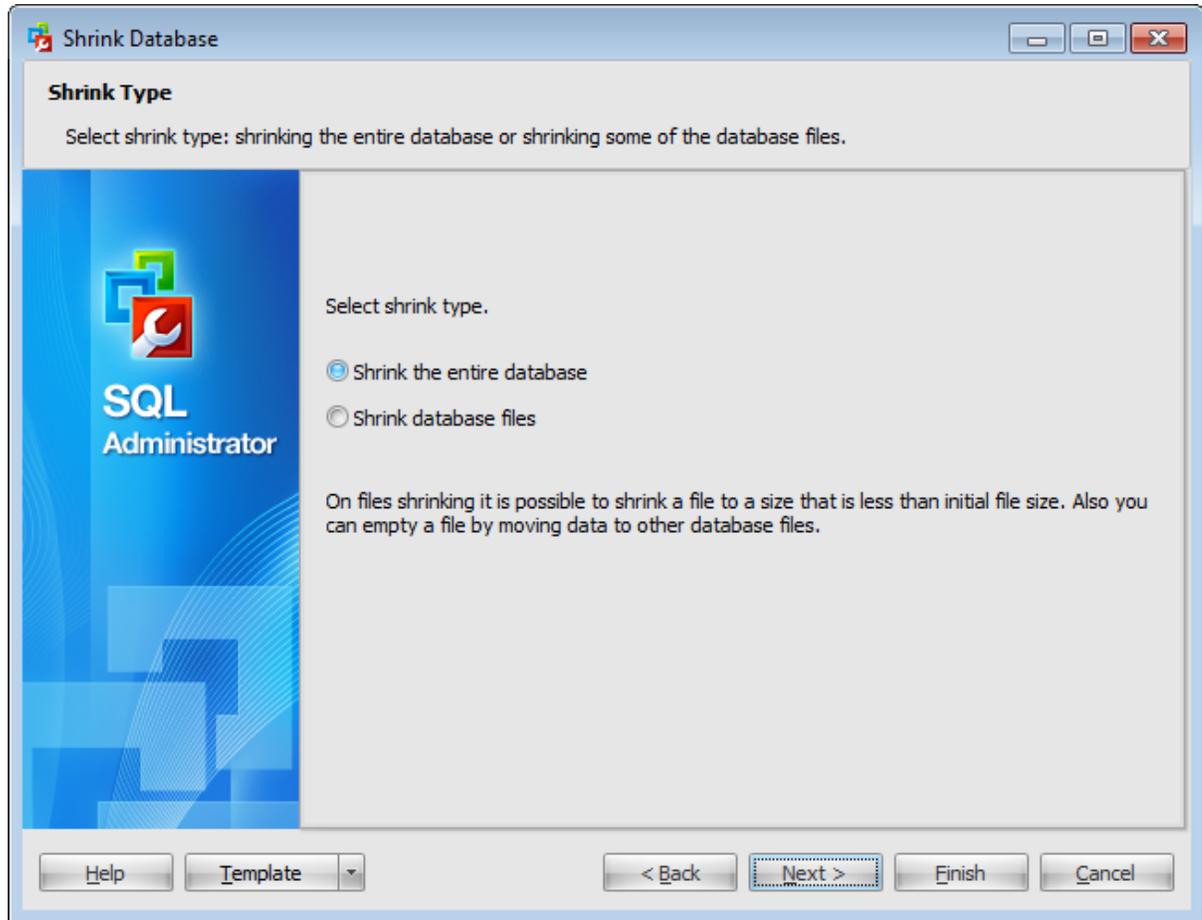
#### Databases

To choose a database to shrink select it in the list and set the respective flag.

Click the **Next** button to proceed to the [Setting shrink type](#) step of the wizard (if you have selected a single database), or to the [Specifying shrink database options](#) step (if you have selected several databases).

## 4.8.2 Setting shrink type

Use this step to specify a shrink type.



Select the preferable shrink mode:  **Shrink the entire database** (*SHRINKDATABASE*) or  **Shrink database files** (*SHRINKFILE*).

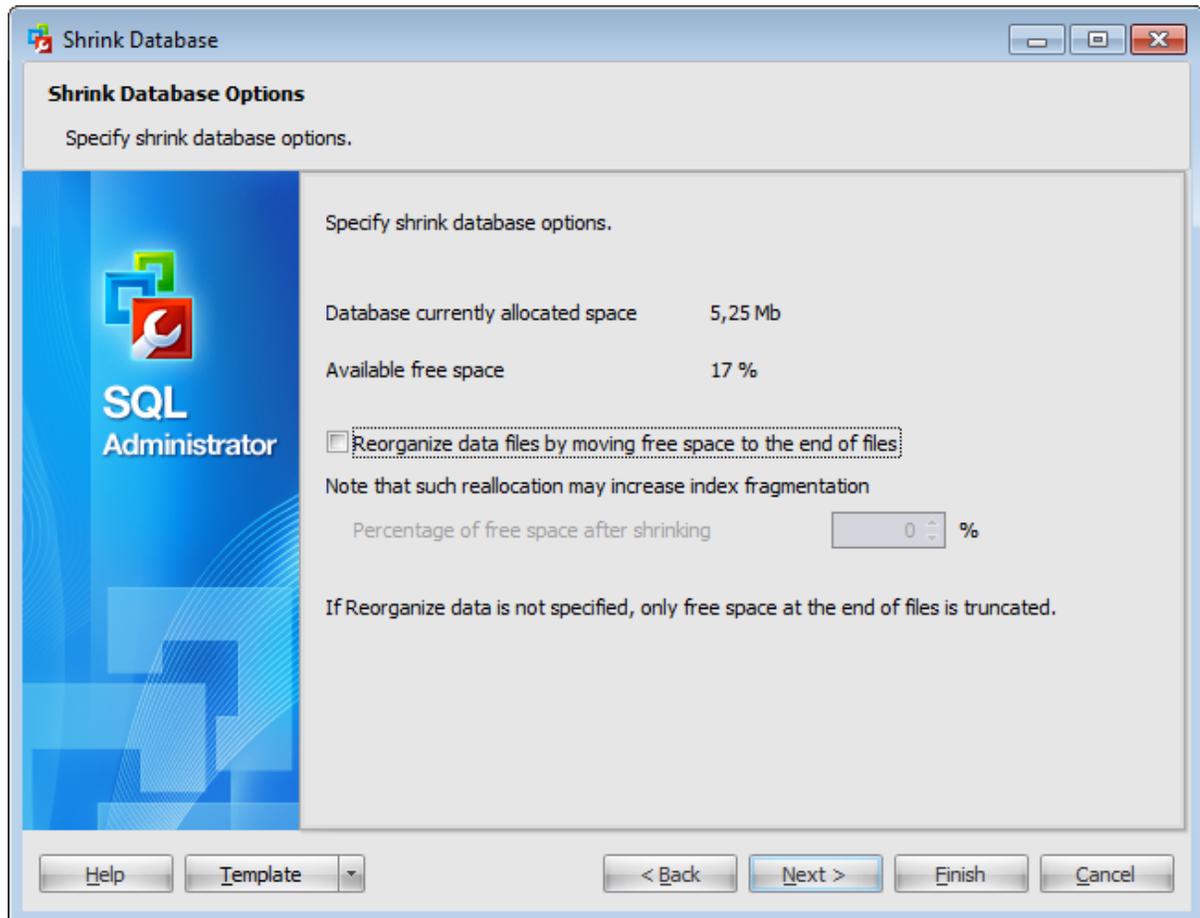
When using the DBCC SHRINKDATABASE statement, you cannot shrink a whole database to be smaller than its original size.

However, you can shrink the individual database files to a smaller size than their initial size by using the DBCC SHRINKFILE statement. You must shrink each file individually, instead of trying to shrink the whole database.

Click the **Next** button to proceed to the [Specifying shrink database options](#) step of the wizard (if you have selected the *Database* shrink mode), or to the [Specifying shrink file options](#) step (if you have selected the *Files* shrink mode).

### 4.8.3 Specifying shrink database options

This step of the wizard provides you with information on the **allocated** and **free space**, and allows you to specify the **shrink action** to be taken on the database. This step is available only if  **Shrink the entire database** type was set on the [second step](#). If you have selected a single database to shrink on the [first step](#), then the following window appears.



This window provides you with the information about the *currently allocated space* and the *available free space* values pertaining to the database being shrunk.

**Reorganize data files by moving free space to the end of files**

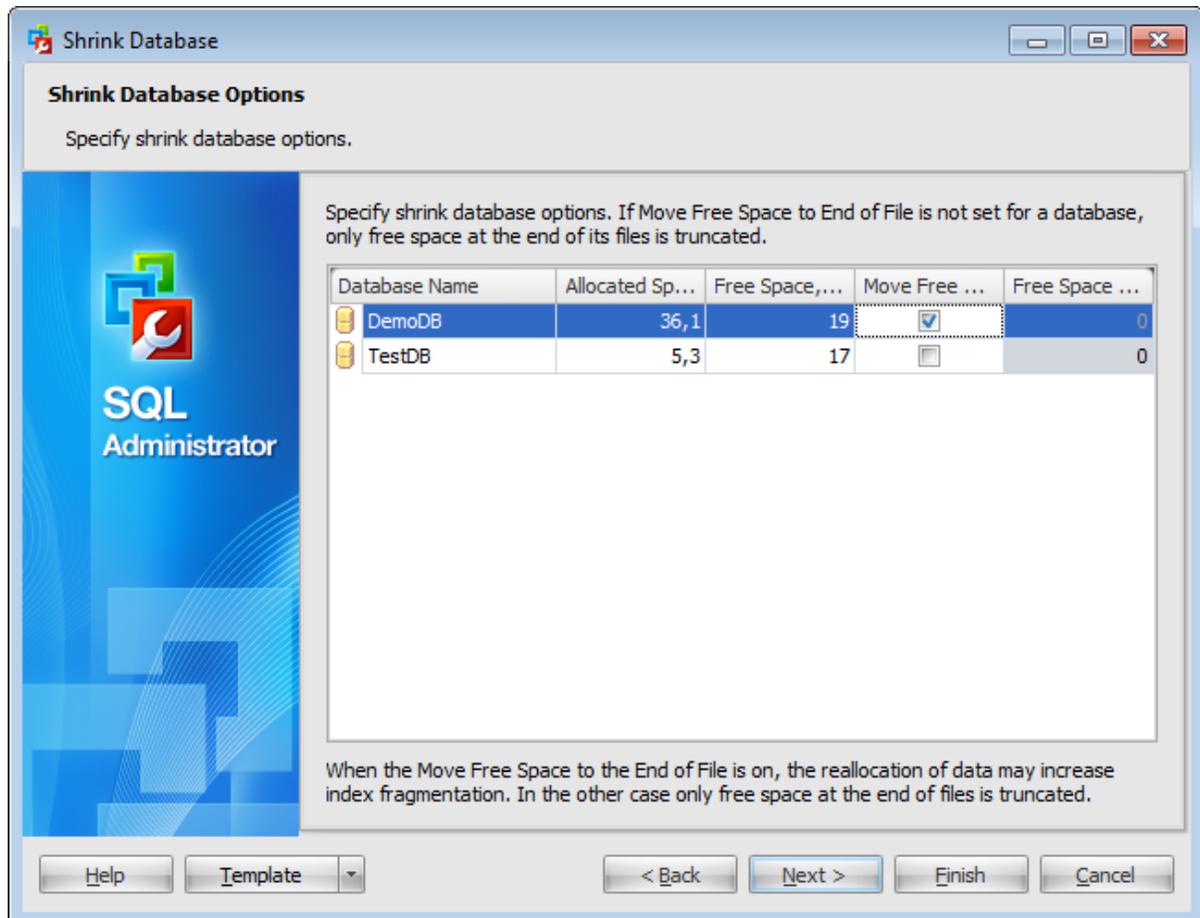
Check this option to recover space by moving pages of data from the end of the file to unoccupied space closer to the front of the file.

This can cause index fragmentation and can slow the performance of queries that search a range of the index. To eliminate the fragmentation, consider rebuilding the indexes on the file after shrinking.

**Percentage of free space after shrinking**

Specify the percentage of free space that will be left in the database file after the database is shrunk.

If you have selected several databases to shrink on the [first step](#), then the step looks the following way:



This grid displays information about **allocated** and **free** space for each database.

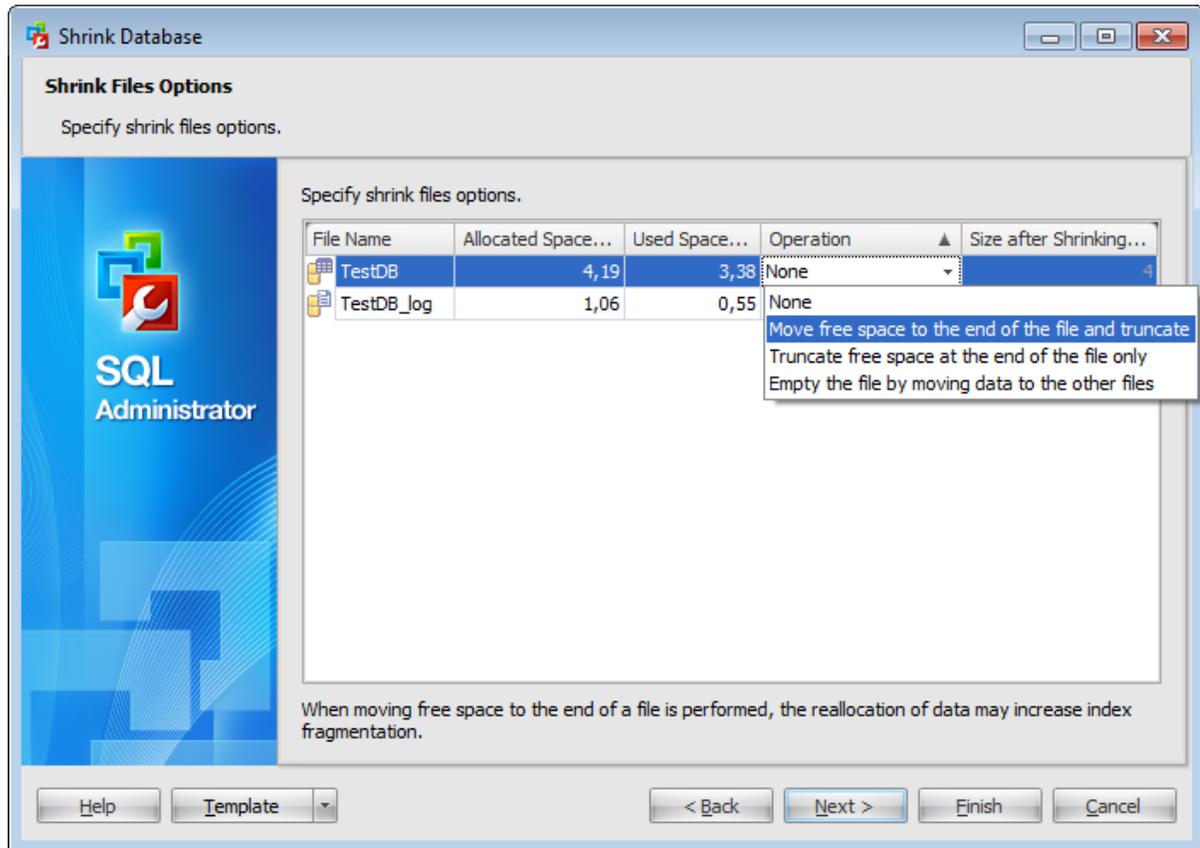
You can set  **Move Free Space to End of File** option for each database by selecting it in the list and setting the respective flag.

Specify the percentage of **Free Space after Shrinking** in the corresponding cells. This amount of free space will be left in the database file after the database has been shrunk.

Click the **Next** button to proceed to the [Setting schedule options](#) step of the wizard.

#### 4.8.4 Specifying shrink file options

This step of the wizard provides you with information on the **allocated** and **used space** by files, and allows you to specify the **shrink action** for each of the database files. This step is available only if you have selected  **Shrink database files** on the [second step](#).



You can set a shrink **operation** to perform for every file by choosing a file from the list and clicking the corresponding cell.

The following operations can be set:

- *Move free space to the end of the file and truncate*

Moves allocated pages from the end of a data file to unallocated pages in the front of the file.

- *Truncate free space at the end of the file only*

Releases all free space at the end of the file to the operating system but does not perform any page movement inside the file.

- *Empty the file by moving data to the other file*

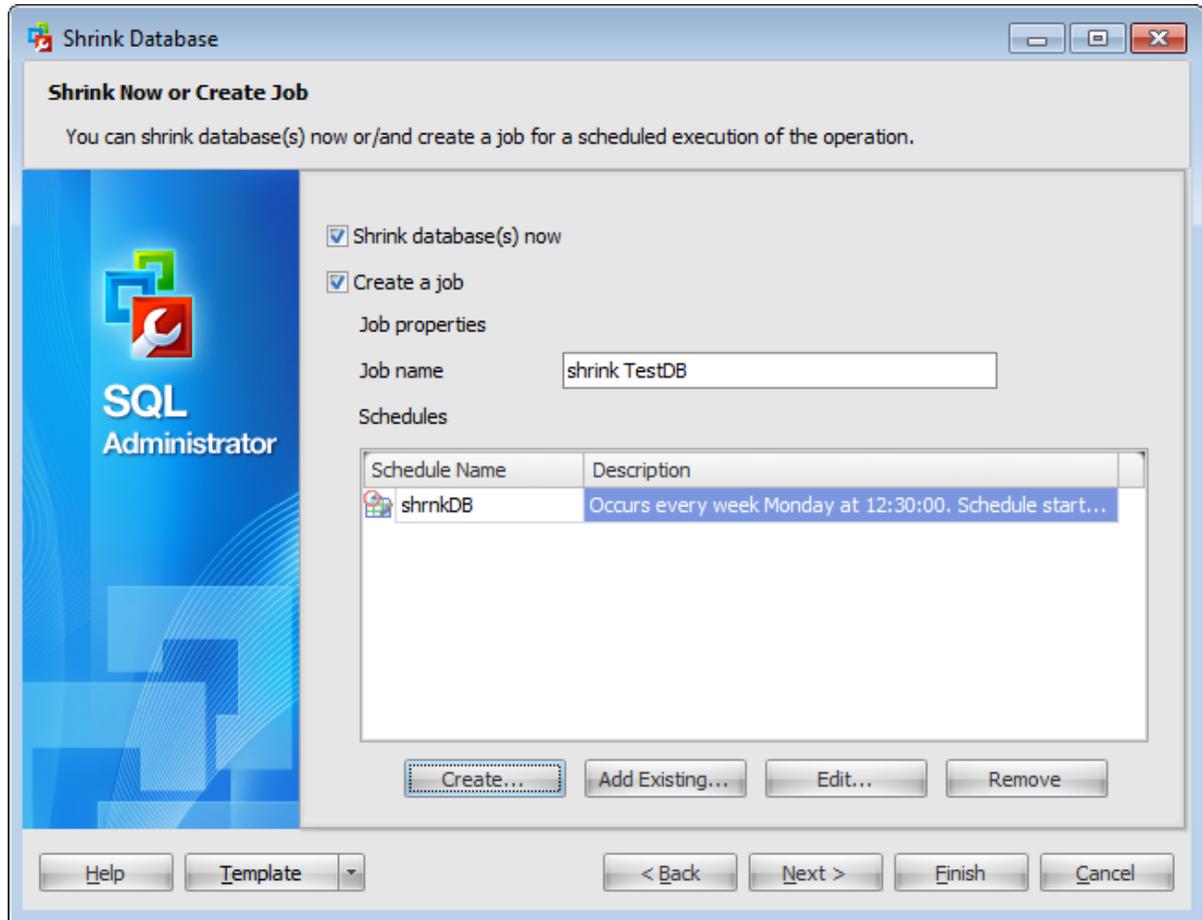
Migrates all data from the specified file to other files in the same filegroup.

You can set a file **size after shrinking** by choosing a file from the list and clicking the corresponding cell. Note that you can shrink a file to a size that is less than the size specified when it was created. This resets the minimum file size to the new value.

Click the **Next** button to proceed to the [Setting schedule options](#) step of the wizard.

### 4.8.5 Setting schedule options

Using this step you can define when the shrinking will be executed.



**Shrink database(s) now**

Check this option to start the shrink process after all necessary options are set.

**Create a job**

Select this option to create a **scheduled** shrink.

Define **job properties**:

**Job name**

Enter a name of the new shrink job.

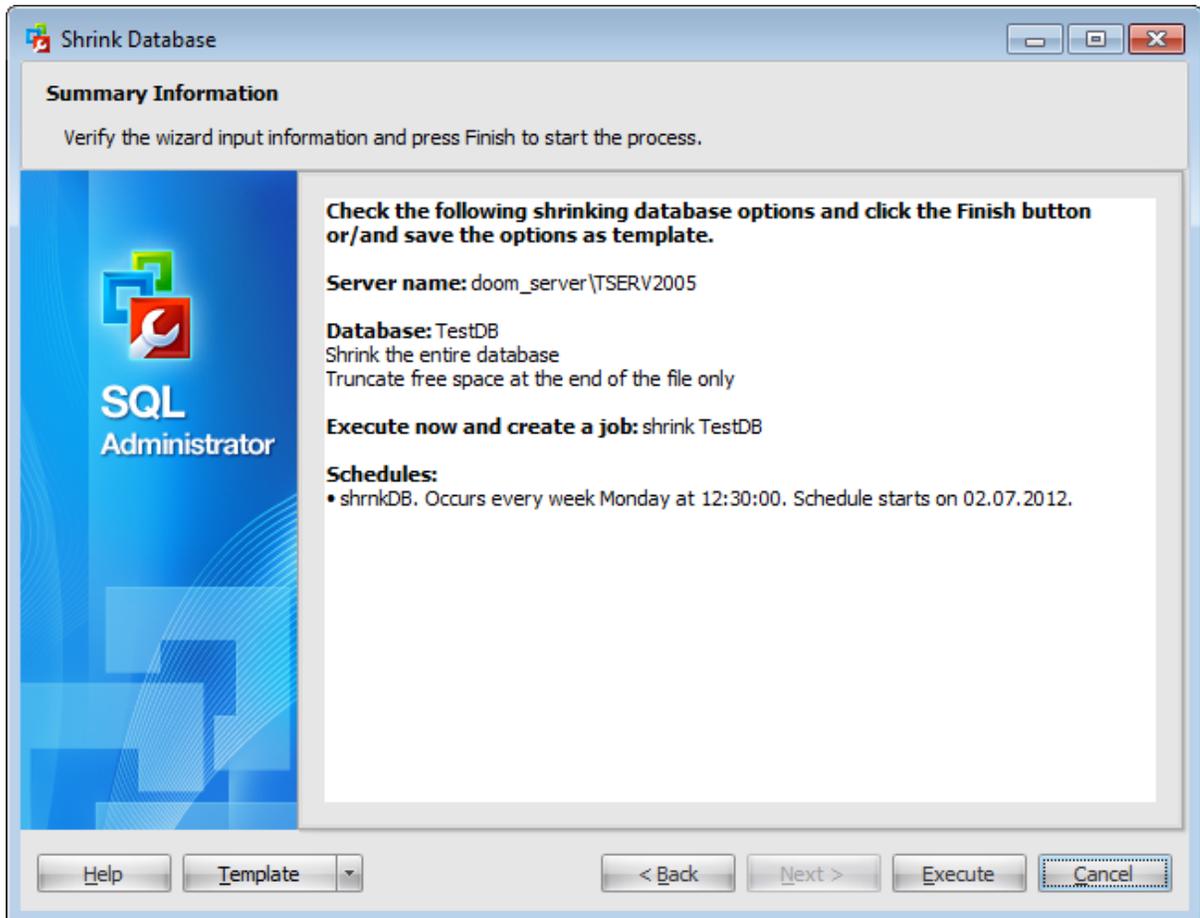
**Schedules**

This list displays all schedules that cause the shrink job to begin running. You can add existing schedule or [create](#) a new one.

Click the **Next** button to proceed to the [Shrinking database](#) step of the wizard.

#### 4.8.6 Viewing summary information

This step of the wizard is intended to inform you that all necessary options have been set, and you can start the shrink database/files process.

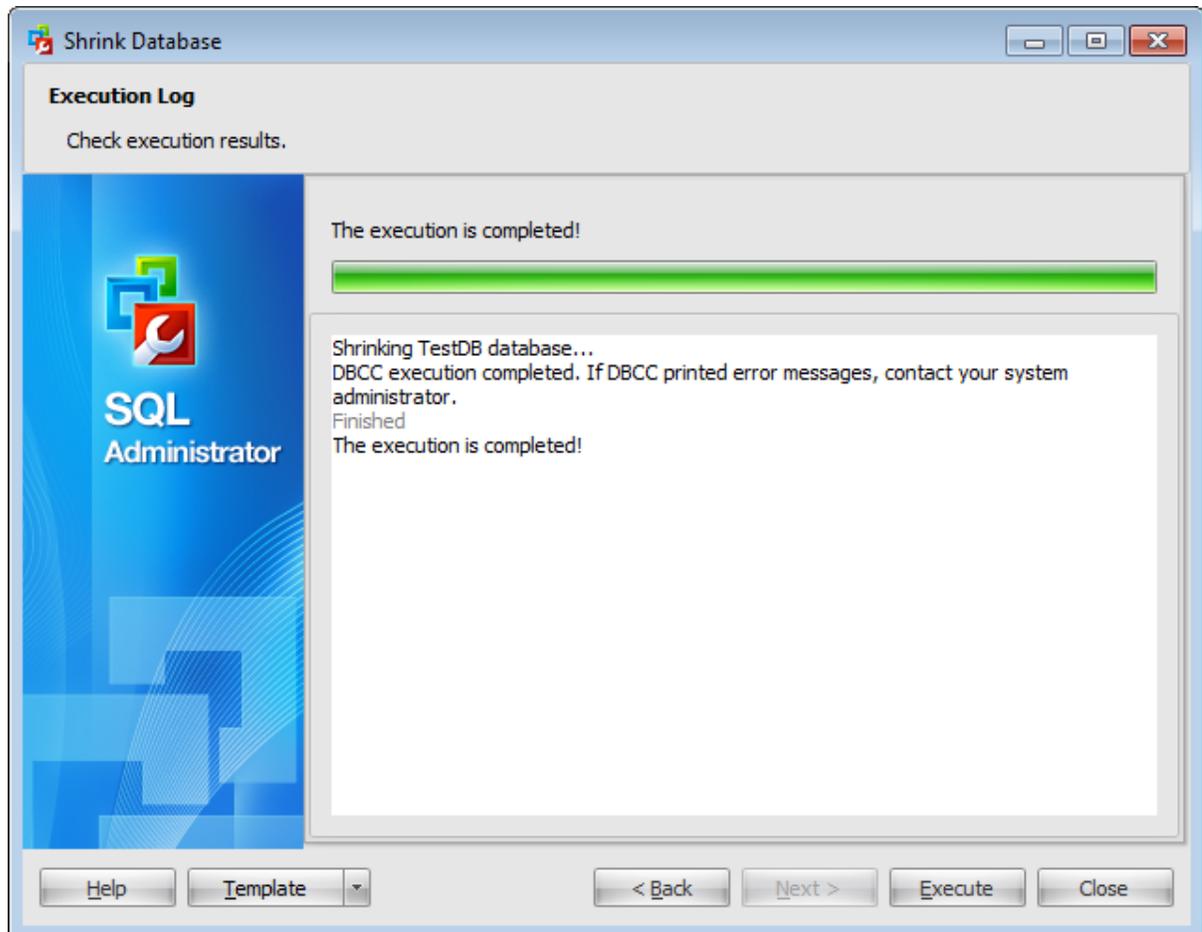


If necessary, you can save a [template](#) for future use.

After verifying the choices made in the wizard click the **Execute** button to run the process. You will be automatically proceeded to the [Executing shrink database task](#) step.

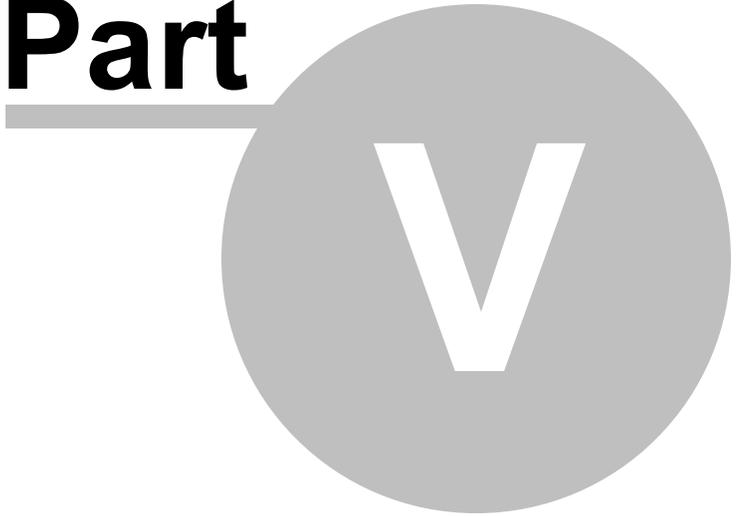
### 4.8.7 Executing shrink database task

The log area allows you to view the log of operation and errors (if any).



Using the context menu of the log area you can clear execution log, select the whole log and copy selected text.

**Part**

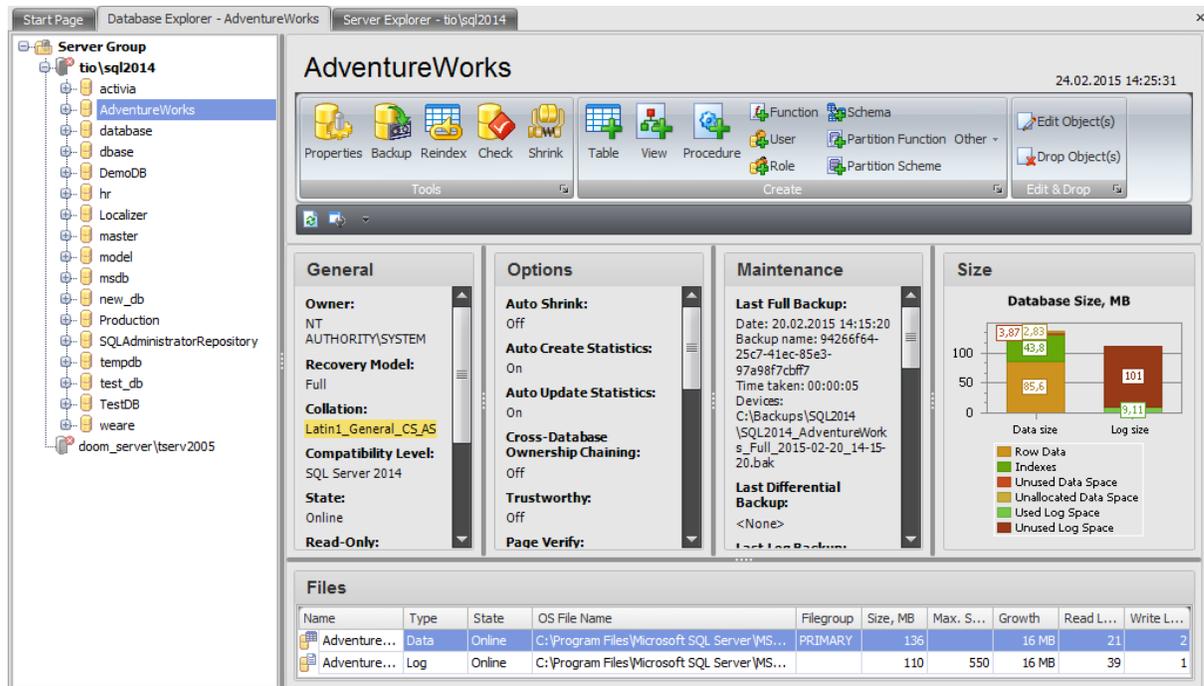


## 5 Database Objects Management

SQL Administrator provides powerful tools to manage **database objects**.

To obtain detailed information concerning SQL Server database and objects, refer to the official Microsoft® SQL Server™ documentation.

**Note:** Before working with database and objects in SQL Administrator you should register a server in the [Database Explorer](#) and connect to it.



After you have connected to the server in the Database Explorer the list of databases expands.

When you select the database in the Database Explorer its information is presented on the work area.

Using the buttons on the ribbon you can:

- open the [Database Editor](#)
- perform main database [maintenance operations](#)
- manage database objects (create, edit and drop)

On the work area of the database section the following database information can be found:

### General

In this section you can view general information about the database: *Owner, Recovery Model, Collation, Compatibility Level, State, Read-only, User Access, Guest Account, Creation Date.*

### Options

In this section some of the database [options](#) and their values are listed.

### Maintenance

Here you can find information about database maintenance operations. For database backups (full, differential, transaction log) the following information is presented:

- *Date*
- *Backup name*
- *Time taken*
- *Devices*

Also this section displays the date of the last successful [check](#) and most fragmented indexes that should be [rebuilt or reorganized](#).

### Size

Here the diagram of the space usage by database files is presented. You can see how much space is occupied by *Row Data* and *Indexes*; *Unallocated and Unused Data Space*; *Used* and *Unused Log Space*.

### Files

This grid displays the list of [database files](#). Using the context menu you can create a new file, edit or drop the existing file and print the list of files.

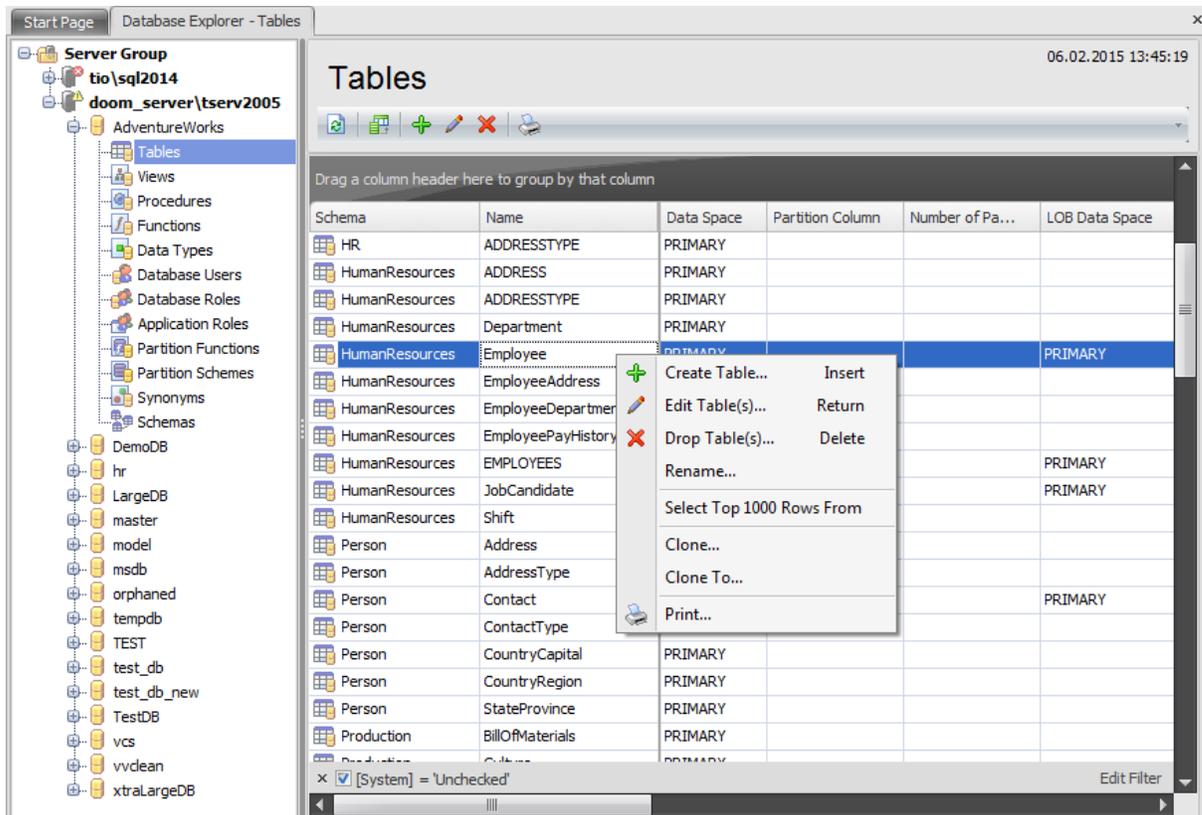
To view the list of database objects double-click it in the tree and then select the necessary object group.

## 5.1 Tables

The **Tables** section of the Database Explorer is used for managing tables in the selected database.

Use the context menu to:

-  [create](#) a new table
-  [edit](#) the selected table(s)
-  drop selected table(s)
- rename the selected table
- select top 1000 rows from the selected table
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database tables as a grid with the following columns: *Schema, Name, Data Space, Partition Column, Number of Partitions, LOB Data Space, Replicated, Filetable, Memory-optimized, Durability, Change Tracking, Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Table in the [Table Editor](#), double-click it in the grid. Alternatively, you can right-click the Table alias and select the **Edit Table(s)** context menu item.

### 5.1.1 Table Editor

**Table Editor** is the basic SQL Administrator tool for working with [tables](#). It opens automatically in the 'new table' mode when you create a new table and is available on editing an existing one.

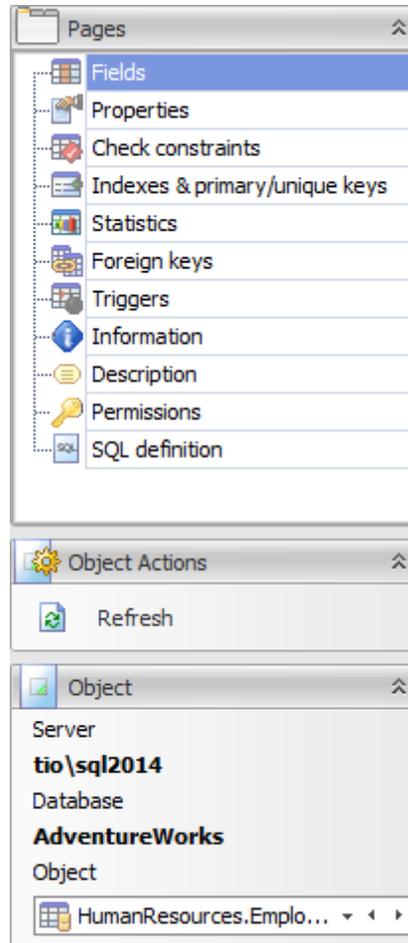
**Table Editor** allows you to create, edit and drop table [fields](#), [indexes](#), [foreign keys](#) and other table subobjects and manage table [properties](#).

To open a table in **Table Editor**, double-click it in [list of tables](#).

- [Using Navigation bar](#)
- [Fields](#)
- [Properties](#)
- [Filetable options](#)
- [Check constraints](#)
- [Indexes and primary/unique keys](#)
- [Statistics](#)
- [Foreign keys](#)
- [Triggers](#)
- [Information](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.1.1.1 Using Navigation bar

The **Table Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Table Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Object

This field displays the database that is currently being edited. However, you can change the edited table by using the drop-down list of tables or by clicking the arrow-left or arrow-right buttons.



### 5.1.1.2 Fields

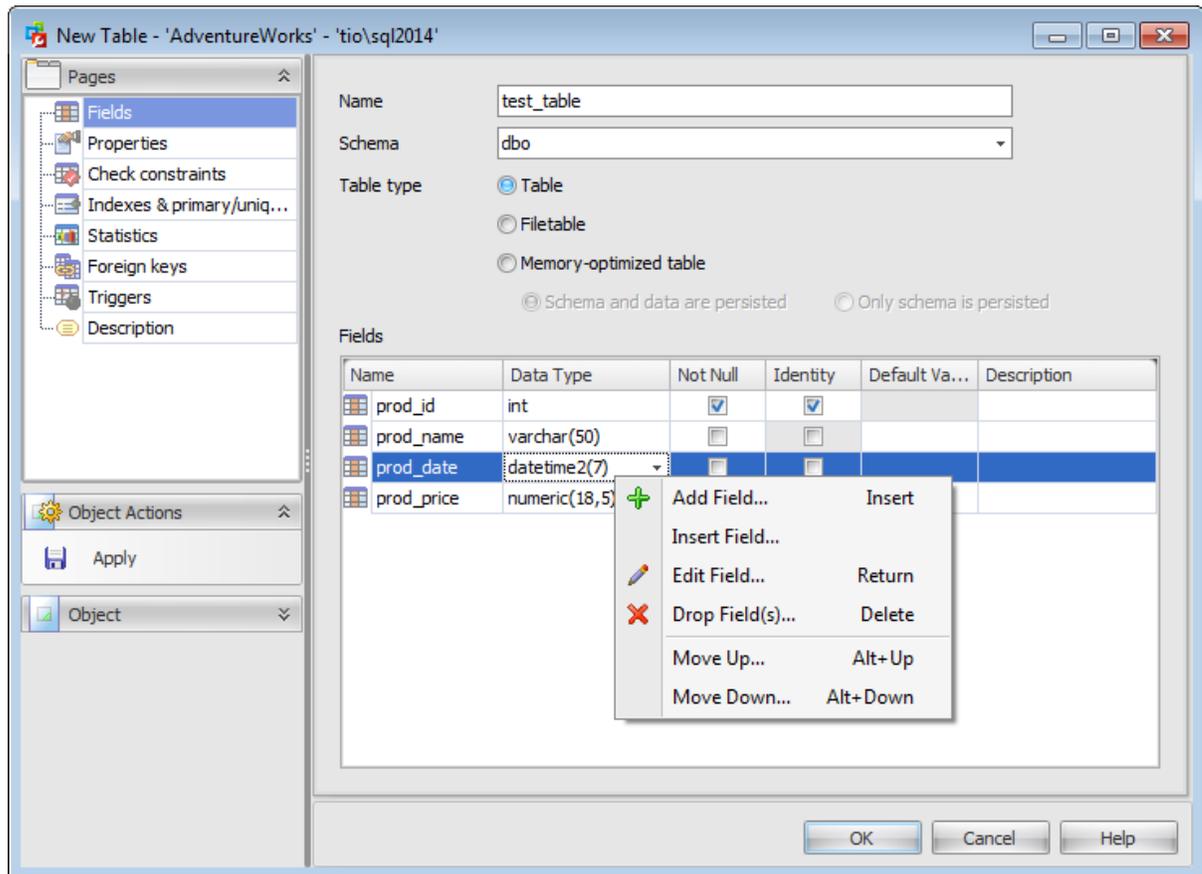
The **Fields** tab is intended for managing table [fields](#) and setting main table properties.

#### Name

Enter a name for the new table. Note that table names must comply with the rules for identifiers and must be unique within the [schema](#). A table name can contain a maximum of 128 characters.

#### Schema

Use the drop-down list to select the [schema](#) to which the table belongs. The default selection is the default schema for the current [user](#).



#### Table type

##### Table

This option indicates that a common SQL Server is created.

##### Filetable

Select this option to enable file and directory storage in the database. A FileTable is a specialized user table with a pre-defined schema that stores FILESTREAM data, as well as file and directory hierarchy information and file attributes. You do not specify columns because a Filetable has a fixed schema.

If you have selected to create a filetable, you must specify [Filetable options](#).

**Memory-optimized table**

Select this option to enable memory-optimized table creation. Memory-optimized tables are in memory user tables, the schema of which is persisted on disk similar to other user tables.

If you have selected to create a memory-optimized table, you must specify [memory-optimized table indices](#).

**Schema and data are persisted**

This value indicates that the table is a durable memory-optimized table. This option provides durability of both schema and data.

**Only schema is persisted**

This value indicates that the table is non-durable. This option ensures durability of the table schema, including indexes.

Such tables are not logged and their data is not persisted on disk.

### Fields

The **Fields** list provides the following attributes of each field of the new table:

*Name*

*Data Type*

*Not Null*

*Identity*

*Default Value*

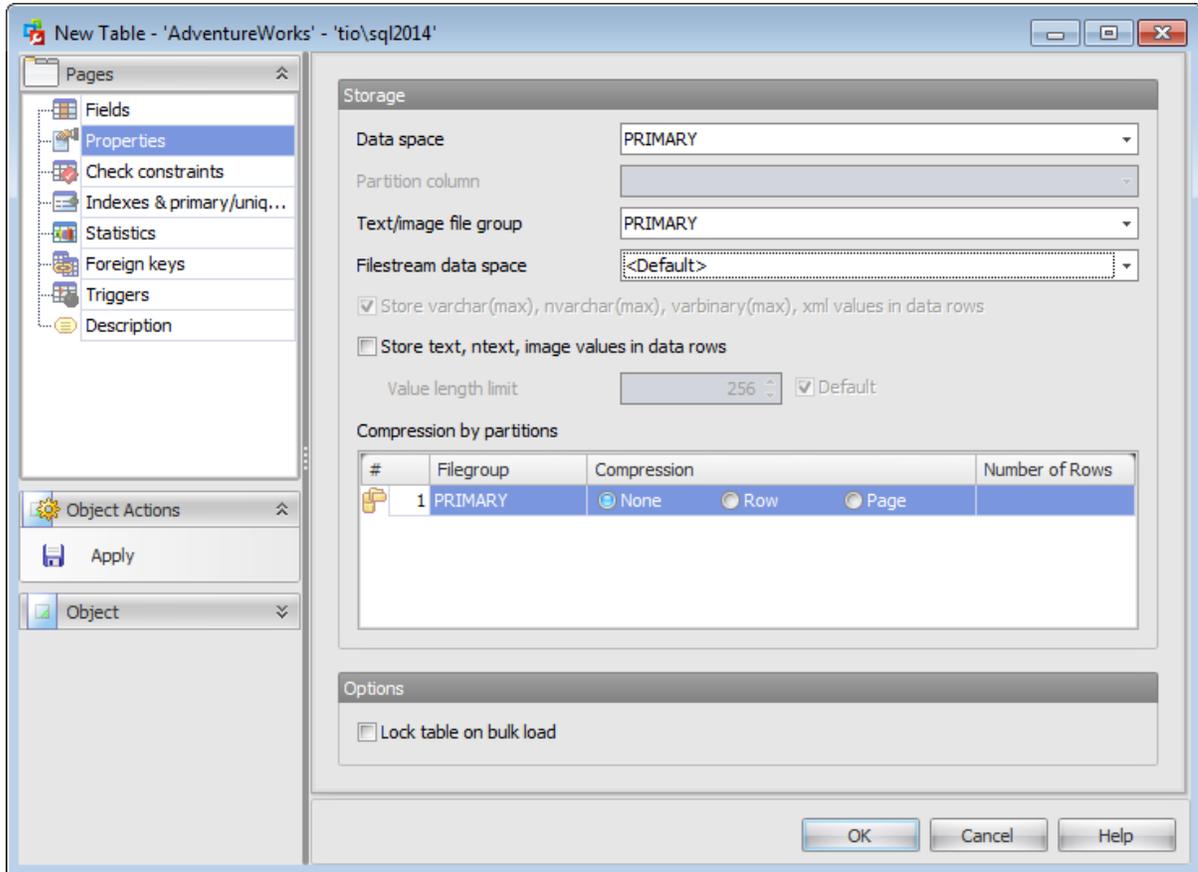
*Description*

Double-click a field to open [Field Editor](#) for editing the field.

Right-click within the **Fields** area to display the context menu allowing you to *add, insert, edit, delete and reorder* fields.

### 5.1.1.3 Properties

The **Properties** tab allows you to view/edit *Table options* (*Storage attributes, Table lock usage*).



#### Storage

##### Data space

Use the drop-down list to specify the new table data storage (a file group or a [partition scheme](#)).

##### Partition column

This drop-down list is only available if a [partition scheme](#) is specified for **Data space**. Select a [field](#) that will be used as the partitioning column for the table.

##### Text/image file group

Use the drop-down list to select the file group where data of the *text* and *image* types will be stored.

##### Filestream data space

Select the needed [Filestream filegroup](#) from the drop-down list. This option is available only for SQL Server 2008 and higher.

**Store varchar(max), nvarchar(max), varbinary(max) and xml fields in data rows**

Varchar(max), nvarchar(max), varbinary(max), xml types and large UDT values are stored directly in the data row, up to a limit of 8000 bytes and as long as the value can fit in the record. If the value does not fit in the record, a pointer is stored in-row and the rest is stored out of row in the LOB storage space.

**Store text, ntext, image values in data rows**

When this option is checked new text, ntext, or image strings are stored directly in the data row. All existing BLOB (text, ntext, or image data) will be changed to text in row format when the BLOB value is updated.

When this option is on, the **value length limit** defaults to 256 bytes. You can set this value in bounds from 24 through 7000.

**Compression by partitions**

Use this area to set compression options for the table. Compression type can be different for each table partition. Set the partition compression to *Row* or *Page*. Set *None* to leave a partition uncompressed.

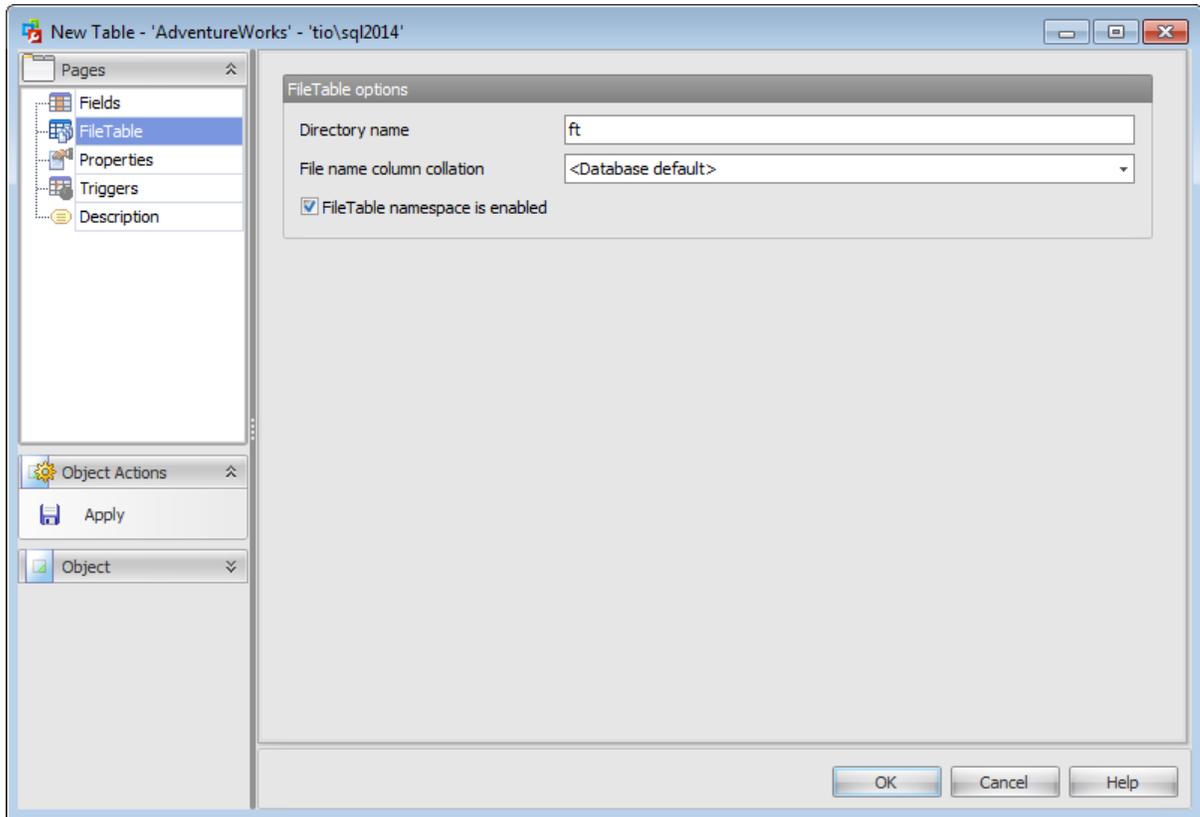
**Options**

**Lock table on bulk load**

When disabled (the default), it causes the bulk load process on user-defined tables to obtain row locks. When enabled, it causes the bulk load processes on user-defined tables to obtain a bulk update lock.

#### 5.1.1.4 Filetable options

Use the **FileTable** tab to configure settings for the FileTable being created.



##### **Directory name**

Specifies the windows-compatible FileTable directory name. This name should be unique among all the FileTable directory names in the database. If this value is not specified, the name of the filetable is used.

##### **File name column collation**

Select the name of the collation to be applied to the *Name* column in the FileTable. If this value is not specified, the database default collation is used.

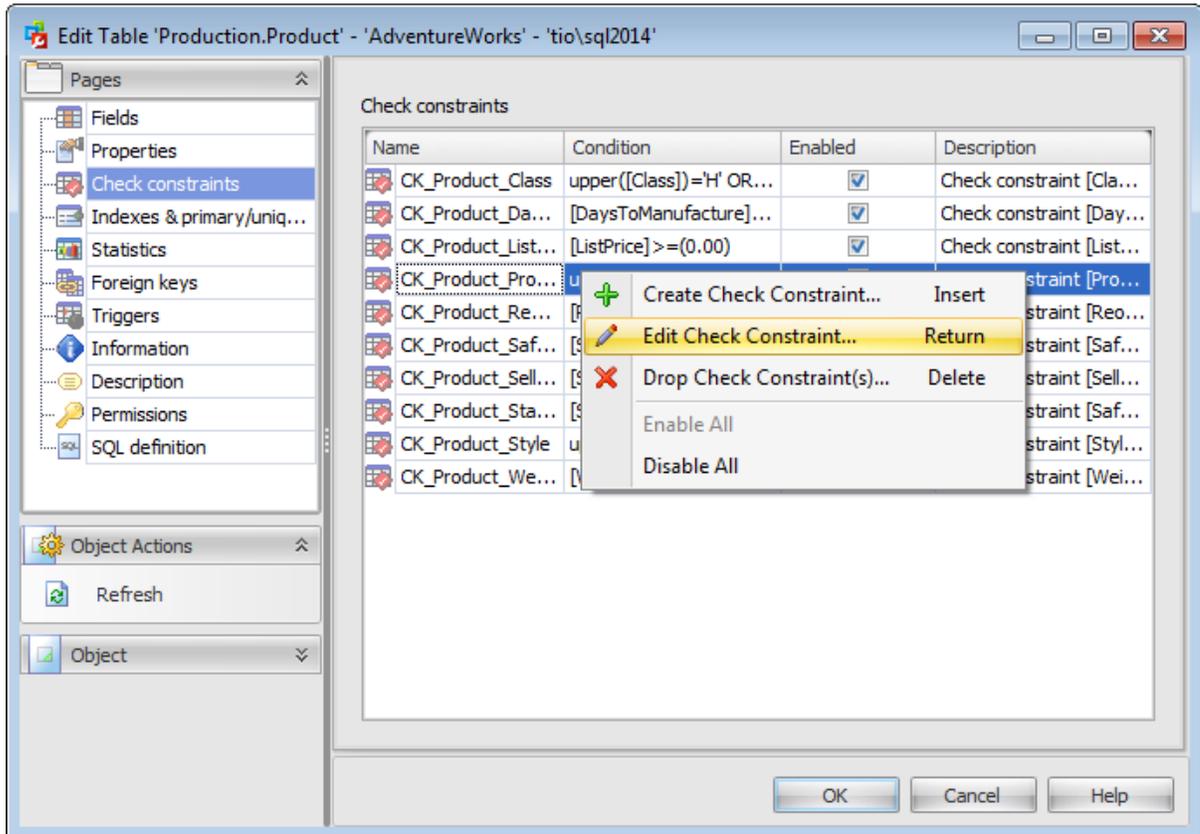
##### **FileTable namespace is enabled**

Disabling this option disables all the system-defined constraints and triggers that were created with the FileTable.

### 5.1.1.5 Check constraints

The **Check constraints** tab is provided for managing table [check constraints](#).

Right-click a check to display the context menu allowing you to *create new, edit, drop or enable/disable* all checks.



The **Check constraints** list provides the following attributes of each check constraint of the table:

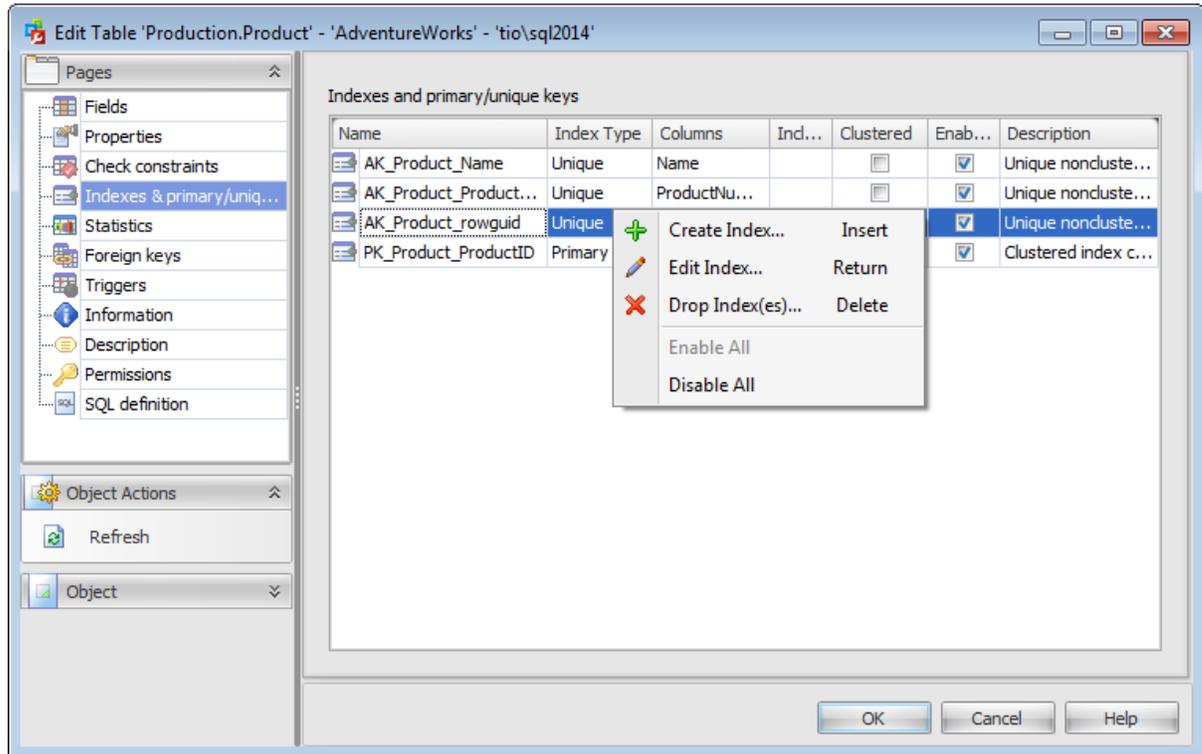
*Name*  
*Condition*  
*Enabled*  
*Description*

Double-click a check to open [Check Editor](#) for editing the check constraint.

### 5.1.1.6 Indexes and primary/unique keys

The **Indexes and primary/unique keys** tab is provided for managing table [indexes](#).

Right-click an index to display the context menu allowing you to *create new, edit, drop or enable/disable* all indexes.



The **Indexes and primary/unique keys** list provides the following attributes of each index of the table:

*Name*

*Index Type*

*Columns*

*Included Columns*

*Clustered*

*Enabled*

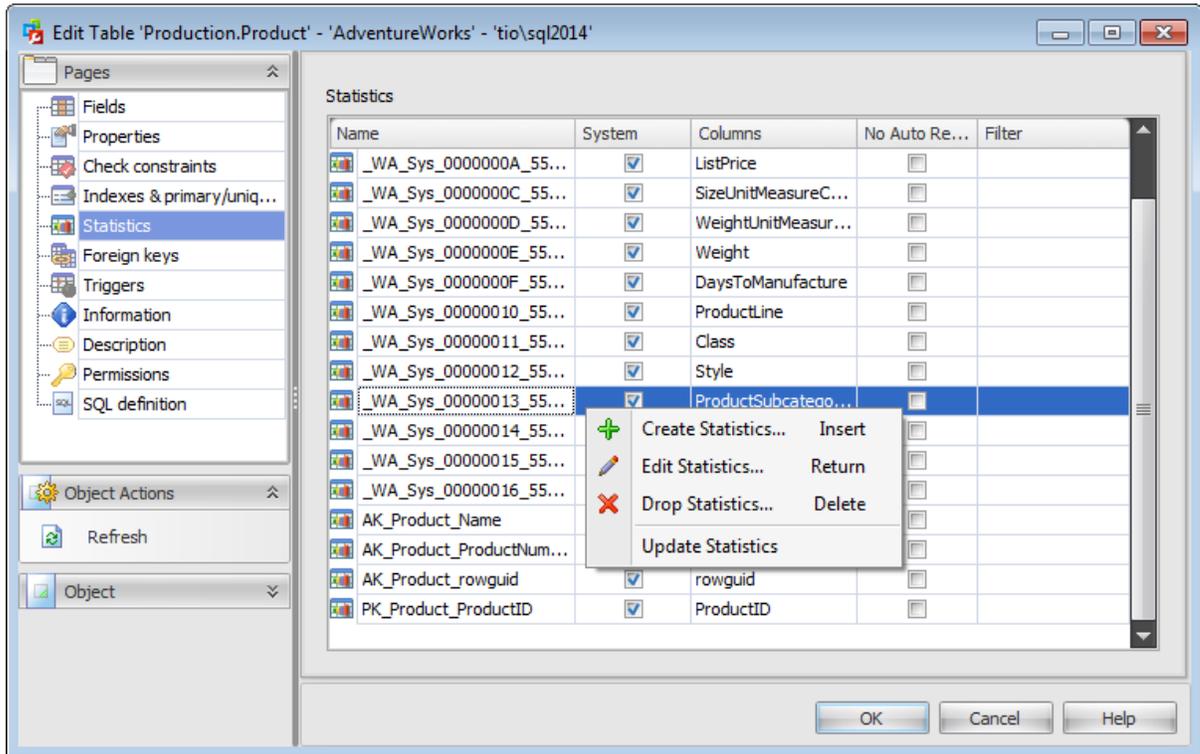
*Description*

Double-click an index to open [Index Editor](#) for editing the index.

### 5.1.1.7 Statistics

The **Statistics** tab is provided for managing table [statistics](#).

Right-click statistics to display the context menu allowing you to *create new, edit, drop or Update Statistics*.



The **Statistics** list provides the following attributes of each statistics of the table:

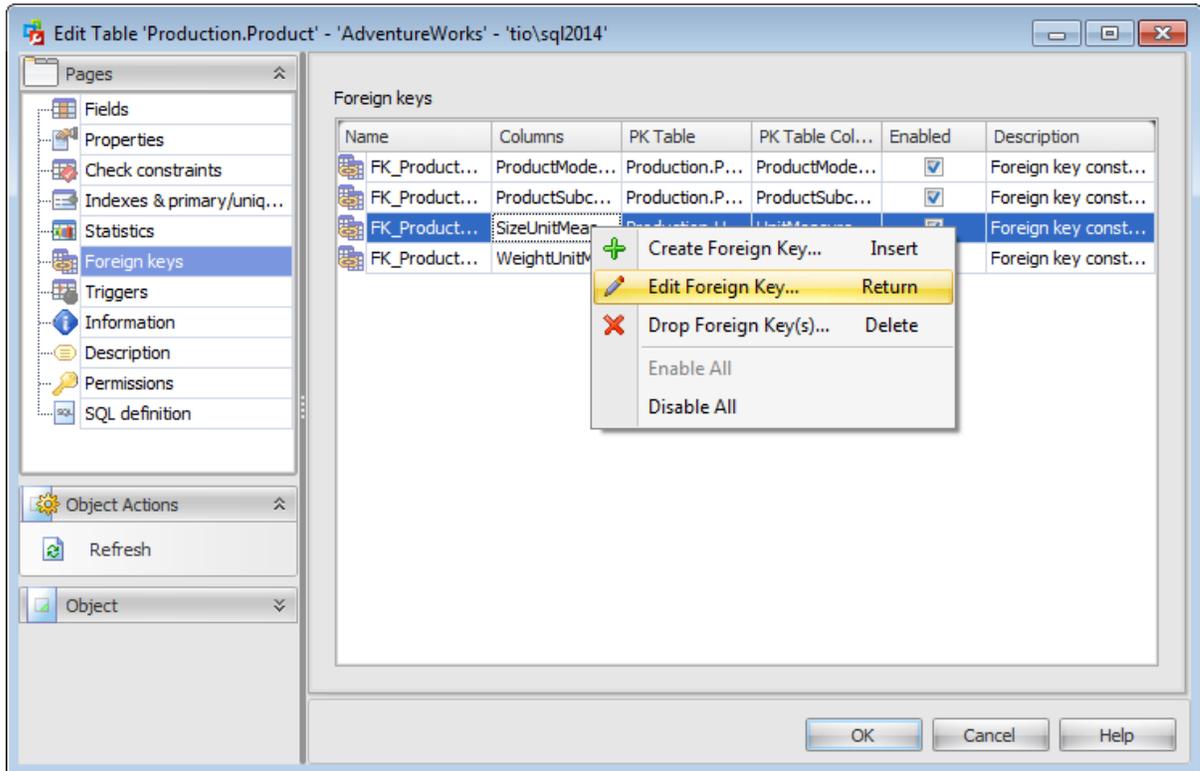
*Name*  
*System*  
*Columns*  
*No Auto Recompute*  
*Filter*

Double-click statistics to open [Statistics Editor](#) for editing the statistics.

### 5.1.1.8 Foreign keys

The **Foreign keys** tab is provided for managing table [foreign keys](#).

Right-click a foreign key to display the context menu allowing you to *create new, edit, drop or enable/disable* all foreign keys.



The **Foreign keys** list provides the following attributes of each foreign key of the table:

*Name*

*Columns*

*PK Table*

*PK Table Columns*

*Enabled*

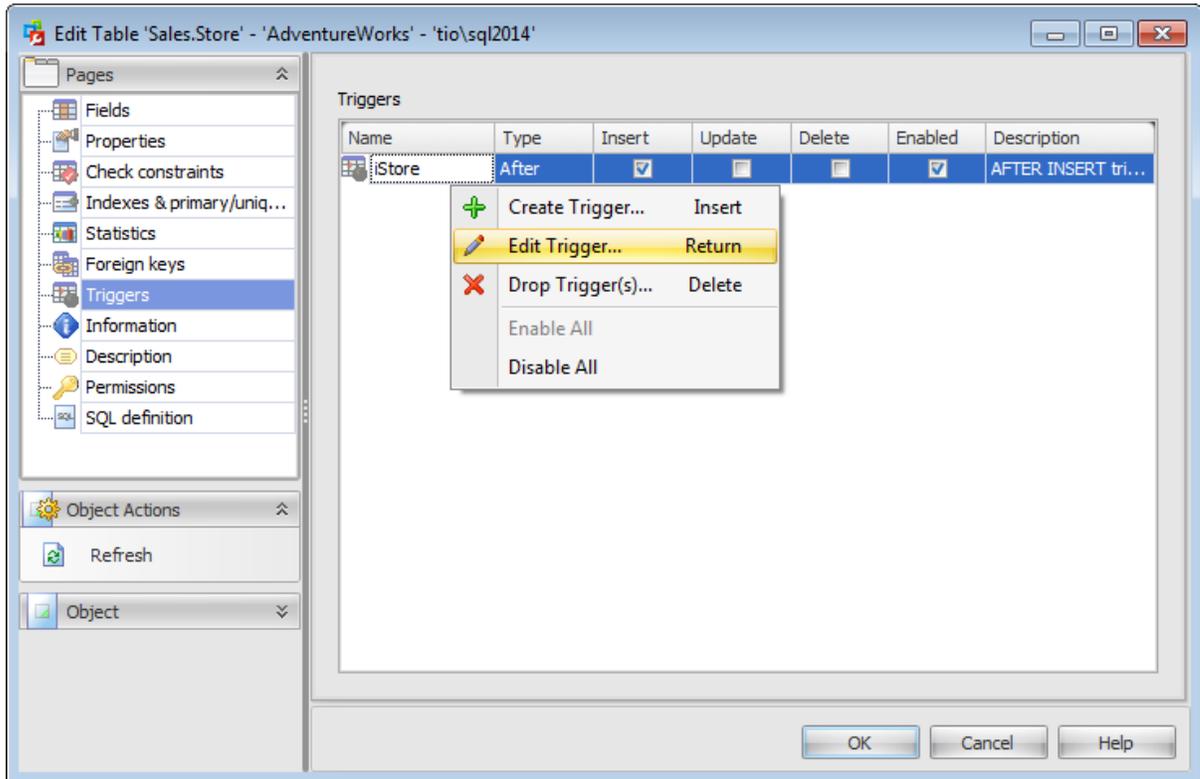
*Description*

Double-click a foreign key to open [Foreign key Editor](#) for editing the foreign key.

### 5.1.1.9 Triggers

The **Triggers** tab is provided for managing table [triggers](#).

Right-click the area to display the context menu allowing you to *create new, edit, drop or enable/disable* all triggers.



The **Triggers** list provides the following attributes of each trigger of the table:

*Name*  
*Type*  
*Insert*  
*Update*  
*Delete*  
*Enabled*  
*Description*

Double-click a trigger to open [Trigger Editor](#) for editing the trigger.

### 5.1.1.10 Information

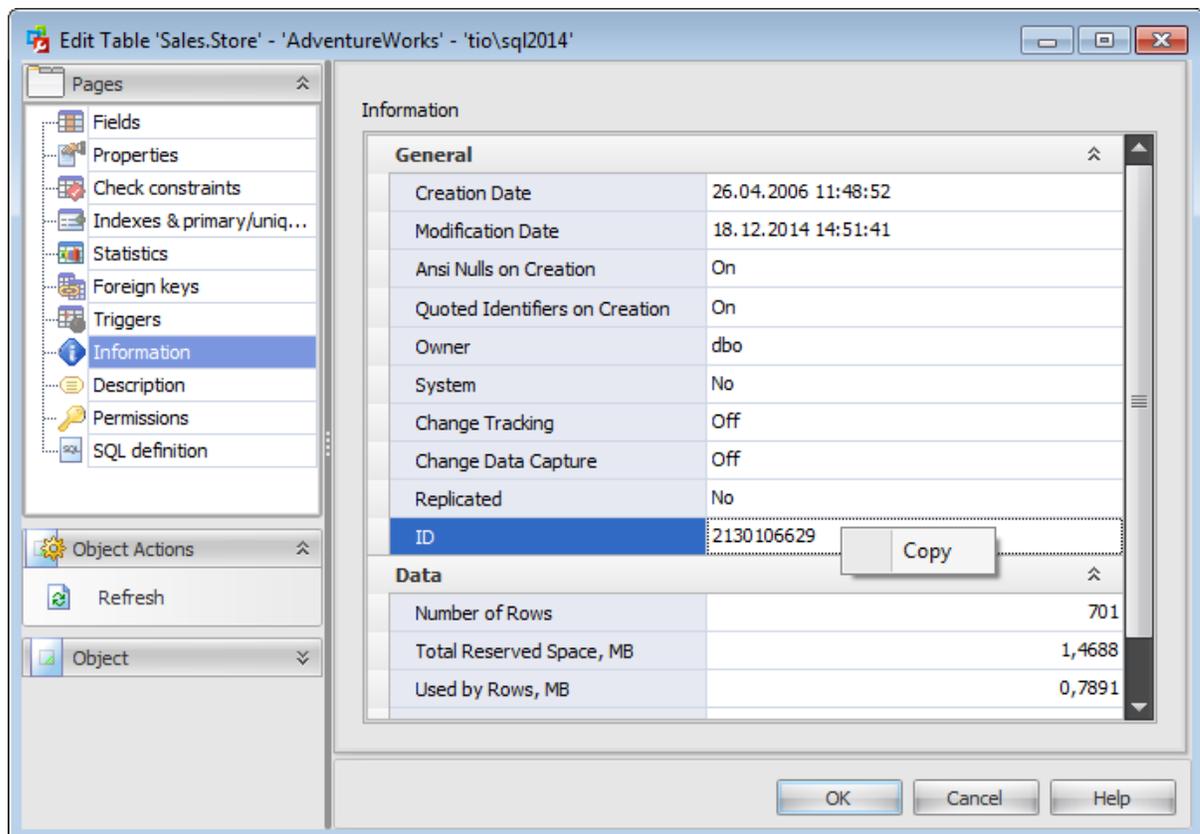
The **Information** tab allows you to view the table summary:

#### General

*Creation Date*  
*Modification Date*  
*ANSI Nulls on Creation (On/Off)*  
*Quoted Identifier on Creation (On/Off)*  
*Owner*  
*System (Yes/No)*  
*Change Tracking (Yes/No)*  
*Change Data Capture (Yes/No)*  
*Replicated (Yes/No)*  
*ID*

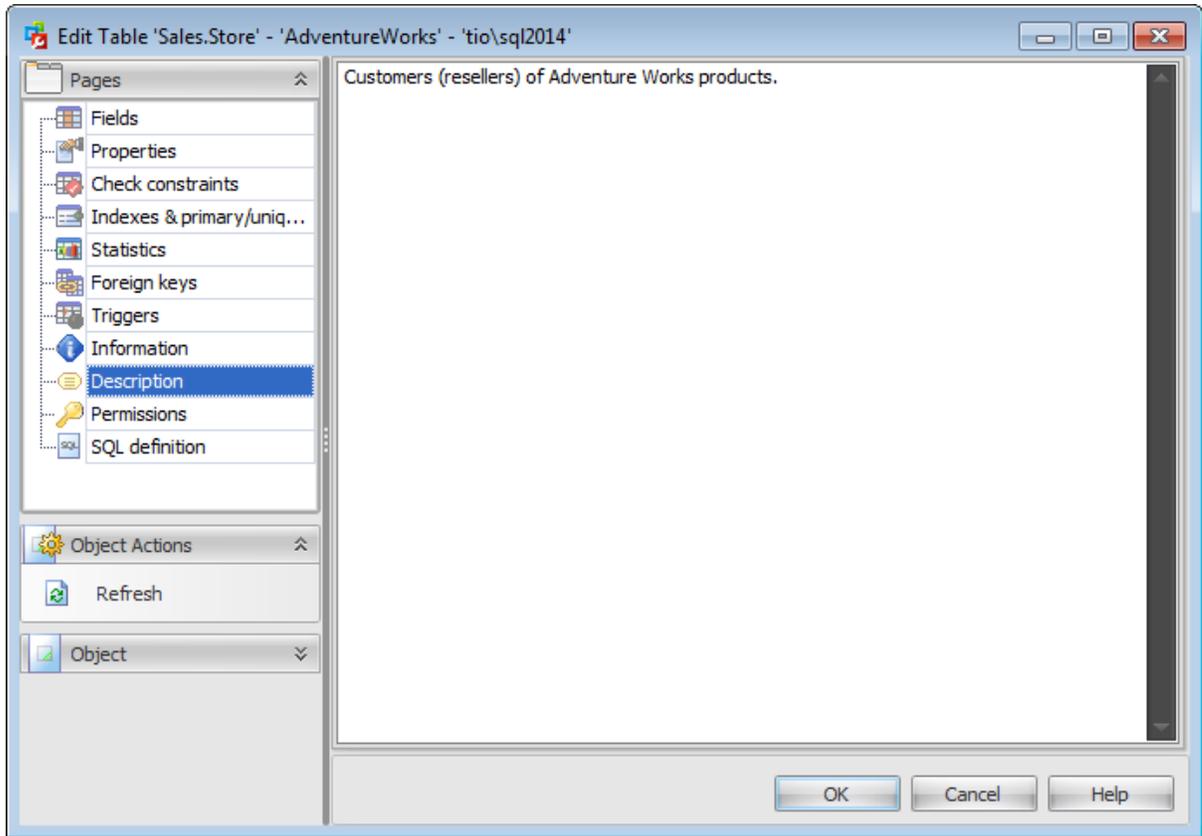
#### Data

*Number of Rows*  
*Total Reserved Space, MB*  
*Used by Rows, MB*  
*Used by Indexes, MB*



### 5.1.1.11 Description

The **Description** tab allows you to view and edit the comment for the table (optional).

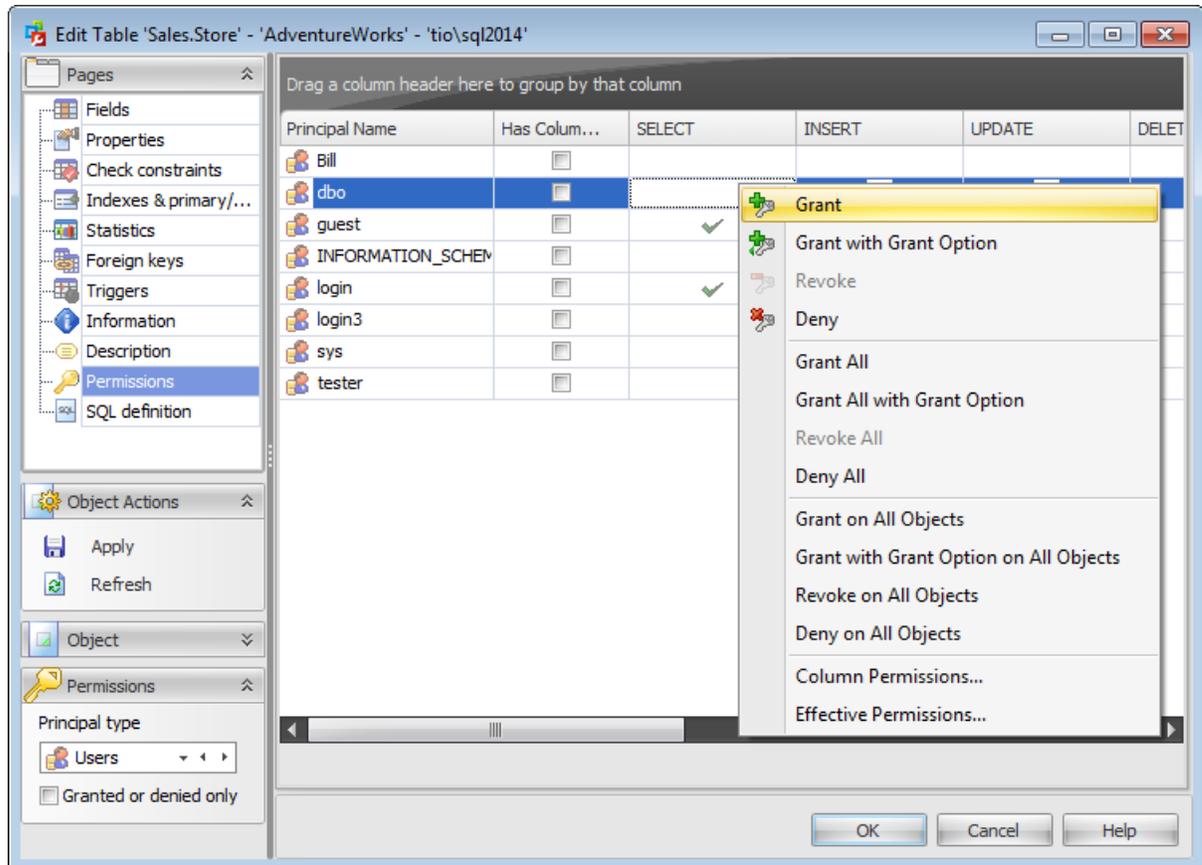


You can save changes made in this area by clicking the **Apply** item on the [Navigation bar](#).

**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.1.1.12 Permissions

The **Permissions** tab of the **Table Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [column permissions](#) and viewing [effective permissions](#).

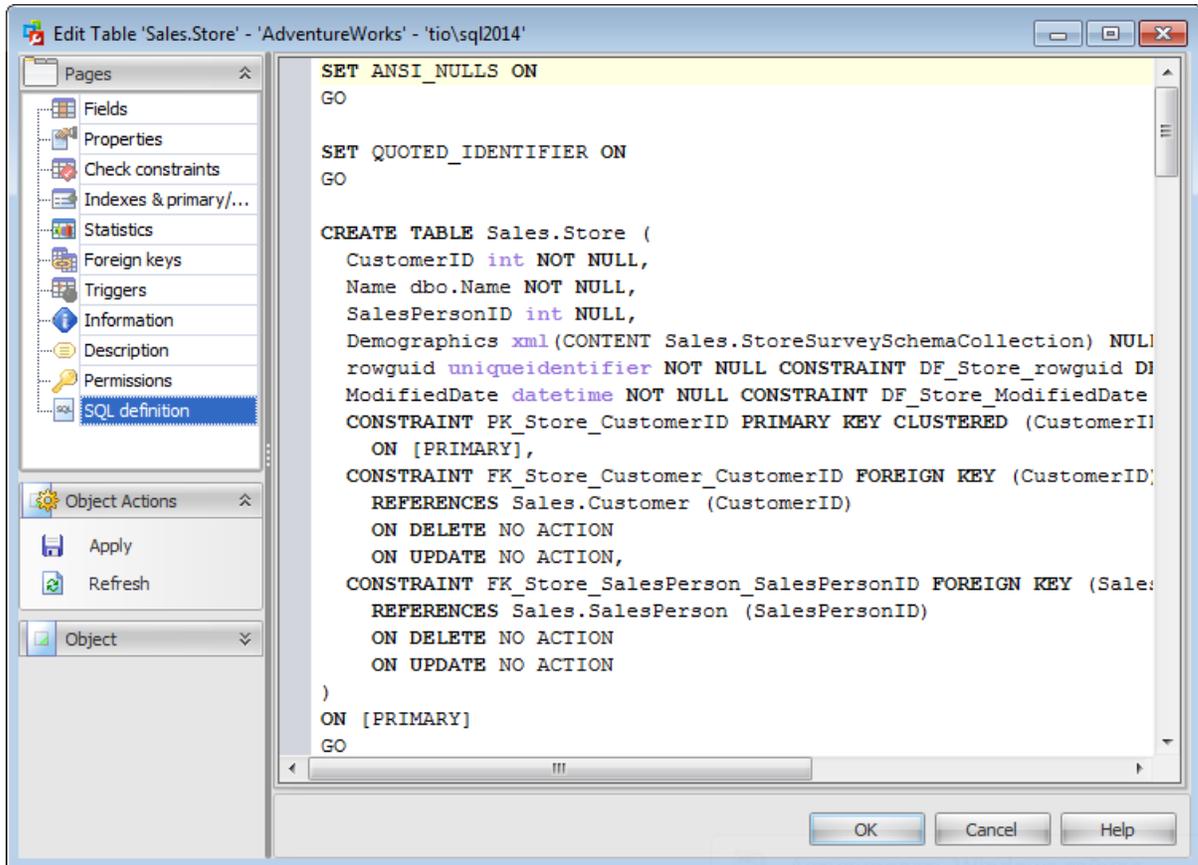
---

**See also:**

[Database Permissions](#)

### 5.1.1.13 SQL definition

The **SQL definition** tab of the **Table Editor** window allows you to view table DDL.



## 5.1.2 Field Editor

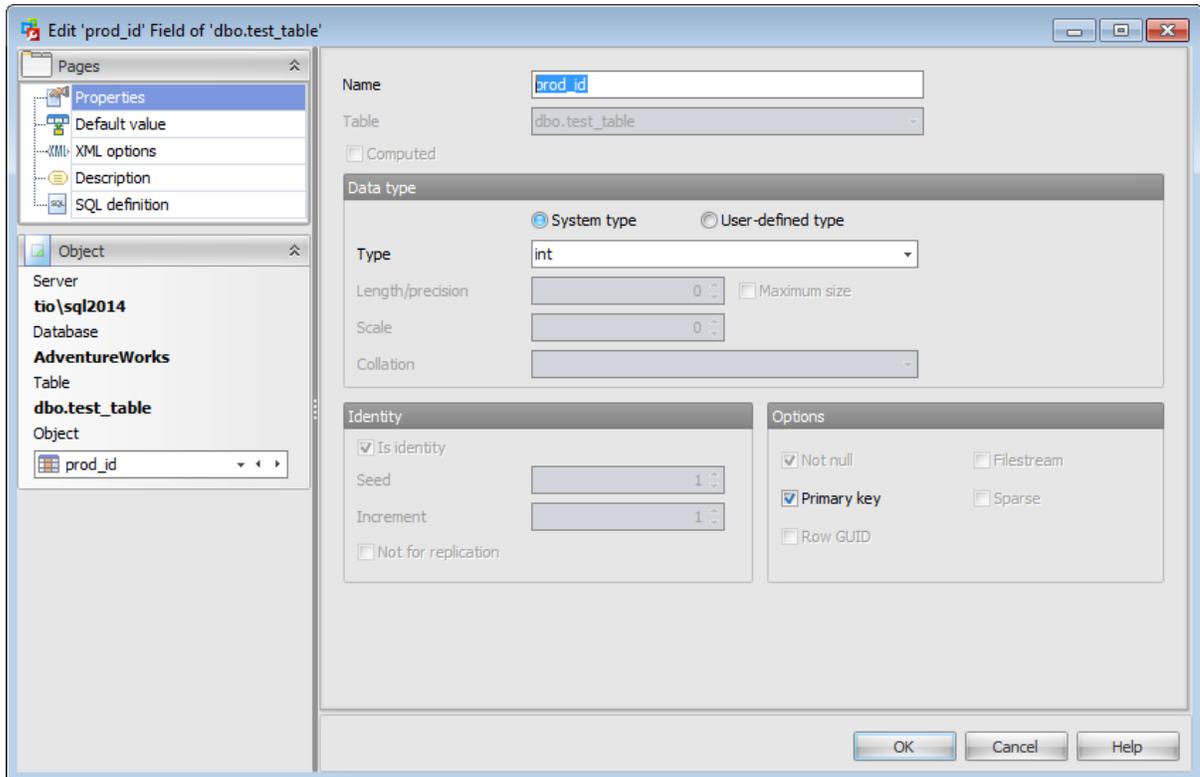
**Field Editor** allows you to specify field definition and set field properties. It opens automatically when you create a new field and is available on editing an existing one.

To open a field in **Field Editor**, double-click it in the [Fields list](#), or use the **Edit Field...** item of the context menu within the [Fields](#) tab of [Table Editor](#).

- [Properties](#)
- [Default value](#)
- [XML options](#)
- [Description](#)
- [SQL definition](#)

### 5.1.2.1 Properties

The **Properties** tab is intended for setting main field properties.



#### Name

Enter a name for the new field, or modify the name of the field being edited. Note that the name of a field must be unique among all the field names in the table.

#### Table

In this field the Table name of the field being edited is displayed.

#### Computed

Check this option to create a computed column and set the expression that defines its value.

#### Store computed value in table

This option determines whether computed values are stored in the table or the column is not physically stored in the table.

#### Expression

Specify an expression that defines the value of the computed column. For example, a computed column can have the following definition:

*cost AS price \* qty.*

The expression can be a noncomputed column name, constant, function, variable, and any combination of these connected by one or more operators. The expression cannot be a subquery or include an alias data type.

### Data type

Select whether the field being created/edited is based on a standard data type or a [user-defined type](#).

### Type

Here you can set the field type by selecting it from the drop-down list of the standard SQL Server data types or [user-defined data types](#).

### Length/precision

Specify the size value (for certain types). Check the  **Maximum size** option to use the maximum values set by SQL Server.

### Scale

Defines the maximum number of decimal digits that can be stored to the right of the decimal point of the value (for numeric and decimal data types).

### Collation

Use the drop-down list to specify the collation for text and char columns. By default, the column is assigned the default collation of the database.

### Identity

#### **Is identity**

This option specifies that the new column is an identity column.

### Seed

Specify the value used for the first row that is loaded into the table.

### Increment

Specify the value added to the identity value of the previous row that is loaded.

#### **Not for replication**

This option allows you to specify that the Identity property should not be enabled when a replication agent modifies the table that contains the identity column.

### Options

#### **Not NULL**

Check this option to specify that the values for the column should never contain a null value. *NOT NULL* affects all INSERT and UPDATE operations on a column.

#### **Primary key**

Check this option to include the field into the primary key. Note that if you include a field to a primary key, you should also make it *Not Null*.

A table typically has a column or combination of columns that contain values that uniquely identify each row in the table. This column, or columns, is called the primary key (PK) of the table and enforces integrity of the table.

#### **Row GUID**

This option specifies that the *ROWGUIDCOL* property is added to the column.

*ROWGUIDCOL* indicates that the column is a row GUID column. This option can be set only for *uniqueidentifier* fields.

#### **Filestream**

Check this option to store the data from this column in the [Filestream filegroup](#). This

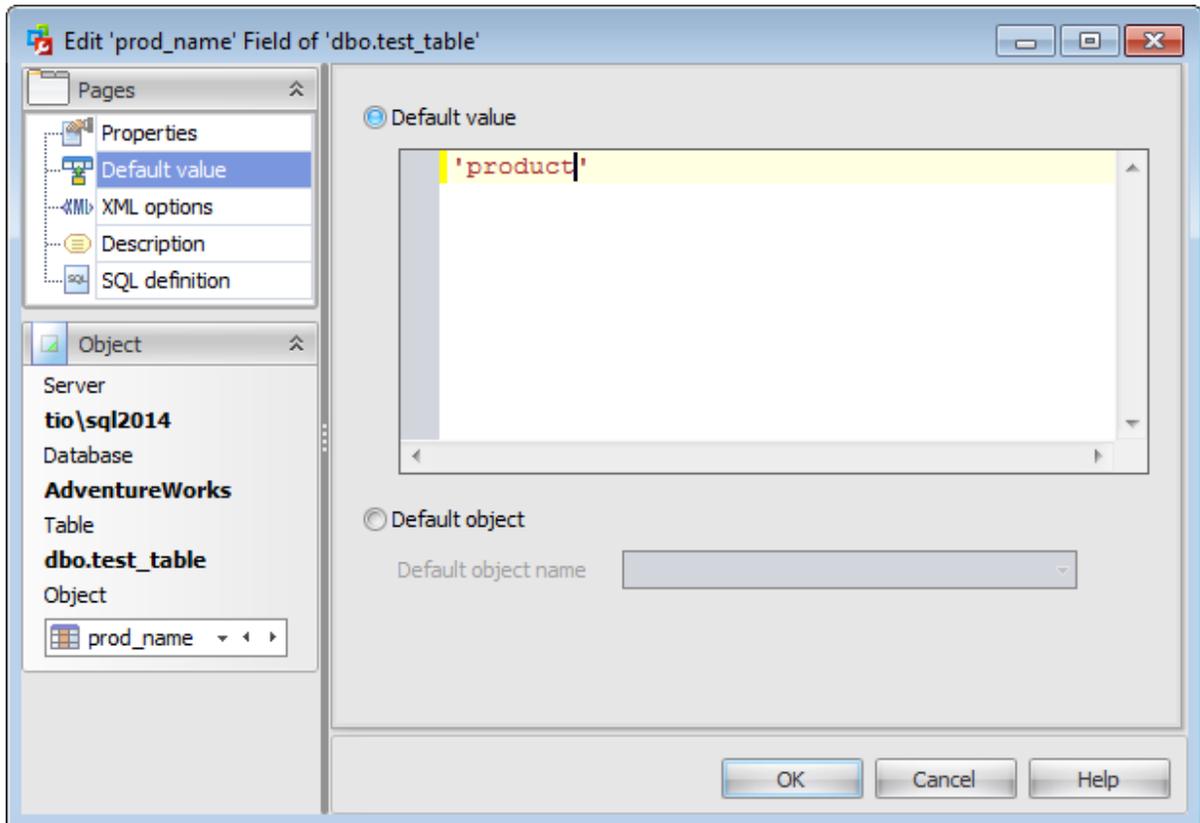
option is available if any filestream file group is specified for the table.

**Sparse**

Check this option to make the column a Sparse column. Sparse columns are ordinary columns that have an optimized storage for null values. Sparse columns reduce the space requirements for null values at the cost of more overhead to retrieve non-null values.

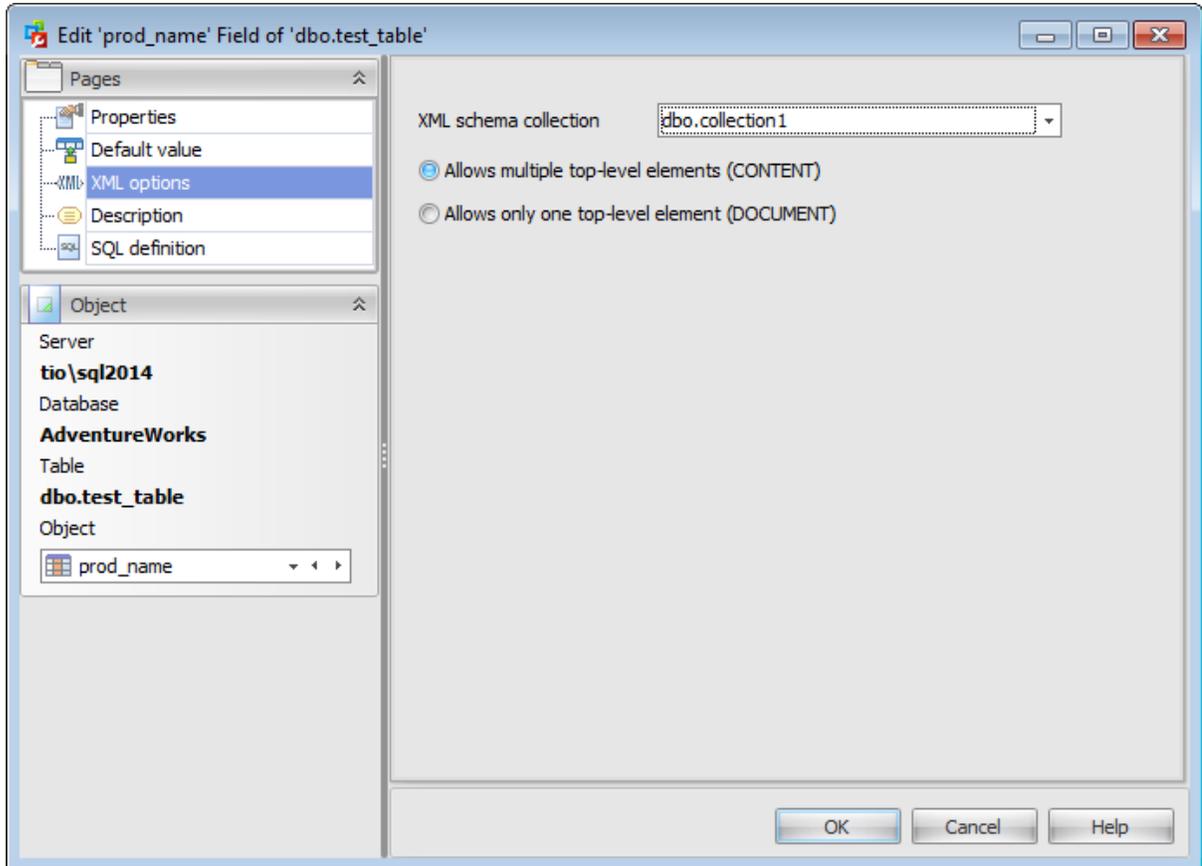
### 5.1.2.2 Default value

The **Default value** tab allows you to define the default value for the column. You can either select the default object from the list of database Defaults or define a default value for the column.



### 5.1.2.3 XML options

The **XML** tab is only available for XML field type and for [user-defined types](#) based on XML type.



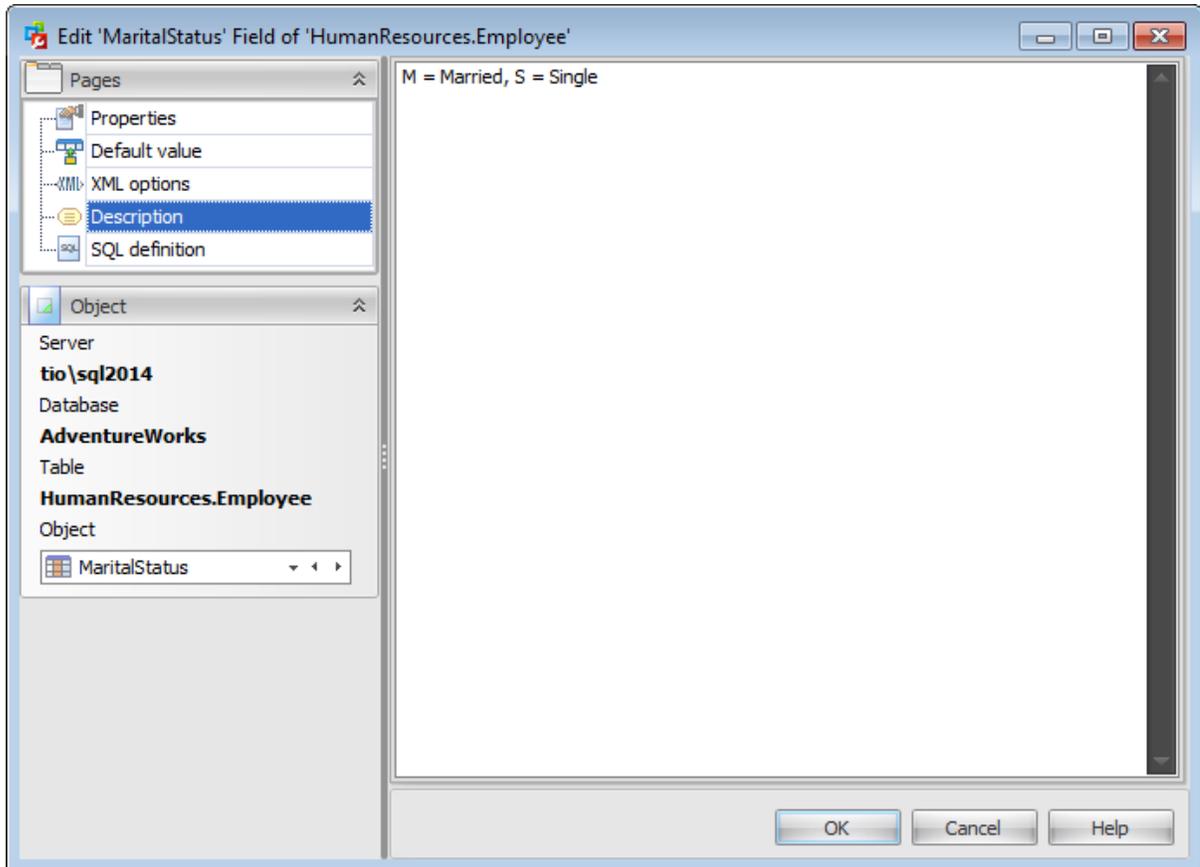
#### XML schema collection

Use the drop-down list to select an XML schema collection to be used by this field.

The **XML schema collection** stores the imported XML schemas. The schemas in the collection are used to validate the XML instances. By default, instances stored in the typed xml column are stored as XML content. For typed xml columns, you can constrain the column to allow only single, top-level elements for each instance stored in it. You can do this by selecting the  **Allows only one top-level element (DOCUMENT)** option.

#### 5.1.2.4 Description

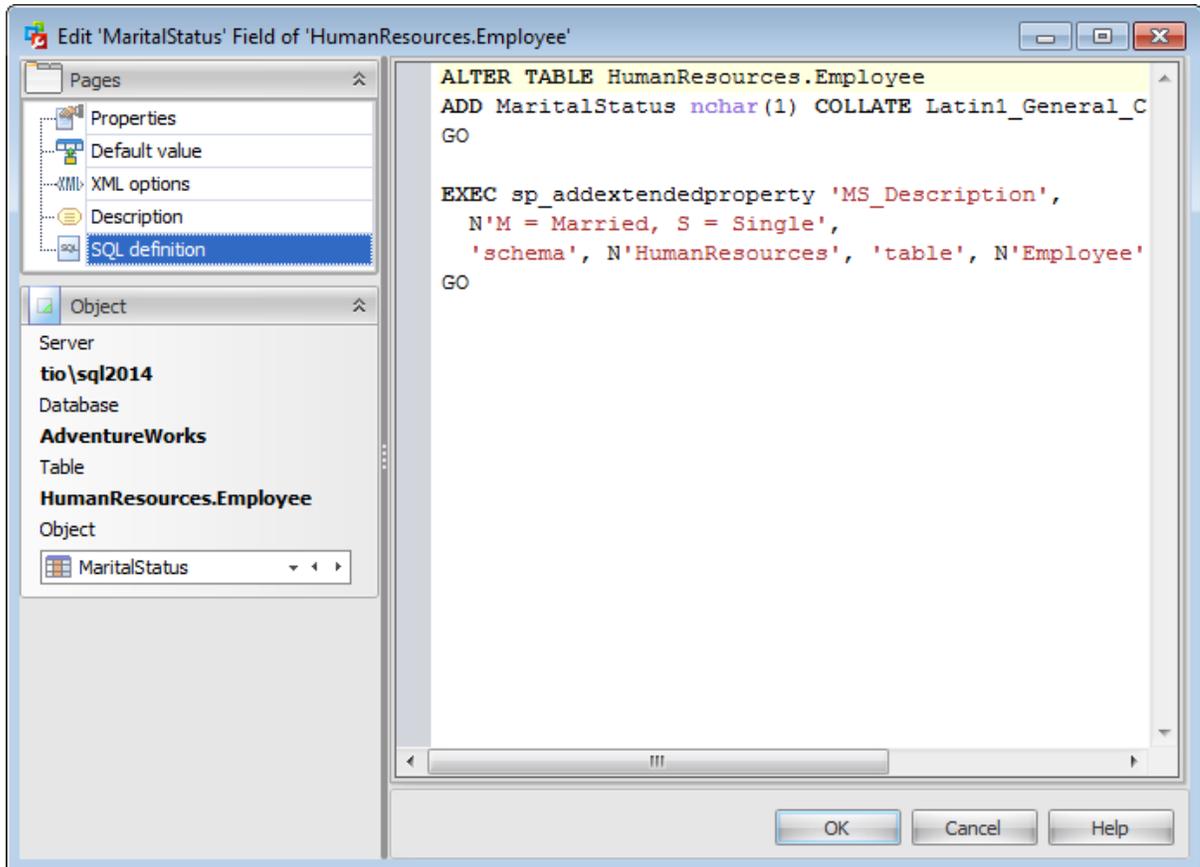
The **Description** tab allows you to view and edit the comment for the field (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.1.2.5 SQL definition

The **SQL definition** tab of the **Field Editor** window allows you to view field DDL.



### 5.1.3 Check Constraint Editor

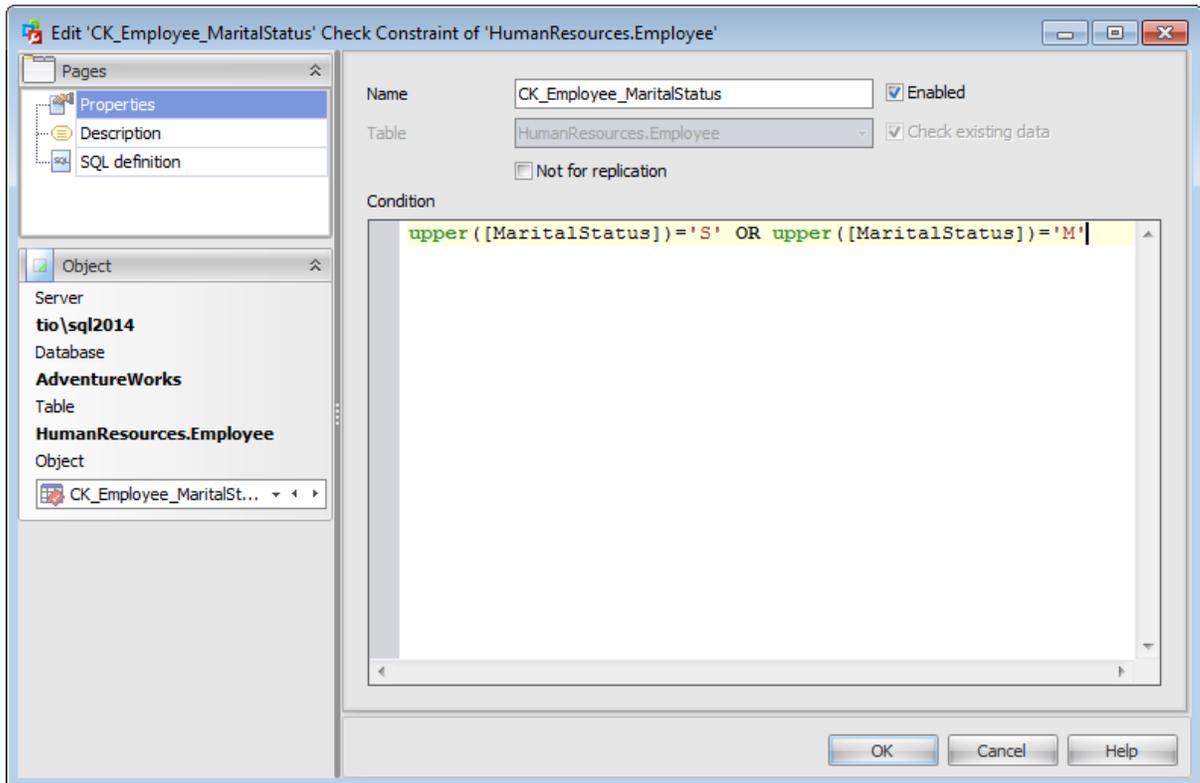
A **Check** specifies an expression producing a Boolean result which new or updated rows must satisfy for an insert or update operation to succeed. Expressions evaluating to TRUE or UNKNOWN succeed. Should any row of an insert or update operation produce a FALSE result an error exception is raised and the insert or update does not alter the database. A check constraint specified as a column constraint should reference that column's value only, while an expression appearing in a table constraint may reference multiple columns.

To open a check constraint in **Check Constraint Editor**, double-click it in the [Checks list](#), or use the **Edit Check Constraint...** item of the context menu within the [Check constraints](#) tab of [Table Editor](#).

- [Properties](#)
- [Description](#)
- [SQL definition](#)

### 5.1.3.1 Properties

Use the **Properties** tab of the **Check Constraint Editor** to create/edit a check constraint and specify its properties.



#### Name

Enter a name for the new check, or modify the name of the check being edited.

#### Table

In this field the Table name of the check being edited is displayed.

#### Enabled

Enables/disables the check immediately after it is created.

#### Check existing data

This option specifies whether the data in the table is or is not validated against a newly added or re-enabled check.

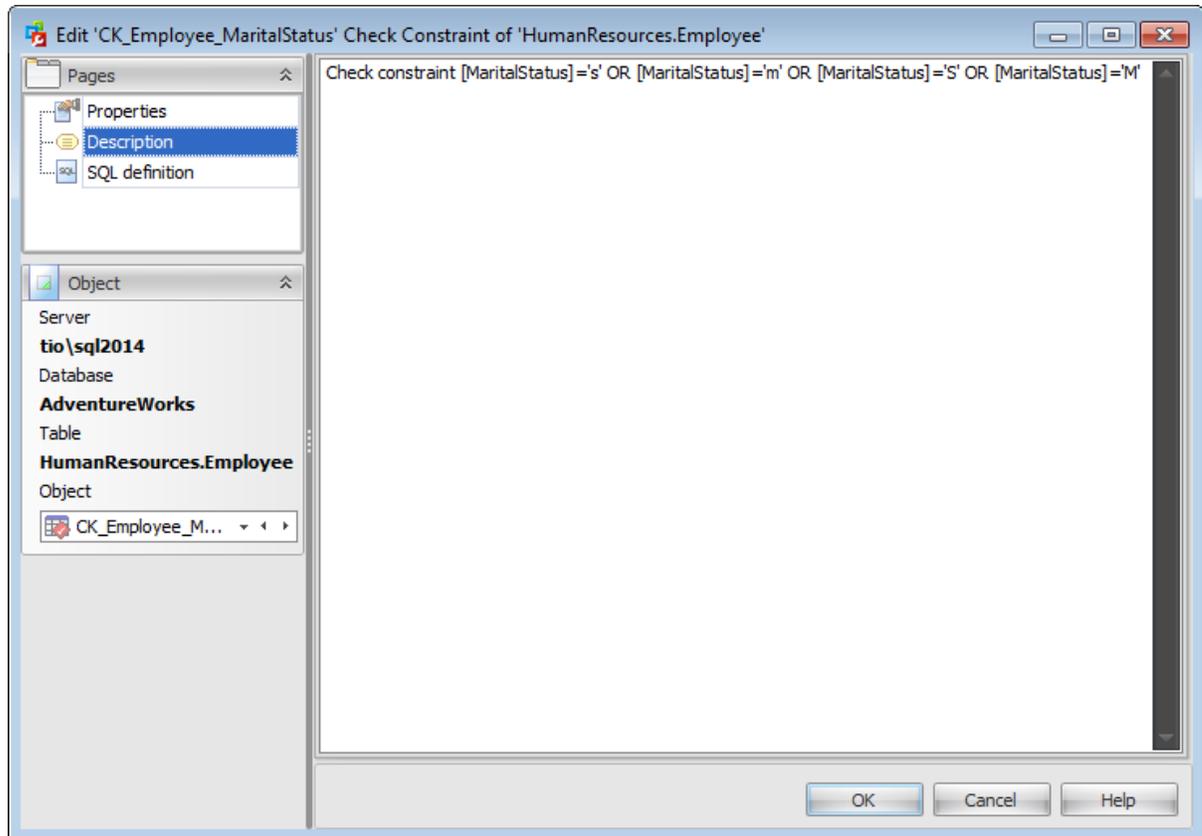
#### Condition

This area represents the condition implied by the check constraint.

You can specify any logical (Boolean) expression that returns *TRUE* or *FALSE* based on the logical operators. For example, *salary >= 15000 AND salary <= 100000*.

### 5.1.3.2 Description

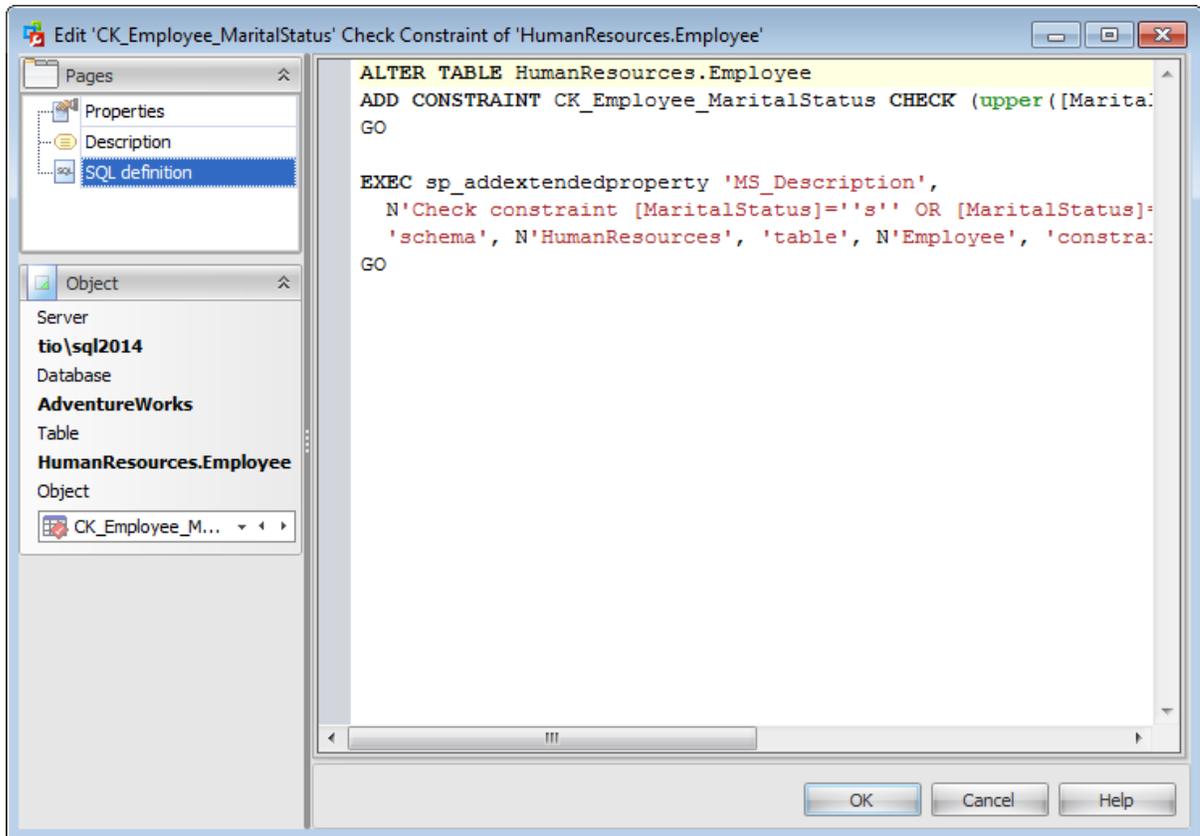
The **Description** tab allows you to view and edit the comment for the check constraints (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.1.3.3 SQL definition

The **SQL definition** tab of the **Check Constraint Editor** window allows you to view check DDL.



### 5.1.4 Index Editor

**Index Editor** allows you to specify index definition and set index properties. It opens automatically when you create a new index and is available on editing an existing one.

To open an index in **Index Editor**, double-click it in the [Indexes and primary/unique keys list](#), or use the **Edit Index...** item of the context menu within the [Indexes and primary/unique keys](#) tab of [Table Editor](#).

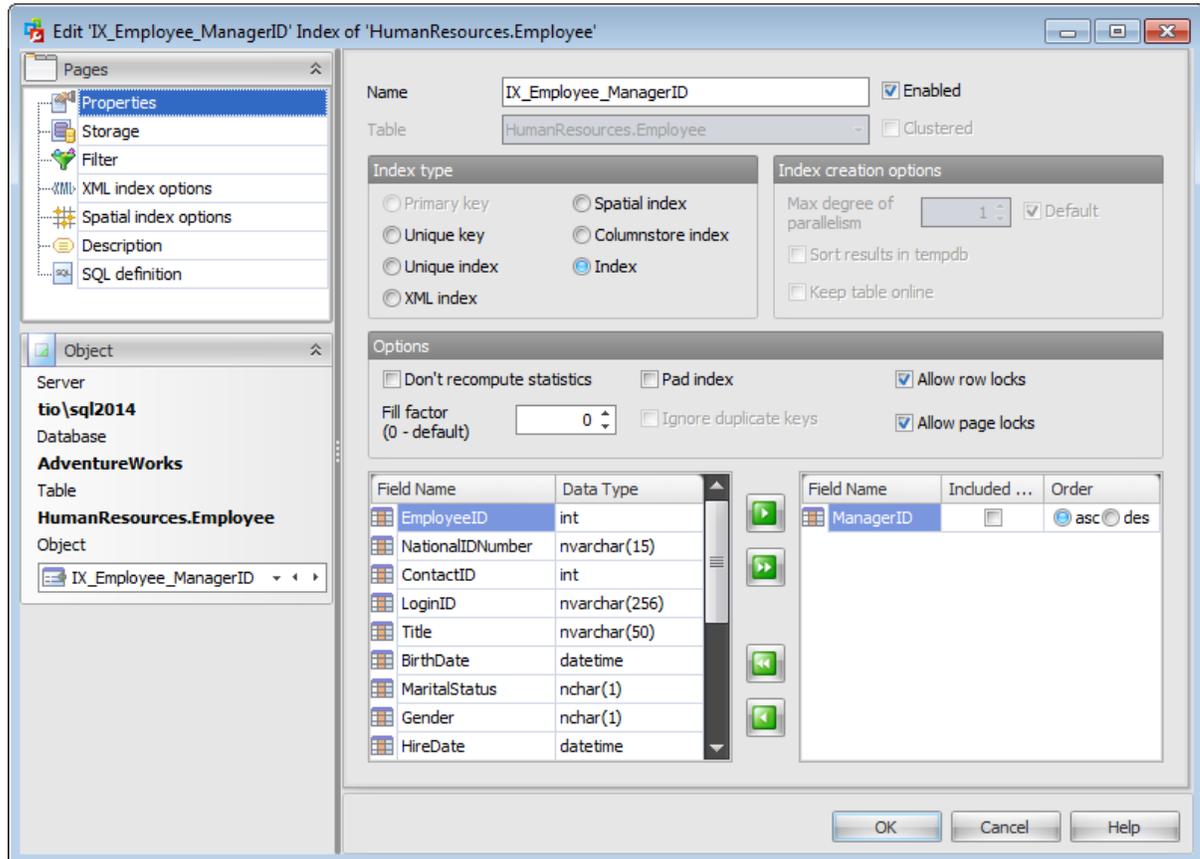
- [Properties](#)
- [Storage](#)
- [Filter](#)
- [XML index options](#)
- [Spatial index options](#)
- [Description](#)
- [SQL Definition](#)

### 5.1.4.1 Properties

Use the **Properties** tab of the **Index Editor** to create/edit an index on a specified [table](#) and specify index properties.

#### Name

Enter a name for the new index, or modify the name of the index being edited.



#### Table

In this field the Table name of the index being edited is displayed.

#### Enabled

Enables/disables the index immediately after it is created.

#### Clustered

Use this option to create an index in which the logical order of the key values determines the physical order of the corresponding rows in a table. The bottom, or leaf, level of the clustered index contains the actual data rows of the table. A [table](#) is allowed one clustered index at a time.

#### Index type

##### Primary Key

This selection indicates that the primary key index is created.

**Unique key**

This selection indicates that the unique key index is created (no two rows are permitted to have the same index value). A clustered index on a view must be unique.

**Unique index**

This selection indicates that the unique index is created (ensures that the index key contains no duplicate values and therefore each row in the table or view is in some way unique). Both clustered and nonclustered indexes can be unique.

**XML index**

This selection indicates that the XML index on the specified XML column is created.

**Spatial index**

This selection indicates that the spatial index is created. These indexes are used by spatial databases (databases which store information related to objects in space).

**Columnstore index**

This selection indicates that the columnstore index is created. These indices group and store data for each column and then join all the columns to complete the whole index.

**Index**

This selection specifies a regular, non-unique index.

### Index creation options

#### Max degree of parallelism

This option allows you to limit the number of processors used in a parallel plan execution. Set a value to restrict the maximum number of processors used in a parallel index operation to the specified number or fewer based on the current system workload. If 0 or

**Default** is specified, the actual number of processors is used.

**Sort in tempdb**

This option specifies whether temporary sort results should be stored in **tempdb**.

**Keep table online**

This option specifies whether underlying tables and associated indexes are available for queries and data modification during the index operation.

### Options

**Do not recompute statistics**

This option specifies that out-of-date index statistics are not automatically recomputed.

**Pad index**

This option determines whether the percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

**Ignore duplicate keys**

This option is available if the Unique option is checked. This option specifies the error response to duplicate key values in a multiple-row insert operation on a unique clustered or unique nonclustered index.

In cases when access patterns are well understood and consistent, limiting the locking levels available for an index can be beneficial. Disallowing a locking level can affect the concurrency for a [table](#) or index.

**Allow row locks**

This option specifies whether row locks are allowed when accessing the index.

**Allow page locks**

This option specifies whether page locks are allowed when accessing the index.

**Hash index options** (for memory-optimized tables only)

**Hash index**

Checking this option indicates that a HASH index is created. A hash index consists of a collection of buckets organized in an array.

**Number of buckets**

Define the number of buckets that should be created in the hash index.

**Fields for index**

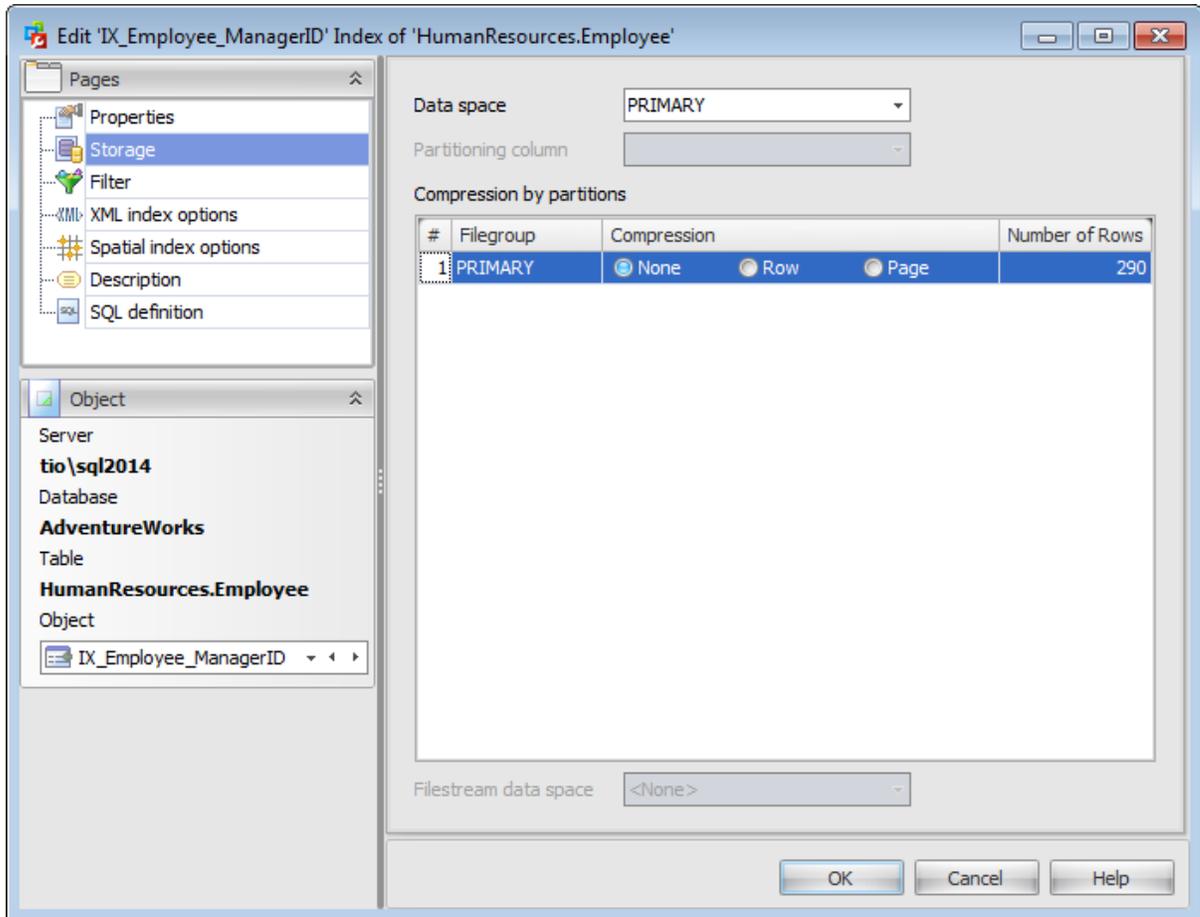
To include field(s) in the index, you need to move it from the **Available** list to the **Selected** list. Use the     buttons or drag-and-drop operations to move the fields from one list to another.

To change the sorting order for a field, select the field in the **Selected** list and change the **Order** value (*ascending/descending*).

Set the  **Included Only** flag to specify that the non-key column is added to the leaf level of the nonclustered index.

### 5.1.4.2 Storage

At the **Storage** tab of the **Index Editor** you need to define the storage for the index.



#### Data space

Use the drop-down list to specify the new index data storage (a file group or a [partition scheme](#) that defines the filegroups onto which the partitions of a partitioned index will be mapped).

**Note:** When creating an index you need not specify the data space; the data space of the [table](#) is used by default.

#### Partitioning column

This drop-down list is only available if a [partition scheme](#) is specified for **Data space**. Select a [field](#) that will be used as the partitioning column for the table.

#### Compression by partitions

Use this section to set the compression options for the index. Compression type can be different for each partition. Set the index compression to *Row* or *Page*. Set *None* to leave a partition uncompressed.

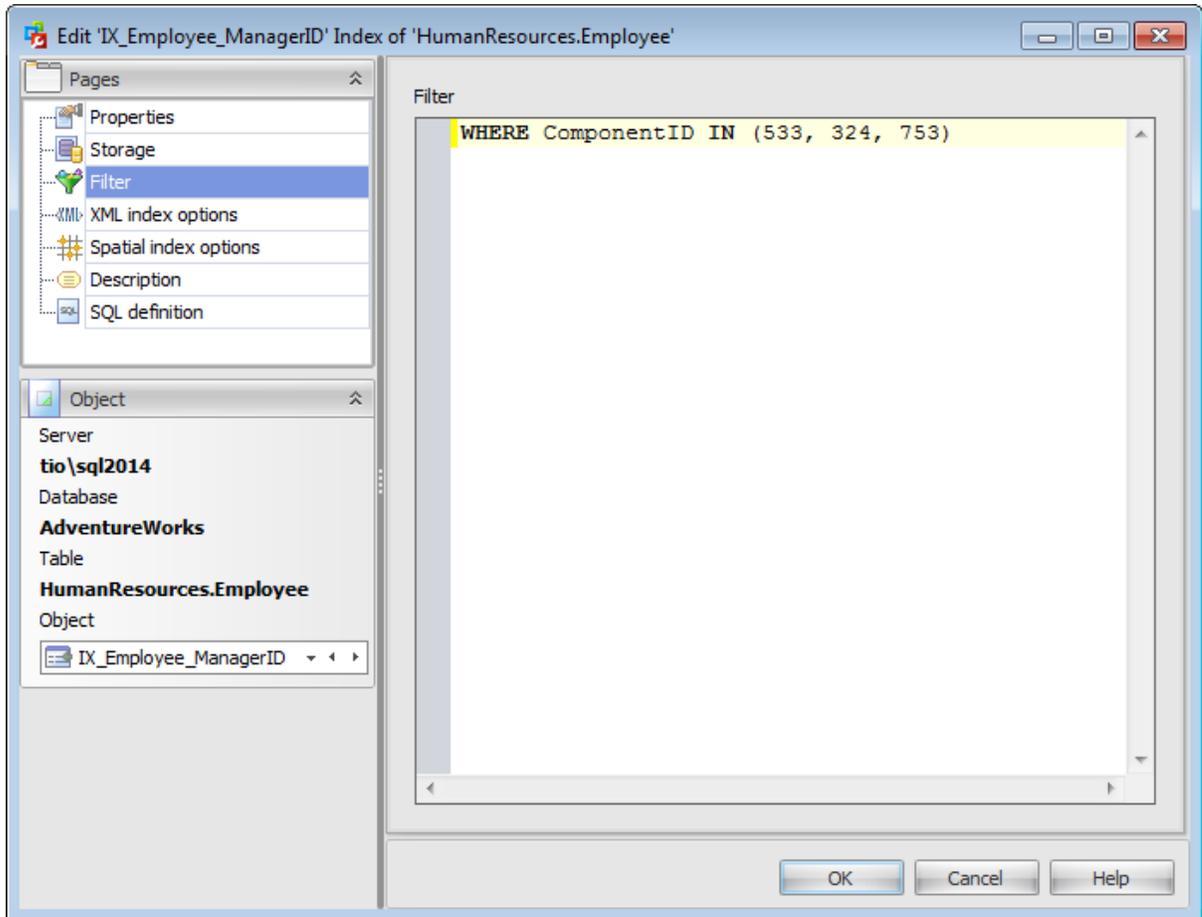
#### Filestream dataspace

Select the needed [Filestream filegroup](#) from the drop-down list. This option is available

only in SQL Server 2008 and higher.

### 5.1.4.3 Filter

Use the **Filter** tab to set the filter predicate for the index.



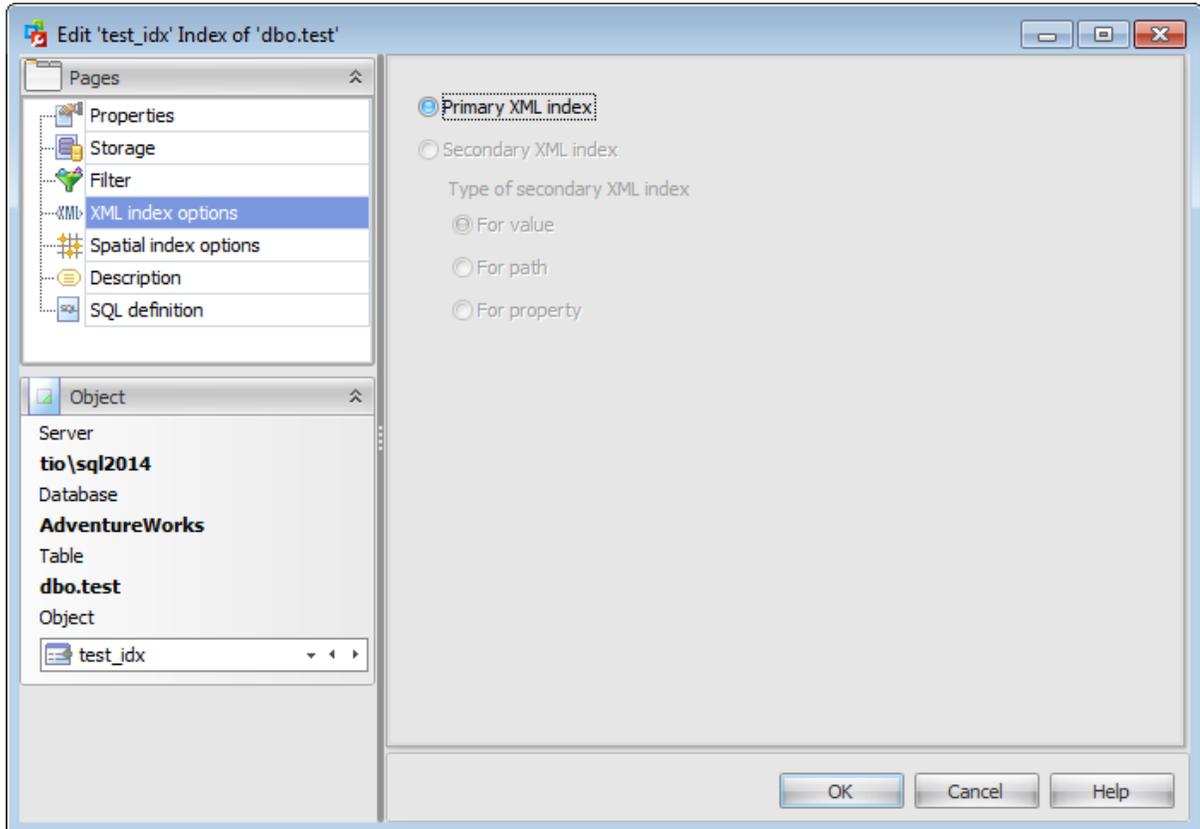
#### Filter

Use this text field to specify which rows to include in the index to create a filtered index. The filtered index must be a nonclustered index on a table. Creates filtered [statistics](#) for the data rows in the filtered index.

The filter predicate uses simple comparison logic and cannot reference a computed column, a UDT column, a spatial data type column, or a hierarchyID data type column. Comparisons using NULL literals are not allowed with the comparison operators. Use the IS NULL and IS NOT NULL operators instead.

#### 5.1.4.4 XML index options

Use the **XML index options** tab to set properties for the XML index being created. The settings on this tab are available only if the  **XML index** was selected on the [Properties](#) tab.



##### **Primary XML index**

A clustered index is created with the clustered key formed from the clustering key of the user table and an XML node identifier.

##### **Secondary XML index**

Secondary XML index can be created only in case the Primary XML index has been already created. Secondary XML index may be one of the following types:

###### *For value*

This selection indicates creation of a secondary XML index on columns where key columns are (node value and path) of the primary XML index. If your workload involves querying for values from XML instances without knowing the element or attribute names that contain the values, VALUE index may be useful.

###### *For path*

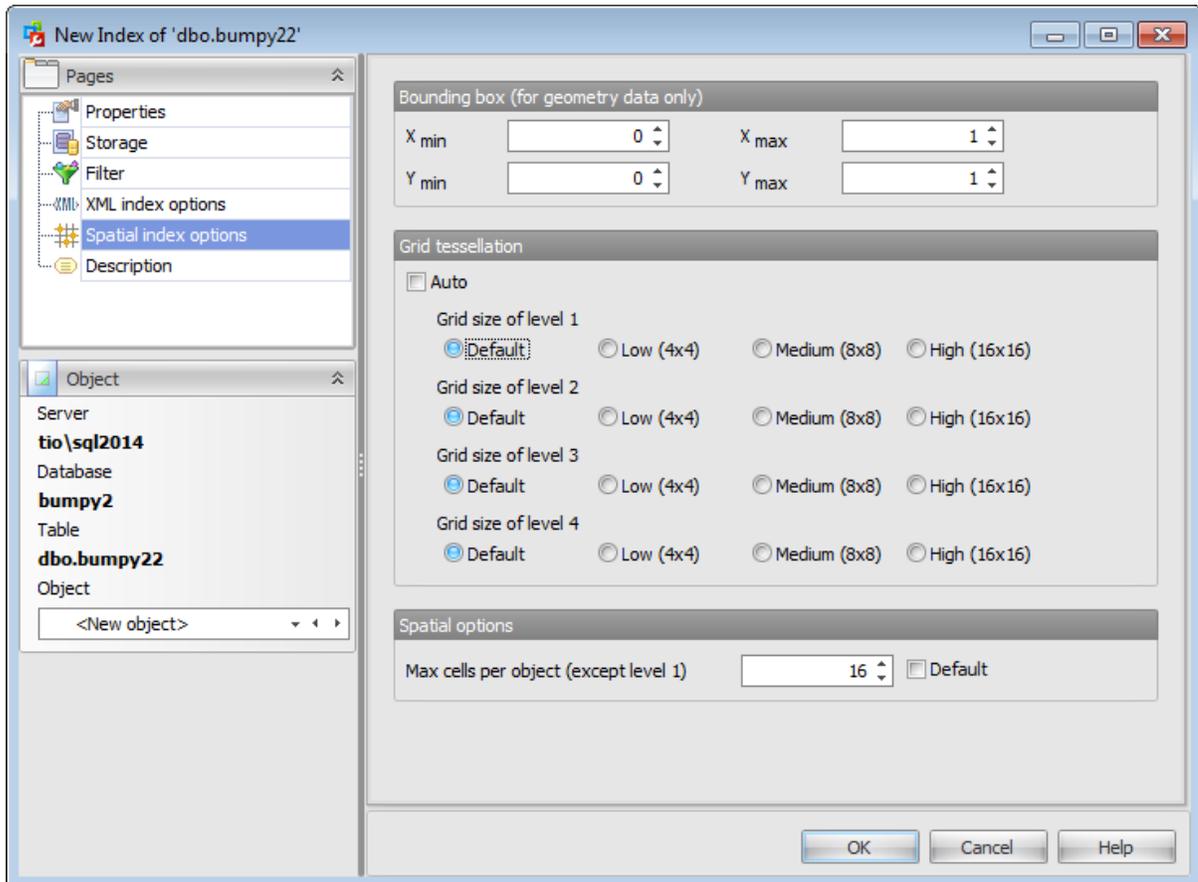
This selection indicates creation of a secondary XML index on columns built on path values and node values in the primary XML index. If your queries generally specify path expressions on xml type columns, a PATH secondary index may be able to speed up the search.

*For property*

This selection indicates creation of a secondary XML index on columns (PK, path and node value) of the primary XML index where PK is the primary key of the base table. Queries that retrieve one or more values from individual XML instances may benefit from a PROPERTY index.

#### 5.1.4.5 Spatial index options

Use the **Spatial** tab to set spatial index options. The settings on this tab are available only if the  **Spatial index** was selected on the [Properties](#) tab. Spatial indices are enabled in SQL Server 2008 and higher.



#### Bounding box (for geometry data only)

Specifies a numeric four-tuple that defines the four coordinates of the bounding box: the x-min and y-min coordinates of the **Lower-left corner**, and the x-max and y-max coordinates of the **Upper-right corner**.

#### Grid tessellation

Define the density of the grid at each level of a tessellation scheme.

##### Automatic grid tessellation

Check this option to set default values for this data type.

#### Grid size of level n

##### Default

If the database compatibility level is set to 100 or lower, then the default is MEDIUM on all levels. When the database compatibility level is set to 110 or higher, then the default is an auto grid scheme.

• **Low (4x4)**

Specifies the lowest possible density for the grid at a given level. LOW equates to 16 cells (a 4x4 grid).

• **Medium (8x8)**

Specifies the medium density for the grid at a given level. MEDIUM equates to 64 cells (an 8x8 grid).

• **High (16x16)**

Specifies the highest possible density for the grid at a given level. HIGH equates to 256 cells (a 16x16 grid).

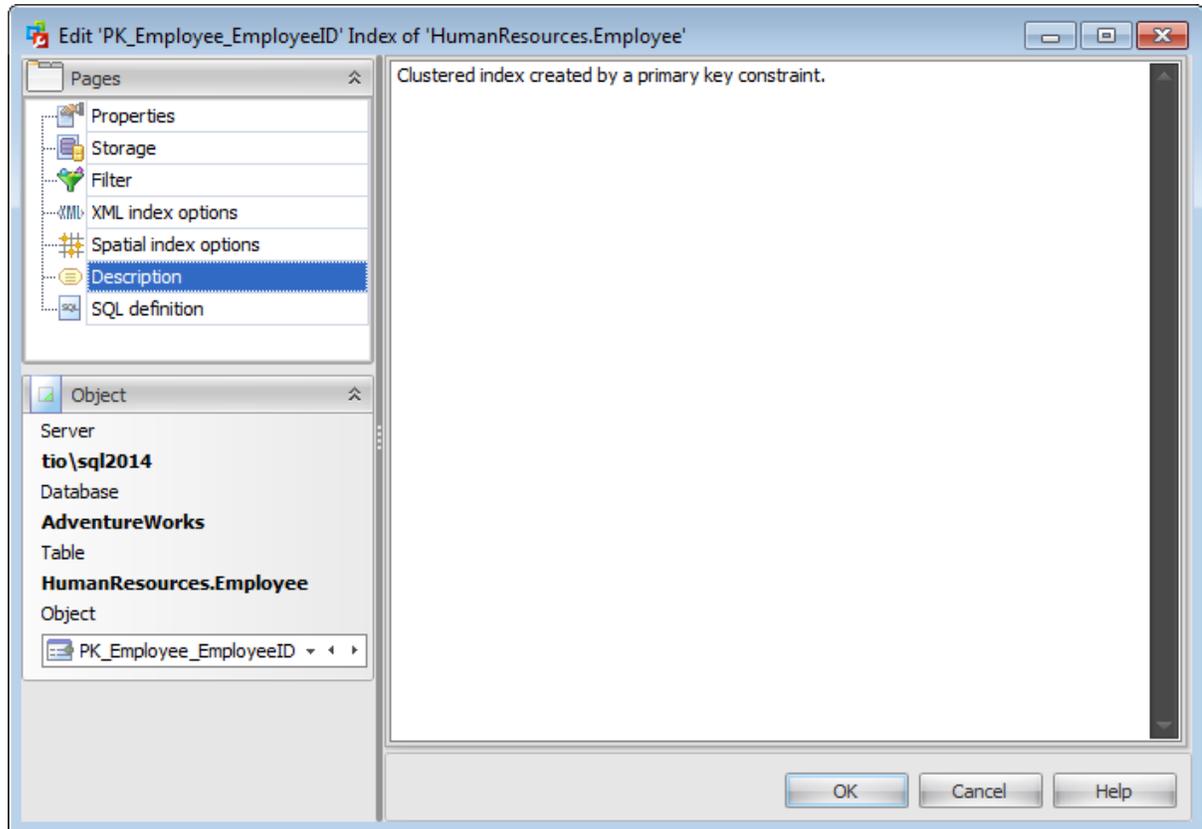
### Spatial options

**Max cells per object (except level 1)**

Set the number of tessellation cells per object that can be used for a single spatial object in the index by the tessellation process. If the  **Default** option is checked then the default value is used for the specified tessellation.

#### 5.1.4.6 Description

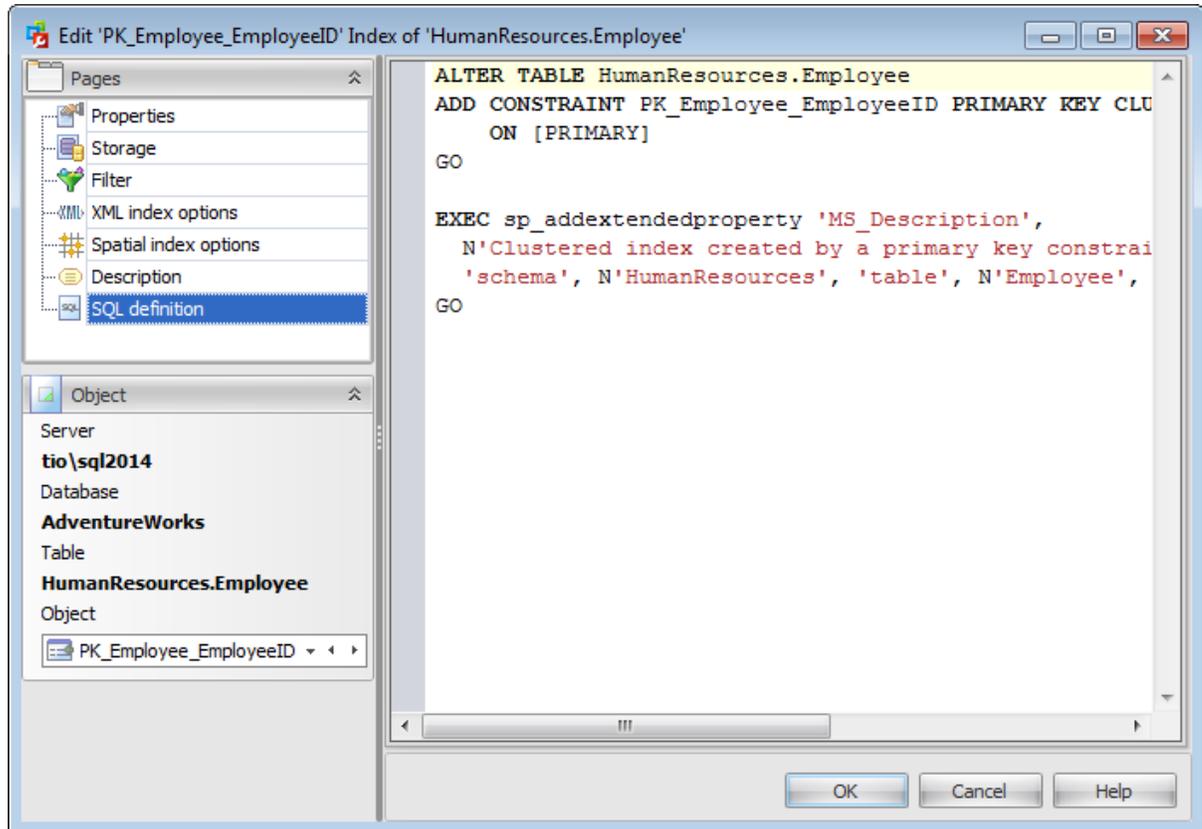
The **Description** tab allows you to view and edit the comment for the index (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

#### 5.1.4.7 SQL Definition

The **SQL definition** tab of the **Index Editor** window allows you to view index DDL.



### 5.1.5 Statistics Editor

**Statistics** for query optimization are objects that contain statistical information about the distribution of values in one or more columns of a table or indexed view. The query optimizer uses these statistics to estimate the cardinality, or number of rows, in the query result. These cardinality estimates enable the query optimizer to create a high-quality query plan. For example, the query optimizer could use cardinality estimates to choose the index seek operator instead of the more resource-intensive index scan operator, and in doing so improve query performance.

To open statistics in **Statistics Editor**, double-click it in the [Statistics list](#), or use the **Edit Statistics...** item of the context menu within the [Statistics](#) tab of [Table Editor](#).

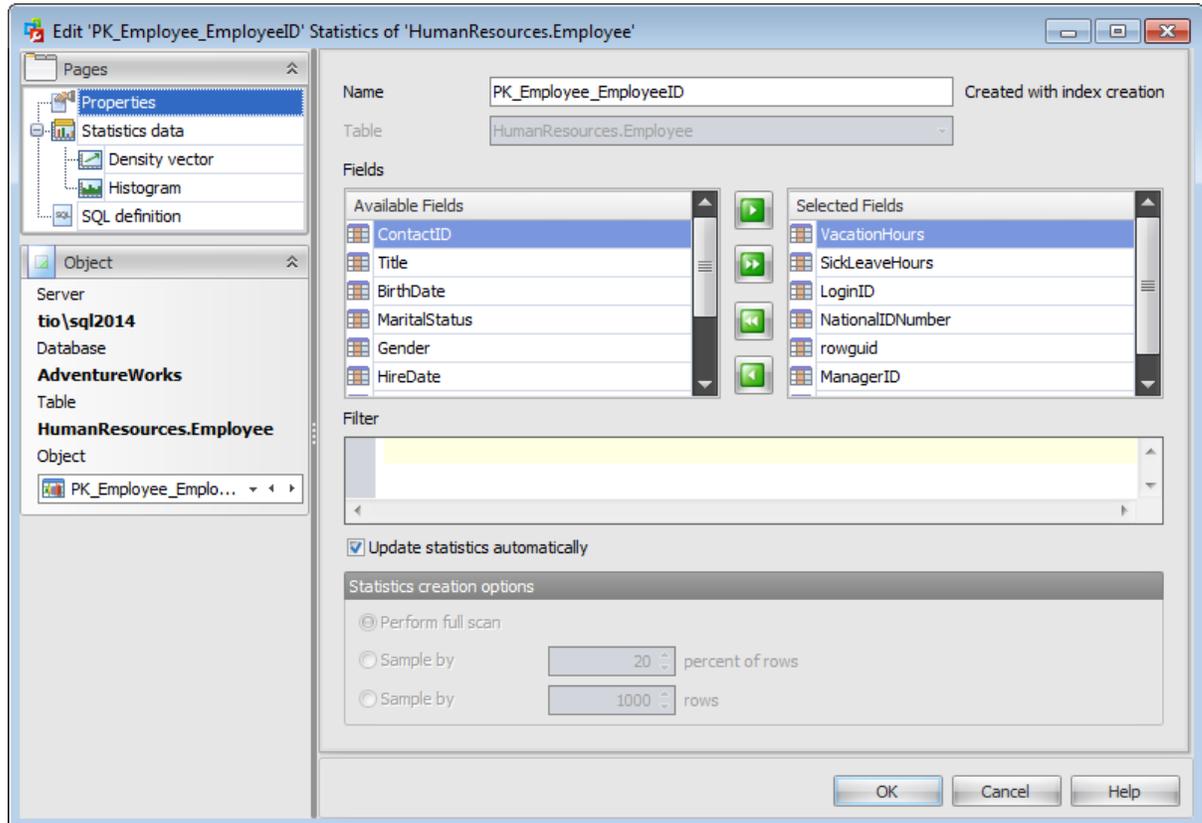
- [Properties](#)
- [Statistics data](#)
- [Density vector](#)
- [Histogram](#)
- [SQL Definition](#)

### 5.1.5.1 Properties

Use the **Properties** tab of the **Statistics Editor** to create/edit statistics on a specified [table](#) and specify its properties.

#### Name

Enter a name for the new statistics, or modify the name of the statistics being edited.



#### Table

In this field the Table name of the statistics being edited is displayed.

#### Fields

To include key field(s) in the statistics, you need to move it from the **Available** list to the **Selected** list. Use the     buttons or drag-and-drop operations to move the fields from one list to another.

#### Filter

Specifies an expression for selecting a subset of rows to include when creating the statistics object. Statistics that are created with a filter predicate are called filtered statistics. The filter predicate uses simple comparison logic and cannot reference a computed column, a UDT column, a spatial data type column, or a hierarchyID data type column. Comparisons using NULL literals are not allowed with the comparison operators. Use the IS NULL and IS NOT NULL operators instead.

#### Update statistics automatically

When this option is on, the query optimizer determines when statistics might be out-of-date and then updates them when they are used by a query. Statistics become out-of-date after insert, update, delete, or merge operations change the data distribution in the table or indexed view.

### Statistics creation options

**Perform full scan**

Compute statistics by scanning all rows in the table or indexed view.

**Sample by ... percent of rows**

Specify the approximate percentage of rows in the table view for the query optimizer to use when it creates statistics.

**Sample by ... rows**

Specify the approximate number of rows in the table for the query optimizer to use when it creates statistics.

### 5.1.5.2 Statistics data

On the **Statistics data** tab you can view the `DBCC SHOW_STATISTICS` command result - current query optimization statistics for a table or indexed view.

`DBCC SHOW_STATISTICS` displays the header, [histogram](#), and [density vector](#) based on data stored in the statistics object.

The screenshot shows the 'Statistics data header' window for the index 'AK\_Employee\_NationalIDNumber' on the table 'HumanResources.Employee'. The window displays a table with the following data:

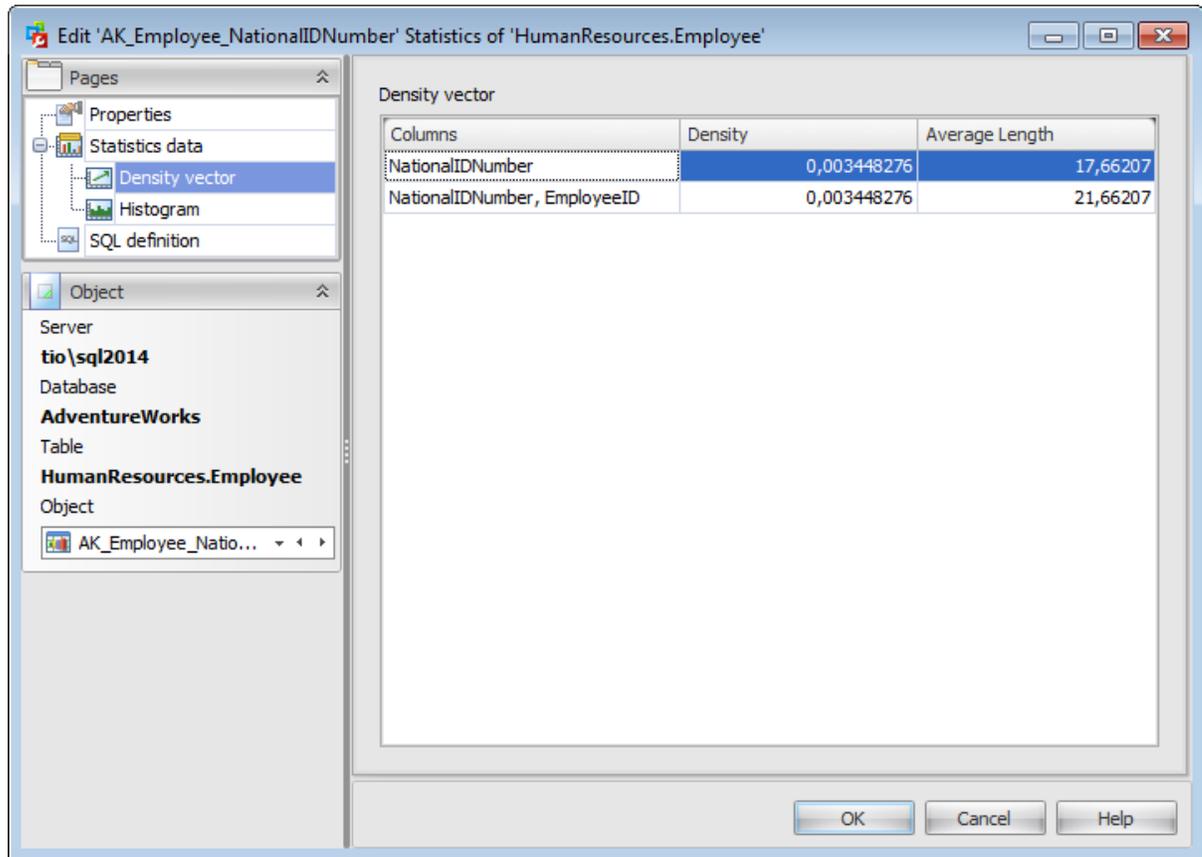
Property Name	Value
Last Update Date	Feb 1 2015 11:31AM
Number of Rows	290
Rows Sampled	290
Number of Histogram Steps	177
Density	1
Average Key Length	21,66207
Contains String Summary	YES
Filter Expression	
Unfiltered Rows	290

The window also shows a tree view on the left with the following structure:

- Pages
  - Properties
  - Statistics data (selected)
  - Density vector
  - Histogram
  - SQL definition
- Object
  - Server
    - tio \sql2014
      - Database
        - AdventureWorks
          - Table
            - HumanResources.Employee
              - Object
                - AK\_Employee\_Natio...

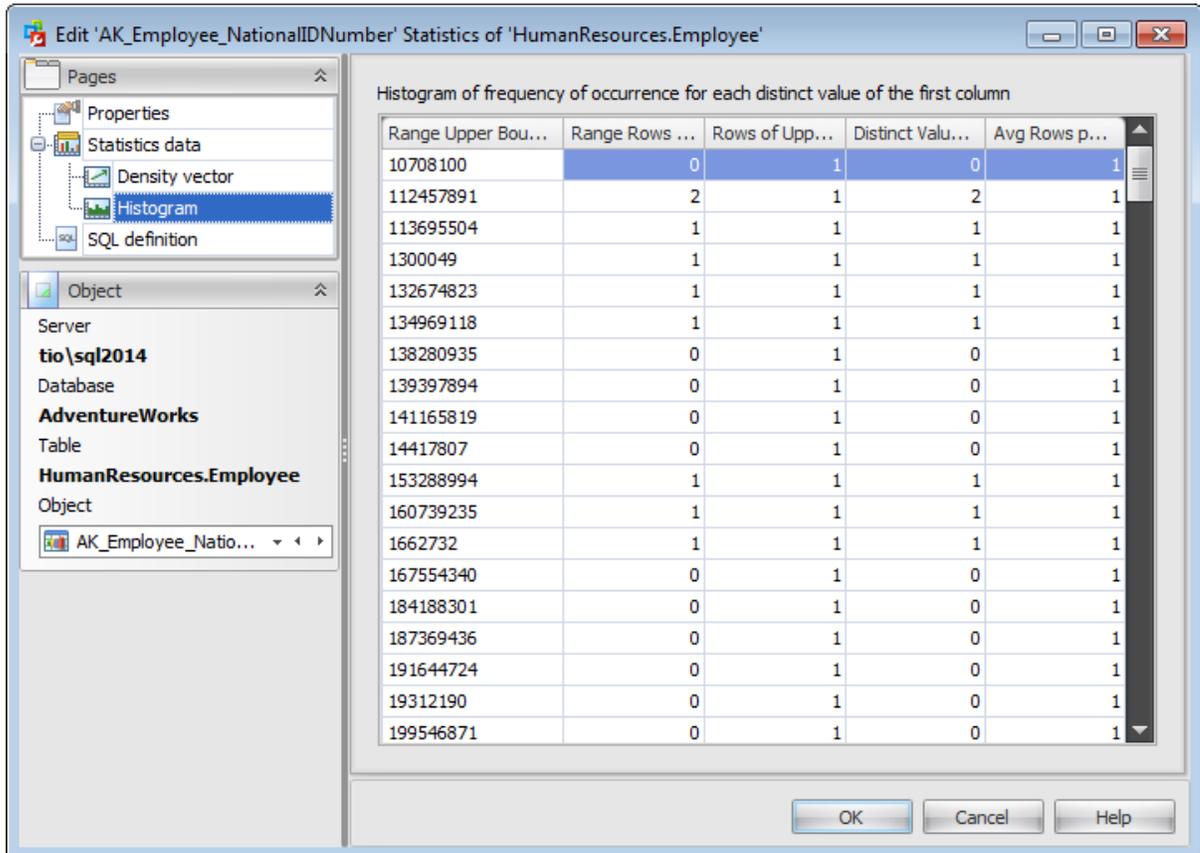
## 5.1.5.2.1 Density vector

The query optimizer uses densities to enhance cardinality estimates for queries that return multiple columns from the same table or indexed view. The density vector contains one density for each prefix of columns in the statistics object.



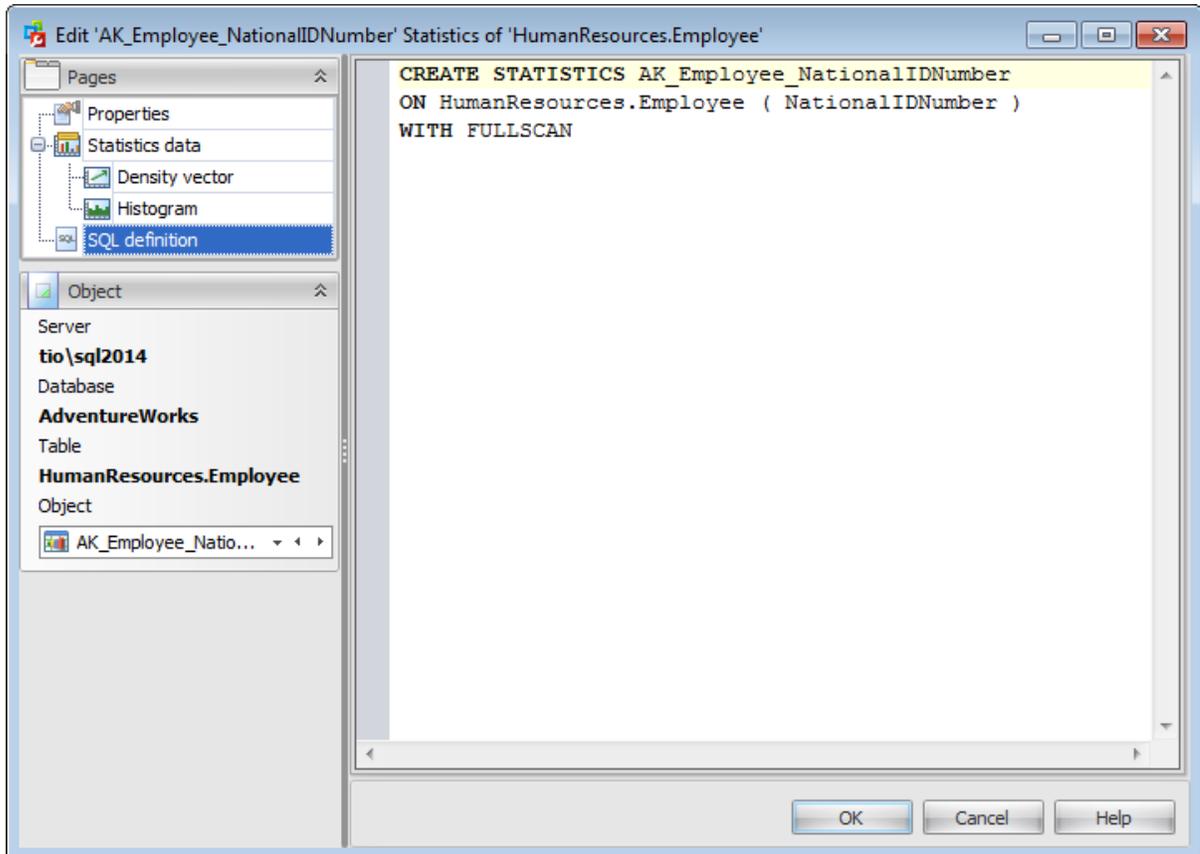
## 5.1.5.2.2 Histogram

A histogram measures the frequency of occurrence for each distinct value in a data set. The query optimizer computes a histogram on the column values in the first key column of the statistics object, selecting the column values by statistically sampling the rows or by performing a full scan of all rows in the table or view.



### 5.1.5.3 SQL Definition

The **SQL definition** tab of the **Index Editor** window allows you to view statistics DDL.



### 5.1.6 Foreign Key Editor

**Foreign Key Editor** allows you to specify foreign key definition and set foreign key properties. It opens when you create a new foreign key or edit an existing one.

To open a foreign key in **Foreign Key Editor**, double-click it in the [Foreign keys list](#), or use the **Edit Foreign Key...** item of the context menu within the [Foreign keys](#) tab of [Table Editor](#).

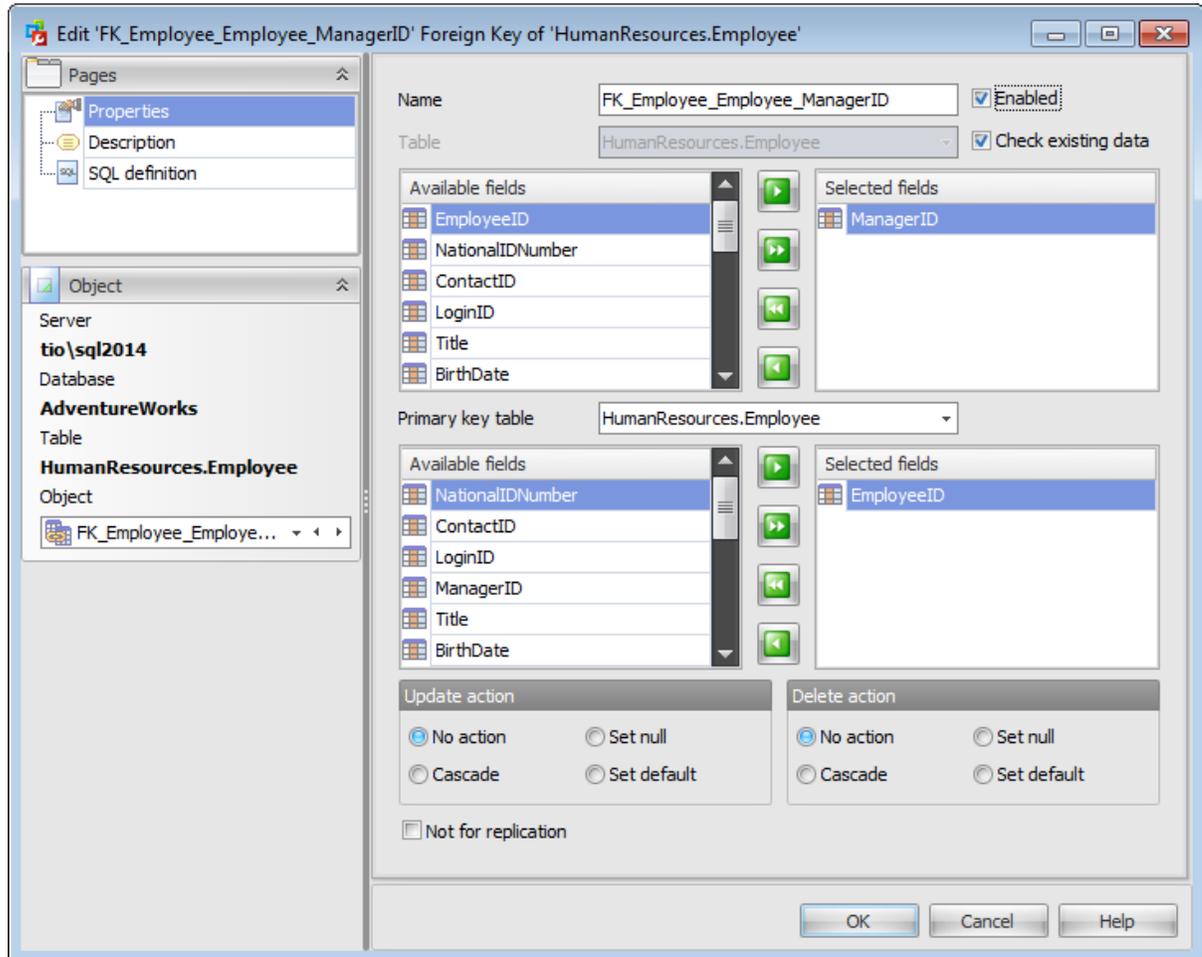
- [Properties](#)
- [Description](#)
- [SQL Definition](#)

### 5.1.6.1 Properties

Use the **Properties** tab of the **Foreign Key Editor** to create/edit a foreign key constraint and specify its properties.

#### Name

Enter a name for the new foreign key, or modify the name of the foreign key being edited.



#### Table

In this field the Table name of the foreign key being edited is displayed.

#### Enabled

Enables/disables the foreign key immediately after it is created.

#### Check existing data

This option specifies whether the data in the table is or is not validated against a newly added or re-enabled foreign key.

#### Update action / Delete action

- *No action*

Produce an error indicating that the deletion or update would create a foreign key

constraint violation. If the constraint is deferred, this error will be produced at constraint check time if there still exist any referencing rows. This is the default action.

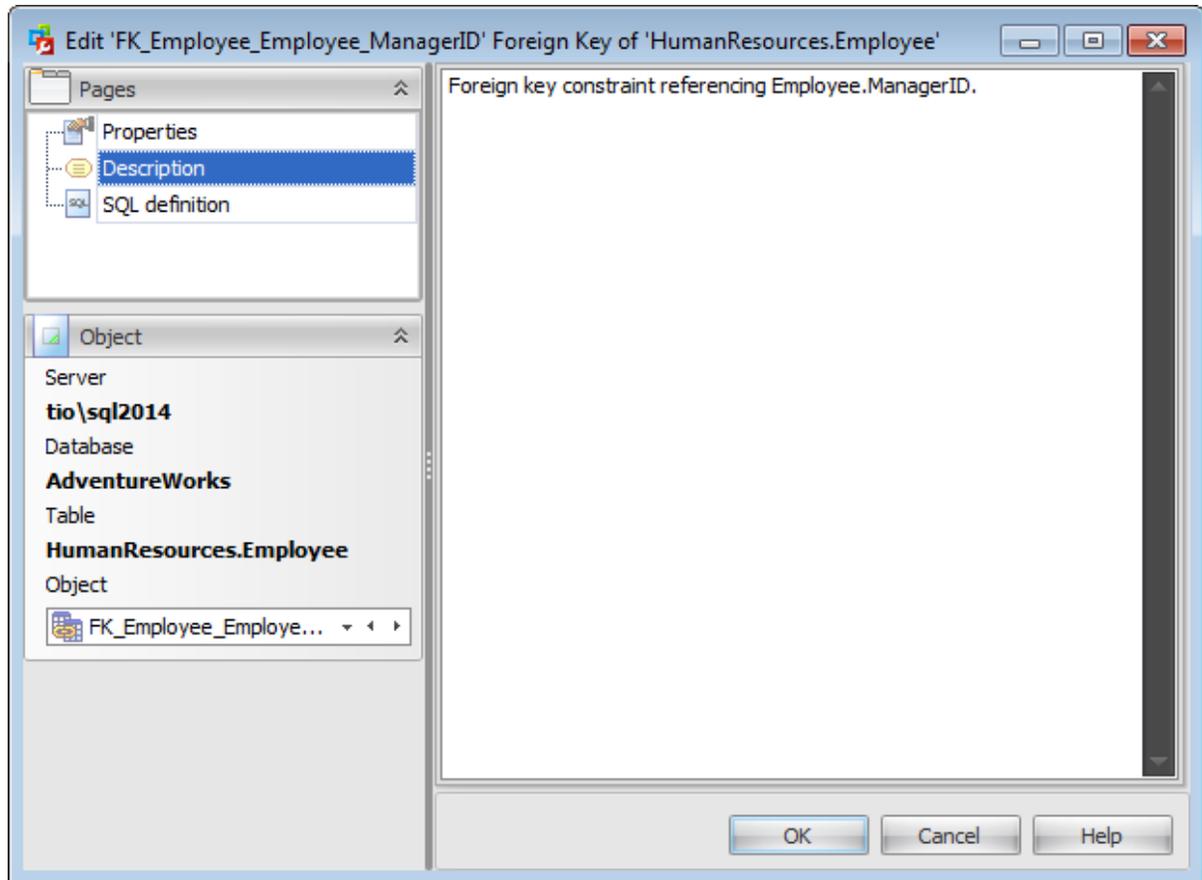
- *Cascade*  
Delete any rows referencing the deleted row, or update the value of the referencing column to the new value of the referenced column, respectively.
- *Set null*  
Set the referencing column(s) to null.
- *Set default*  
Set the referencing column(s) to their default values.

**Not for replication**

This option allows you to specify that the foreign key should not be enabled when a replication agent modifies the table that is involved in the foreign key.

### 5.1.6.2 Description

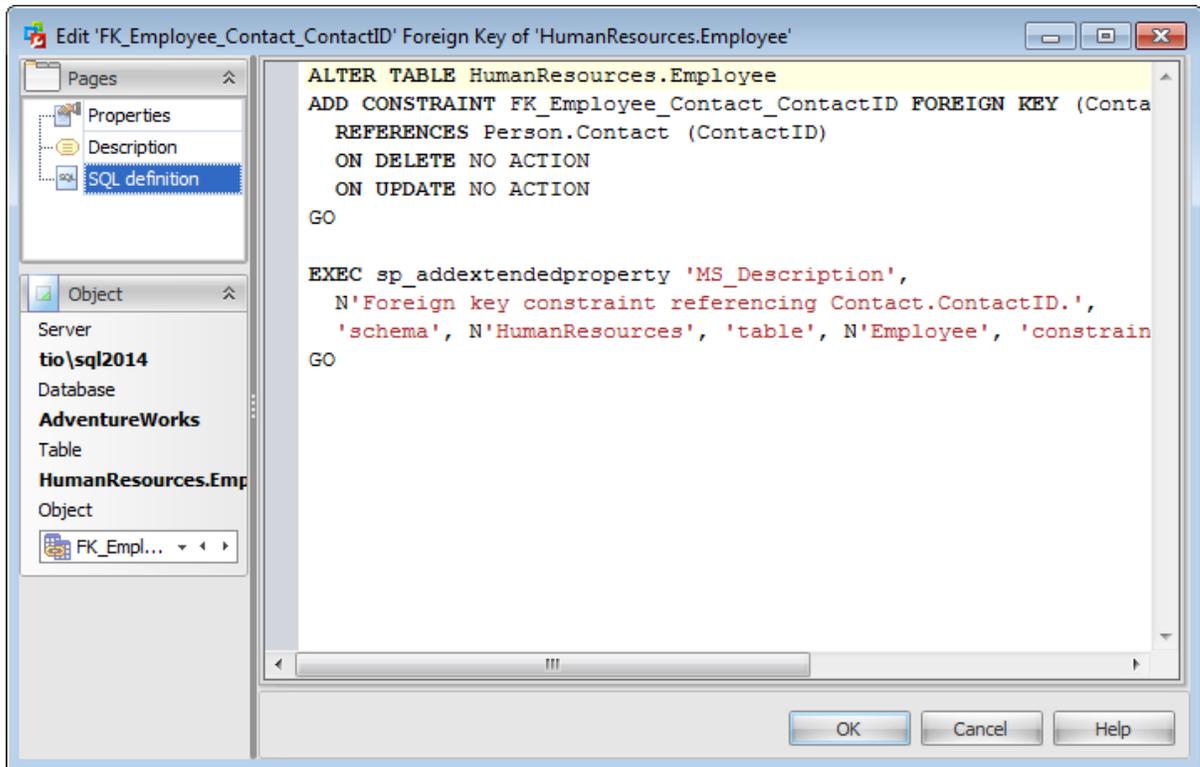
The **Description** tab allows you to view and edit the comment for the foreign key (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

### 5.1.6.3 SQL Definition

The **SQL definition** tab of the **Foreign key Editor** window allows you to view foreign key DDL.



### 5.1.7 Trigger Editor

A **Trigger** is a special kind of [stored procedure](#) that automatically executes when an event occurs in the database server.

**DML Triggers** execute when a user tries to modify data through a data manipulation language (DML) event. DML events are *INSERT*, *UPDATE*, or *DELETE* statements on a [table](#) or [view](#).

**Trigger Editor** allows you to specify trigger definition and set trigger properties. It opens automatically when you create a new trigger and is available on editing an existing one.

To open a trigger in **Trigger Editor**, double-click it in the [Triggers list](#), or use the **Edit Trigger...** item of the context menu within the [Triggers](#) tab of [Table Editor](#).

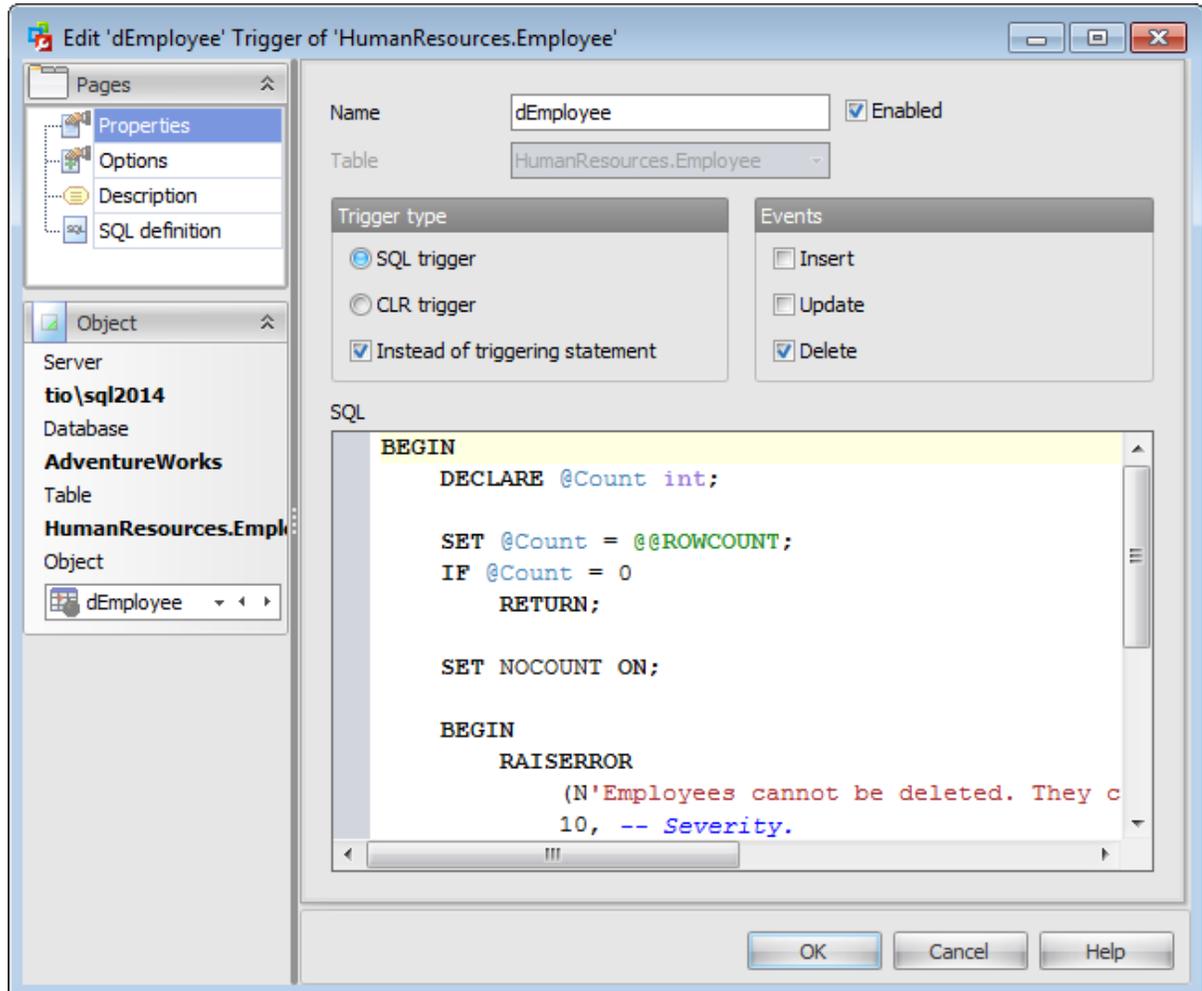
- [Properties](#)
- [Options](#)
- [Description](#)
- [SQL Definition](#)

### 5.1.7.1 Properties

Use the **Properties** tab of the **Trigger Editor** to create/edit a DML trigger and specify its properties.

#### Name

Enter a name for the new trigger, or modify the name of the trigger being edited.



#### Table

In this field the Table name of the trigger being edited is displayed.

#### Enabled

Enables/disables the trigger immediately after it is created. Enabling a trigger causes it to fire when any Transact-SQL statements on which it was originally programmed are executed. A disabled trigger still exists as an object in the current database, but does not fire.

#### Trigger type

Select whether the trigger should be created directly from Transact-SQL statements ( *SQL trigger*) or from methods of assemblies that are created in the Microsoft .NET ( *CLR*)

*trigger*).

**Instead of triggering statement**

This selection specifies that the trigger is executed instead of the triggering SQL statement. Otherwise, it is fired only when all operations specified in the triggering SQL statement have executed successfully.

### Events

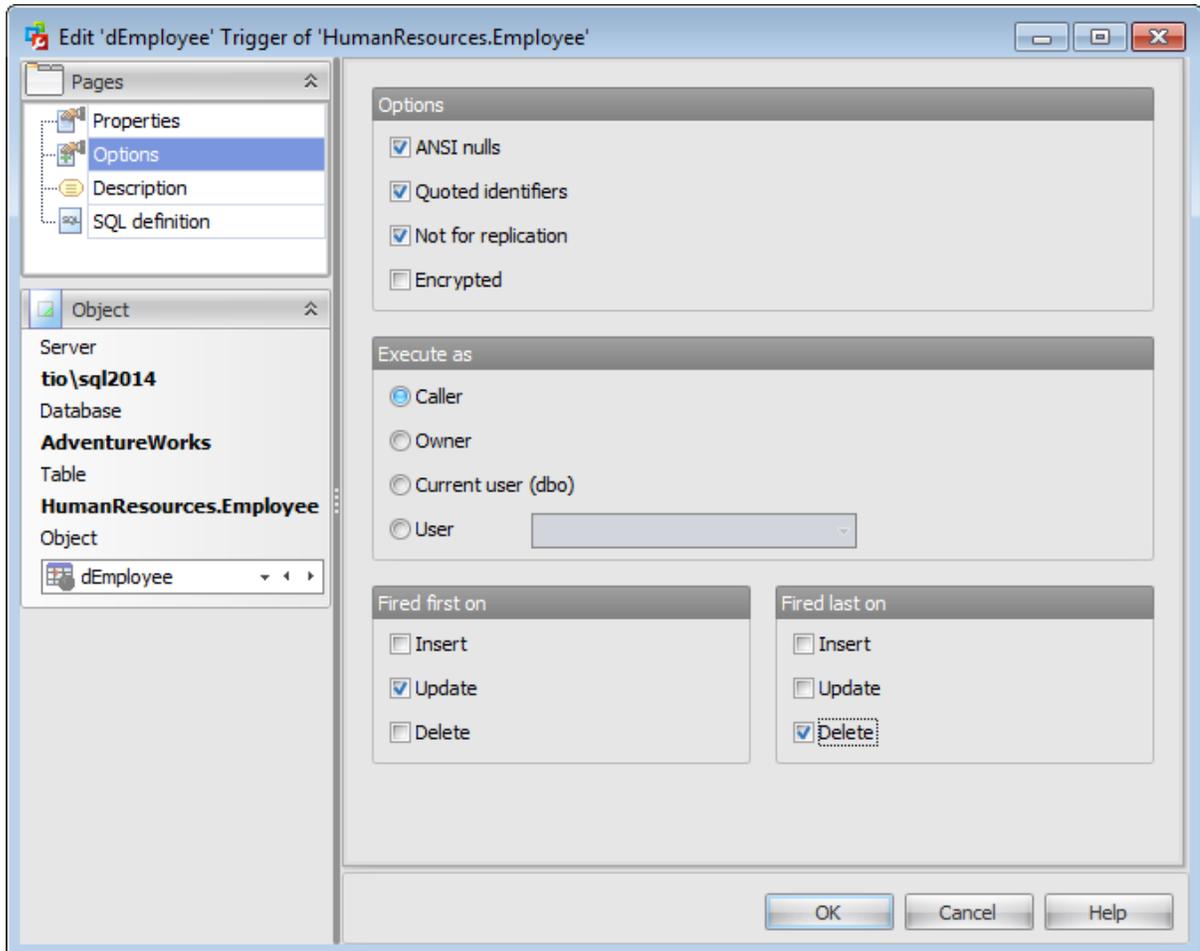
Specify the data modification statements that activate the trigger when it is tried against this table or view: *Insert*, *Update* or *Delete*. At least one option must be specified.

For *SQL trigger* Specify the trigger conditions and actions. Trigger conditions specify additional criteria that determine whether the tried DML, DDL, or logon events cause the trigger actions to be performed.

For a *CLR trigger*, specify the **Method** of an **Assembly** to bind with the trigger. The method must take no arguments and return void. **Class** name must be a valid SQL Server identifier and must exist as a class in the assembly with assembly visibility. If the class has a namespace-qualified name that uses '.' to separate namespace parts, the class name must be delimited by using [ ] or " " delimiters. The class cannot be a nested class.

### 5.1.7.2 Options

Use the **Options** tab to specify additional options of the trigger being created.



#### Options

##### ANSI Nulls

This option specifies ISO compliant behavior of the Equals (=) and Not Equal To (<>) comparison operators when they are used with null values. When this option is switched on, a *SELECT* statement that uses *WHERE column\_name = NULL* returns zero rows even if there are null values in *column\_name*. A *SELECT* statement that uses *WHERE column\_name <> NULL* returns zero rows even if there are nonnull values in *column\_name*.

##### Quoted identifier

This option causes SQL Server to follow the ISO rules regarding quotation mark delimiting identifiers and literal strings. When this option is switched on, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When it is switched off, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers.

**Not for replication**

This option allows you to specify that the trigger should not be executed when a replication agent modifies the table that is involved in the trigger.

 **Encrypted**

You can check this option to encrypt the text of the *CREATE TRIGGER* statement if you want to ensure that other [users](#) cannot view the trigger definition.

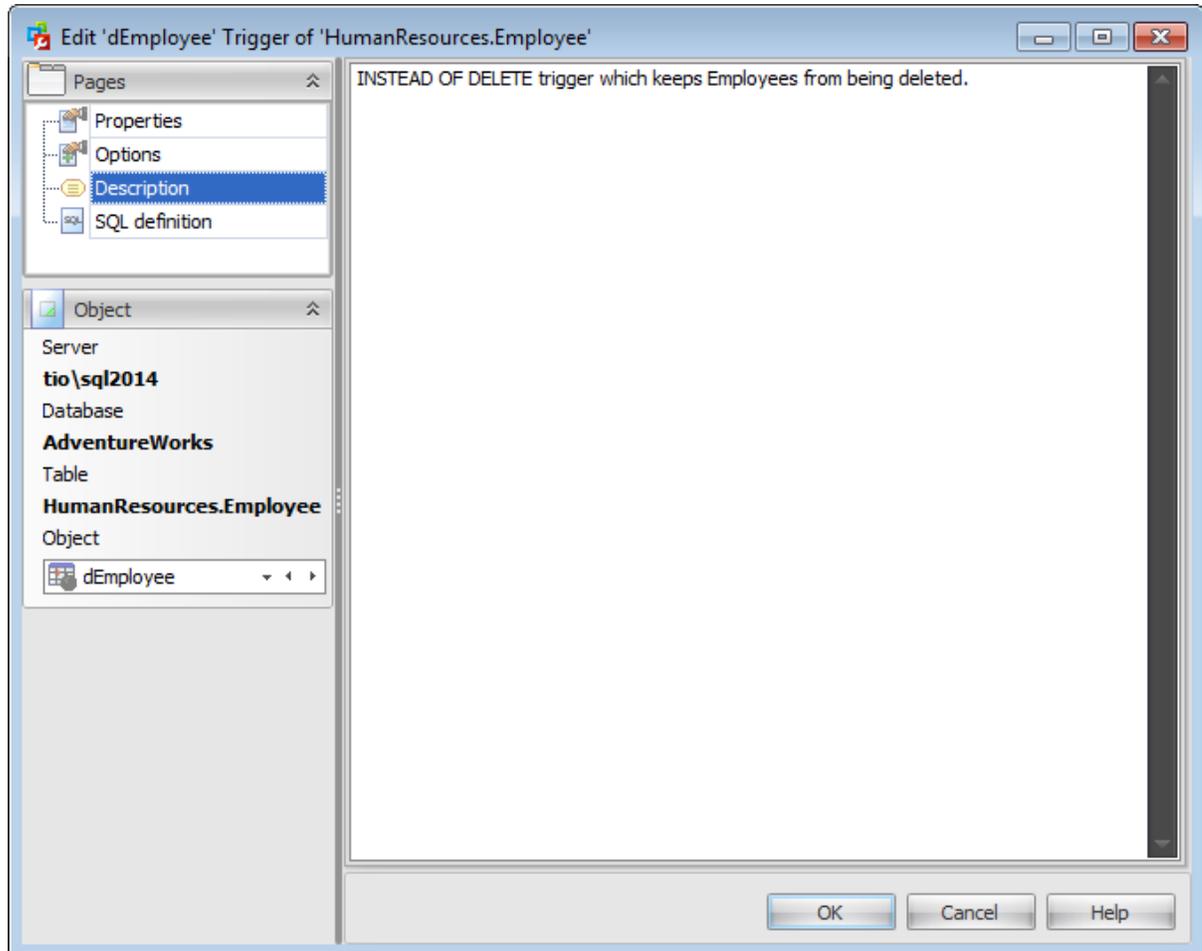
**Execute as**

Specify the security context under which the trigger is to be executed:  *Caller*,  *Owner*,  *Current user* or  *User* (select which [user](#) account should be used to validate permissions on the objects that are referenced by the trigger).

Use the **Fired first on** and **Fired last on** groups of options to specify the *AFTER* triggers that are fired first or last. The *AFTER* triggers that are fired between the first and last triggers are executed in undefined order.

### 5.1.7.3 Description

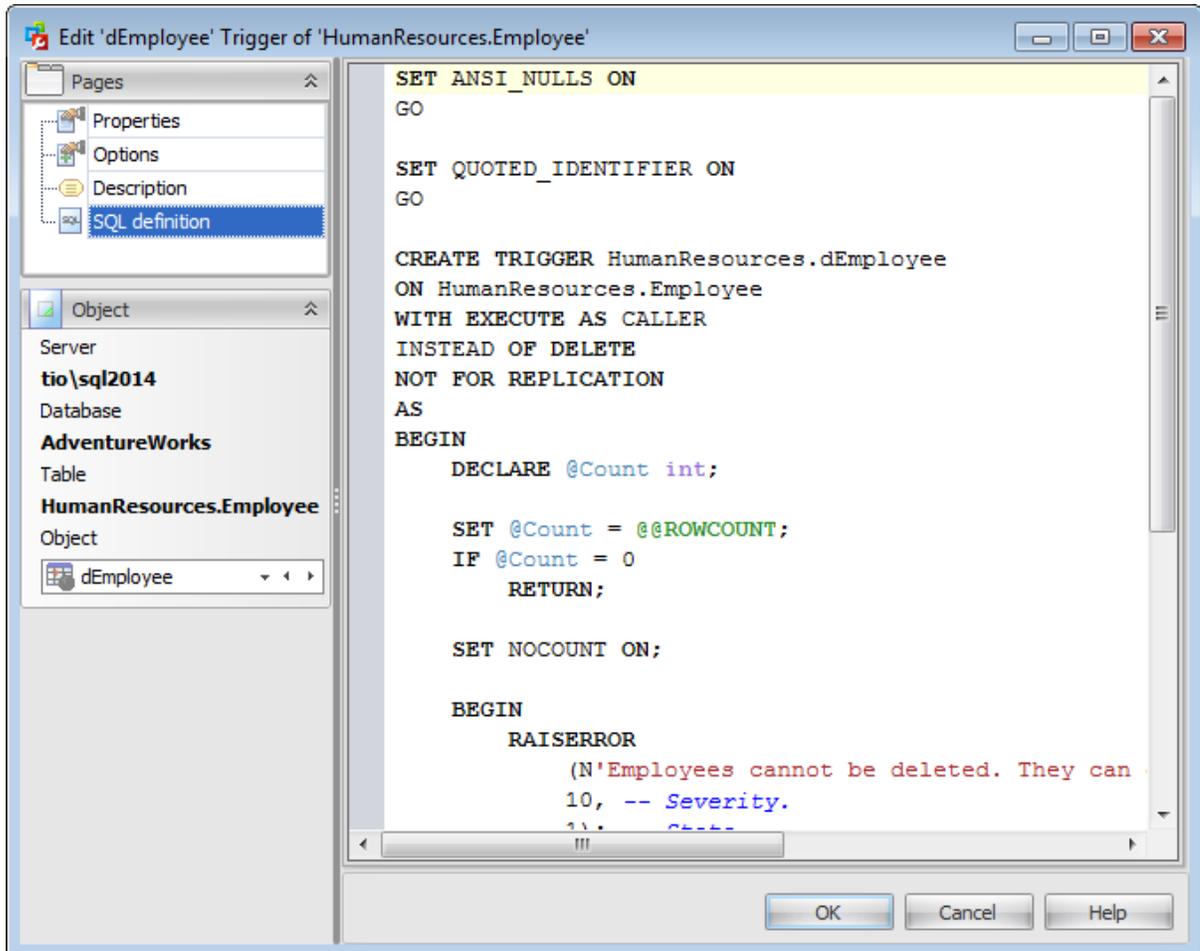
The **Description** tab allows you to view and edit the comment for the trigger (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

#### 5.1.7.4 SQL Definition

The **SQL definition** tab of the **Trigger Editor** window allows you to view foreign key DDL.

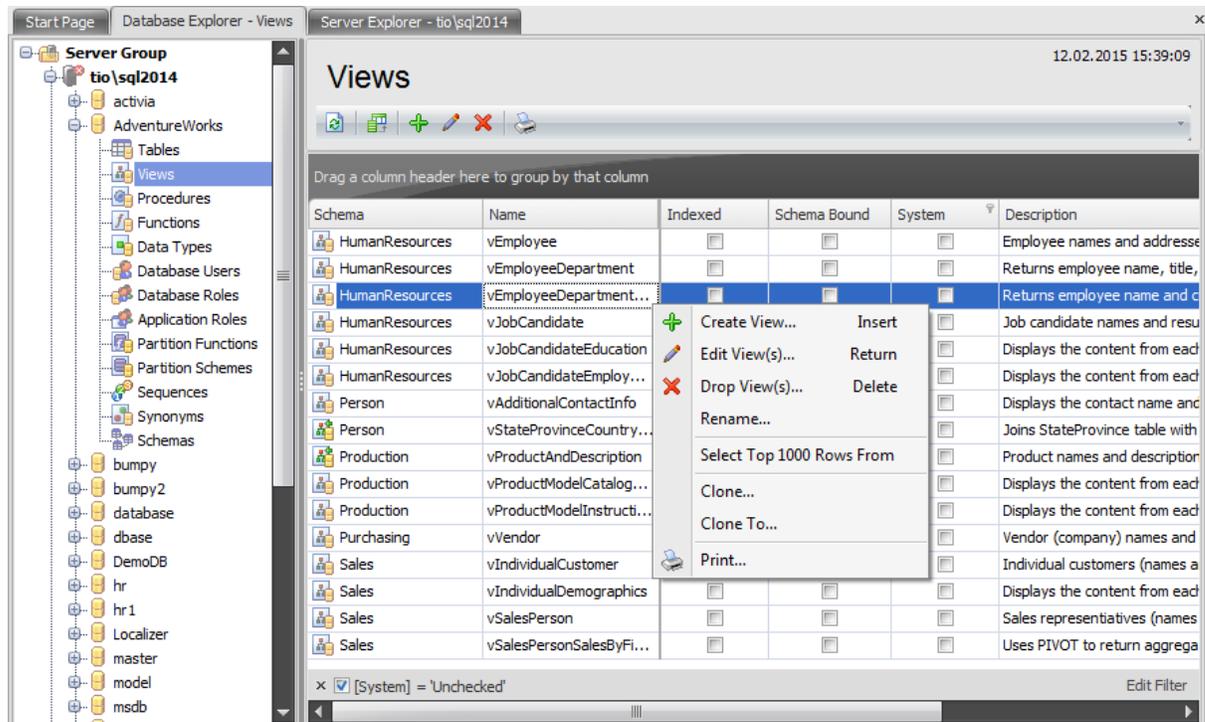


## 5.2 Views

The **Views** section of the Database Explorer is used for managing views in the selected database.

Use the context menu to:

-  [create](#) a new view
-  [edit](#) the selected view(s)
-  drop selected view(s)
- rename the selected view
- select top 1000 rows from the selected view
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database views as a grid with the following columns: *Schema*, *Name*, *Indexed*, *Schema Bound*, *System*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a View in the [View Editor](#), double-click it in the grid. Alternatively, you can right-click the View alias and select the **Edit View(s)** context menu item.

## 5.2.1 View Editor

A **View** is a logical table based on one or more [tables](#) or [views](#). A view contains no data itself. The tables upon which a view is based are called *base tables*.

**Views** are useful for allowing users to access a set of relations ([tables](#)) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

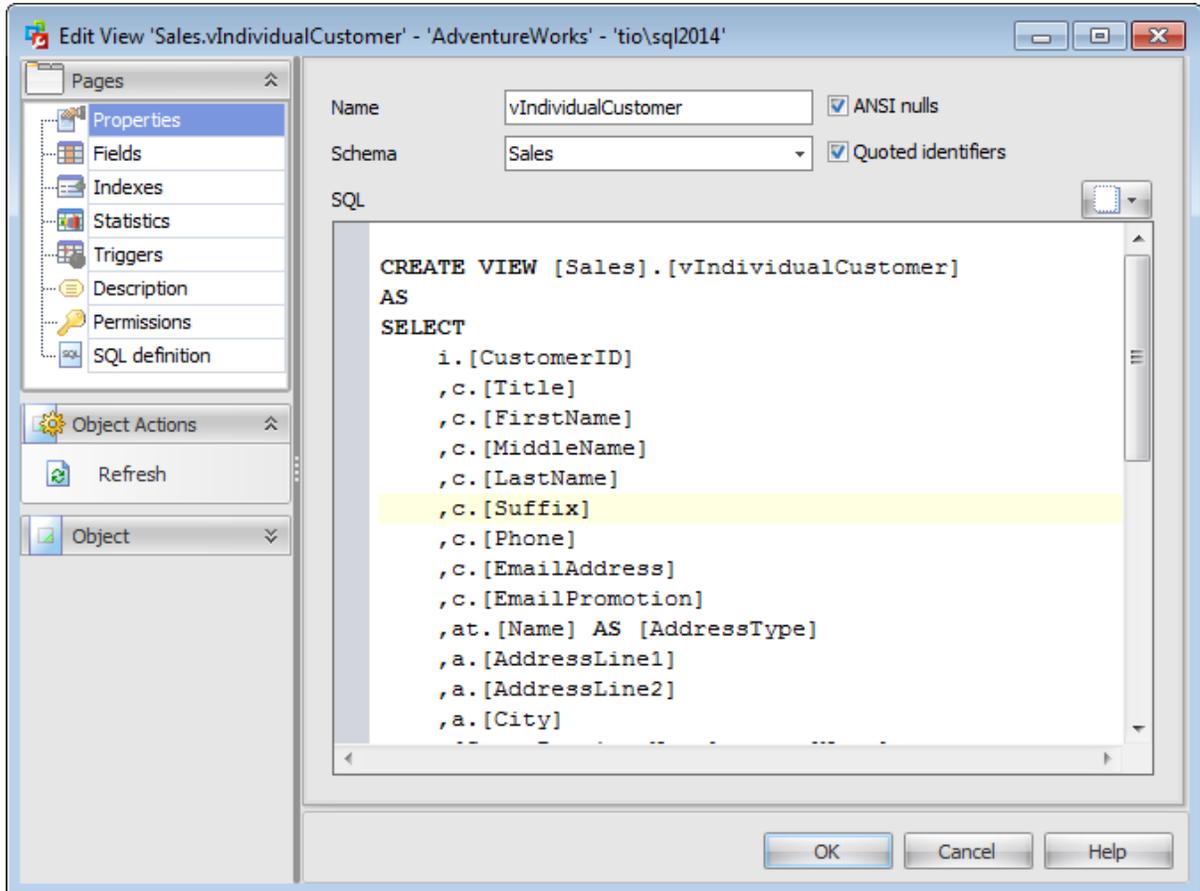
**View Editor** allows you to create new views and define their properties (view name and the SELECT statement it implements). It opens automatically when you create a new view and is available on editing an existing one.

To open a view in **View Editor**, double-click it in the [Views](#) list.

- [Properties](#)
- [Fields](#)
- [Indexes](#)
- [Statistics](#)
- [Triggers](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.2.1.1 Properties

Use the **Properties** tab of the **View Editor** to create/edit a view and specify its definition.



#### Name

Enter a name for the new view. Note that view names must comply with the rules for identifiers and must be unique within the [schema](#). A view name can contain a maximum of 128 characters.

#### Schema

Use the drop-down list to select the [schema](#) which the view belongs to. The default selection is the default schema for the current [user](#).

#### ANSI Nulls

This option specifies ISO compliant behavior of the Equals (=) and Not Equal To (<>) comparison operators when they are used with null values. When this option is switched on, a *SELECT* statement that uses *WHERE column\_name = NULL* returns zero rows even if there are null values in *column\_name*. A *SELECT* statement that uses *WHERE column\_name <> NULL* returns zero rows even if there are nonnull values in *column\_name*.

#### Quoted identifier

This option causes SQL Server to follow the ISO rules regarding quotation mark delimiting identifiers and literal strings. When this option is switched on, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When it is switched off, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers.

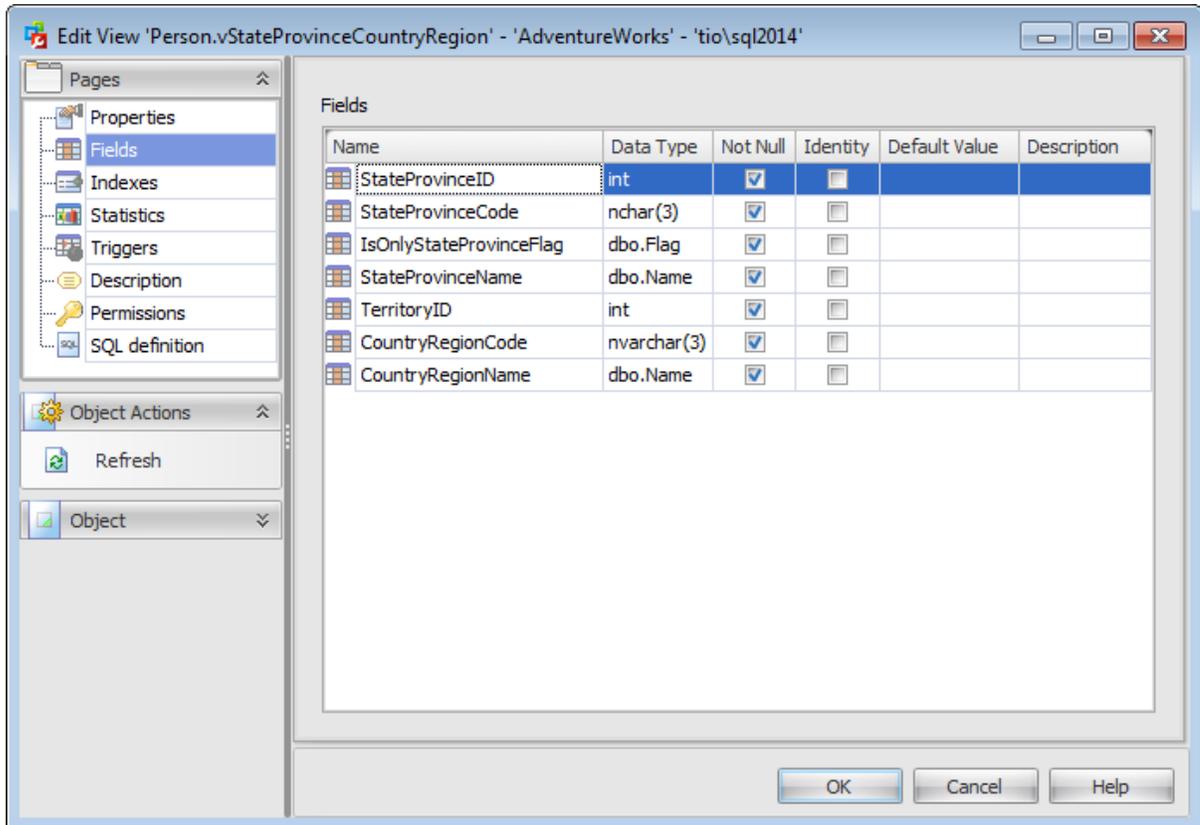
**SQL**

Use this text area to specify view definition as SQL statement.

You can click the  button to replace the text field content with the view definition template.

### 5.2.1.2 Fields

The **Fields** tab is provided for viewing fields represented in the view.



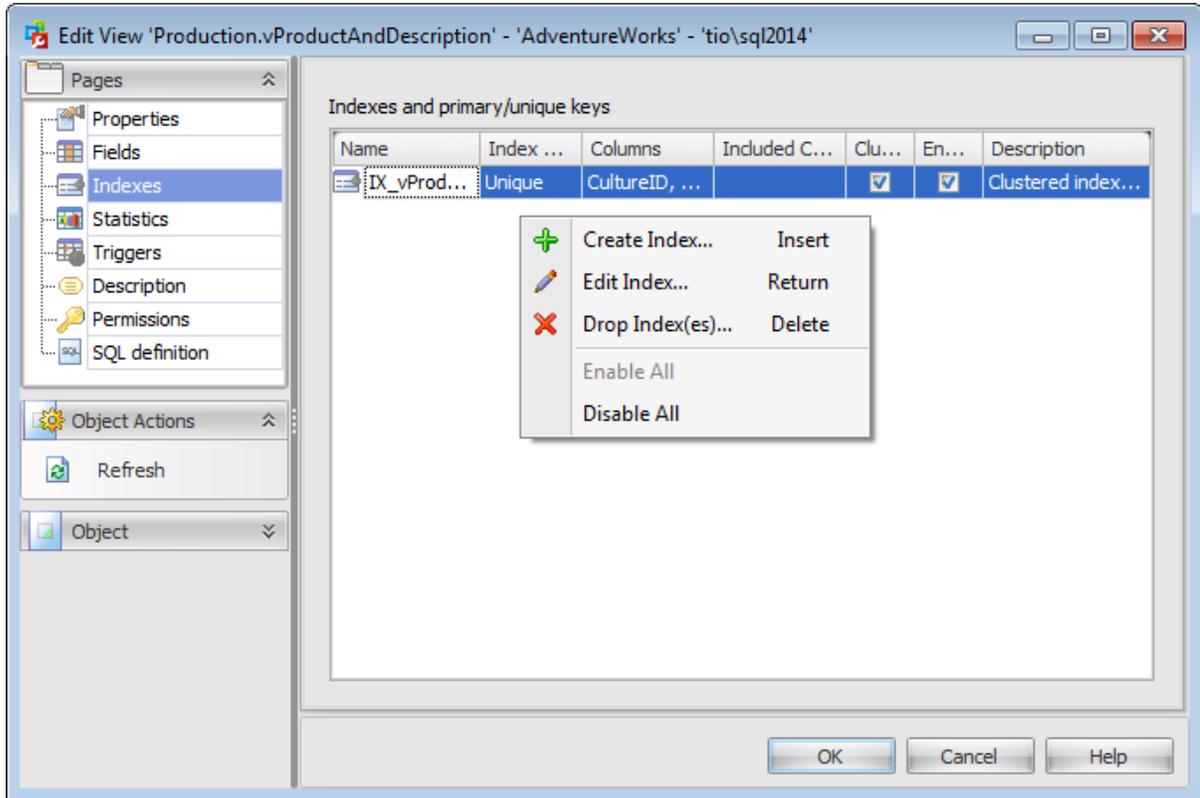
The **Fields** list provides the following attributes of each field of the view:

*Name*  
*Data Type*  
*Not Null*  
*Identity*  
*Default Value*  
*Description*

If necessary, you can also use the **Description** column to supply a *description* for each field.

### 5.2.1.3 Indexes

The **Indexes** tab is provided for viewing indexes represented in the view. Double-click an index to open [Index Editor](#) for editing the index.



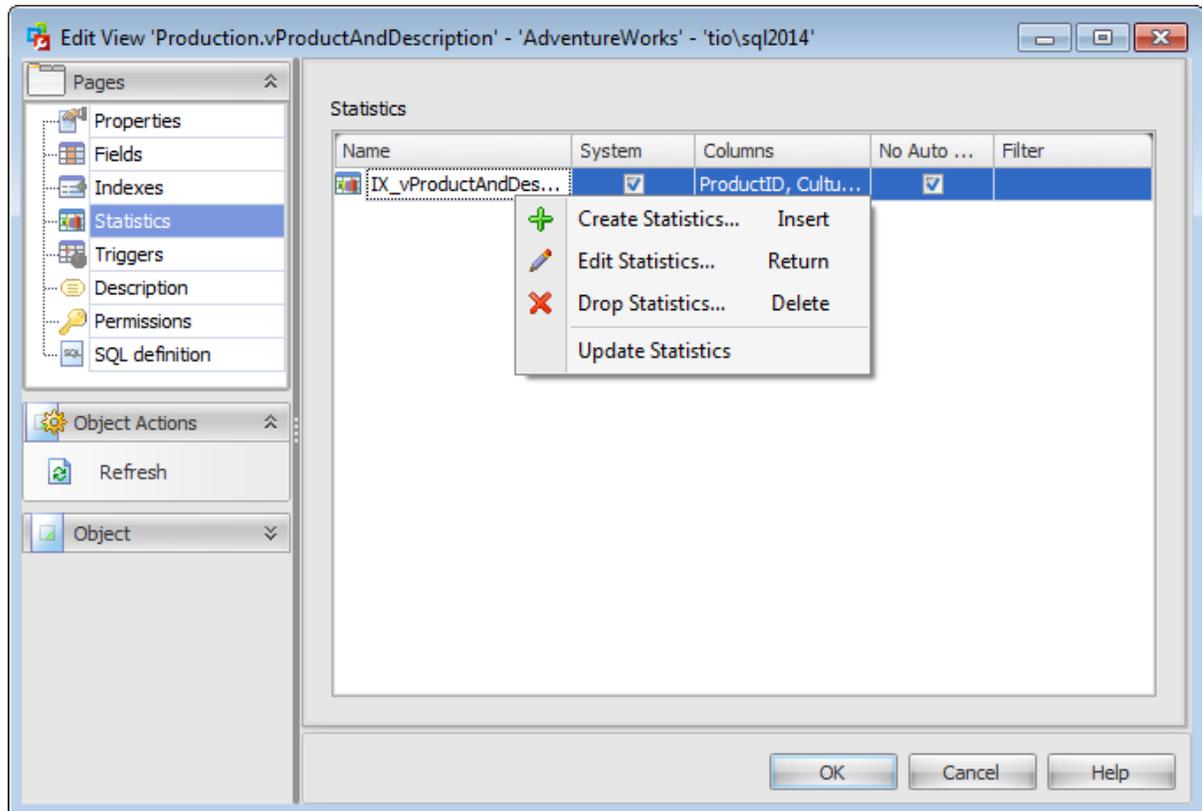
The **Indexes and primary/unique keys** list provides the following attributes of each index of the view:

*Name*  
*Index type*  
*Columns*  
*Included Columns*  
*Clustered*  
*Enabled*  
*Description*

If necessary, you can use the **Description** area to supply a *description* for each index.

### 5.2.1.4 Statistics

The **Statistics** tab is provided for viewing statistics represented in the view. Double-click statistics to open [Statistics Editor](#) for editing the statistics.



The **Statistics** list provides the following attributes of each statistics of the view:

*Name*

*System*

*Columns*

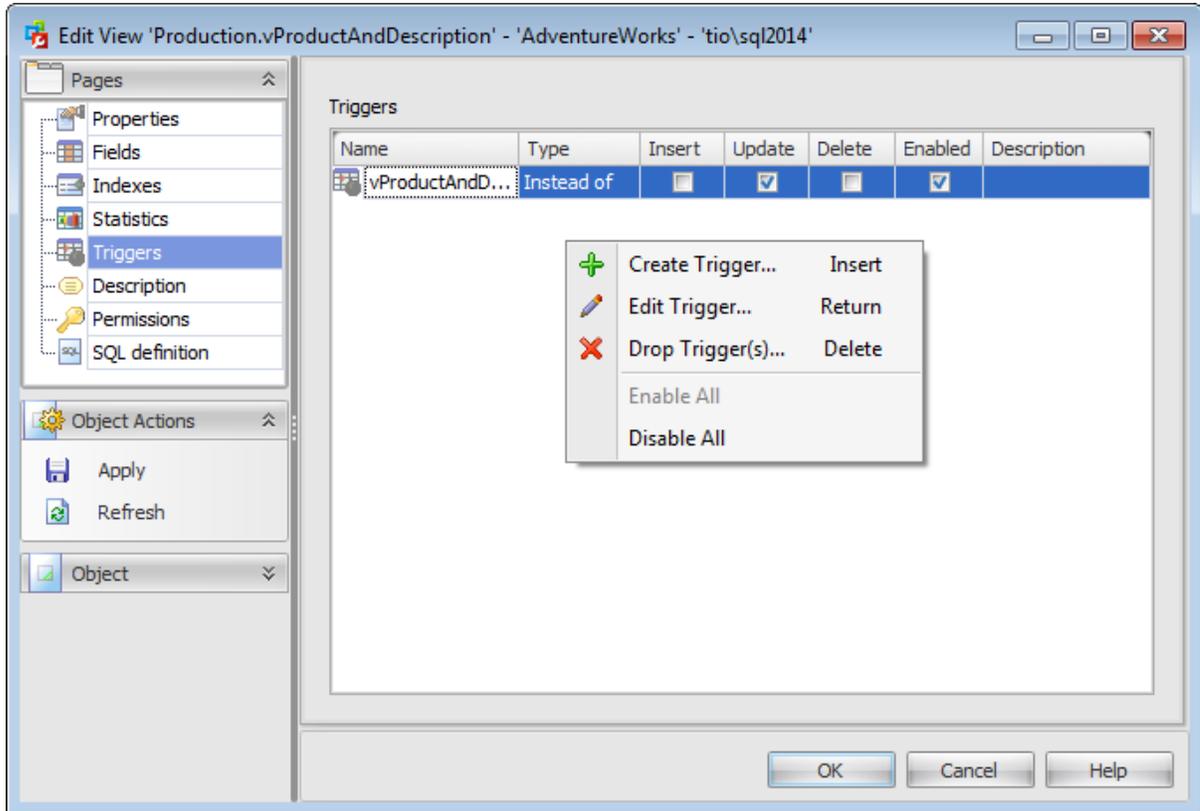
*No Auto Recompute*

*Filter*

Double-click statistics to open [Statistics Editor](#) for editing the statistics.

### 5.2.1.5 Triggers

The **Triggers** tab is provided for managing triggers represented in the view. Double-click a trigger to open [Trigger Editor](#) for editing the trigger.



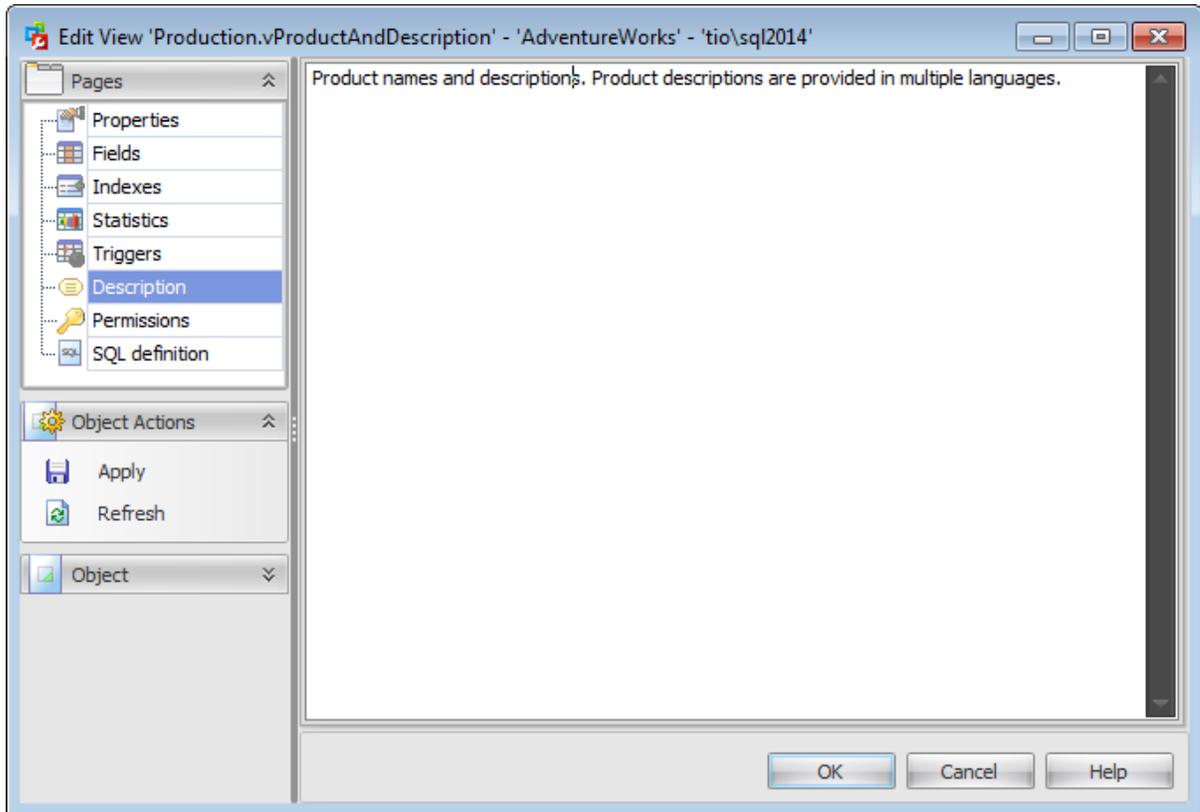
The **Triggers** list provides the following attributes of each trigger of the view:

*Name*  
*Type*  
*Insert*  
*Update*  
*Delete*  
*Enabled*  
*Description*

If necessary, you can also use the **Trigger Description** area to supply a *description* for each trigger.

### 5.2.1.6 Description

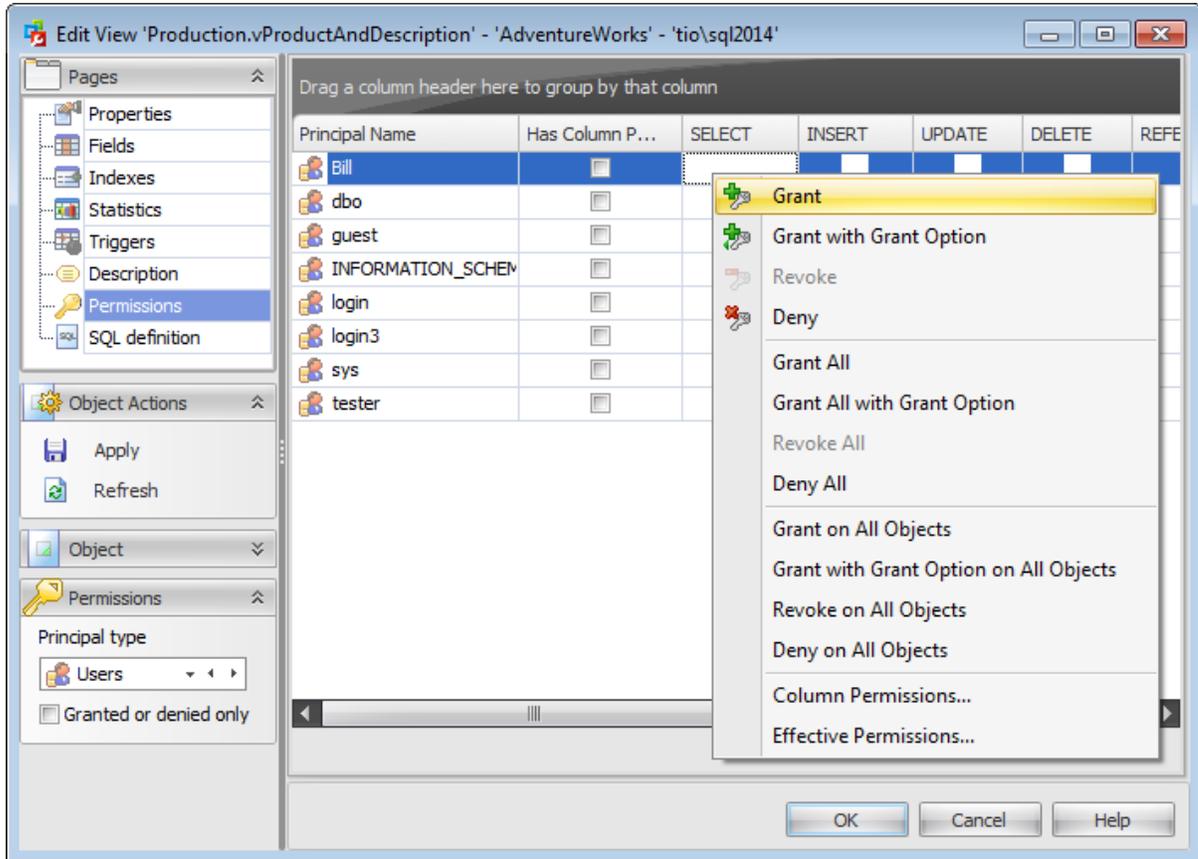
The **Description** tab allows you to view and edit the comment for the view (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

### 5.2.1.7 Permissions

The **Permissions** tab of the **View Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [column permissions](#) and viewing [effective permissions](#).

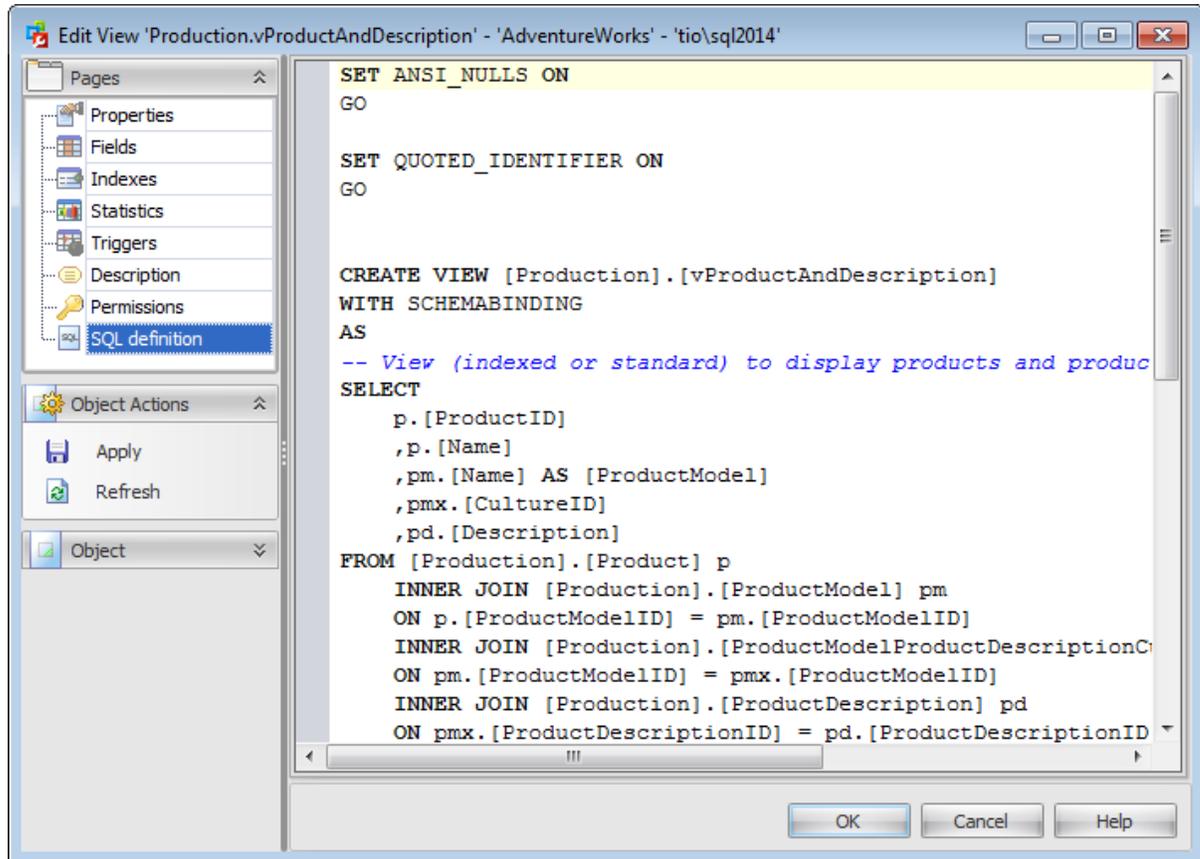
---

**See also:**

[Database Permissions](#)

### 5.2.1.8 SQL Definition

The **SQL definition** tab of the **View Editor** window allows you to view the object DDL.

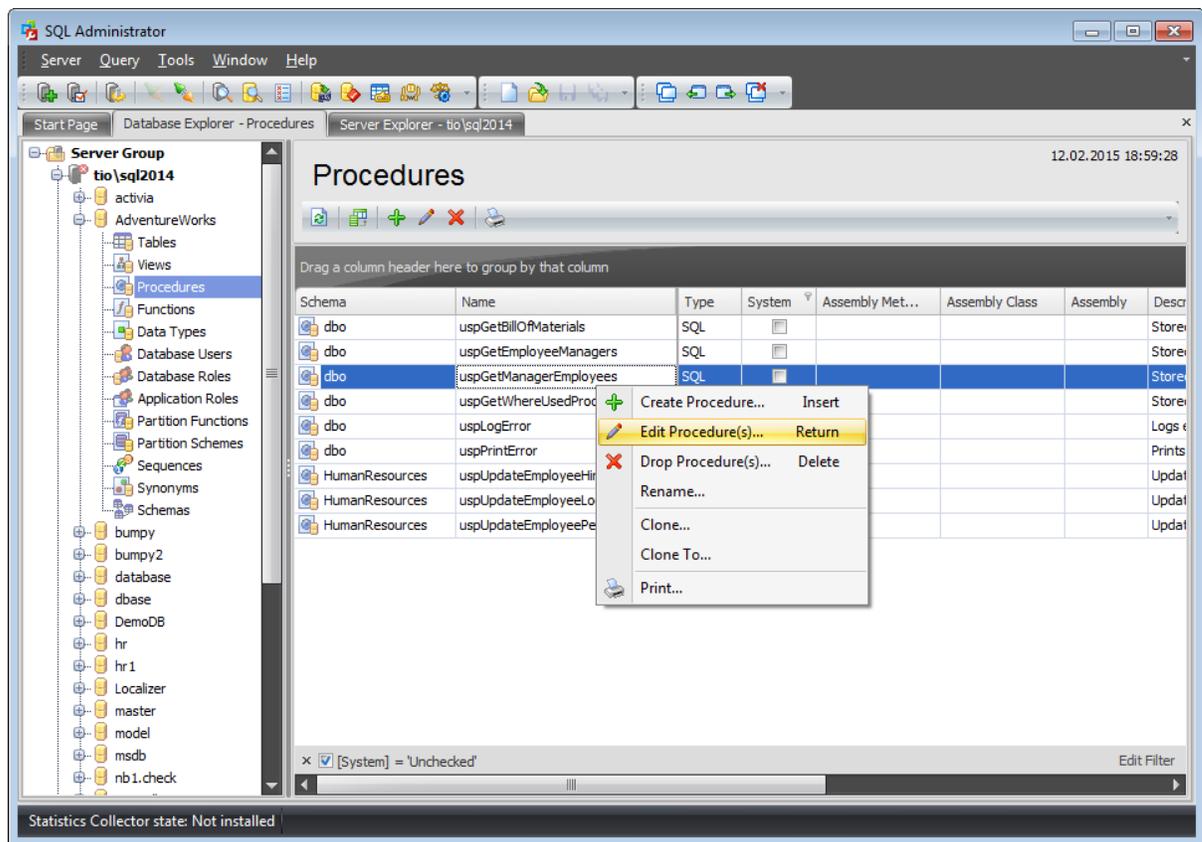


## 5.3 Procedures

The **Procedures** section of the Database Explorer is used for managing procedures in the selected database.

Use the context menu to:

- + [create](#) a new procedure
-  [edit](#) the selected procedure(s)
- drop selected procedure(s)
  - rename the selected procedure
  - create the object copy on the same server
  - clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database procedures as a grid with the following columns: *Schema, Name, Type, System, Assembly Method, Assembly Class, Assembly, Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Procedure in the [Procedure Editor](#), double-click it in the grid. Alternatively, you can right-click the Procedure alias and select the **Edit Procedure(s)** context menu item.

### 5.3.1 Procedure Editor

A **Procedure** is a set of procedural constructs and embedded SQL statements that is stored in the database and can be called by name.

When you create an application with Microsoft SQL Server 2005 (and higher), the Transact-SQL programming language is the primary programming interface between your applications and the Microsoft SQL Server database. When you use Transact-SQL programs, two methods are available for storing and executing the programs. You can store the programs locally and create applications that send the commands to SQL Server and process the results, or you can store the programs as stored procedures in SQL Server and create applications that execute the stored procedures and process the results.

A **stored procedure** is a saved collection of Transact-SQL statements or a reference to a Microsoft .NET Framework common language runtime (CLR) method that can take and return user-supplied parameters. Procedures can be created for permanent use or for temporary use within a session, local temporary procedure, or for temporary use within all sessions, global temporary procedure.

**Procedure Editor** allows you to create new procedure, execute the existing procedure or edit its definition. It opens automatically when you create a new procedure and is available on editing an existing one.

To open a stored procedure in **Procedure Editor**, double-click it in the [Procedures](#) list.

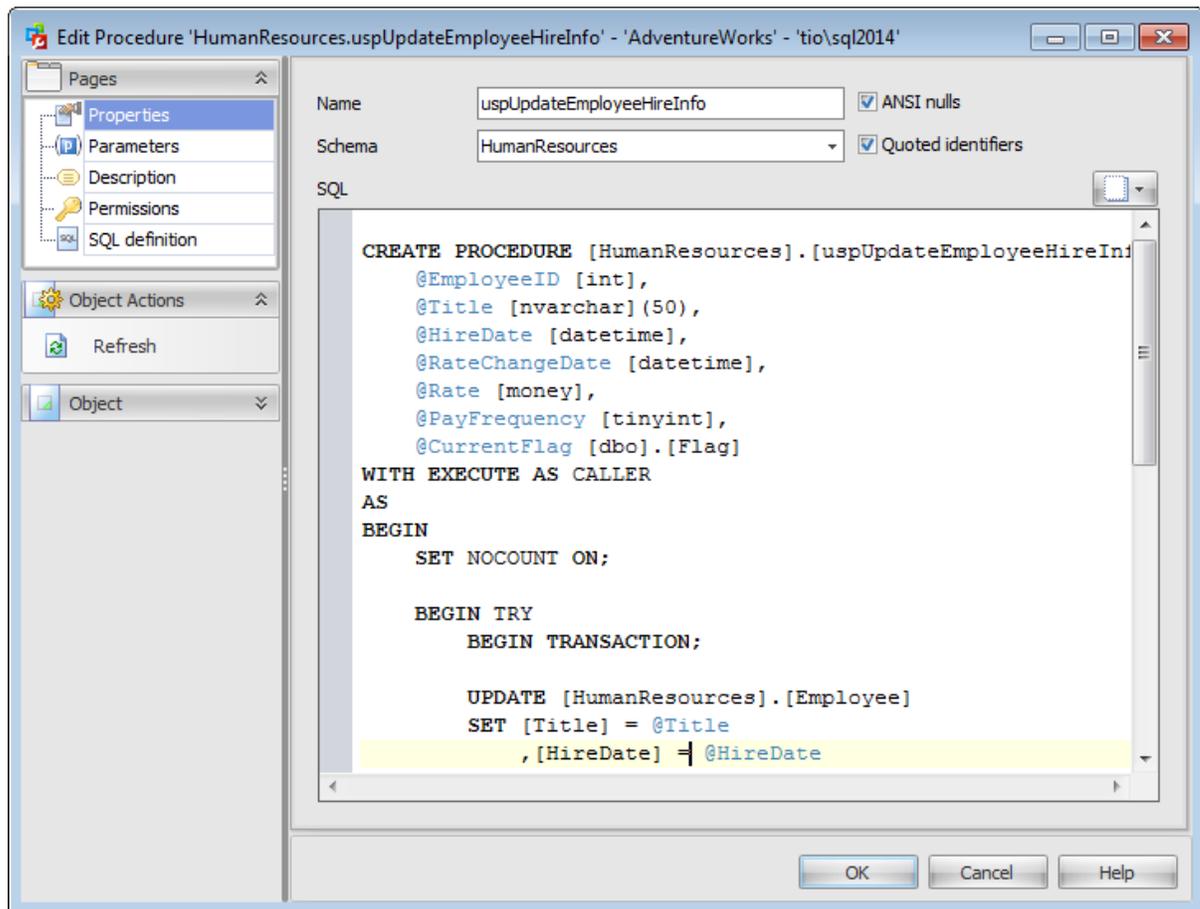
- [Properties](#)
- [Parameters](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.3.1.1 Properties

Use the **Properties** tab of the **Procedure Editor** to create/edit a stored procedure and specify its definition.

#### Name

Enter a name for the new procedure. Note that procedure names must comply with the rules for identifiers and must be unique within the [schema](#). Local or global temporary procedures can be created by using one number sign (#) before procedure\_name (#procedure\_name) for local temporary procedures, and two number signs for global temporary procedures (##procedure\_name).



#### Schema

Use the drop-down list to select the [schema](#) which the procedure belongs to. The default selection is the default schema for the current [user](#).

#### ANSI Nulls

This option specifies ISO compliant behavior of the Equals (=) and Not Equal To (<>) comparison operators when they are used with null values. When this option is switched on, a *SELECT* statement that uses *WHERE column\_name = NULL* returns zero rows even if there are null values in *column\_name*. A *SELECT* statement that uses *WHERE column\_name <> NULL* returns zero rows even if there are nonnull values in *column\_name*.

**Quoted identifier**

This option causes SQL Server to follow the ISO rules regarding quotation mark delimiting identifiers and literal strings. When this option is switched on, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When it is switched off, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers.

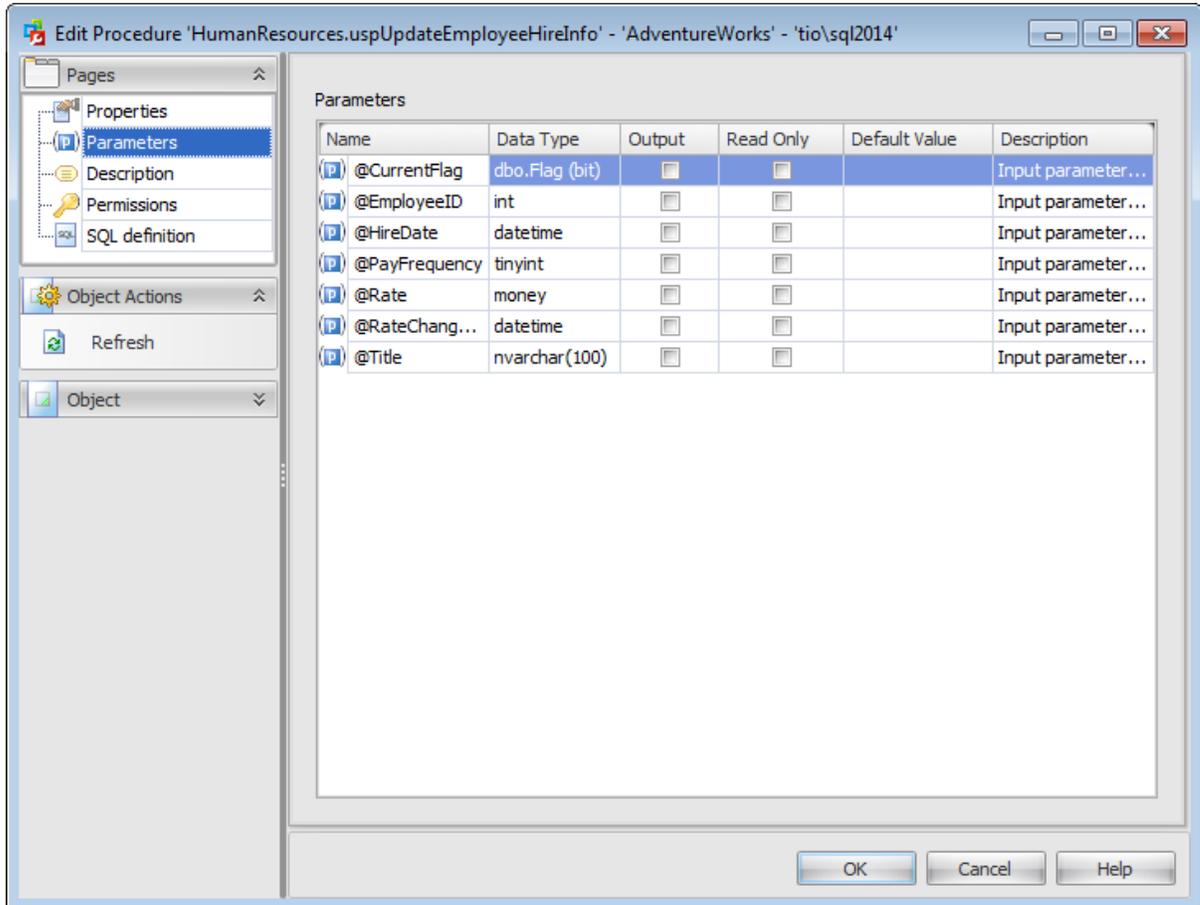
**SQL**

Use this text area to specify procedure definition as SQL statement.

You can click the  button to replace the text field content with the procedure definition template.

### 5.3.1.2 Parameters

The **Parameters** tab is provided for browsing the list of parameters of the stored procedure.



The parameter list provides the following attributes of each parameter used in the procedure:

*Name* (parameter name specified by using an at (@) sign as the first character)

*Data Type* (the parameter data type)

*Output* (indicates that the parameter is an output parameter)

*Read Only*

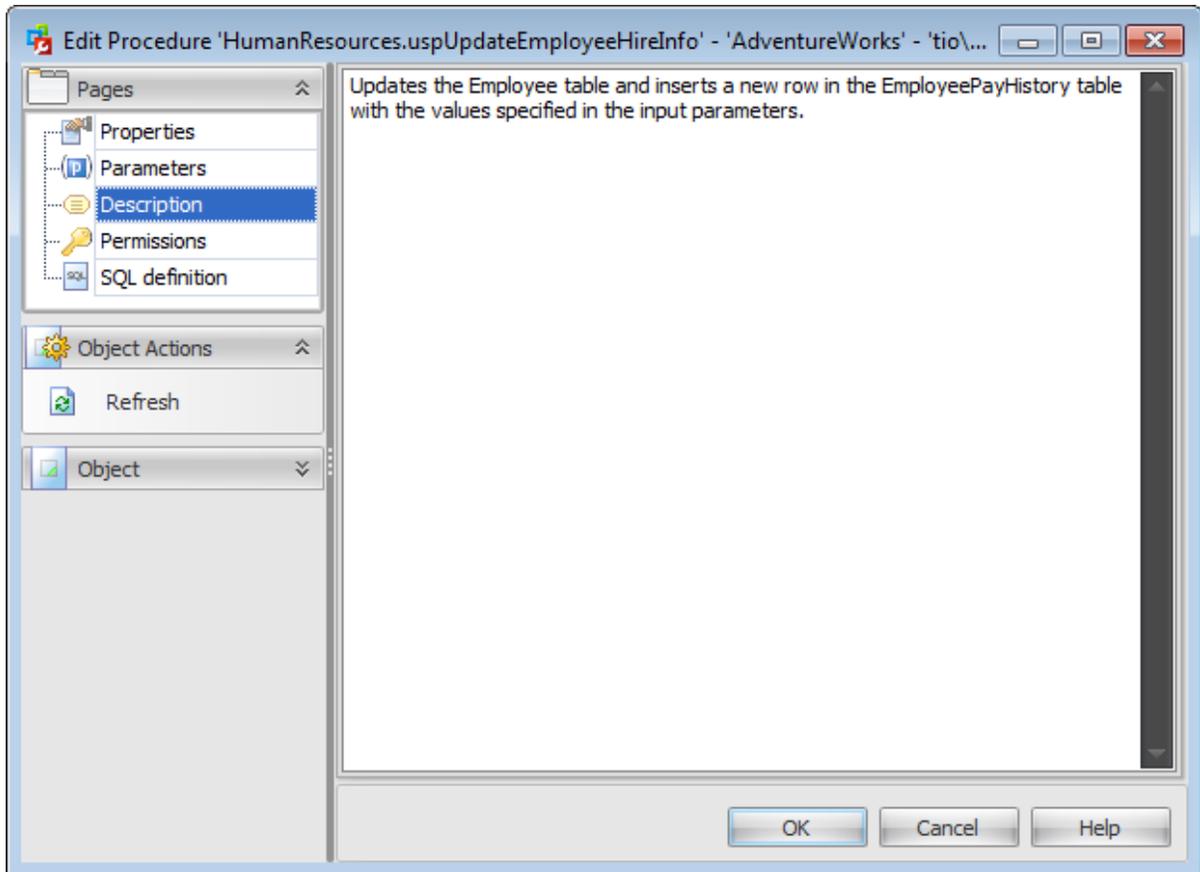
*Default Value* (a default value for the parameter)

*Description*

If necessary, you can also use the **Description** column to supply a *description* for each parameter.

### 5.3.1.3 Description

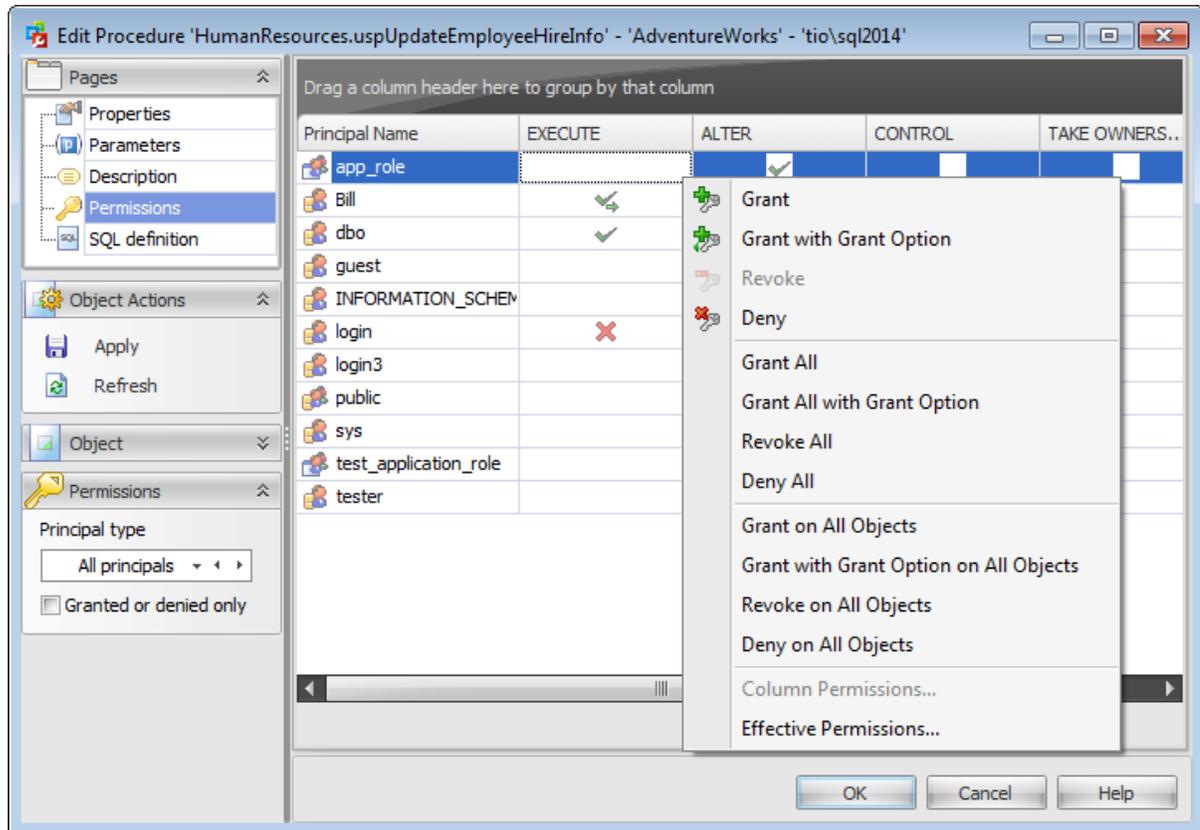
The **Description** tab allows you to view and edit the comment for the procedure (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.3.1.4 Permissions

The **Permissions** tab of the **Procedure Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

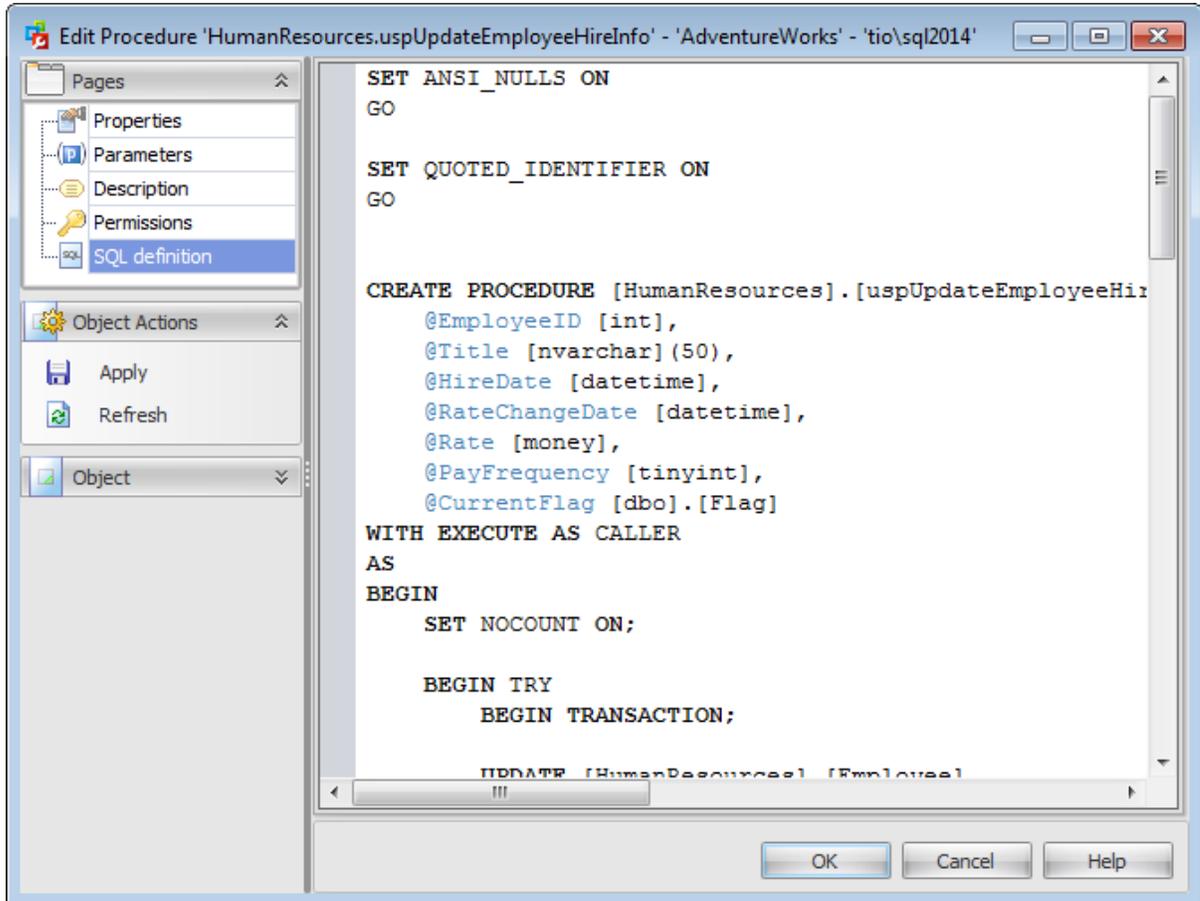
**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).



### 5.3.1.5 SQL Definition

The **SQL definition** tab of the **Procedure Editor** window allows you to view the object DDL.

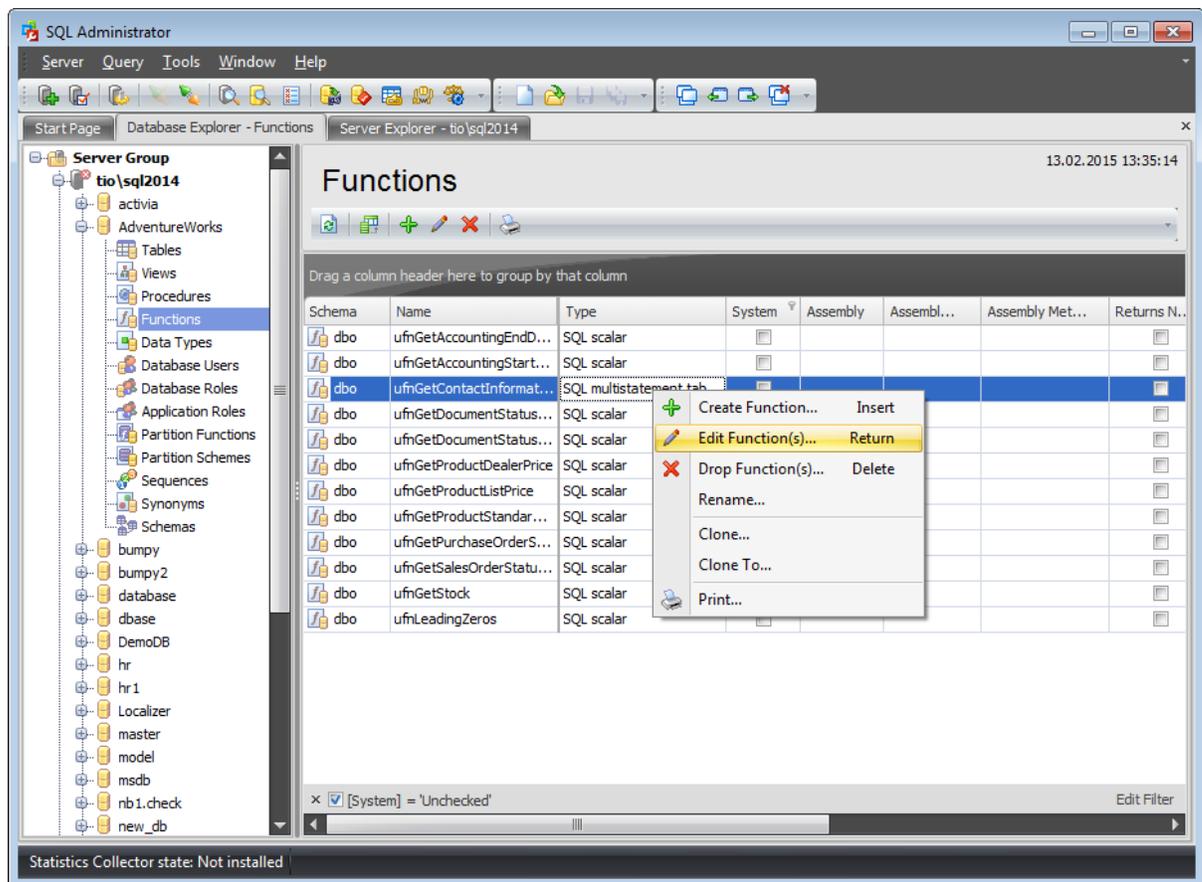


## 5.4 Functions

The **Functions** section of the Database Explorer is used for managing functions in the selected database.

Use the context menu to:

- + [create](#) a new function
-  [edit](#) the selected function(s)
- drop selected function(s)
- rename the selected function
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database functions as a grid with the following columns: *Schema, Name, Type, System, Assembly Method, Assembly Class, Assembly, Returns Null on Null input, Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Function in the [Function Editor](#), double-click it in the grid. Alternatively, you can right-click the Function alias and select the **Edit Function(s)** context menu item.

### 5.4.1 Function Editor

A **Function** is a mapping embodied as a program (the function *body*) that can be invoked by using zero or more input values (*arguments*) to a single value (the *result*). A **Function** is stored as a database object providing reusable code.

The Transact-SQL programming language provides the following types of functions:

*Rowset functions* (can be used like table references in an SQL statement)

*Aggregate functions* (operate on a collection of values but return a single, summarizing value)

*Ranking functions* (return a ranking value for each row in a partition)

*Scalar functions* (operate on a single value and then return a single value; scalar functions can be used wherever an expression is valid)

**Function Editor** allows you to define function properties. It opens automatically when you create a new function and is available on editing an existing one.

To open a function in **Function Editor**, double-click it in the [Functions](#) list.

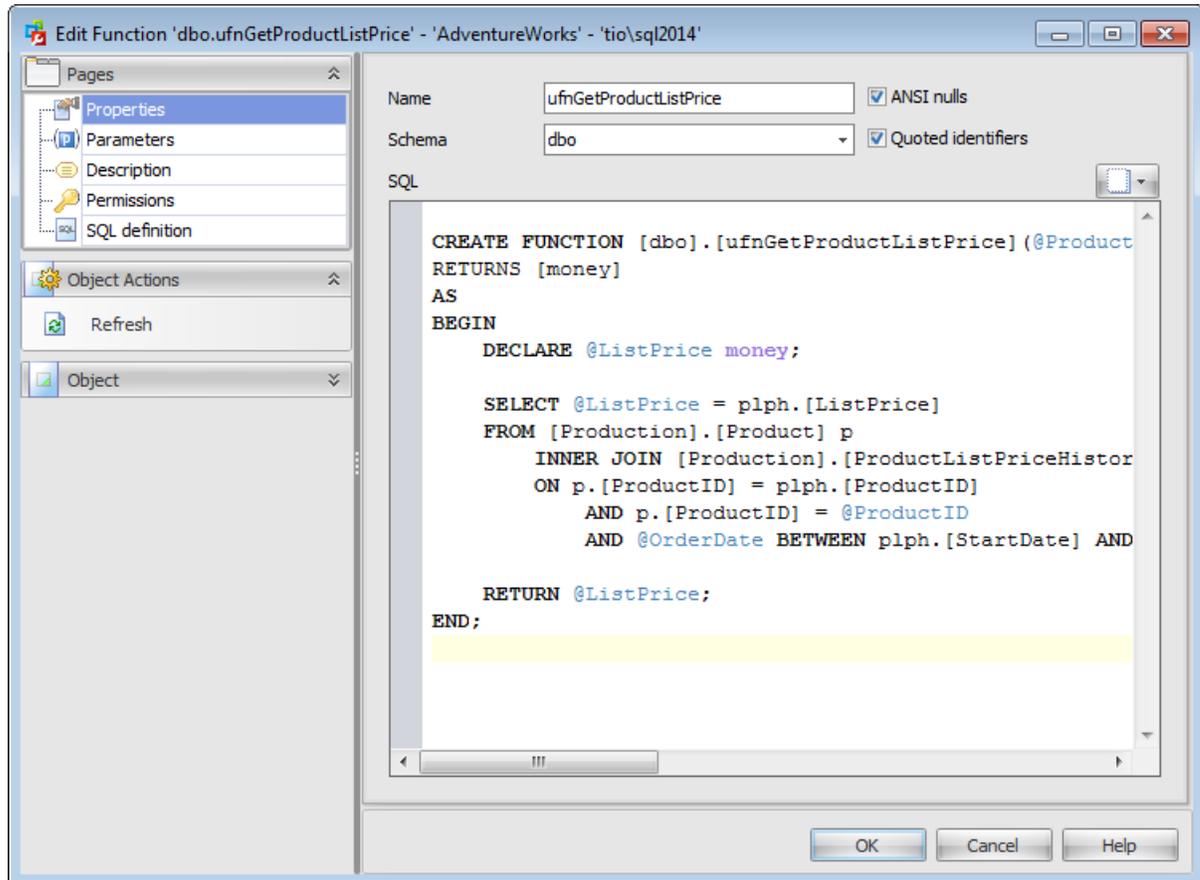
- [Properties](#)
- [Parameters](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.4.1.1 Properties

Use the **Properties** tab of the **Function Editor** to create/edit a function and specify its definition.

#### Name

Enter a name for the new function. Note that function names must comply with the rules for identifiers and must be unique within the [schema](#).



#### Schema

Use the drop-down list to select the [schema](#) which the function belongs to. The default selection is the default schema for the current [user](#).

#### ANSI Nulls

This option specifies ISO compliant behavior of the Equals (=) and Not Equal To (<>) comparison operators when they are used with null values. When this option is switched on, a *SELECT* statement that uses *WHERE column\_name = NULL* returns zero rows even if there are null values in *column\_name*. A *SELECT* statement that uses *WHERE column\_name <> NULL* returns zero rows even if there are nonnull values in *column\_name*.

#### Quoted identifier

This option causes SQL Server to follow the ISO rules regarding quotation mark delimiting

identifiers and literal strings. When this option is switched on, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When it is switched off, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers.

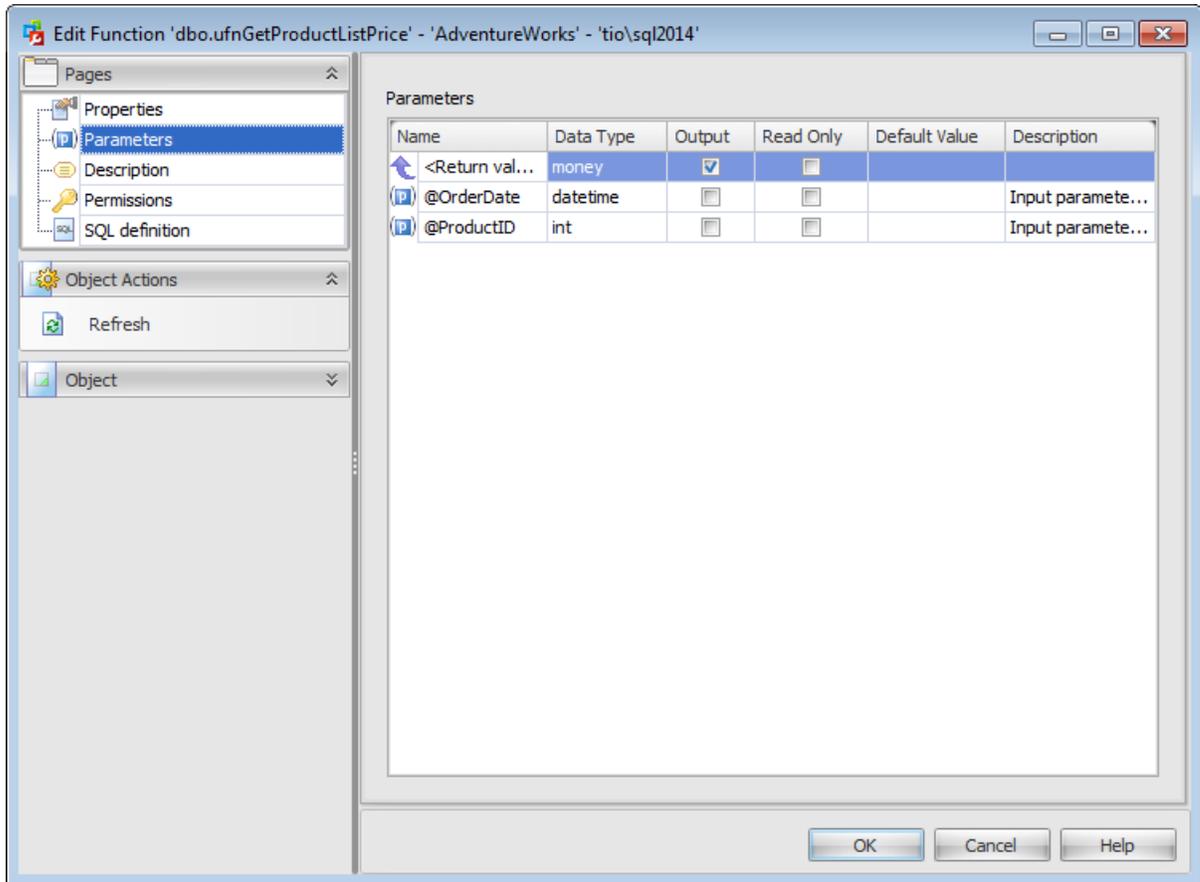
**SQL**

Use this text area to specify function definition as SQL statement.

You can click the  button to replace the text field content with the function definition template.

### 5.4.1.2 Parameters

The **Parameters** tab is provided for browsing the list of parameters of the function.



The parameter list provides the following attributes of each parameter used in the function:

*Name* (parameter name specified by using an at (@) sign as the first character)

*Data Type* (the parameter data type)

*Output* (indicates that the parameter is an output parameter)

*Read Only*

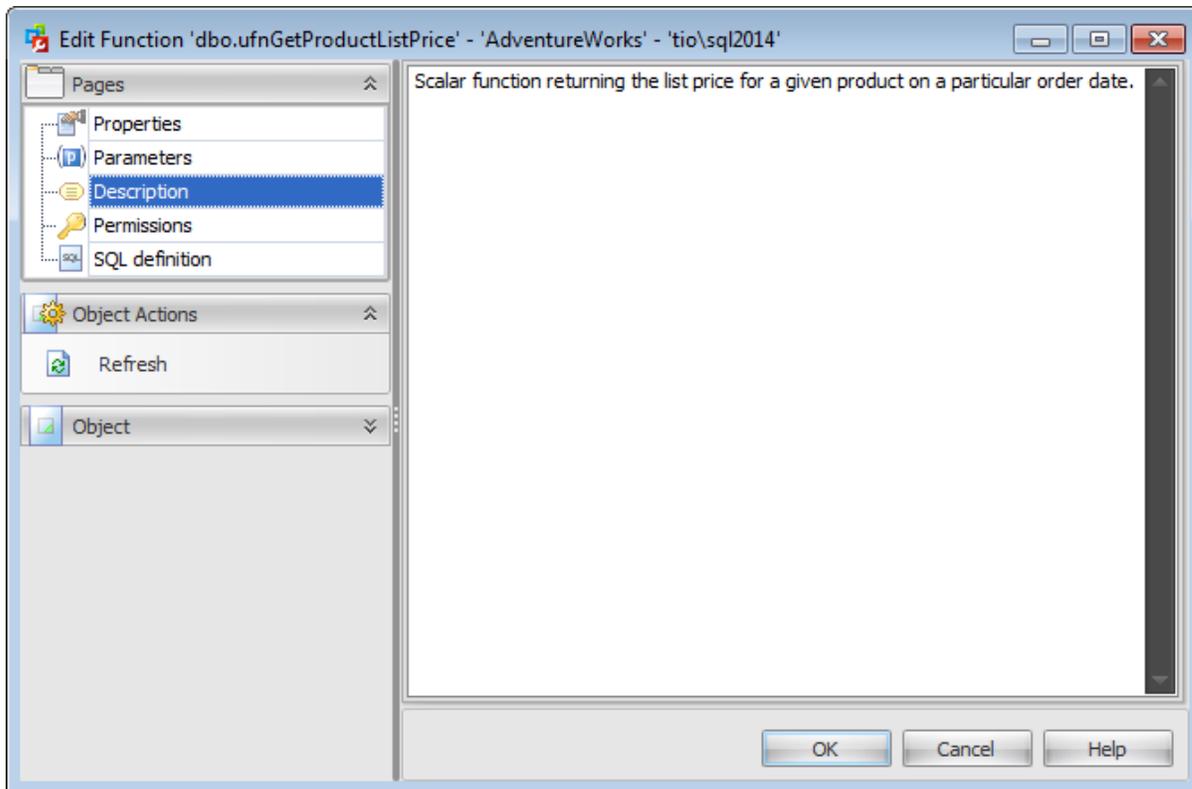
*Default Value* (a default value for the parameter)

*Description*

If necessary, you can also use the **Description** column to supply a *description* for each parameter.

### 5.4.1.3 Description

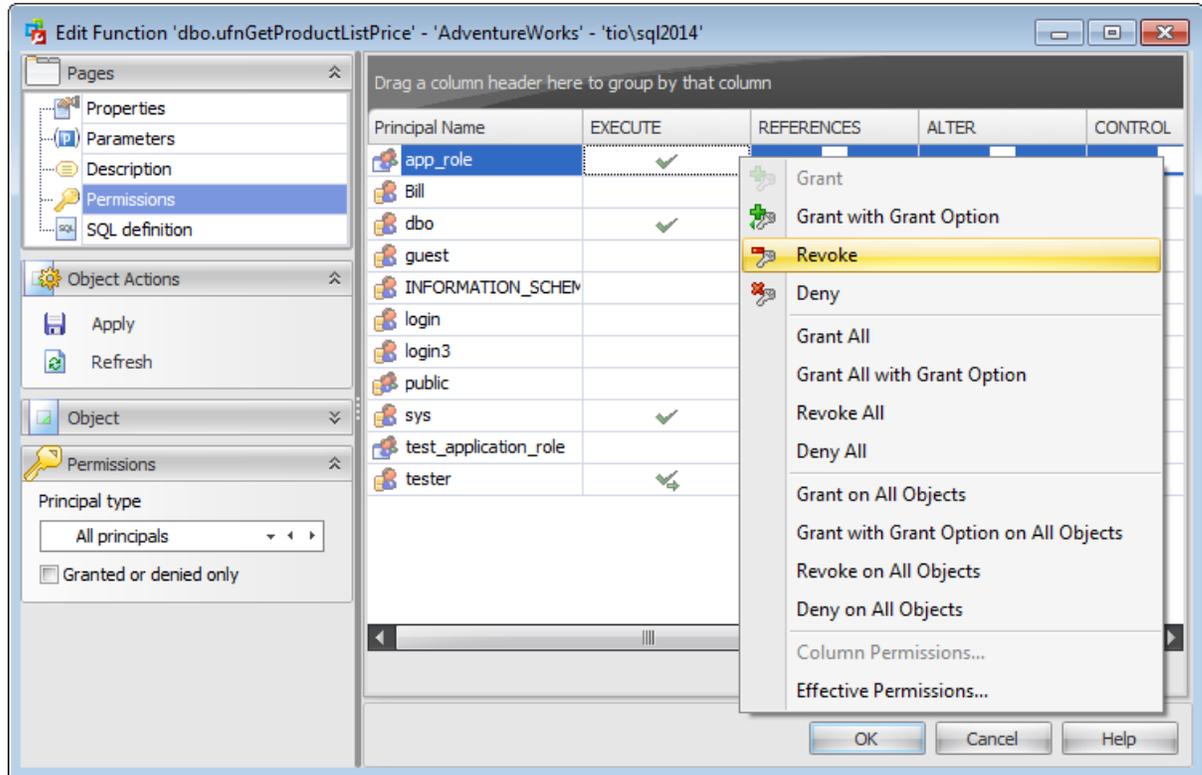
The **Description** tab allows you to view and edit the comment for the function (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

#### 5.4.1.4 Permissions

The **Permissions** tab of the **Function Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

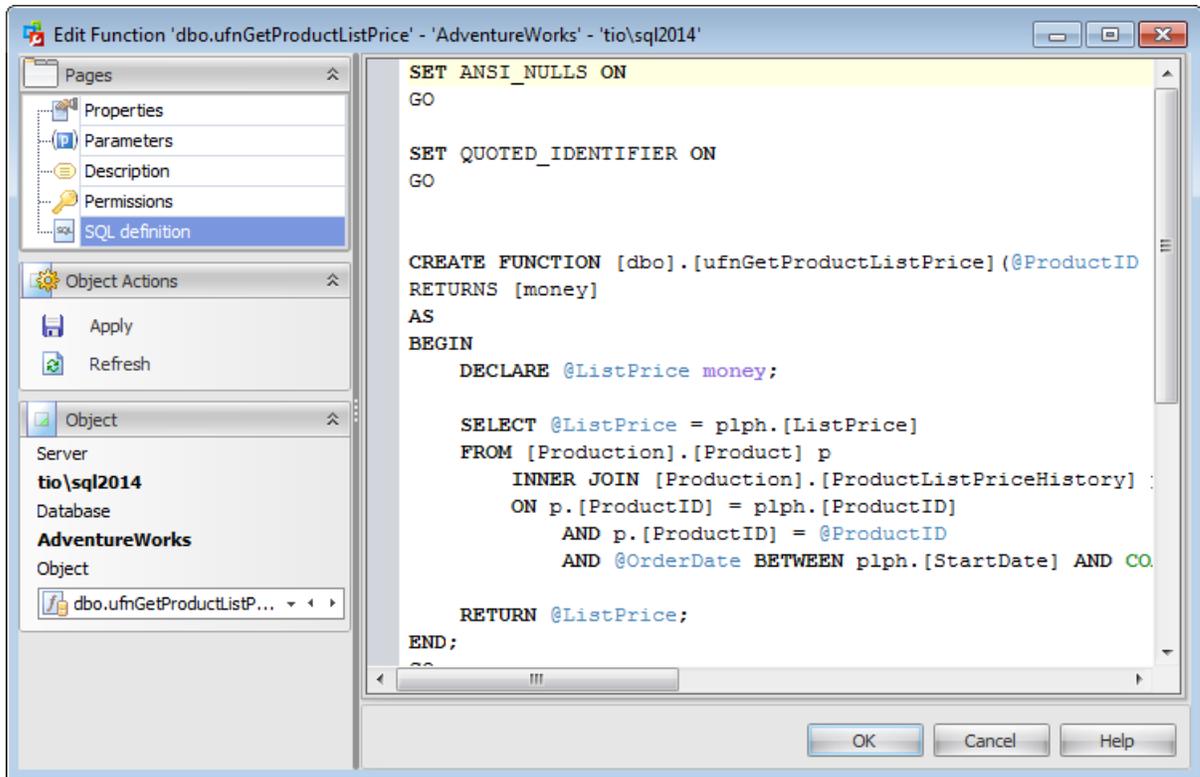
- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).

### 5.4.1.5 SQL Definition

The **SQL definition** tab of the **Function Editor** window allows you to view the object DDL.

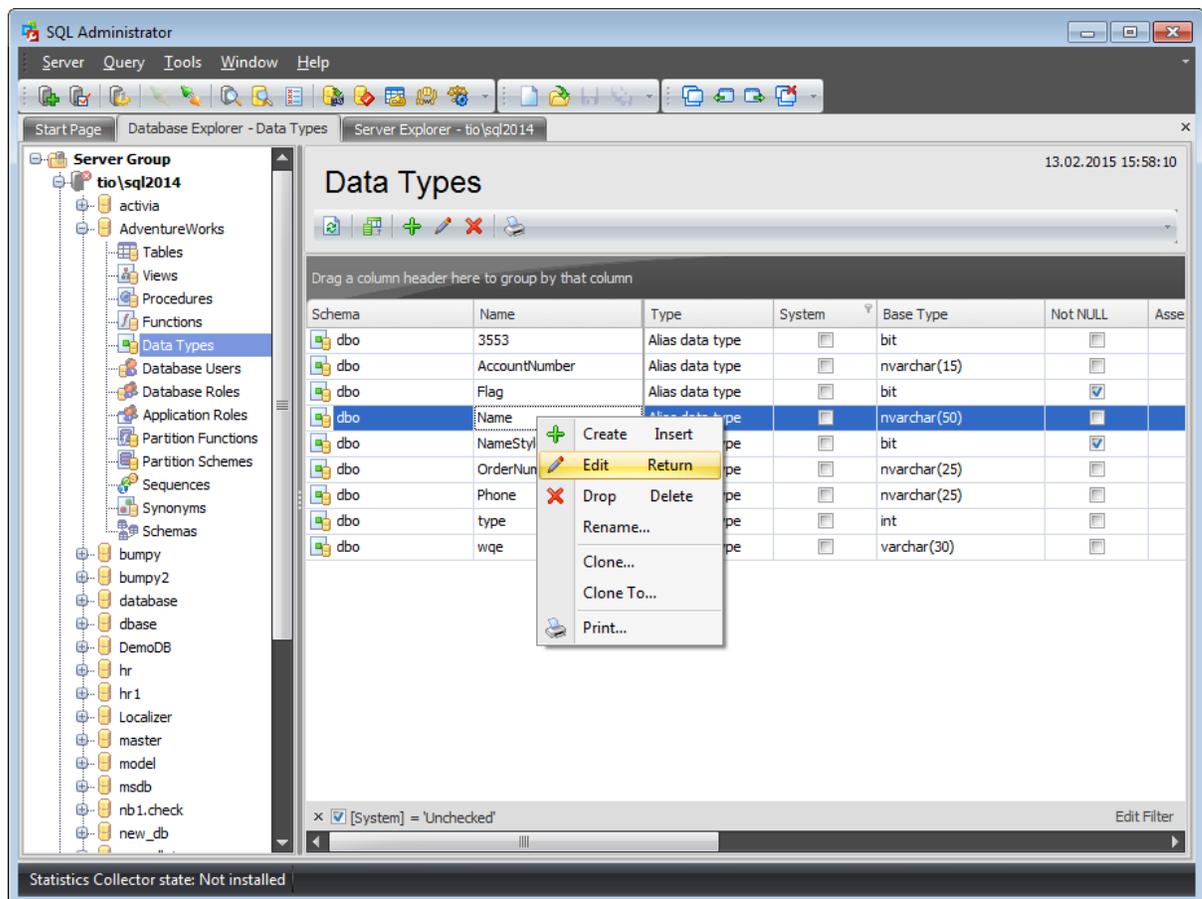


## 5.5 Data Types

The **Data Types** section of the Database Explorer is used for managing data types in the selected database.

Use the context menu to:

- + [create](#) a new data type
-  [edit](#) the selected data type(s)
- drop selected data type(s)
- rename the selected data type
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database data types as a grid with the following columns: *Schema, Name, Type, System, Base Type, Not NULL, Assembly Class, Assembly, Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Data type in the [Data Type Editor](#), double-click it in the grid. Alternatively, you can right-click the data type alias and select the **Edit** context menu item.



### 5.5.1 Data Type Editor

Use the **Data Type Editor** to create an alias data type or a user-defined type in the current database in SQL Server Database. The implementation of an alias data type is based on a SQL Server native system type. A user-defined type is implemented through a class of an assembly in the Microsoft .NET Framework common language runtime (CLR).

**UDT Editor** allows you to define user-defined type properties. It opens automatically when you create a new type and is available on editing an existing one.

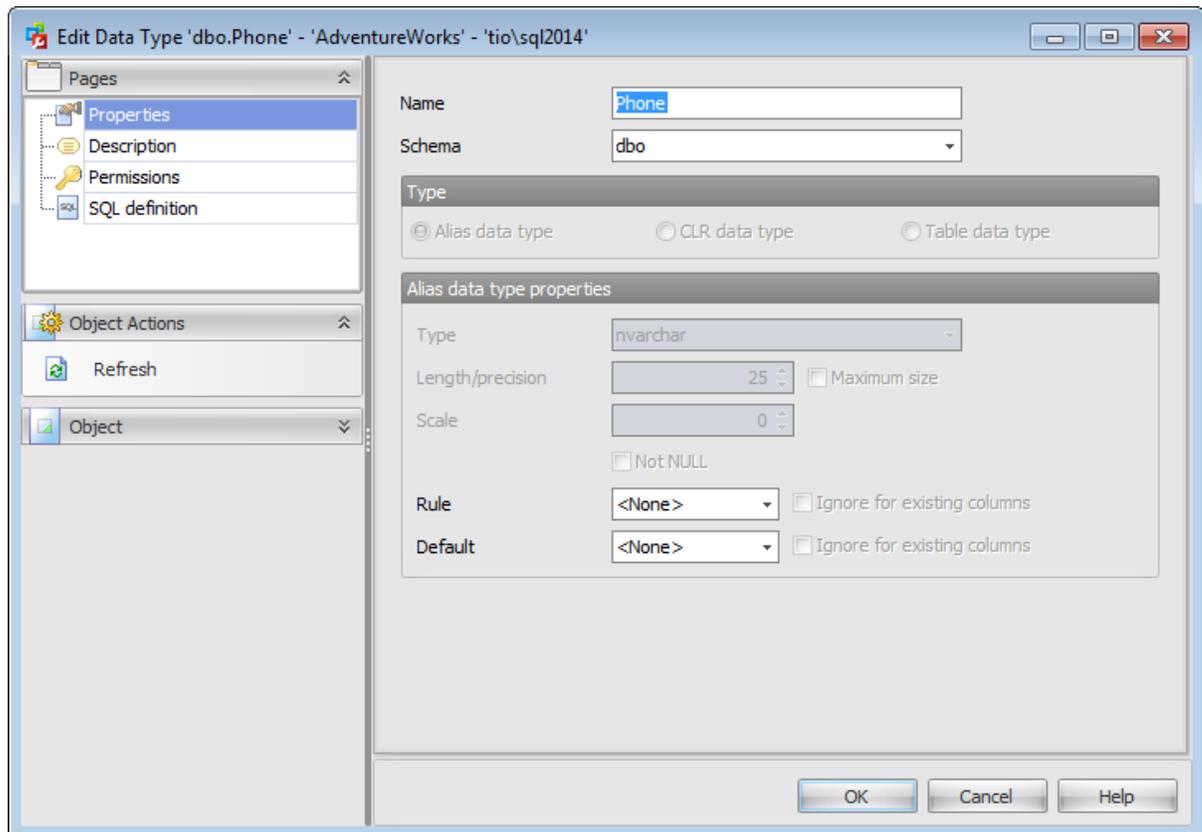
To open a user-defined type in **UDT Editor**, double-click it in [Data Types](#) list.

### 5.5.1.1 Properties

Use the **Properties** tab of the **Data Type Editor** to create/edit a data type and specify its properties.

#### Name

Enter a name for the new data type. Note that data type names must comply with the rules for identifiers and must be unique within the [schema](#).



#### Schema

Use the drop-down list to select the [schema](#) which the data type belongs to. The default selection is the default schema for the current [user](#).

#### Type

On creating a new user-defined type, you should first specify its implementation kind.

##### • Alias data type

This selection specifies that an alias data type will be created on the basis of an existing SQL Server data type.

##### • CLR data type

This selection specifies that the user-defined will be taken from an assembly.

##### • Table data type

This selection specifies that a new user-defined table type will be created.

### Alias data type properties

#### Type

Use the drop-down list to select SQL Server supplied data type on which the alias data type is based (*base\_type*). The base type of an existing Data Type is not editable.

#### Length/precision

Specify the size value (for certain types). Check  **Maximum size** to use the maximum values set by SQL Server.

#### Scale

Defines the maximum number of decimal digits that can be stored to the right of the decimal point of the value (for numeric and decimal data types).

#### Not Null

Specify whether the Data Type can accept NULL values. The nullability of an existing Data Type is not editable.

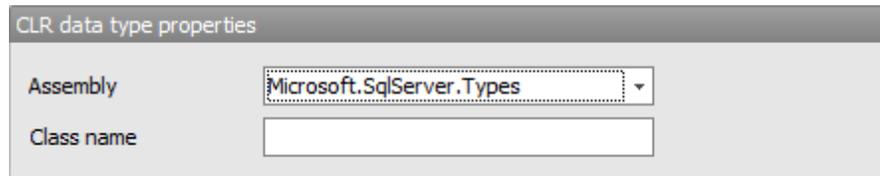
#### Rule

Use the drop-down list to select a rule to bind to the user-defined data type (optional). Check  **Ignore for existing columns** to apply the selected rule only to newly added columns.

#### Default

Use the drop-down list to select a default to bind to the user-defined data type (optional). Check  **Ignore for existing columns** to apply the selected default only to newly added columns.

### CLR Data Type



CLR data type properties

Assembly: Microsoft.SqlServer.Types

Class name:

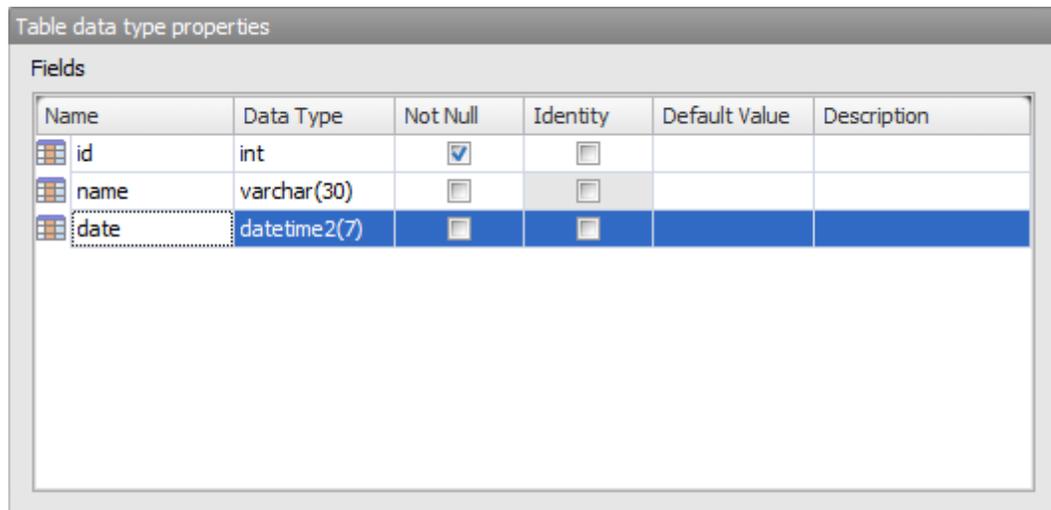
#### Assembly

Use the drop-down list to select the SQL Server assembly that references the implementation of the Data Type in the common language runtime (CLR). The assembly must have been already created in the database.

#### Class name

Specifies the class within the assembly that implements the Data Type. The class name can be a namespace-qualified name (if the programming language that is used to write the class uses the concept of namespaces, such as C#).

### Table Data Type



Name	Data Type	Not Null	Identity	Default Value	Description
id	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
name	varchar(30)	<input type="checkbox"/>	<input type="checkbox"/>		
date	datetime2(7)	<input type="checkbox"/>	<input type="checkbox"/>		

### Fields

This area lists the columns of the Data Type as a grid.

Double-click a field to open [Field Editor](#) for editing the field.

Right-click within the **Fields** area to display the context menu allowing you to *add* new, *insert*, *edit* using [Field Editor](#), *delete* the selected field, or move it up/down within the list.

The **Fields** list provides the following attributes of each field of the Data Type:

*Name*

*Data Type*

*Not Null*

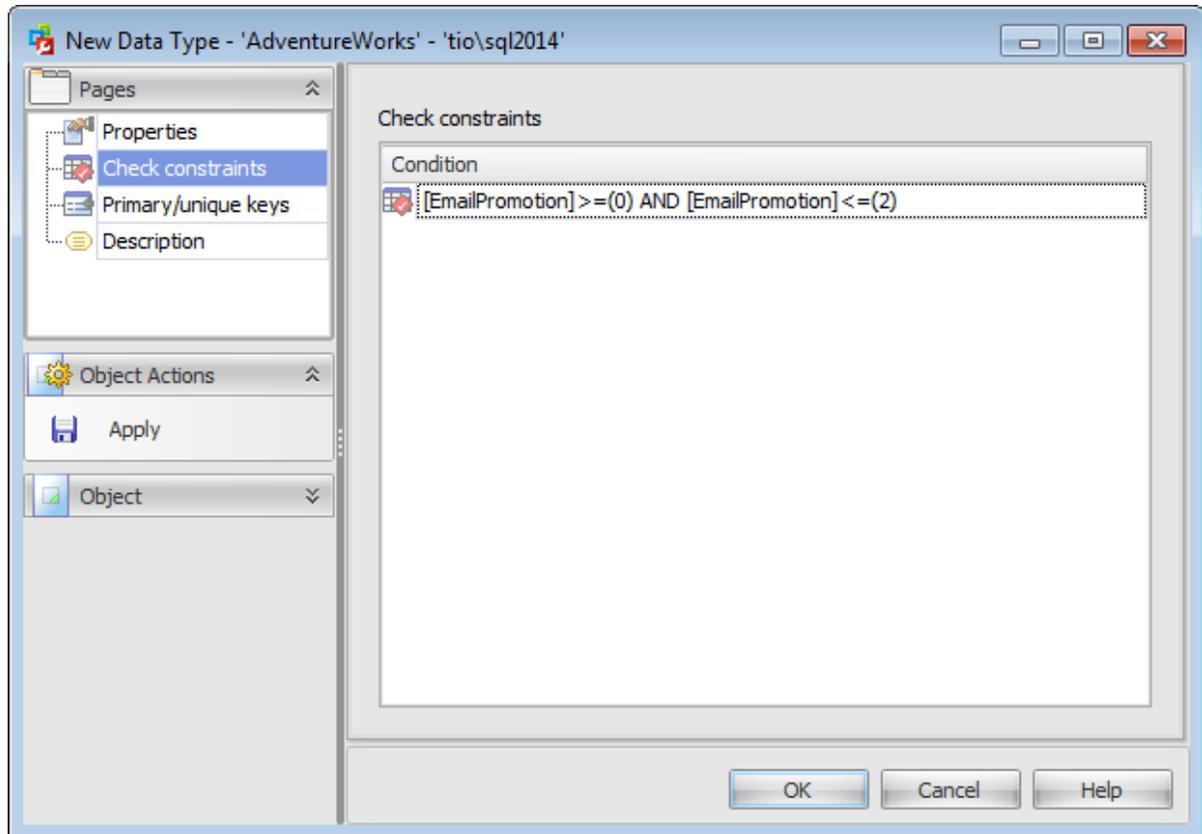
*Identity*

*Default Value*

*Description*

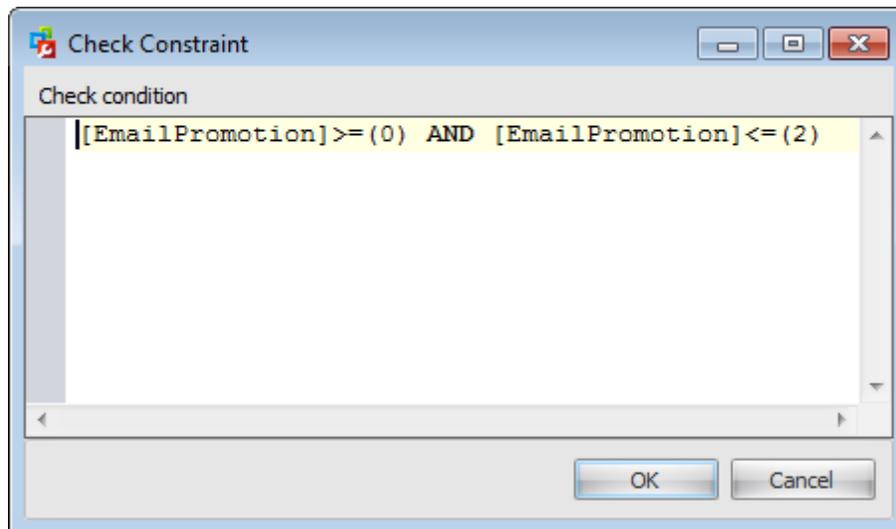
### 5.5.1.2 Check constraints

The **Check constraints** tab is provided for viewing and editing data type checks. This tab is only available for Table Data Type.



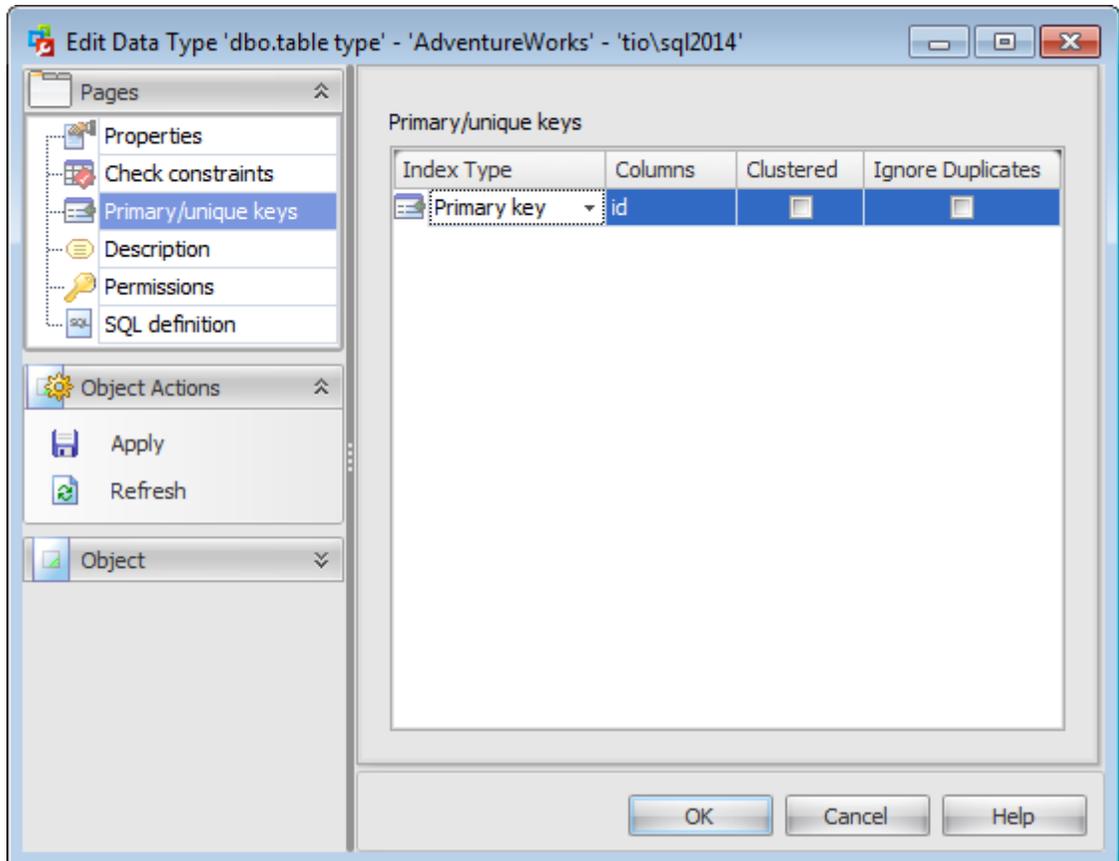
Right-click a check to display the context menu allowing you to *create* new, *edit*, or *drop* the selected check constraint.

Double-click a check to edit the condition of the check.



### 5.5.1.3 Primary/unique keys

The **Primary/unique keys** tab is provided for viewing and editing data type primary/unique keys. This tab is only available for Table Data Type.

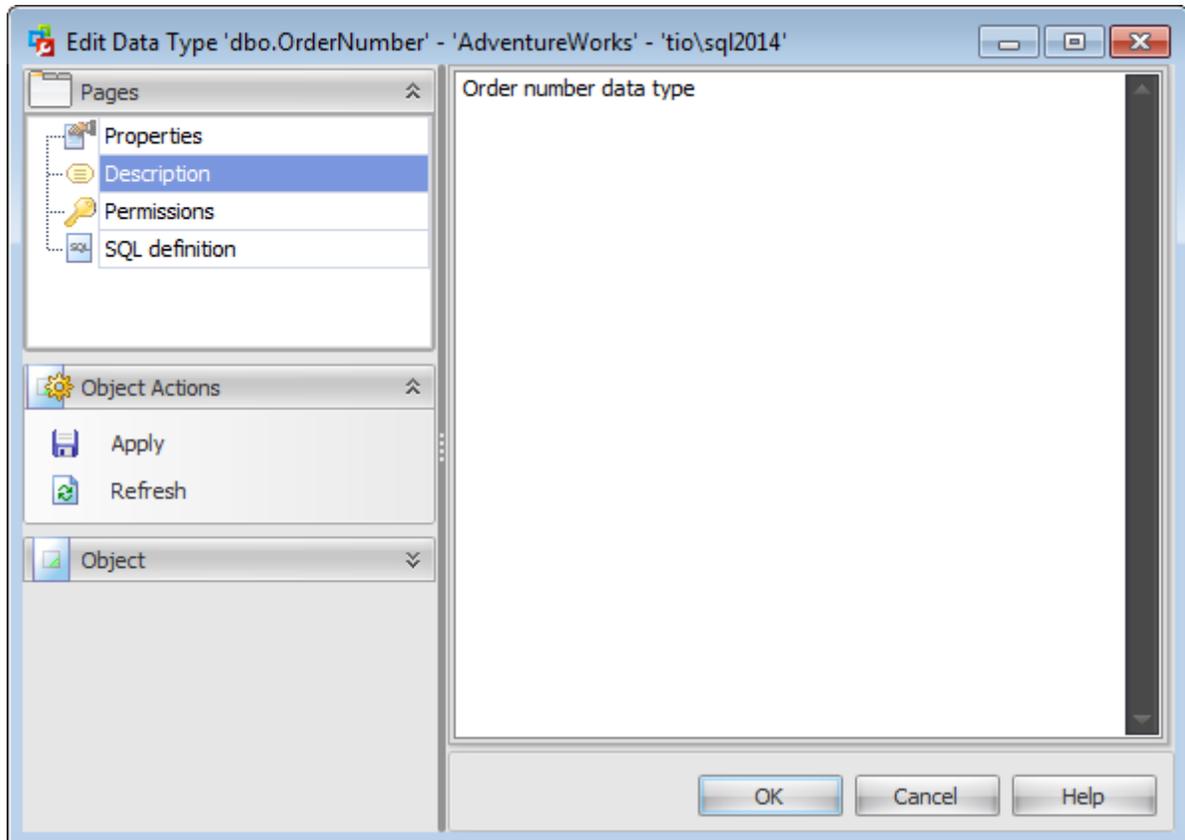


Right-click a key to display the context menu allowing you to *create new*, *edit*, or *drop* the selected key.

Double-click a key to edit the index in the [Index Editor](#).

#### 5.5.1.4 Description

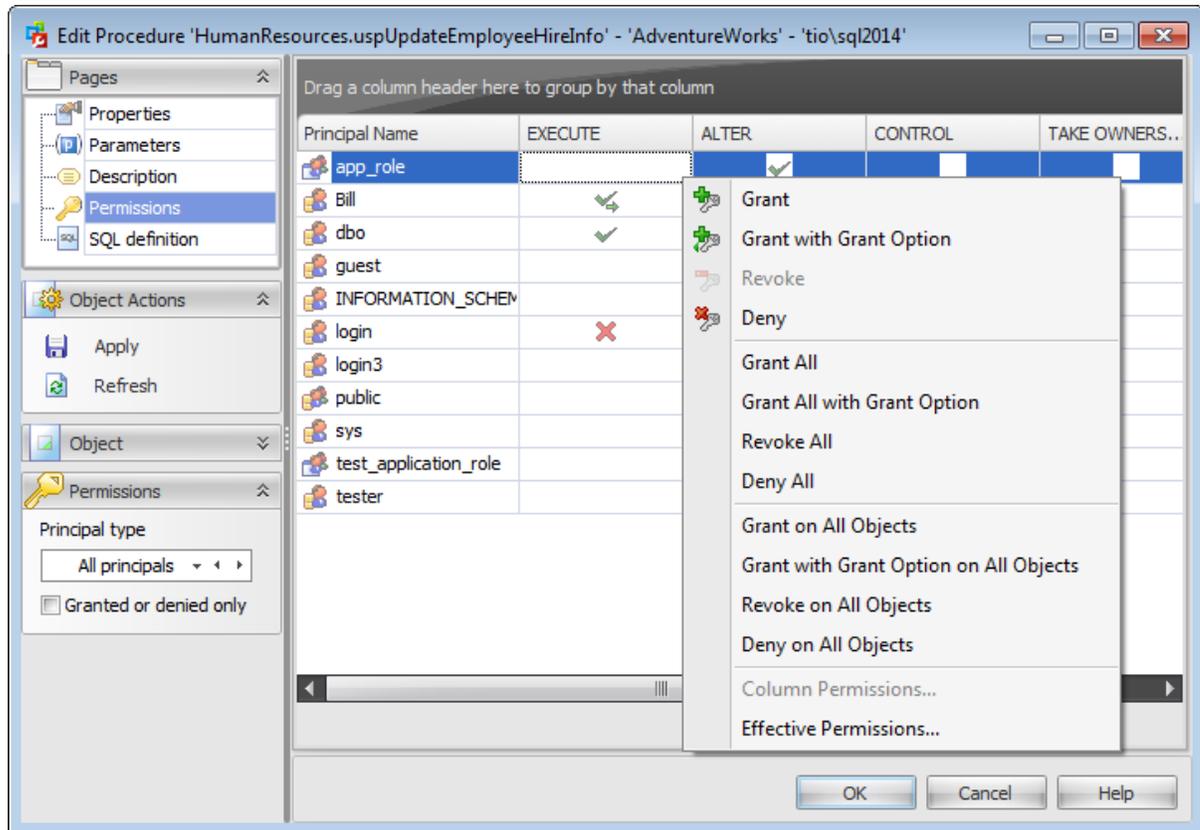
The **Description** tab allows you to view and edit the comment for the data type (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty* 'MS\_Description'.

### 5.5.1.5 Permissions

The **Permissions** tab of the **Data Type Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

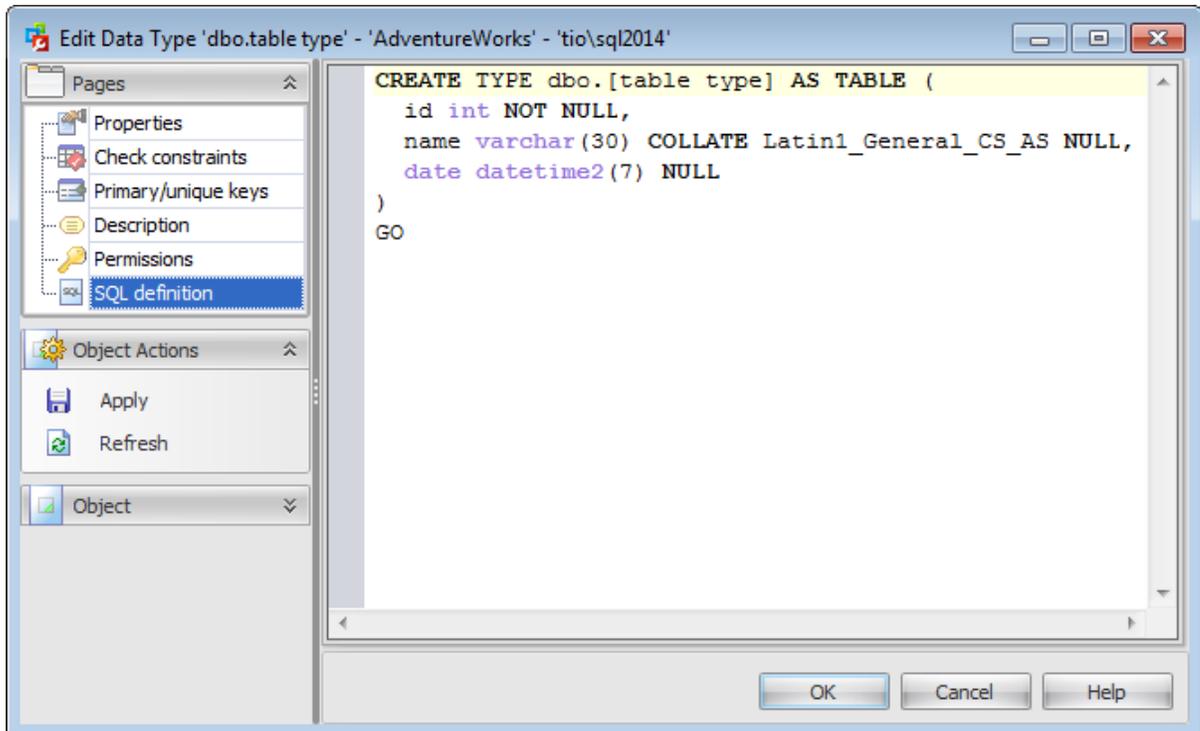
**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).



### 5.5.1.6 SQL Definition

The **SQL definition** tab of the **Data Type Editor** window allows you to view the object DDL.



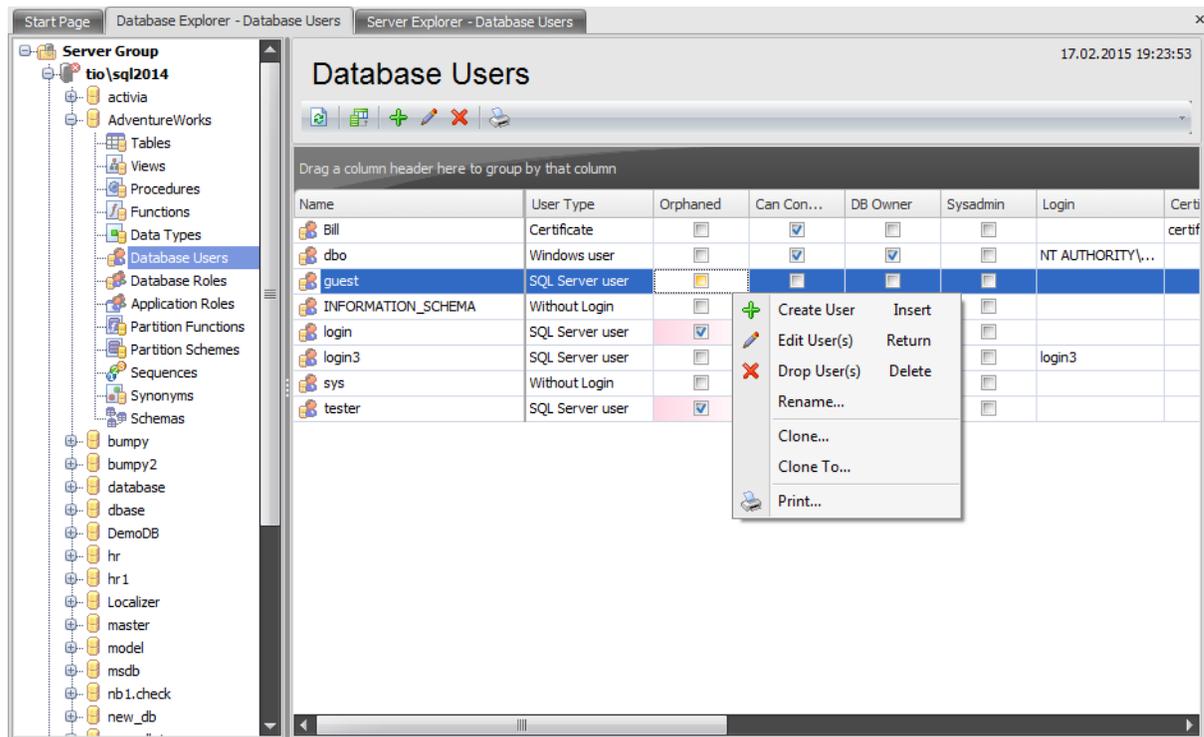
## 5.6 Database Users

A **Database User** is a database-level security principal enabling object access permission control at the finest level of granularity. A user represents a single SQL Server login within the scope of the database in which the user is defined.

Use the context menu to:

- + [create](#) a new user
-  [edit](#) the selected user(s)
- drop selected user(s)
- rename selected user
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a user, 'Enter' button to edit user (s) and 'Delete' button to drop user(s).



The list displays the existing database users as a grid with the following columns: *Name*, *User type*, *Orphaned*, *Can Connect*, *DB Owner*, *Sysadmin*, *Login*, *Certificate*, *Asymmetric Key*, *Default Schema*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Database User in the [User Editor](#), double-click it in the grid. Alternatively, you can right-click the Database User alias and select the **Edit User(s)** context menu item.

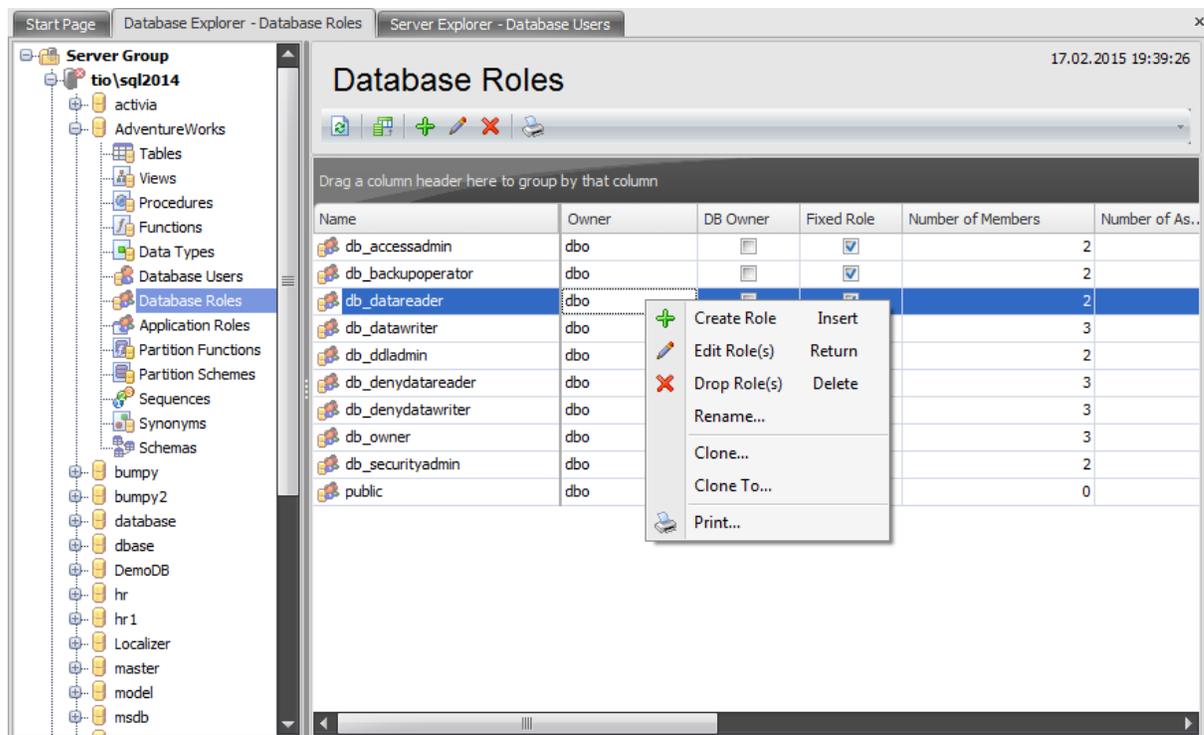
## 5.7 Database Roles

All [users](#) interact with SQL Server within the context of a database-level principal - **Database Role**. A user can belong to multiple groups and have multiple roles, and the operations that are permitted by each role determine the actions that the user can perform. It is necessary to create SQL Server database roles when a group of users needs to perform a specified set of activities in the database.

Use the context menu or buttons on the work area toolbar to manage the list of database roles:

-  [create](#) a new database role
-  [edit](#) selected database role(s)
-  drop selected database role(s)
- rename selected database role
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a role, 'Enter' button to edit role (s) and 'Delete' button to drop role(s).



The list displays the existing database roles as a grid with the following columns: *Name*, *Owner*, *DB Owner*, *Fixed Role*, *Created*, *Modified*, *Number of Assigned Roles*, *Number of Members*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Database Role in the [Role Editor](#), double-click it in the grid. Alternatively, you

can right-click the Database Role alias and select the **Edit Role(s)** context menu item.

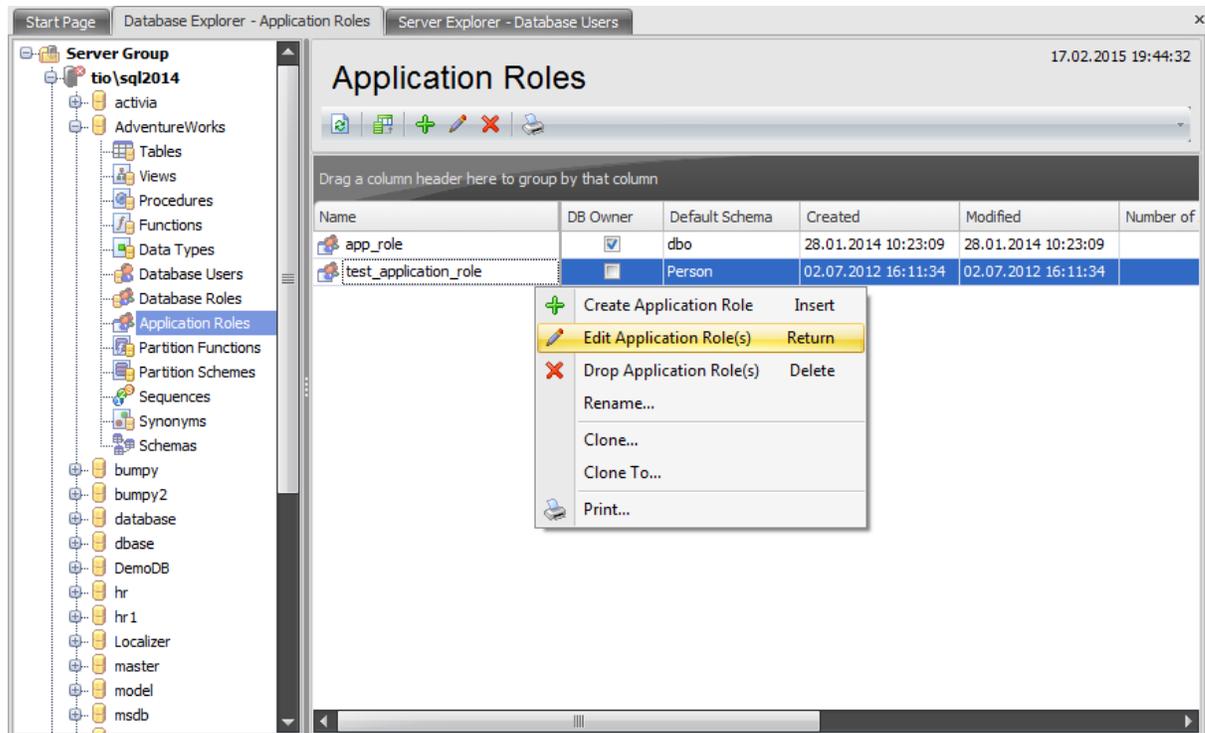
## 5.8 Application Roles

An **Application Role** is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application. Unlike database roles, application roles contain no members and are inactive by default. Because application roles are a database-level principal, they can access other databases only through permissions granted in those databases to guest. Therefore, any database in which guest has been disabled will be inaccessible to application roles in other databases.

Use the context menu or buttons on the work area toolbar to manage application roles:

-  [create](#) a new application role
-  [edit](#) selected application role(s)
-  drop selected application role(s)
- rename selected application role
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a role, 'Enter' button to edit role (s) and 'Delete' button to drop role(s).



The list displays the existing application roles as a grid with the following columns: *Name*, *DB Owner*, *Default Schema*, *Created*, *Modified*, *Number of Assigned Roles*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

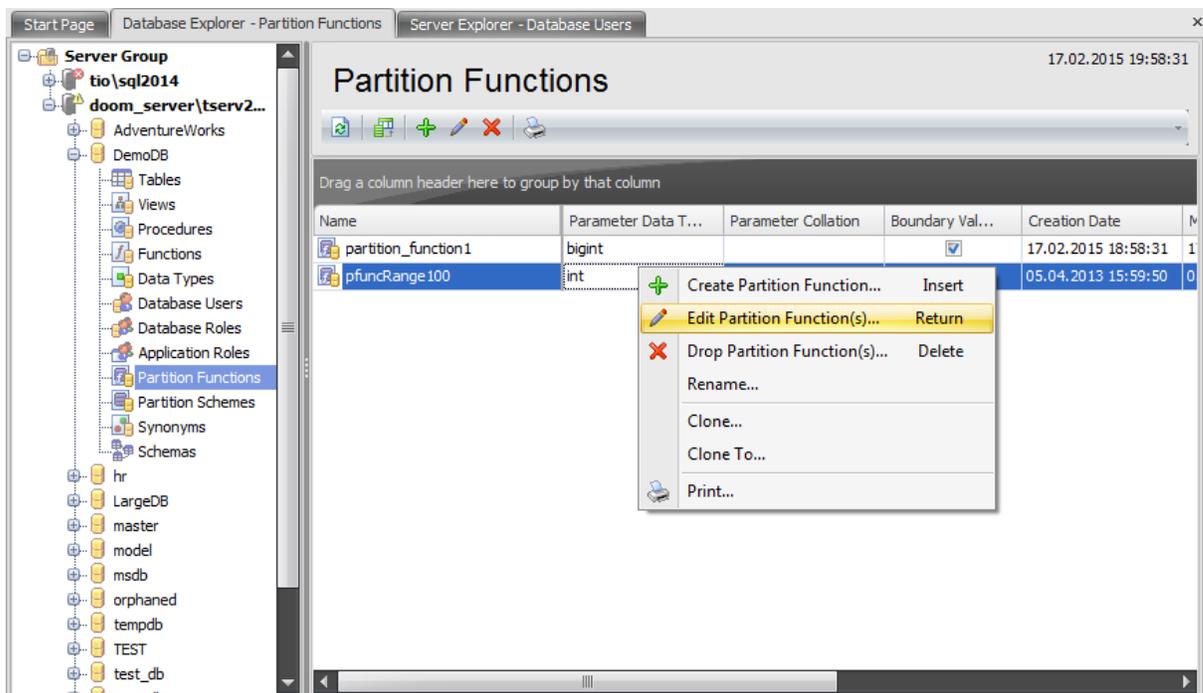
To open an Application Role in the [Application role Editor](#), double-click it in the grid. Alternatively, you can right-click the Application Role alias and select the **Edit Application Role(s)** context menu item.

## 5.9 Partition Functions

The **Partition Functions** section of the Database Explorer is used for managing partition functions in the selected database.

Use the context menu to:

- + [create](#) a new partition function
-  [edit](#) the selected partition function(s)
- drop selected partition function(s)
- rename the selected partition function
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database partition functions as a grid with the following columns: *Name*, *Parameter Data Type*, *Parameter Collation*, *Boundary Value*, *Creation Date*, *Modification Date*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Partition Function in the [Partition Function Editor](#), double-click it in the grid. Alternatively, you can right-click the partition function alias and select the **Edit Partition Function(s)** context menu item.

### 5.9.1 Partition Function Editor

A **Partition function** specifies how a [table](#) or [index](#) is partitioned. The function maps the domain into a set of partitions. When creating a partition function, you specify the number of partitions, the partitioning column, and the range of partition column values for each partition. Note that when you specify the partitioning column, you can only specify one.

**Partition Function Editor** allows you to define partition function properties. It opens automatically when you create a new partition function and is available on editing an existing one.

To open a partition function in **Partition Function Editor**, double-click it in the [Partition Functions](#) list.

- [Properties](#)
- [Description](#)
- [SQL Definition](#)

### 5.9.1.1 Properties

Use the **Properties** tab of the **Partition Function Editor** to create/edit a partition function and specify its properties.

**Name**

Enter a name for the new partition function, or view/edit the name of the partition function being edited.

**Parameter****Data type**

Specifies the data type of the column used for partitioning.

**Length/precision**

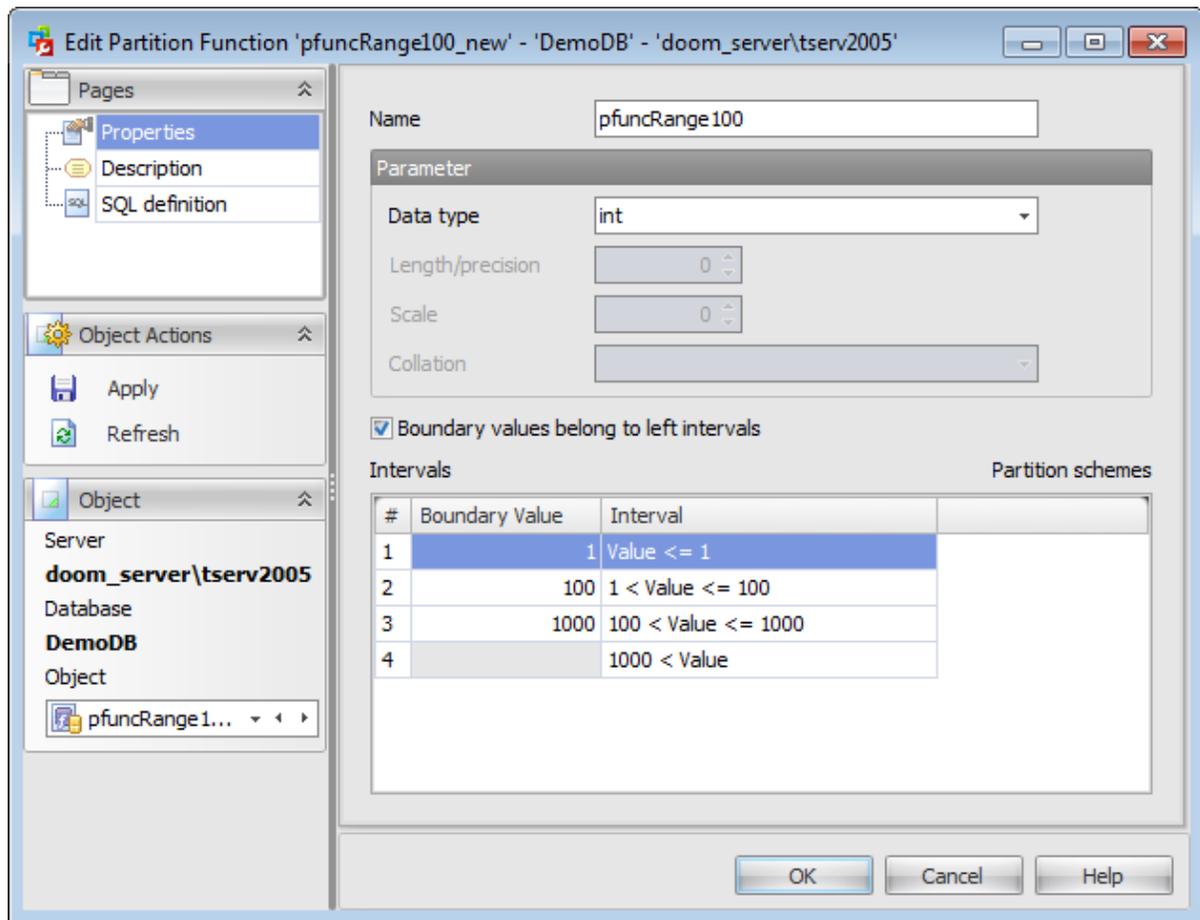
Specifies the size of the data type. This box is not editable if the size of the selected data type is fixed or not applicable.

**Scale**

Specifies the maximum number of decimal digits that can be stored to the right of the decimal point. This box is editable only for *numeric* and *decimal* data types only.

**Collation**

Defines the collation for *text*, *char*, *nchar*, *varchar*, *nvarchar* values.



**Boundary values belong to left intervals**

This option specifies to which side of each boundary value interval, left or right, the **boundary value** belongs, when interval values are sorted by the Database Engine in ascending order from left to right. If checked, *LEFT* is specified.

### Boundary values

This area lists the boundary values specified by the partition function.

To add a value, right-click within the *Intervals* area and select the **+ Add Boundary Value** popup menu item.

To remove a value, right-click the value within the *Intervals* area and select the **- Delete Value** popup menu item.

Use the **↑ Move Boundary Value up** and the **↓ Move Boundary Value down** popup menu items to reorder the intervals.

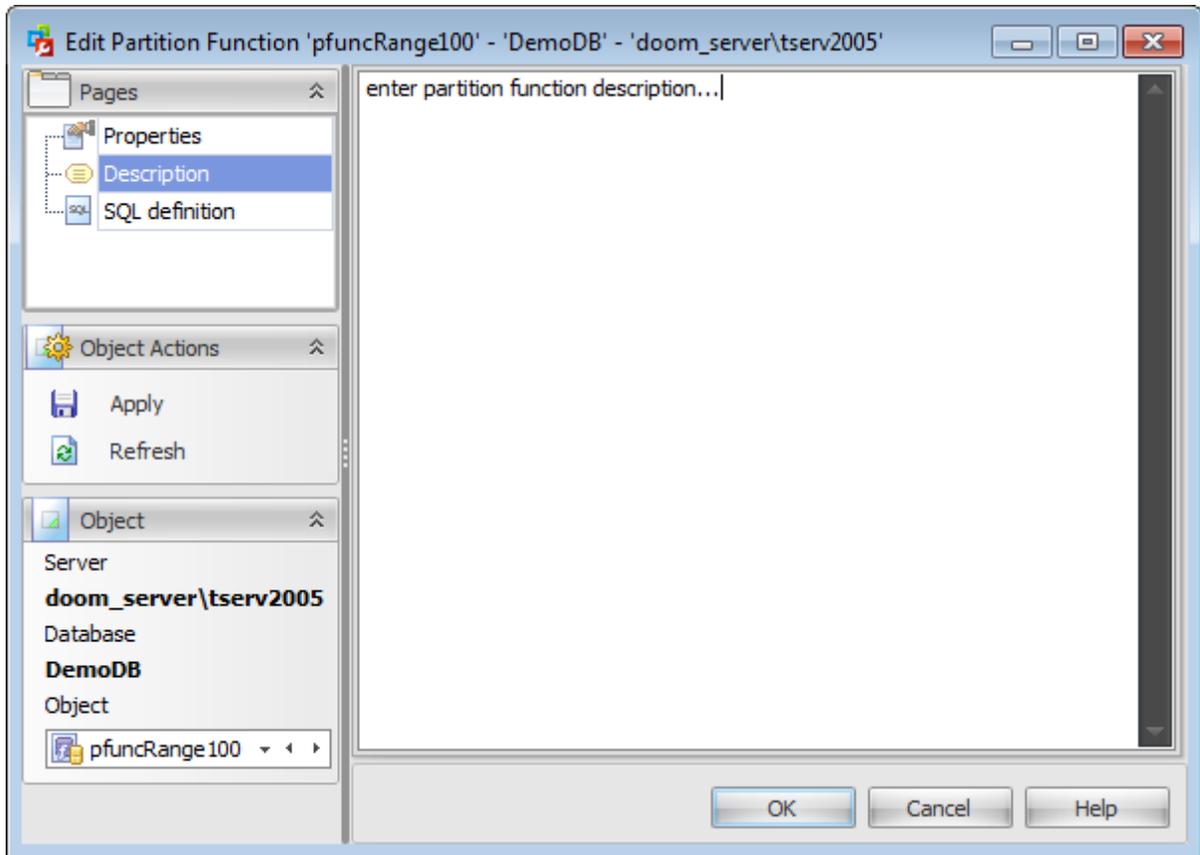
To add a partition scheme which will use the current partition function, right-click within the *Intervals* area and select the **+ Create Partition Scheme...** popup menu item.

To drop the partition scheme which uses the current partition function, right-click within

the *Intervals* area and select **Delete Partition Scheme** popup menu item.

### 5.9.1.2 Description

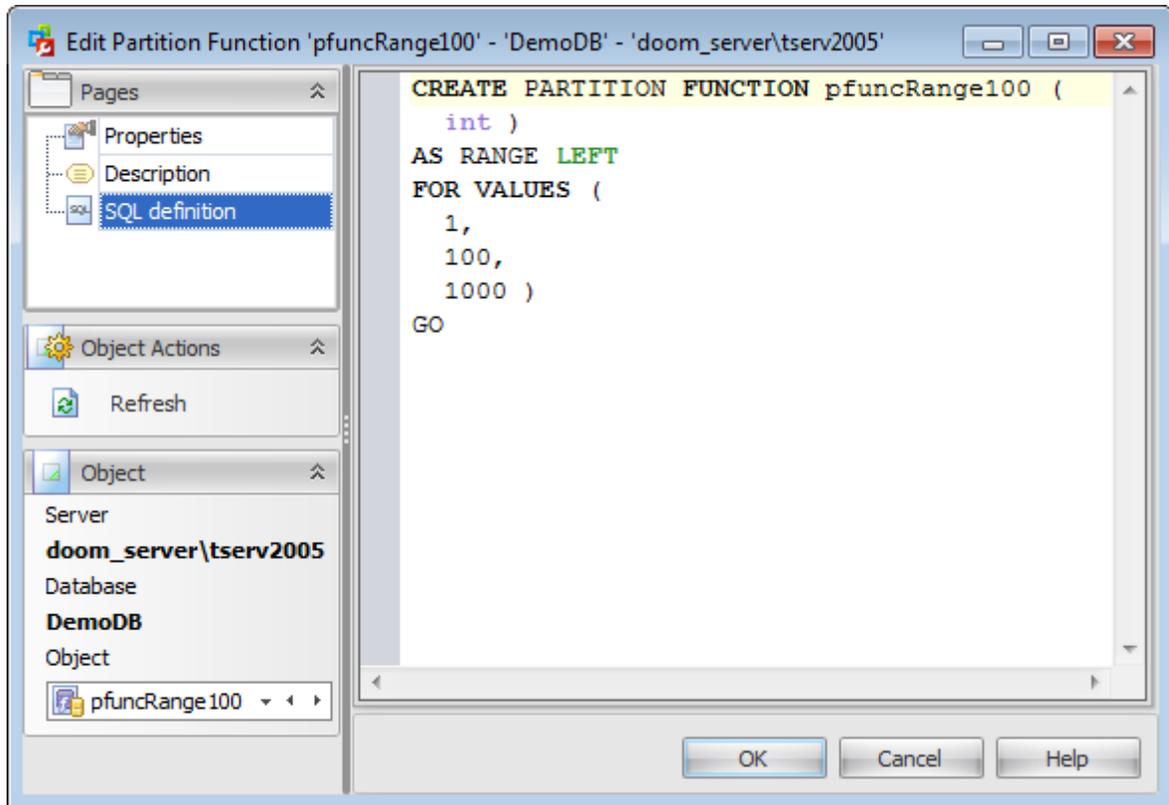
The **Description** tab allows you to view and edit the comment for the partition function (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.9.1.3 SQL Definition

The **SQL definition** tab of the **Partition Function Editor** window allows you to view the object DDL.

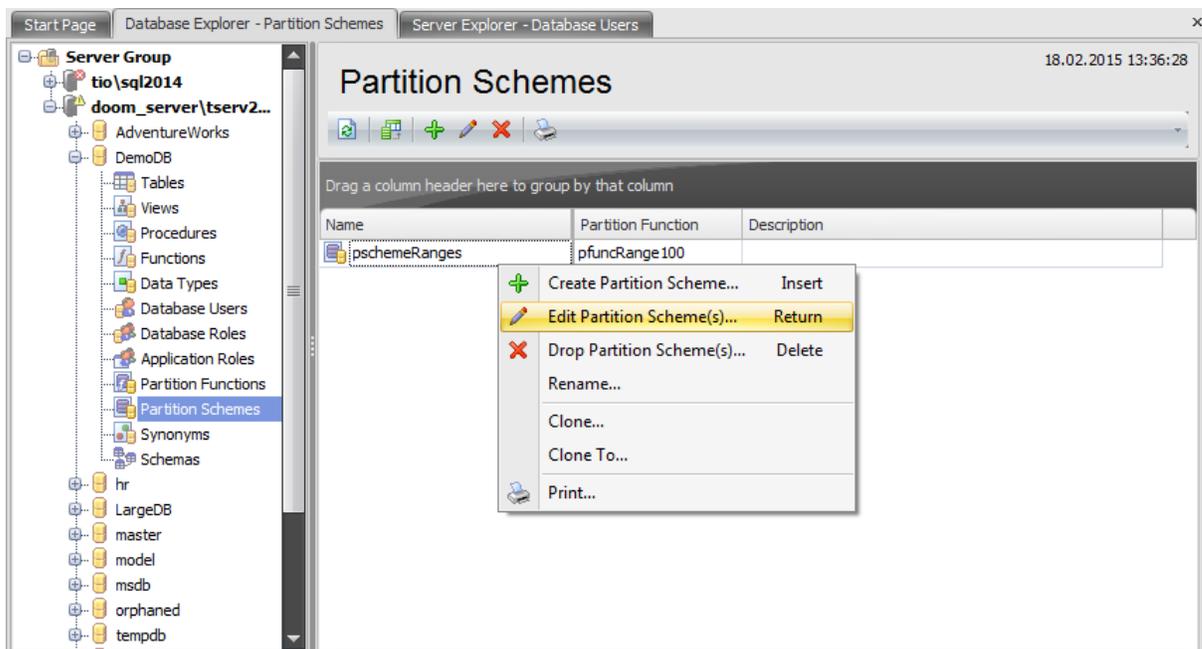


## 5.10 Partition Schemes

The **Partition Schemes** section of the Database Explorer is used for managing partition schemes in the selected database.

Use the context menu to:

- + [create](#) a new partition scheme
-  [edit](#) the selected partition scheme(s)
- drop selected partition scheme(s)
  - rename the selected partition scheme
  - create the object copy on the same server
  - clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database partition schemes as a grid with the following columns: *Name*, *Partition Function*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Partition Scheme in the [Partition Scheme Editor](#), double-click it in the grid. Alternatively, you can right-click the partition scheme alias and select the **Edit Partition Scheme(s)** context menu item.

### 5.10.1 Partition Scheme Editor

A **Partition scheme** maps the partitions produced by a [partition function](#) to a set of filegroups that you define. A partition scheme can use only one partition function. However, a partition function can participate in more than one partition scheme.

**Partition Scheme Editor** allows you to define partition scheme properties. It opens automatically when you create a new partition scheme and is available on editing an existing one.

To open a partition scheme in **Partition Scheme Editor**, double-click it in the [Partition Schemes](#) list.

- [Properties](#)
- [Description](#)
- [SQL Definition](#)

### 5.10.1.1 Properties

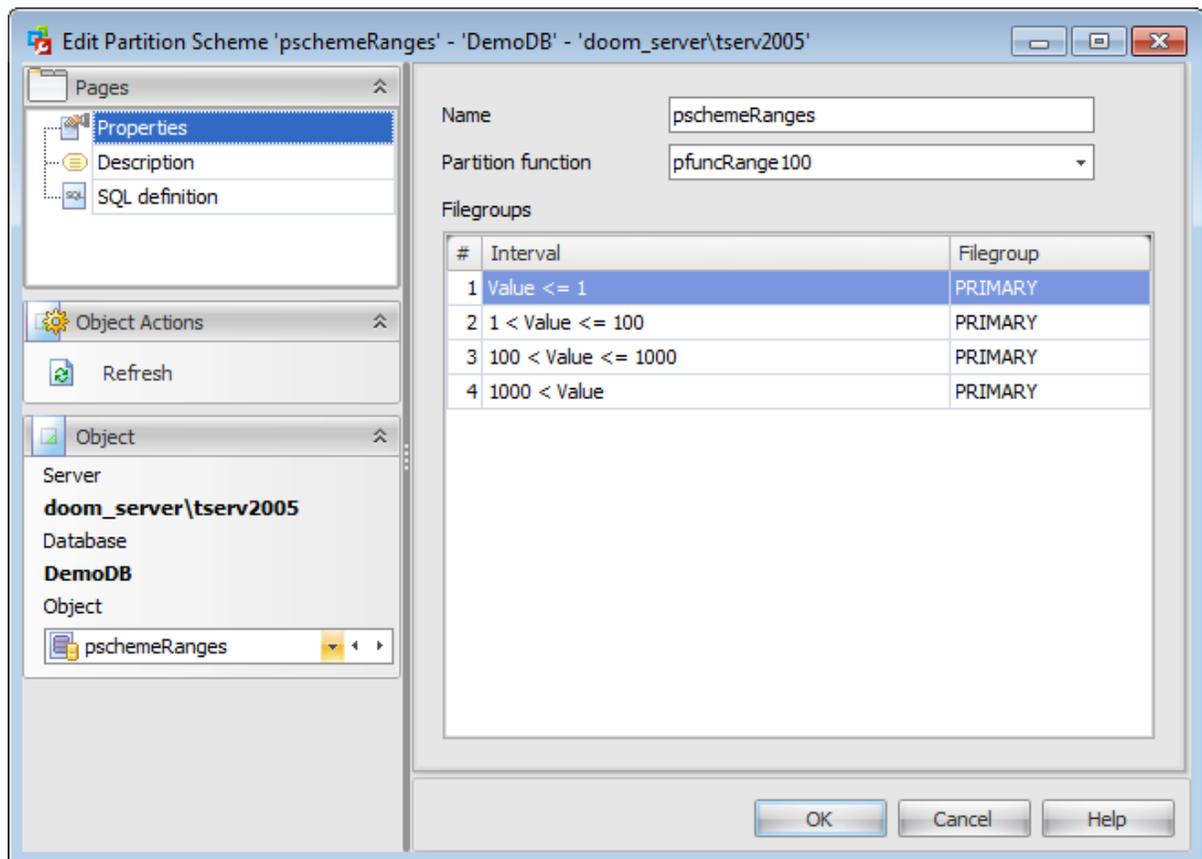
Use the **Properties** tab of **Partition Scheme Editor** to create/edit a partition scheme and specify its properties.

#### Name

Enter a name for the new partition scheme, or modify the name of the partition scheme being edited.

#### Partition function

Use the drop-down list to select the name of the [partition function](#) that will use the partition scheme.



#### Filegroups

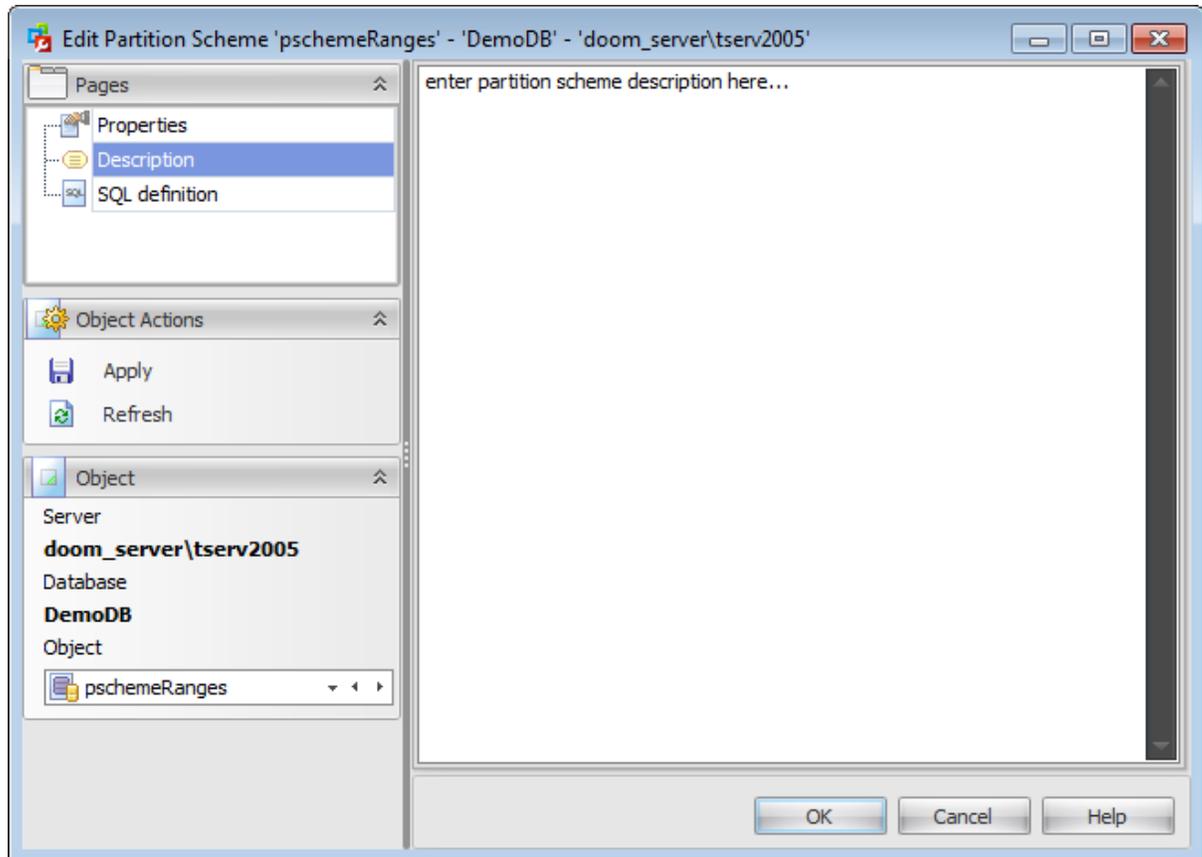
Partitions created by the [partition function](#) are mapped to the filegroups specified in the partition scheme.

Select the filegroup name in the **Filegroup** drop-down list for each of the intervals specified by the [partition function](#). If *PRIMARY* is selected, the partition is stored in the primary filegroup.

**Hint:** Filegroups can be created on the [Filegroups](#) page of the [Database Editor](#) dialog.

### 5.10.1.2 Description

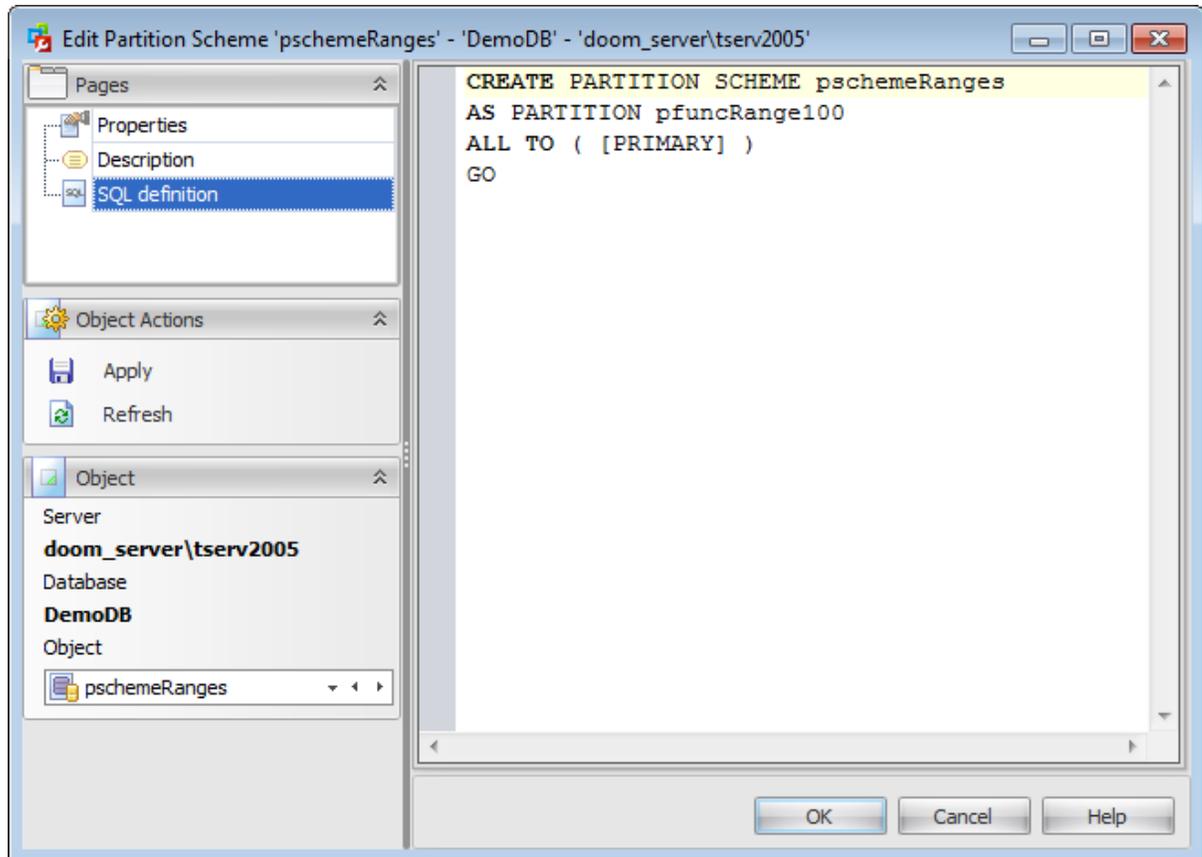
The **Description** tab allows you to view and edit the comment for the partition scheme (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty 'MS\_Description'*.

### 5.10.1.3 SQL Definition

The **SQL definition** tab of the **Partition Scheme Editor** window allows you to view the object DDL.

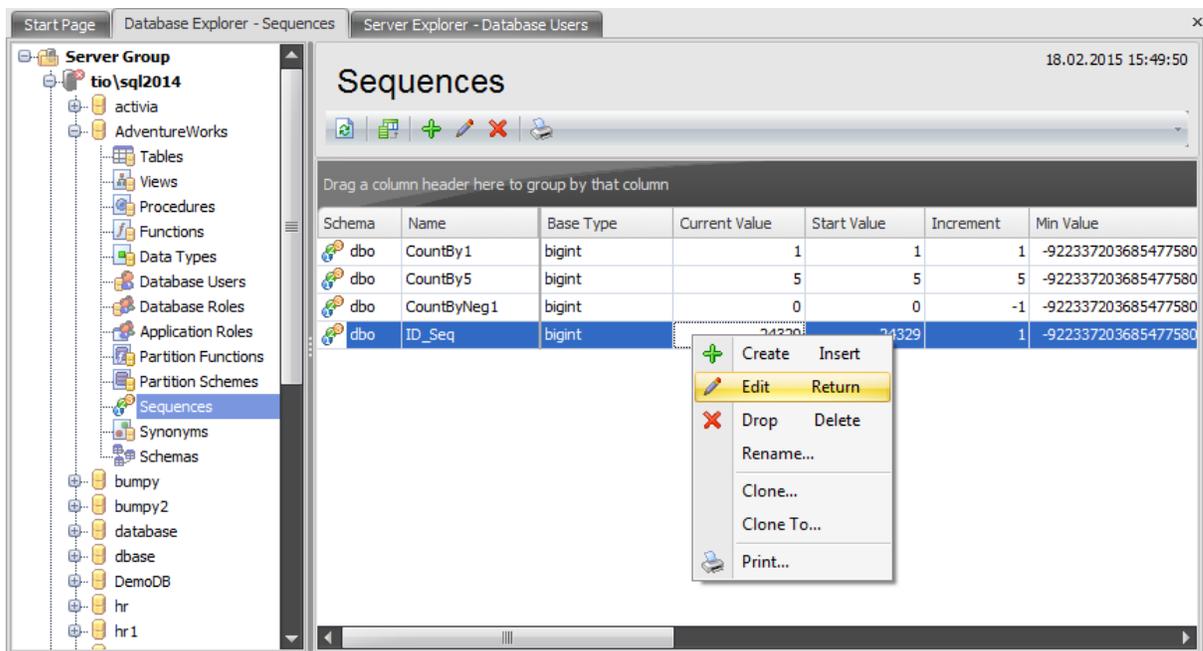


## 5.11 Sequences

The **Sequences** section of the Database Explorer is used for managing sequences in the selected database.

Use the context menu to:

-  [create](#) a new sequence
-  [edit](#) the selected sequence(s)
-  drop selected sequence(s)
- rename the selected sequence
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database sequences as a grid with the following columns: *Schema, Name, Base Type, Current Value, Start Value, Increment, Min Value, Max Value, Cyclic, Cache, Exhausted, Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Sequence in the [Sequence Editor](#), double-click it in the grid. Alternatively, you can right-click the sequence alias and select the **Edit** context menu item.

### 5.11.1 Sequence Editor

A **Sequence** is a database object from which multiple users may generate unique integers. You can use sequences to automatically generate primary key values. When a sequence number is generated, the sequence is incremented, independently of the transaction committing or rolling back. If two users concurrently increment the same sequence, then the sequence numbers each user acquires may have gaps, because sequence numbers are being generated by the other user. One user can never acquire the sequence number generated by another user. After a sequence value is generated by one user, that user can continue to access that value regardless of whether the sequence is incremented by another user.

**Sequence Editor** allows you to define sequence properties. It opens automatically when you create a new sequence and is available on editing an existing one (see [Create sequence](#) and [Edit sequence](#) for details).

To open a sequence in the **Sequence Editor**, double-click it in the [Sequences](#) list.

- [Properties](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.11.1.1 Properties

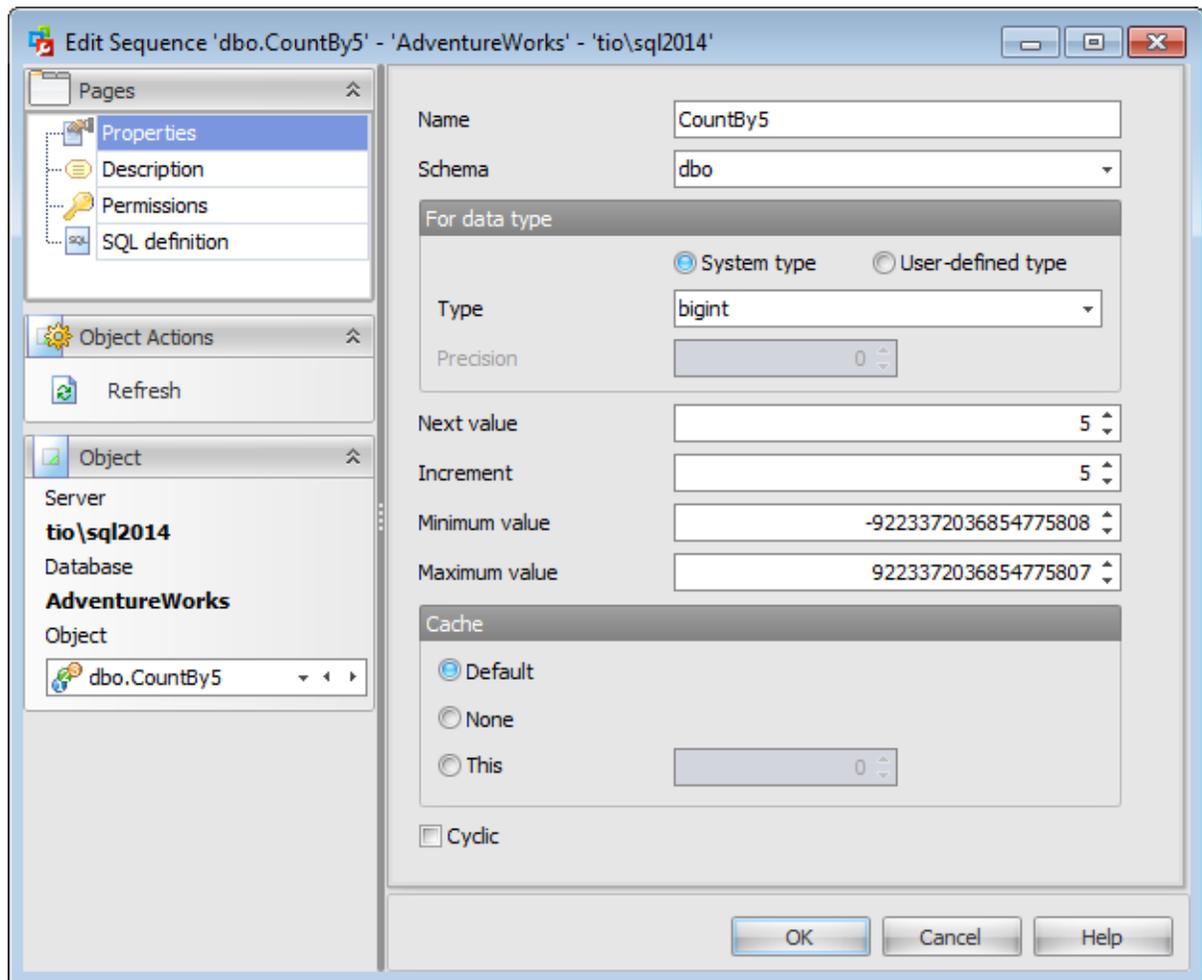
Use the **Properties** tab of **Synonym Editor** to create/edit a synonym and specify its properties.

#### Name

Specify the name of the sequence to be created, or view the name of the sequence being edited. Note that the sequence name must be unique within its schema.

#### Schema

Use the drop-down list to select the [schema](#) in which the sequence is created.



#### For data type

Select whether to create a sequence for the  *System type* or the  [User-defined type](#).

#### Type

Specify the type of generated sequence numbers.

#### Precision

Specify the size value (for the *decimal* type).

**Next value**

Specify the first sequence number to be generated. Use this setting to start an ascending sequence at a value greater than its minimum or to start a descending sequence at a value less than its maximum.

**Increment**

Specify which value is added to the current sequence value to create a new value. A positive value will make an ascending sequence, a negative one - a descending sequence. The default value is 1.

**Minimum value**

Specify the minimum value of the sequence.

**Maximum value**

Specify the maximum value for the sequence.

**Cache**

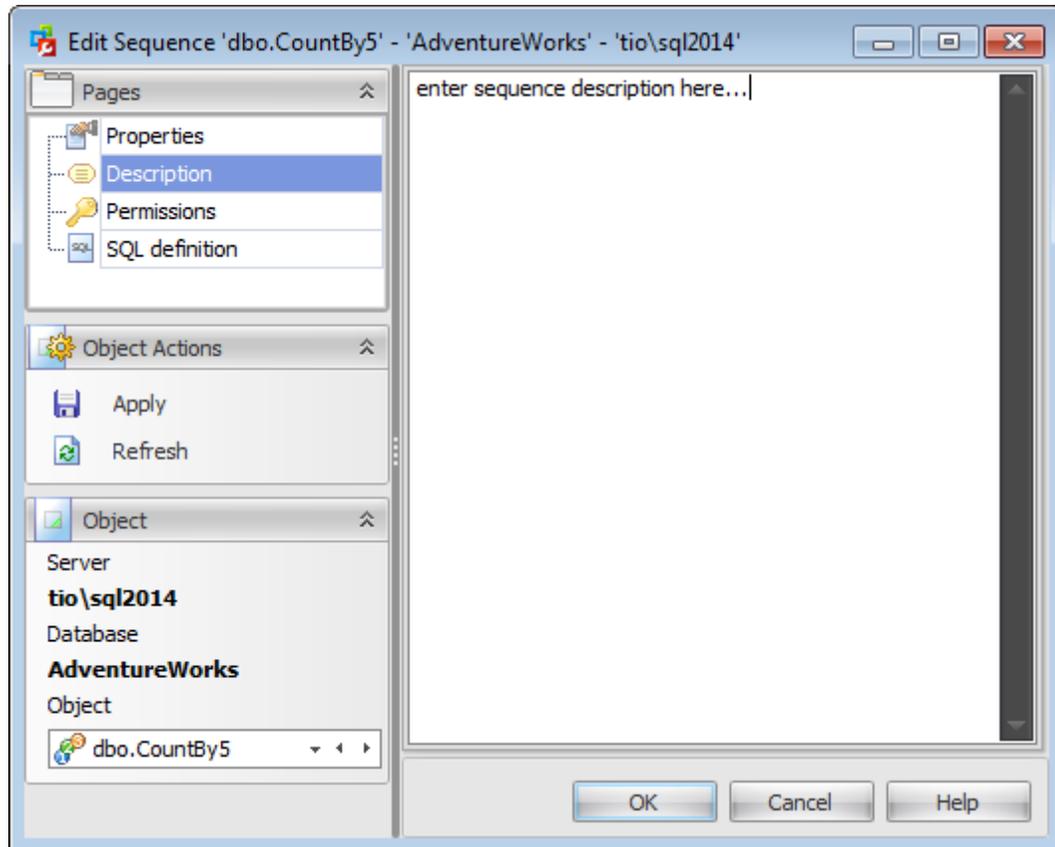
Specify how many sequence numbers are to be preallocated and stored in memory for faster access in the  *This* field or select  *Default* or  *None*.

 **Cyclic**

This option indicates that the sequence continues to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates its minimum value. After a descending sequence reaches its minimum, it generates its maximum value.

### 5.11.1.2 Description

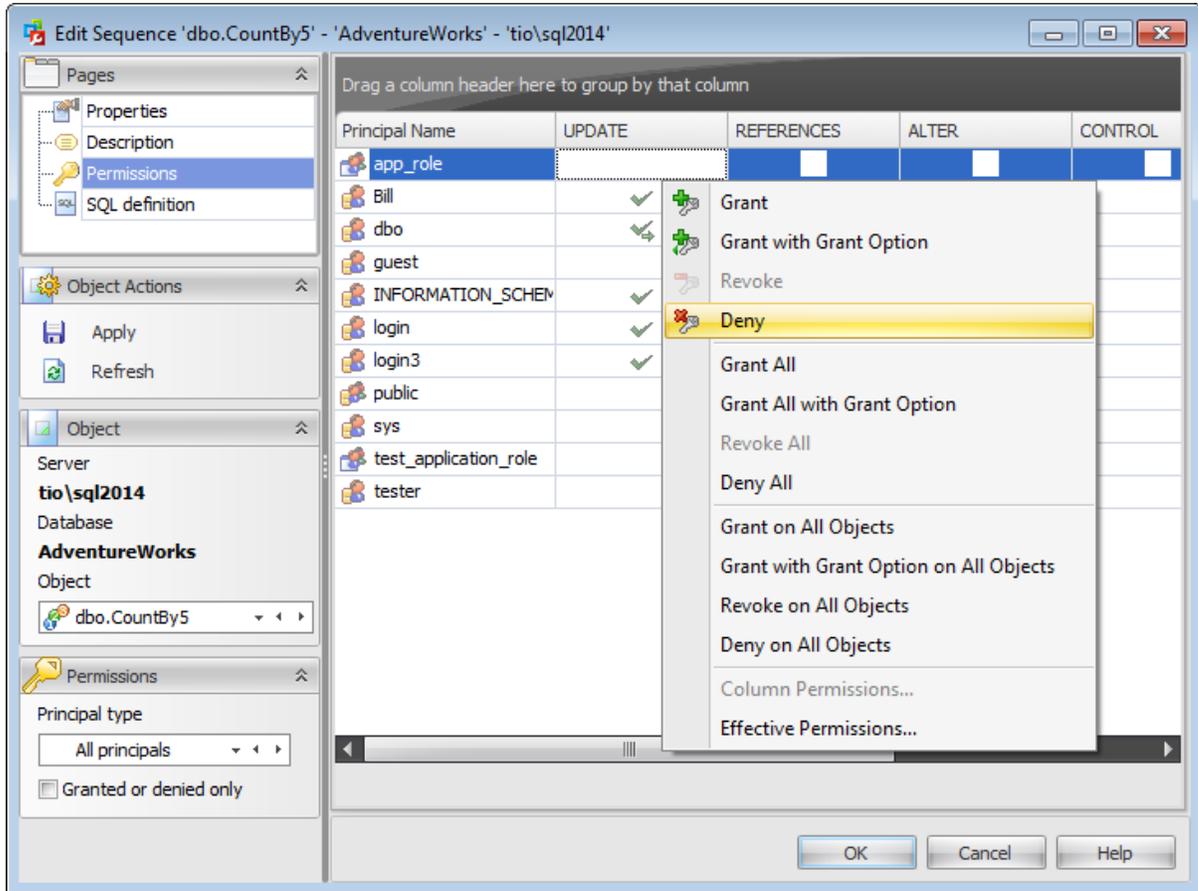
The **Description** tab allows you to view and edit the comment for the sequence (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

### 5.11.1.3 Permissions

The **Permissions** tab of the **Sequence Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

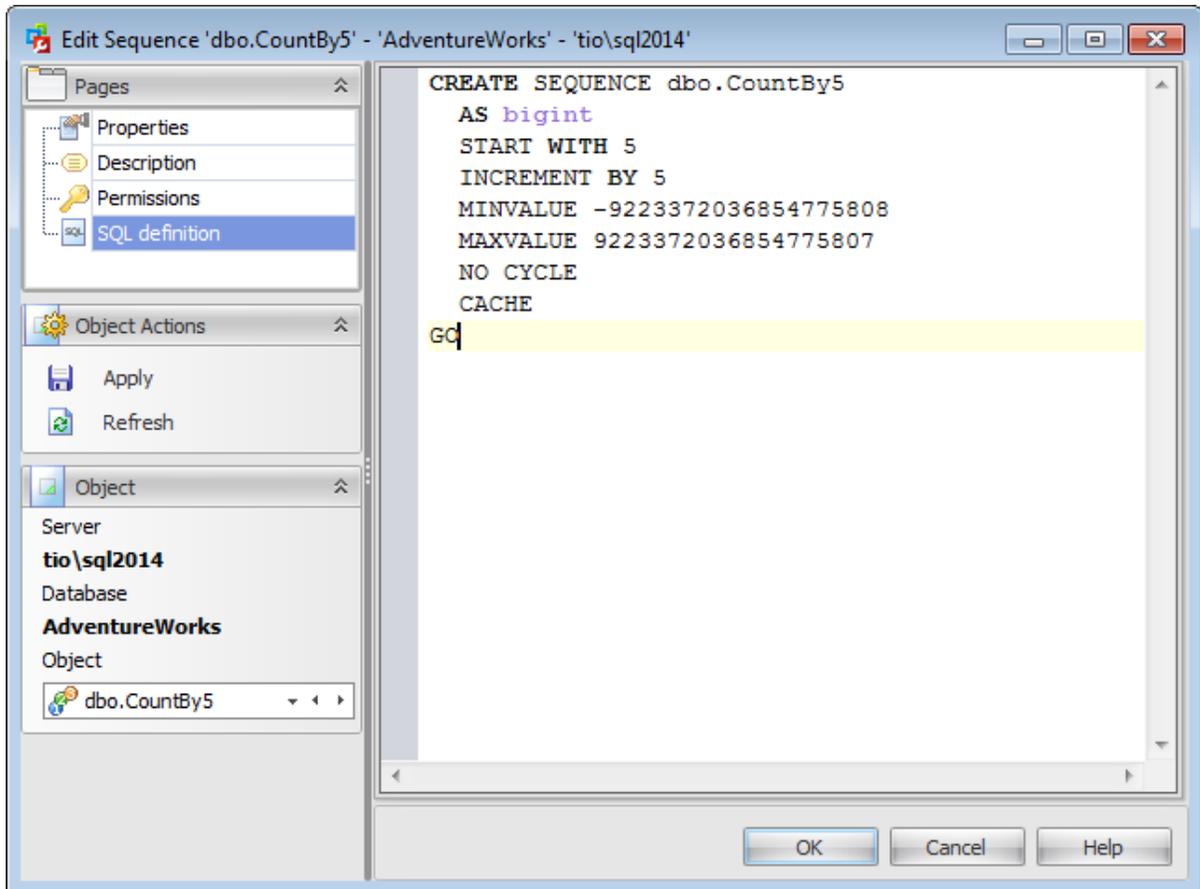
- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).

#### 5.11.1.4 SQL Definition

The **SQL definition** tab of the **Sequence Editor** window allows you to view the object DDL.

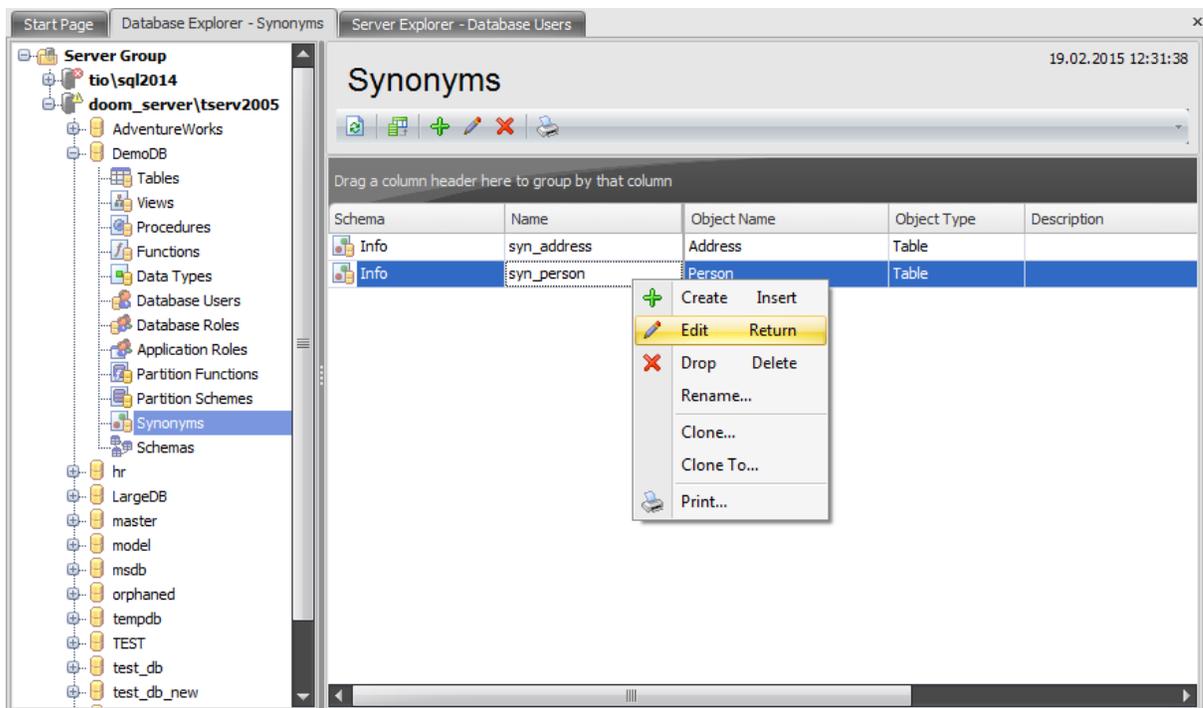


## 5.12 Synonyms

The **Synonyms** section of the Database Explorer is used for managing synonyms in the selected database.

Use the context menu to:

- + [create](#) a new synonym
-  [edit](#) the selected synonym(s)
- drop selected synonym(s)
  - rename the selected synonym
  - create the object copy on the same server
  - clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database synonyms as a grid with the following columns: *Schema*, *Name*, *Object Name*, *Object Type*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Synonym in the [Synonym Editor](#), double-click it in the grid. Alternatively, you can right-click the synonym alias and select the **Edit** context menu item.

### 5.12.1 Synonym Editor

You can use **Synonyms** in place of their referenced base object in a number of SQL statements and expression contexts. When working with synonyms in these contexts, the base object is affected. For example, if you have a synonym that references a base object that is a table, when you insert a row into the synonym, you are actually inserting a row into the referenced table.

**Synonym Editor** allows you to define synonym properties. It opens automatically when you create a new synonym and is available on editing an existing one.

To open a synonym in **Synonym Editor**, double-click it in the [Synonyms](#) list.

- [Properties](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.12.1.1 Properties

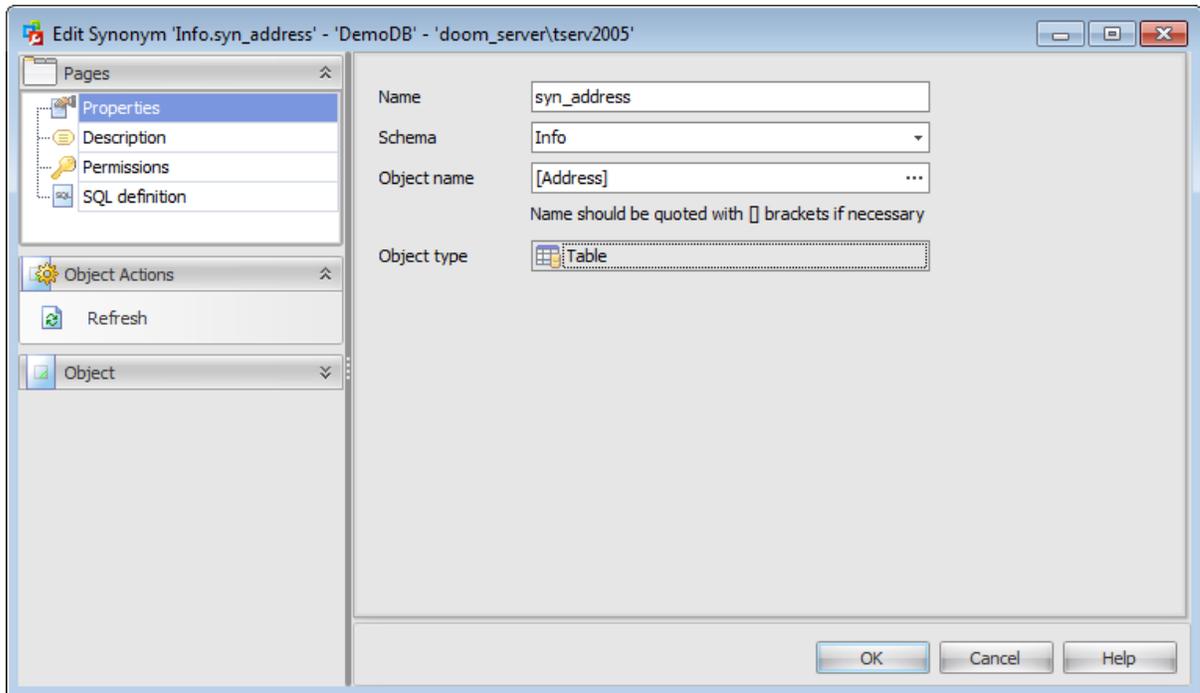
Use the **Properties** tab of the **Synonym Editor** to create/edit a synonym and specify its properties.

#### **Name**

Enter a name for the new synonym, or modify the name of the synonym being edited.

#### **Schema**

Use the drop-down list to select the [schema](#) in which the synonym is created.

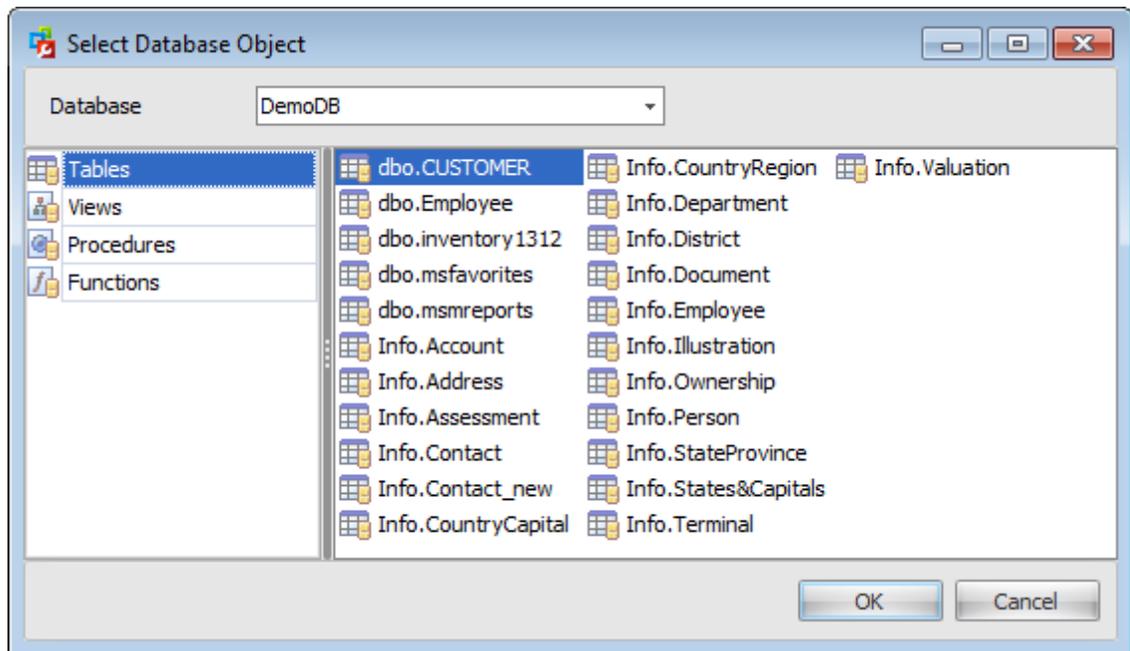


#### **Object name**

This box displays the base object that the synonym references. To change the object, click the ellipsis button.

The object name can be edited manually if necessary.

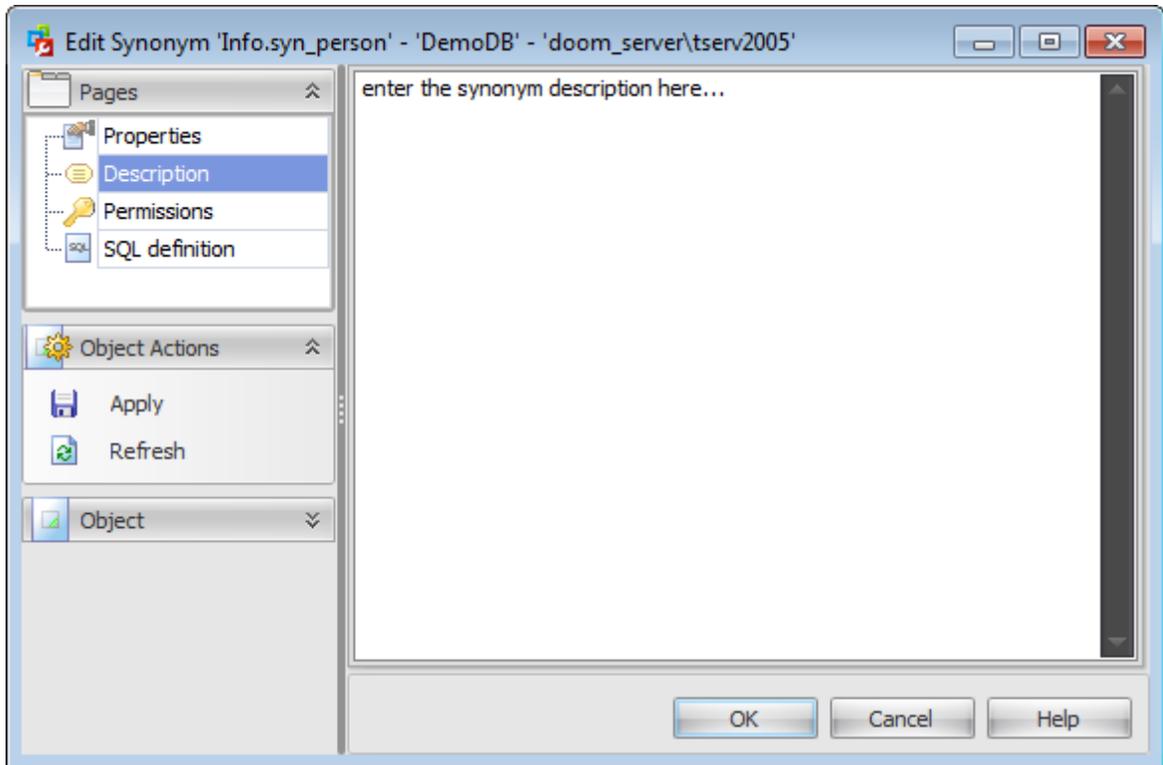
The **Select Database Object** dialog allows you to specify a database object to use as the referenced object.



After you have selected the necessary object, its type will be displayed in the **Object type** field.

### 5.12.1.2 Description

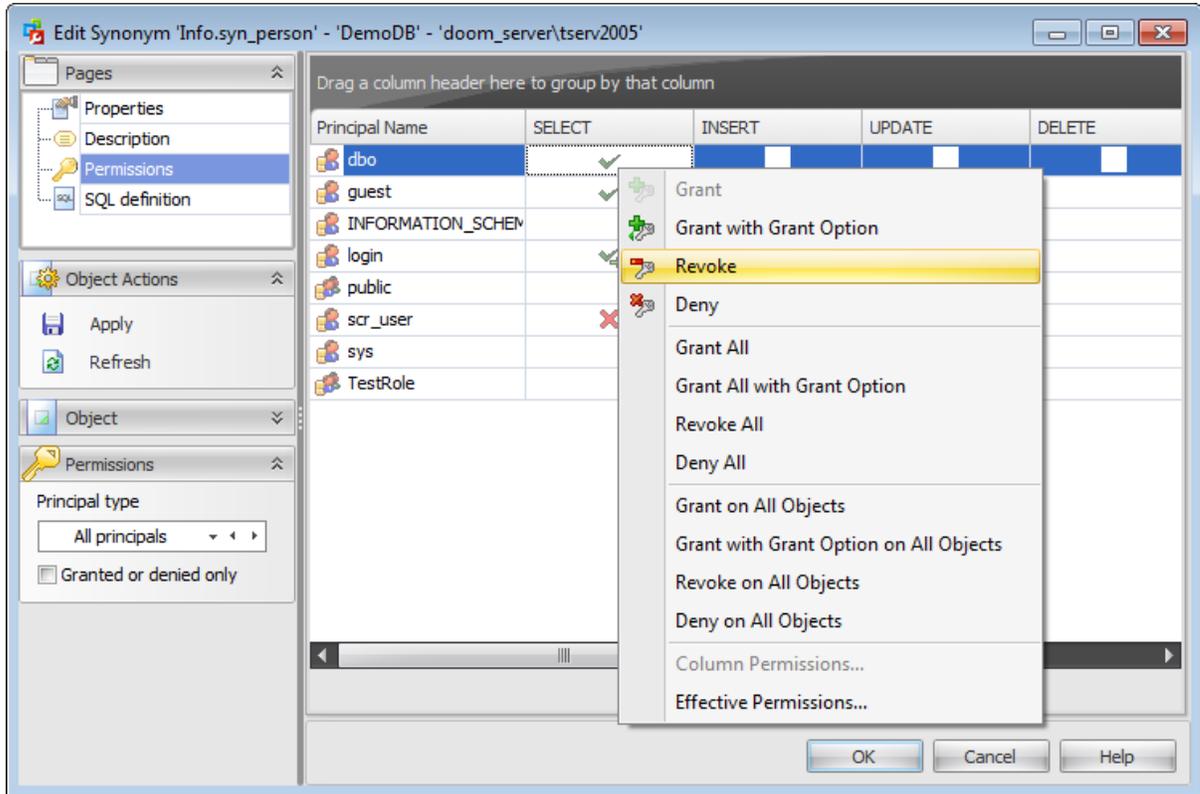
The **Description** tab allows you to view and edit the comment for the synonym (optional).



**Note:** Changing object description is performed by means of `sp_addextendedproperty 'MS_Description'`.

### 5.12.1.3 Permissions

The **Permissions** tab of the **Synonym Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

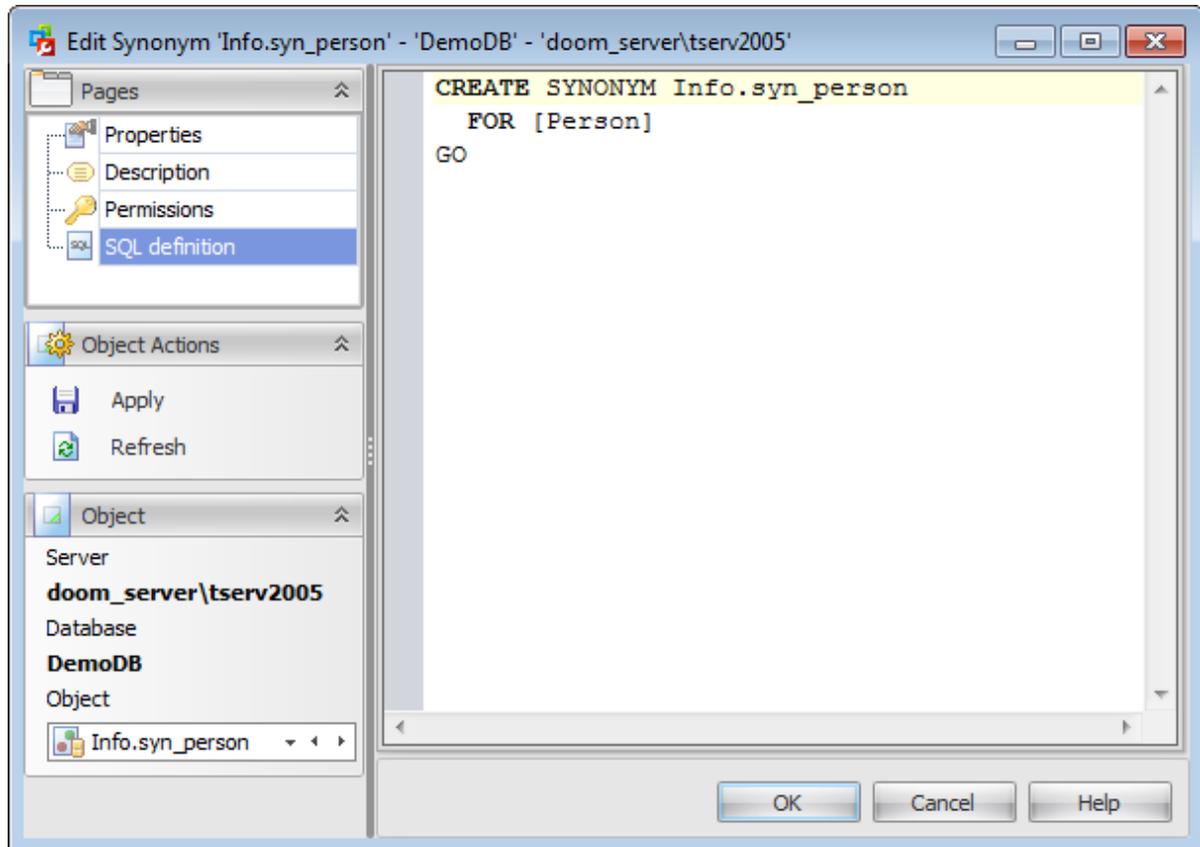
- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).

#### 5.12.1.4 SQL Definition

The **SQL definition** tab of the **Synonym Editor** window allows you to view the object DDL.

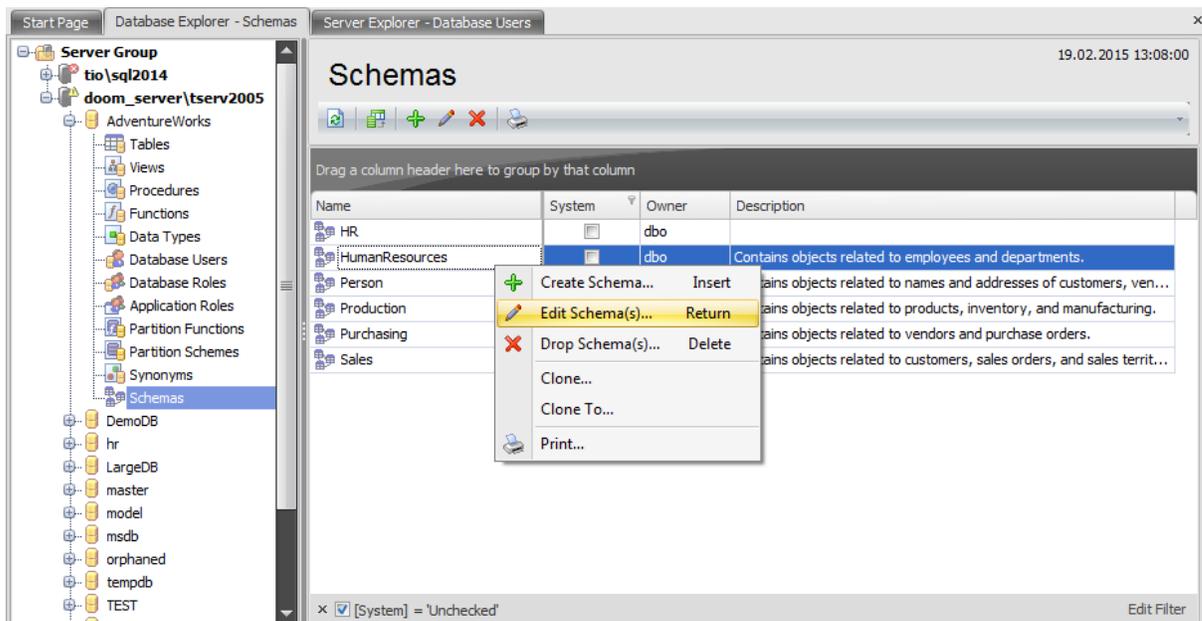


## 5.13 Schemas

The **Schemas** section of the Database Explorer is used for managing schemas in the selected database.

Use the context menu to:

-  [create](#) a new schema
-  [edit](#) the selected schema(s)
-  drop selected schema(s)
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report



The list displays the existing database schemas as a grid with the following columns: *Name*, *System*, *Owner*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Schema in the [Schema Editor](#), double-click it in the grid. Alternatively, you can right-click the schema alias and select the **Edit Schema(s)** context menu item.

### 5.13.1 Schema Editor

A **Schema** is essentially a namespace: it can be defined as a collection of database objects that form a single namespace. A namespace is a set in which each element has a unique name.

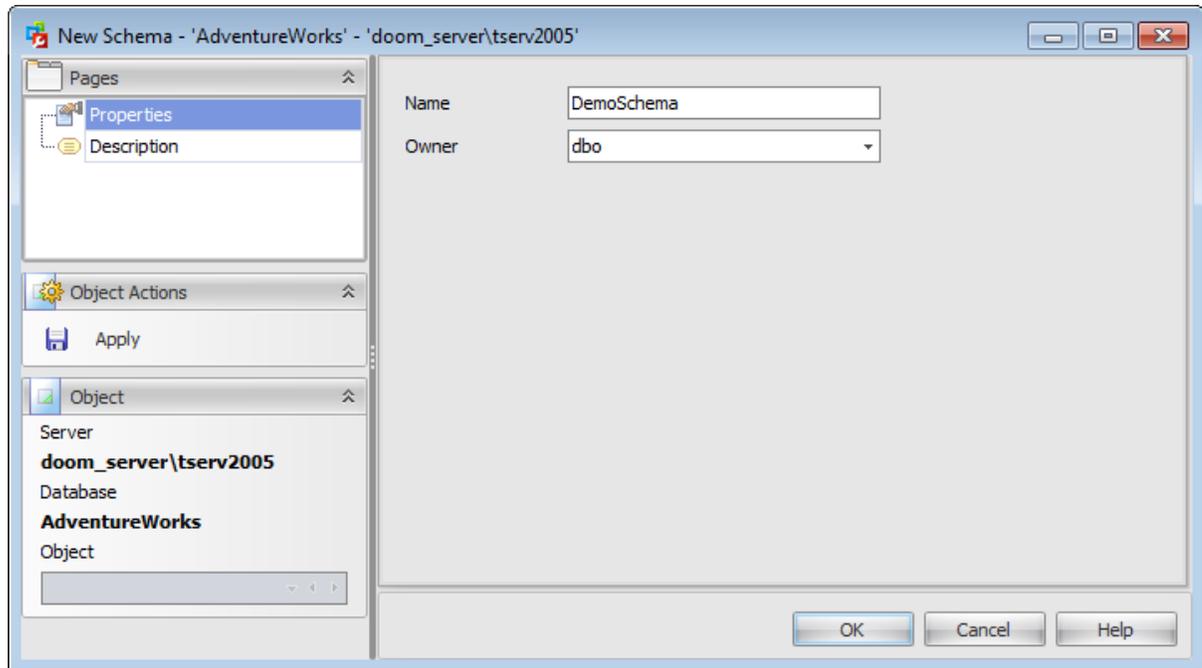
**Schema Editor** allows you to manage SQL Server schemas efficiently. It opens automatically when you create a new schema and is available on editing an existing one.

To open a schema in **Schema Editor**, double-click it in the [Schemas](#) list.

- [Properties](#)
- [Description](#)
- [Permissions](#)
- [SQL Definition](#)

### 5.13.1.1 Properties

Use the **Properties** tab of **Schema Editor** to create/edit a schema and specify its properties.



#### **Name**

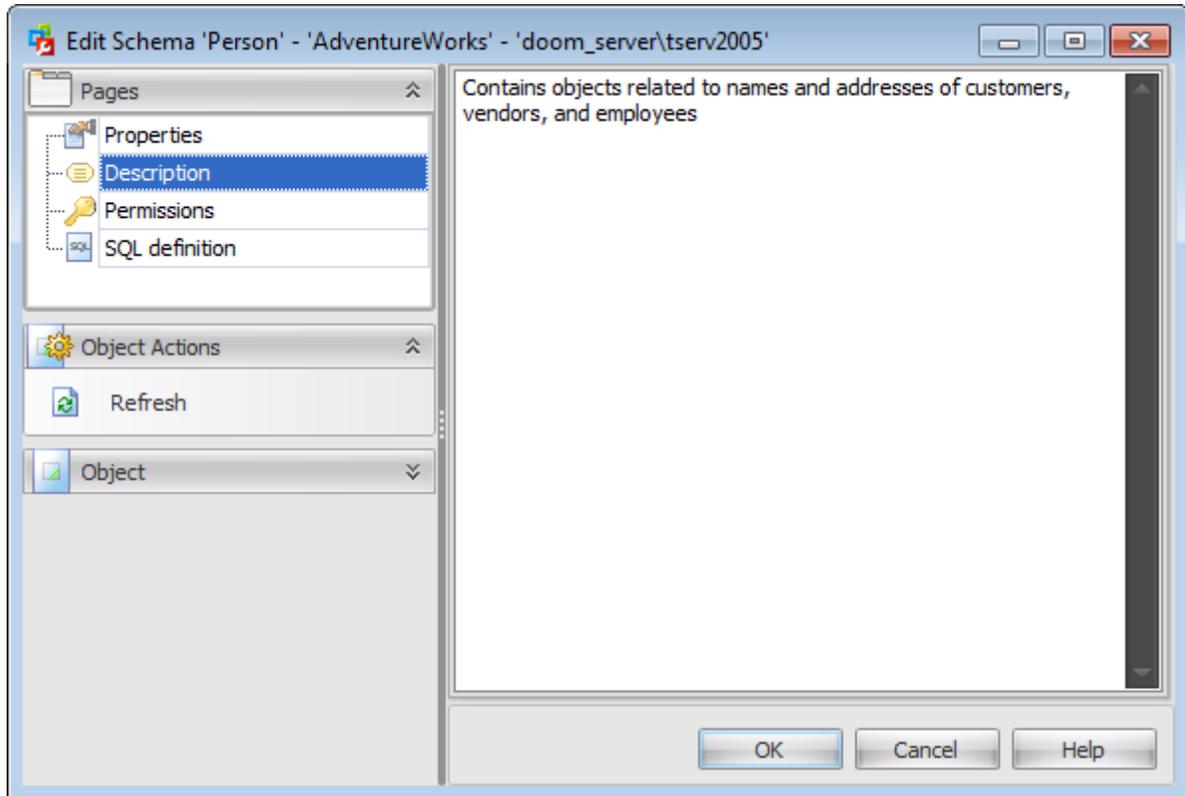
Specify the name by which the schema is identified within the database.

#### **Owner**

Specify the name of the database-level principal ([user](#) or [role](#)) that will own the schema. This principal may own other schemas, and may not use the current schema as its default schema.

### 5.13.1.2 Description

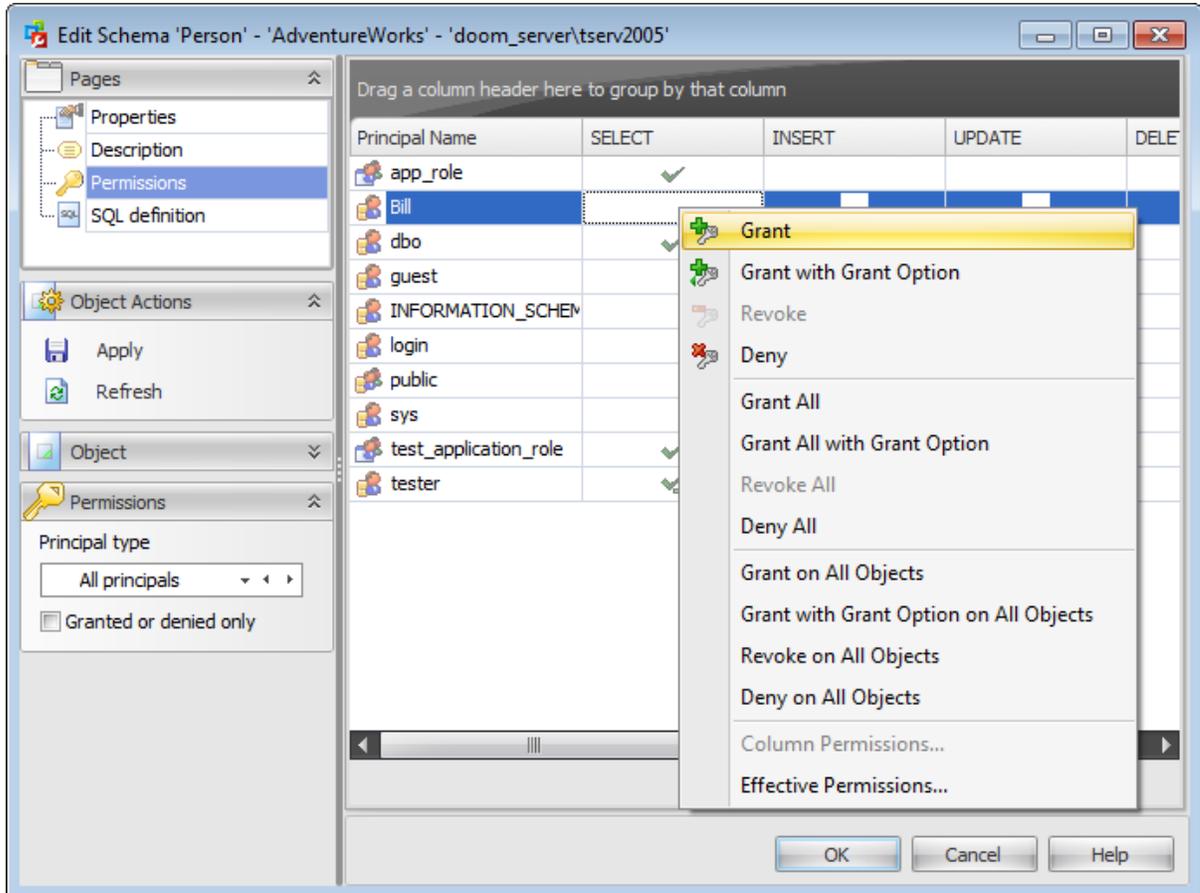
The **Description** tab allows you to view and edit the comment for the schema (optional).



**Note:** Changing object description is performed by means of *sp\_addextendedproperty* 'MS\_Description'.

### 5.13.1.3 Permissions

The **Permissions** tab of the **Schema Editor** window allows you to set permissions to [users](#), [roles](#) and [application roles](#).



#### Principal Name

This column contains the name of the database-level principals ([users](#), [database roles](#) or [application roles](#)).

**Note:** If you double-click the object name then the [User Editor](#) / [Database Role Editor](#) / [Application Role Editor](#) appears.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

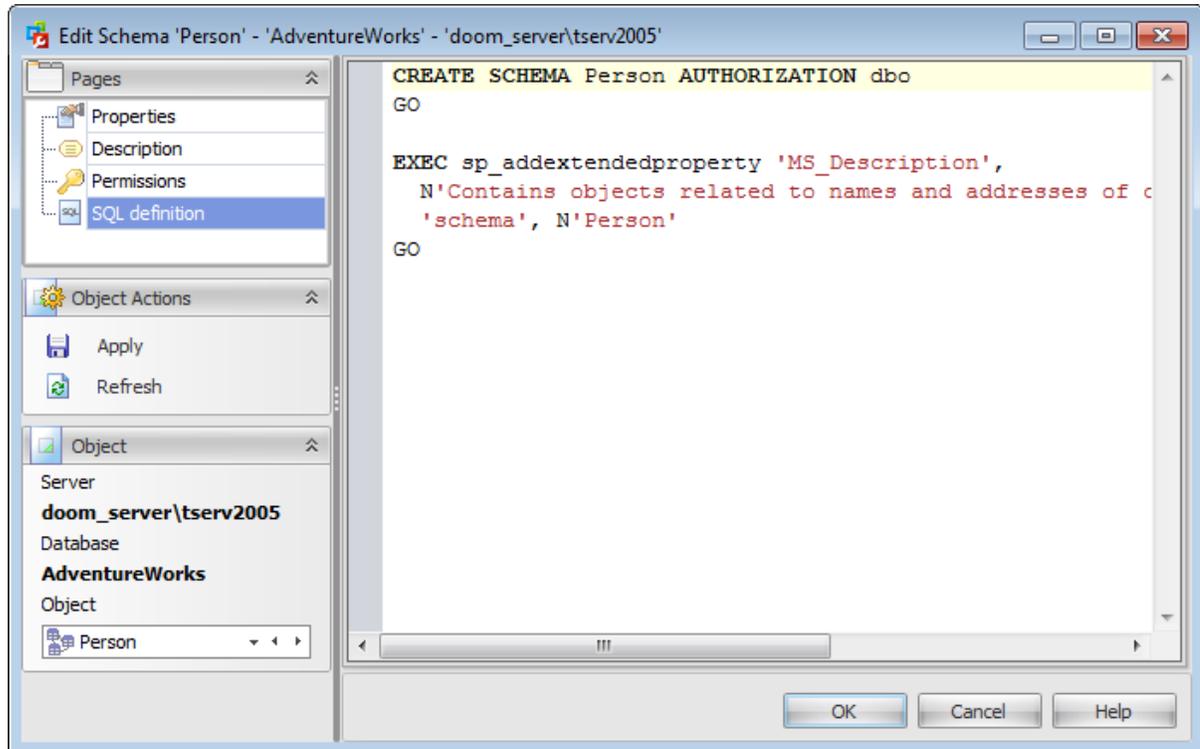
- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Using the context menu you can access the dialog for managing [effective permissions](#).

#### 5.13.1.4 SQL Definition

The **SQL definition** tab of the **Schema Editor** window allows you to view the object DDL.





**Part**

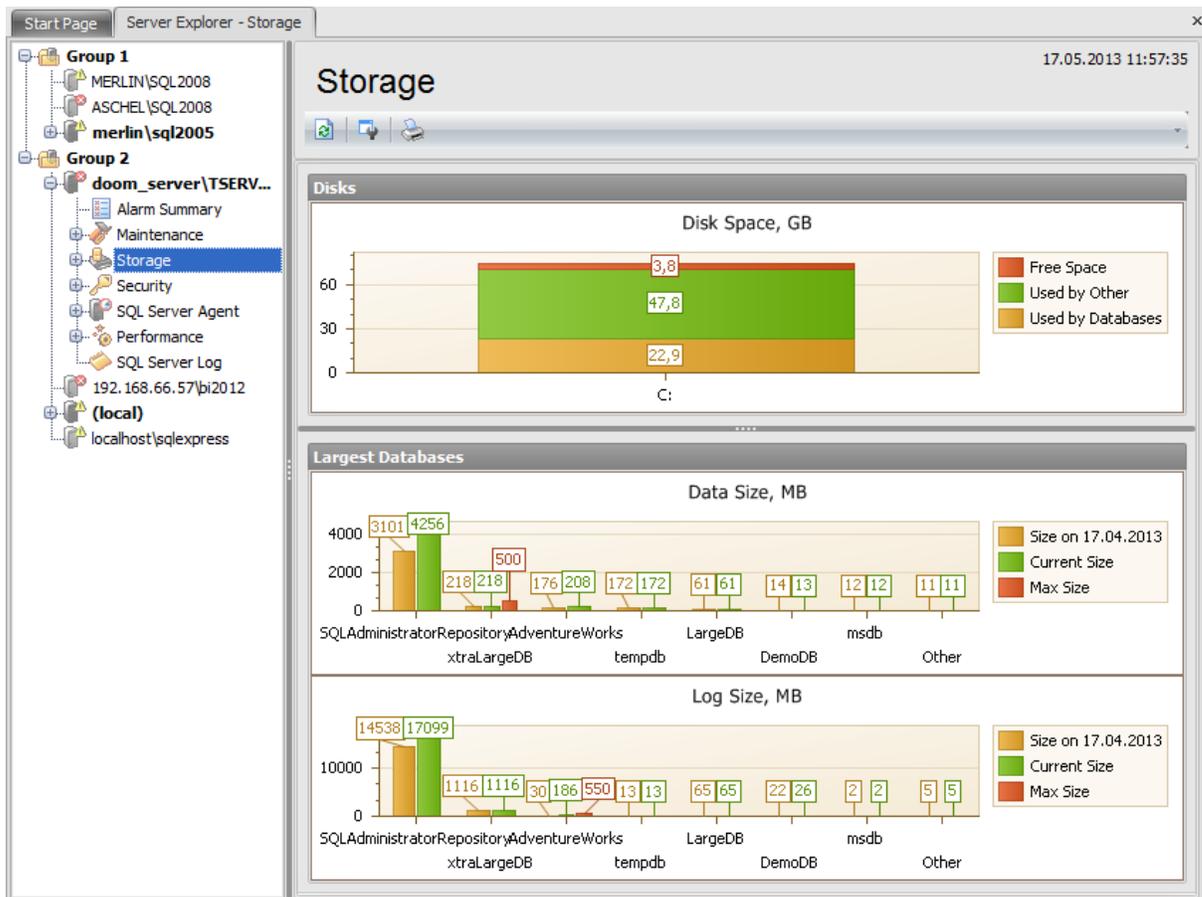


## 6 Storage

The **Storage** section of the [Server Explorer](#) displays diagrams built on the basis of the information about disks space and databases size.

The following nodes are situated in this section:

- [Databases](#)
- [Database Snapshots](#)
- [Disk Space Usage](#)
- [Space Usage by Objects](#)



### Disk Space, GB

This area shows information about the disk space usage on the server. The number of bars in the bar chart depends on the number of disks on the server computer. Each bar is separated in three sections: *Disk Space Used by Databases*, *Disk Space Used by Other* (disk space used by data other than databases), *Free Disk Space*.

### Data Size, MB

This section provides information on the data files size of largest databases on the server. There are three bars for each databases: *Size on* (the data size that was a month ago), *Current Data File Size*, *Max Data File Size* (if it is specified).

**Log Size, MB**

This section provides information on the log files size of largest databases on the server. There are three bars for each databases: *Size on (the log size that was a month ago)*, *Current Log File Size*, *Max Log File Size* (if it is specified).

**Note:** If there is no information on the database size in the repository pertaining to this date, then the displayed size is the size on the nearest date to that.

---

**See also:**[Getting Started](#)[Servers](#)[Database Maintenance](#)[Security](#)[SQL Server Agent](#)[Performance](#)[SQL Server Log](#)[Query Editor](#)[Options](#)

## 6.1 Database Editor

The **Database Editor** dialog allows you to view/edit a number of properties which can be changed to optimize database performance.

To open the dialog, double-click the database alias in any list of databases or right-click it and select the  **Edit Database...** context menu item.

Lists of databases can be found in the following sections:

- [Backup](#)
- [Check Databases](#)
- [Rebuild/Reorganize indexes](#)
- [Storage | Databases](#)
- [Security](#)

A number of database-level options that determine the characteristics of the database can be set for each database. Only the system administrator, database owner, members of the **sysadmin** and **dbcreator** fixed [server roles](#) and **db\_owner** fixed [database roles](#) can modify these options. These options are unique to each database and do not affect other databases.

The list of **Database Editor** tabs is displayed on the [navigation bar](#):

- [Setting general options](#)
- [Managing filegroups](#)
- [Editing database options](#)
- [Configuring change tracking](#)
- [Viewing database properties](#)
- [Setting permissions](#)
- [Viewing SQL definition](#)

---

### See also:

[Databases](#)

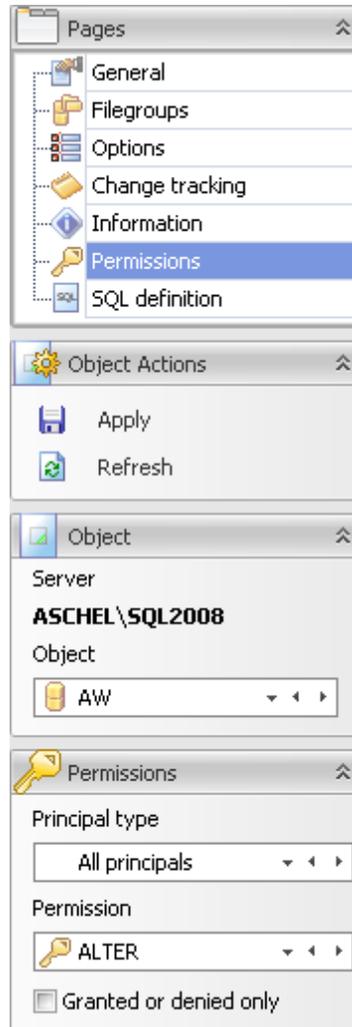
[Database Snapshots](#)

[Disk Space Usage](#)

[Space Usage by Objects](#)

## 6.1.1 Using Navigation bar

The **Database Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



### Pages

This group contains the list of **Database Editor** tabs which are clusters of properties that can be configured.

### Object Actions

 apply changes

 refresh the content of the active tab

### Object

#### Server

This field displays the name of the server where the edited object resides.

#### Object

This field displays the database that is currently being edited. However, you can change the edited database by using the drop-down list of databases or by clicking the arrow-left or arrow-right buttons.

### Permissions

This group of options appears on setting permissions.

#### **Principal type**

Use the drop-down list or arrow-left or arrow-right buttons to select the type of the principal which you want to grant permissions.

#### **Permission**

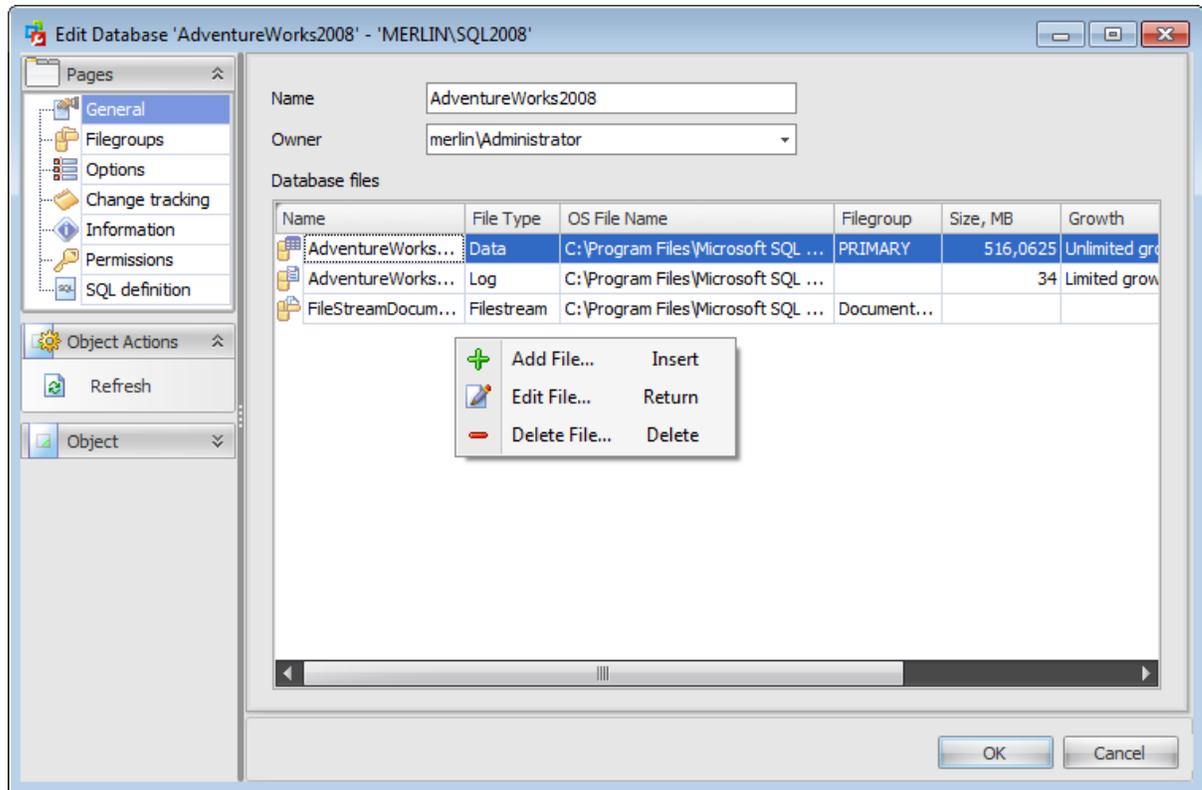
Select the permission which you want to grant to principals of the selected type by using the drop-down list of databases or by clicking the arrow-left or arrow-right buttons.

#### **Granted or denied only**

Check this option for displaying only those principals that have granted or denied permissions.

## 6.1.2 Setting general options

The **General** section of the **Edit Database** window allows you to view and edit main database properties.



### Name

Edit the database name.

### Owner

Select a database owner from the drop-down list. The database owner is a user that has implied permissions to perform all activities in the database.

### Database files

SQL Server databases have three types of files:

- Primary data files

The primary data file is the starting point of the database and points to the other files in the database. Every database has one primary data file. The recommended file name extension for primary data files is *\*.mdf*.

- Secondary data files

Secondary data files make up all the data files, other than the primary data file. Some databases may not have any secondary data files, while others have several secondary data files. The recommended file name extension for secondary data files is *\*.ndf*.

- Log files

Log files hold all the log information that is used to recover the database. There must be at least one log file for each database, although there can be more than one. The recommended file name extension for log files is *\*.ldf*.

In SQL Server, the locations of all the files in a database are recorded in the primary file of the database.

### Database files

This list displays information about all data files.

The **Name** is a logical name used to refer to the physical file in all Transact-SQL statements. The logical file name must conform to the rules for SQL Server identifiers and must be unique within the database.

The **Filegroup** is the named collection of files that is used to help with data placement and administrative tasks.

The **File Type** defines the type of an operating-system file.

The **OS File Name** is the name of the physical file including the directory path. The path must follow the rules for operating system file names.

Double-click the  button to open the [File Properties](#) dialog.

### Size, MB

Use the spinner control to specify the original size of the selected database file (in megabytes).

### Filegroup

Use the drop-down list to specify a [file\\_group](#) the file belongs to. The *Primary* file group contains the primary data file and any other files not specifically assigned to another file group. All pages for the system tables are allocated in the primary file group. *User-defined* file groups are any other file groups specified by the user.

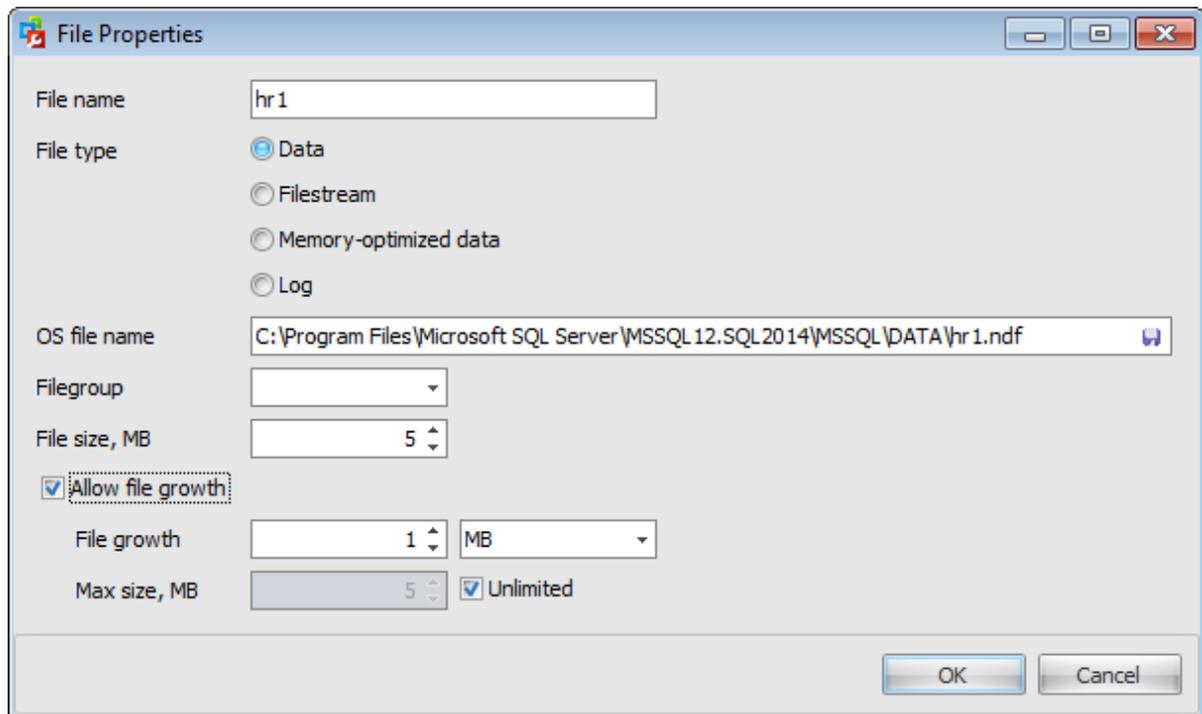
### Growth

This field shows whether the file size is limited or unlimited and the max size of the limited growth file.

Use the context menu to manage a list of database. It allows you to:

-  add a new database file
-  edit the selected database file
-  delete the selected database file

When adding a new file or editing the existing one the **File Properties** window appears. It allows you to enter or edit the database file parameters.



### File name

The **Name** is a logical name used to refer to the physical file in all Transact-SQL statements. The logical file name must conform to the rules for SQL Server identifiers and must be unique within the database.

### File type

Select one of the three types of files:

- Data*
- Filestream*
- Memory-optimized data*
- Log*

The **OS File Name** is the name of the physical file including the directory path. The path must follow the rules for operating system file names.

Click the  button to open the **Save File As** dialog where you can choose among files and directories stored on the SQL Server.

### Filegroup

Use the drop-down list to specify a [file group](#) the file belongs to. The *Primary* file group contains the primary data file and any other files not specifically assigned to another file group. All pages for the system tables are allocated in the primary file group. *User-defined* file groups are any other file groups specified by the user.

### File size, MB

Use the spinner control to specify the original size of the selected database file (in megabytes).

**Allow file growth**

Check this option to enable automatic file growth.

**File growth**

Specify a **file growth** increment (in *megabytes* or in *percent*).

**Max size, MB**

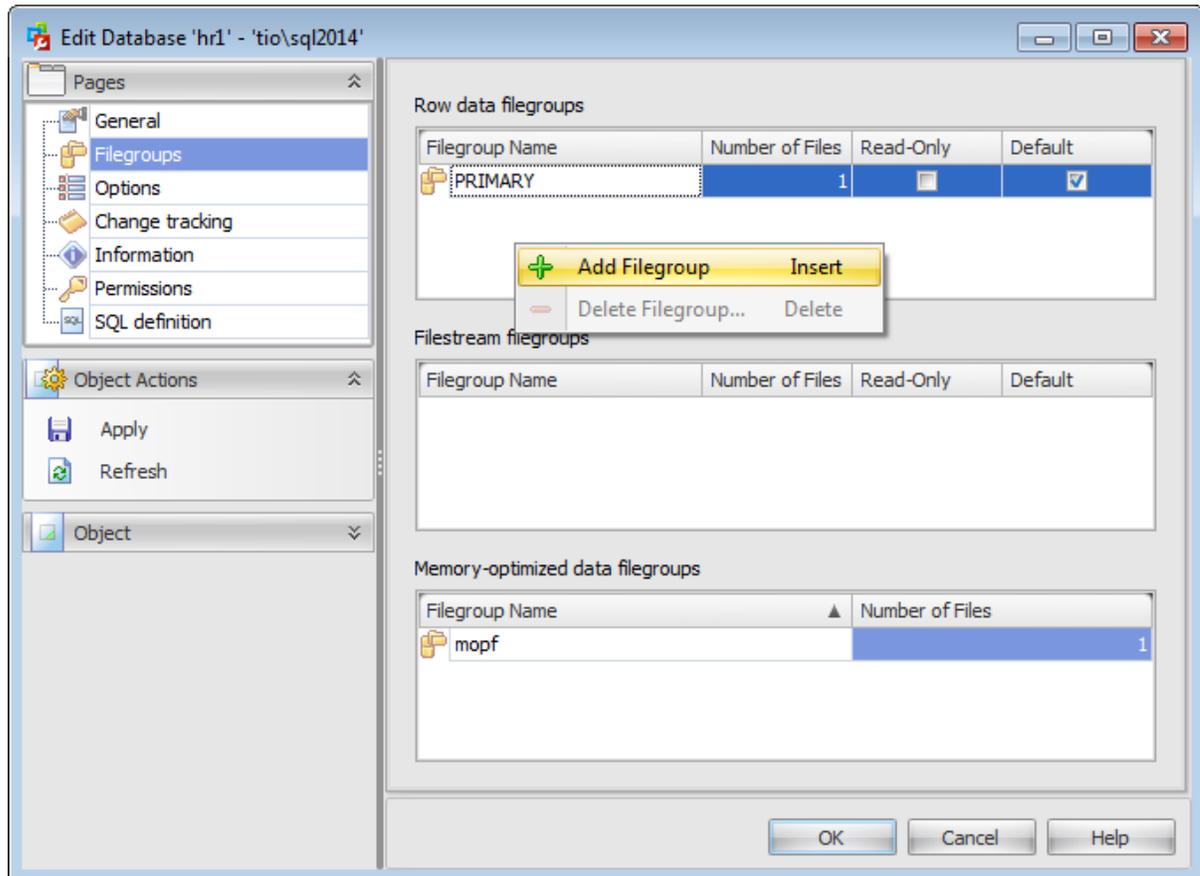
Each file can also have **maximum file size** specified.

If  **Unlimited** is specified, the file can continue to grow until it has used all available space on the disk.

### 6.1.3 Managing filegroups

Database objects and files can be grouped together in *filegroups* for allocation and administration purposes.

Using the popup menu you can **+ Add** or **- Delete** filegroups.



All data files are stored in filegroups:

- *PRIMARY*: the filegroup that contains the primary file. All system tables are allocated to the primary filegroup.
- *User-defined*: any filegroup that is specifically created by the user when first creating or later altering the database.

Filegroup types are separated into *Row data filegroups* and *Filestream filegroups*. Row data filegroups contain regular data and log files. Filestream filegroups contain Filestream data files. These data files store information about how binary large object (BLOB) data is stored on the file system when you are using Filestream storage. The options are the same for both types of filegroups.

#### Read Only

Specifies that the filegroup is read-only. Updates to objects that reside in this filegroup are not allowed. The *PRIMARY* filegroup cannot be marked read-only.

#### Default

When objects are created in the database without specifying to which filegroup they belong, they are assigned to the default filegroup. At any time, exactly one filegroup is designated as the default filegroup. The files in the default filegroup must be large enough to hold any new objects not allocated to other filegroups. Initially, the *PRIMARY* filegroup is the default filegroup.

For SQL Server 2014 databases *Memory-optimized data filegroups* area is available for editing as well.

You need to specify the name of the memory-optimized data filegroup of the database in the **Filegroup Name** column.

The number of files is increased automatically after adding memory-optimized data files on the [General](#) tab.

All these filegroup parameters are edited directly in the corresponding fields.

## 6.1.4 Editing database options

### Recovery Model

SQL Server recovery models are used for backup and restore operations. A *recovery model* is a database property that controls the basic behavior of backup and restore operations for a database: how transactions are logged, whether the transaction log requires backup, what kinds of restore operations are available, etc.

#### *Full*

When this recovery model is set, all transactions are fully logged and all the transaction log records are retained until after they are backed up. In the Enterprise Edition of SQL Server, the full recovery model allows a database to be recovered to the point of failure, assuming that the tail of the log has been backed up after the failure.

#### *Bulk-logged*

When this recovery model is set, most bulk operations are minimally logged (e.g. index creation and bulk loads), and other transactions are fully logged.

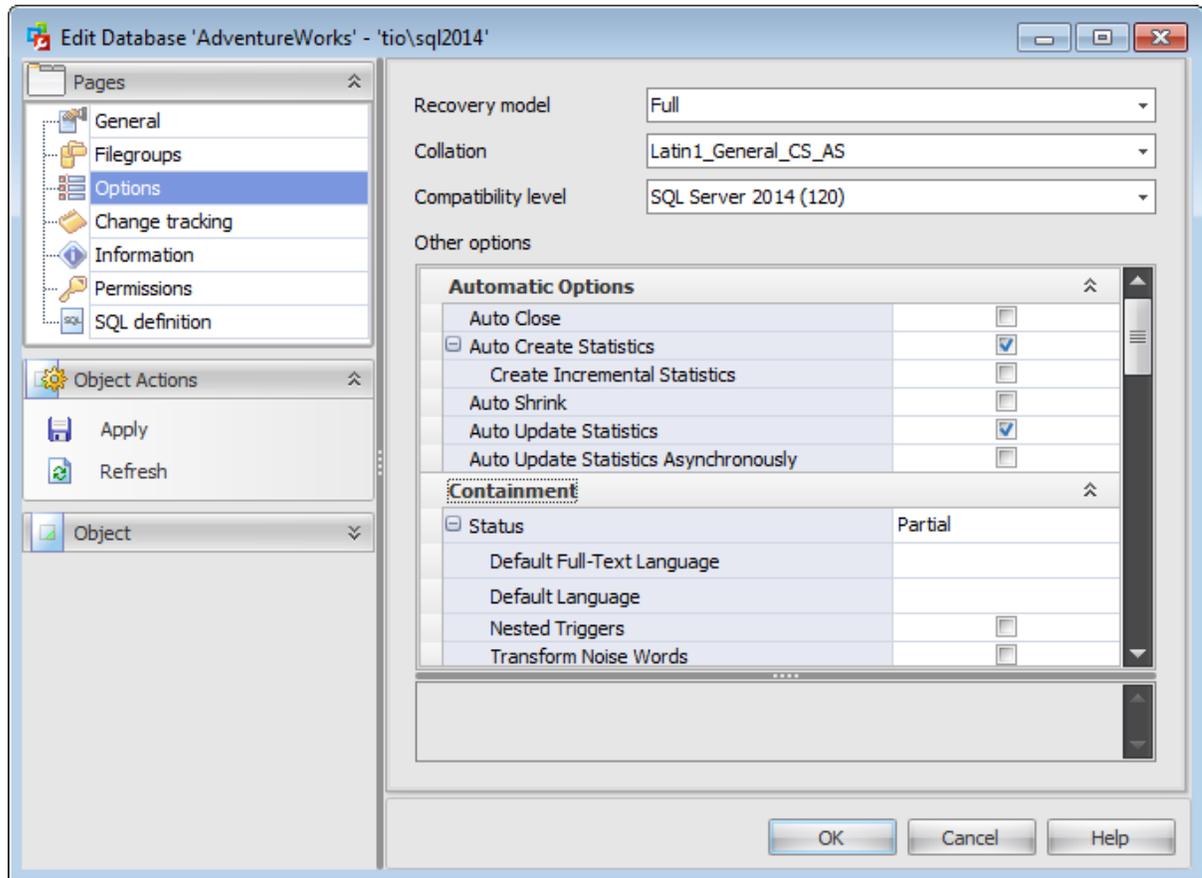
#### *Simple*

When this recovery model is set, most transactions are minimally logged: only the information required to ensure database consistency after a system crash or after restoring a data backup.

### Collation

Use the drop-down list to specify collation for the database being created. By default, the server collation is used.

The **Compatibility Level** option group controls the behavior of an instance of Microsoft® SQL Server™, setting behavior to match a specified version of SQL Server.



## Automatic options

### Auto Close

When this option is enabled, the database is closed and shut down when the last user of the database exits and all processes in the database complete, thereby freeing any resources. The database reopens automatically when a user tries to use the database again. When disabled, the database remains open even if no users are currently using the database.

### Auto Create Statistics

When this option is enabled, statistics are automatically created on columns used in a predicate. Adding statistics improves query performance because the SQL Server query optimizer can better determine how to evaluate a query. If the statistics are not used, SQL Server automatically deletes them. When disabled, statistics are not automatically created; instead, statistics can be manually created.

### Create Incremental Statistics

Specify whether to use the incremental option when per partition statistics are created.

### Auto Shrink

When this option is enabled, the database files are candidates for periodic shrinking. Both data files and log files can be shrunk automatically by SQL Server. *AUTO\_SHRINK* reduces the size of the transaction log only if the database is set to *SIMPLE* recovery model or if the log is backed up. When disabled, the database files are not automatically shrunk

during periodic checks for unused space.

### **Auto Update Statistics**

If this option is checked then any missing statistics that is required on a query for its optimization will be created automatically during the query optimization process.

If the option is not checked then statistics must be created manually.

### **Auto Update Statistics Asynchronously**

When this option is enabled, existing statistics are automatically updated when the statistics become out-of-date because the data in the tables have been changed. When disabled, existing statistics are not automatically updated; instead, statistics can be manually updated.

## **Containment**

### **Status**

When this option is set to *Partial* - the database is contained database. In a contained databases, some settings usually configured at the server level can be configured at the database level.

### **Default Full-Text Language**

Specifies a default language for full-text indexed columns. Linguistic analysis of full-text indexed data is dependent on the language of the data. The default value of this option is the language of the server.

### **Default Language**

The default language for all new contained database users, unless otherwise specified.

### **Nested Triggers**

Allows triggers to fire other triggers.

### **Transform Noise Words**

Suppress an error message if noise words, that is stopwords, cause a Boolean operation on a full-text query to return zero rows.

### **Two Digit Year Cut-off**

Indicates the highest year number that can be entered as a two-digit year. Possible values are from 1753 to 9999. Default value is 2049.

## **Cursor options**

### **Close Cursor on Commit**

When this option is enabled, any open cursors are closed automatically (in compliance with SQL-92) when a transaction is committed. When disabled, cursors remain open across transaction boundaries, closing only when the connection is closed or when they are explicitly closed.

### **Default Cursor Scope**

LOCAL

The scope of the cursor is local to the batch, stored procedure, or trigger in which the cursor was created. The cursor name is valid only within this scope. The cursor can be referenced by local cursor variables in the batch, stored procedure, or trigger, or a stored procedure OUTPUT parameter.

GLOBAL

The scope of the cursor is global to the connection. The cursor name can be referenced

in any stored procedure or batch executed by the connection.

### Database State

#### Database State

*ONLINE*

The database is open and available for use (enabled by default).

*OFFLINE*

The database is closed and shutdown, and marked offline. The database cannot be modified while it is in the offline state.

*EMERGENCY*

The database is marked *READ\_ONLY*, logging is disabled, and only members of the **sysadmin** fixed server role can access the database.

#### Read-Only

When this option is enabled, the database is in read-only mode. Users can retrieve data from the database, but cannot modify the data, because a read-only database does not allow data modifications.

#### User Access

*MULTI\_USER*

Allows all users with the appropriate permissions to connect to the database (enabled by default).

*RESTRICTED\_USER*

Allows only members of the **db\_owner** fixed database role and **dbcreator** and **sysadmin** fixed server roles to connect to the database, but it does not limit their number.

*SINGLE\_USER*

Allows one user at a time to connect to the database. All other user connections are broken.

### External Access Options

#### Cross-Database Ownership Chaining

When this option is checked, *DB\_CHAINING* is enabled and the database can be the source or target of a cross-database ownership chain. When the option is unchecked, the database cannot participate in cross-database ownership chaining. You can change this option for user databases, but not for system databases.

#### Trustworthy

When this option is on, database modules (for example, views, user-defined functions, or stored procedures) that use an impersonation context are allowed to access resources outside of the database.

### FILESTREAM

#### Directory name

The directory name for the FILESTREAM data associated with the selected database.

#### Non-transactional Access Level

Specify one of the following options for non-transactional access through the file system to FILESTREAM data stored in FileTables: OFF, READ\_ONLY, or FULL.

### Service Broker Options

#### Enable Service Broker

Specifies that Service Broker is enabled for the specified database. The

*IS\_BROKER\_ENABLED* flag is set to *TRUE* in the *sys.databases* catalog view and message delivery is started

### **Honor Broker Priority**

Setting honor broker priority allows non-default priority settings to be honored for sending messages between computers or SQL Server instances, or in error cases.

## **Snapshot Isolation Level Options**

### **Enable Snapshot Isolation**

If this option is enabled, SQL Server will execute snapshot transactions against the specified database when transactions are explicitly set to the *SNAPSHOT* transaction isolation level.

### **Elevate Isolation to Snapshot on Memory-Optimized Tables**

If this option is enabled then the minimum isolation level for accessing memory-optimized tables is *SNAPSHOT*. Otherwise, the isolation level is not elevated.

### **Read-Committed Snapshot**

If this option is enabled, transactions setting the read-committed isolation level use row versioning instead of locking.

## **SQL Options**

### **ANSI NULL Default**

When this option is enabled, all user-defined data types or columns that are not explicitly defined as *NOT NULL* during a *CREATE TABLE* or *ALTER TABLE* statement default to allowing null values. Columns that are defined with constraints follow constraint rules regardless of this option.

### **ANSI Nulls**

When this option is enabled, all comparisons to a null value evaluate to *NULL* (unknown). When disabled, comparisons of non-Unicode values to a null value evaluate to *TRUE* if both values are *NULL*.

### **ANSI Padding**

When this option is enabled, trailing blanks in character values inserted into *varchar* or *nvarchar* columns and trailing zeros in binary values inserted into *varbinary* columns are not trimmed. Values are not padded to the length of the column. When disabled, the trailing blanks (for *varchar* or *nvarchar*) and zeros (for *varbinary*) are trimmed. This option affects only the definition of new columns.

### **ANSI Warnings**

When this option is enabled, errors or warnings are issued when conditions such as "divide by zero" occur or null values appear in aggregate functions. When disabled, no warnings are raised when null values appear in aggregate functions, and null values are returned when conditions such as "divide by zero" occur.

### **Arithmetic Abort**

Check this option to terminates a query when an overflow or divide-by-zero error occurs during query execution.

If this option is not checked then a warning message about an occurred mistake appears, but a query or transaction continue to execute as if there were no errors.

### **Concatenate Null Yields Null**

Check this option to returns NULL when concatenating a NULL value with a string.

**Numeric Round-Abort**

Check this option to generate an error when a loss of precision occurs in an expression. If this option is not checked then a loss of precision does not cause an error,

**Quoted Identifiers**

Differentiates between single and double quotation marks when evaluating an expression.

**Recursive Triggers**

When this option is enabled, triggers are allowed to fire recursively. When disabled, triggers cannot be fired recursively.

**Miscellaneous Options****Date Correlation Optimization**

If this option is enabled, SQL Server maintains correlation statistics between any two tables in the database that have *DATETIME* columns and are linked by a one-column foreign key constraint.

**Database Encryption**

Enable the option to use encryption. Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and decrypted when read into memory.

**Note:** This option is available only for SQL Server 2008 databases and higher.

**Delayed durability**

Controls whether transactions commit fully durable or delayed durable. When it is set to *DISABLED*, all transactions are fully durable. Any durability options set in an atomic block or commit statement are ignored. If it is set to *ALLOWED*, transactions are either fully durable or delayed durable, depending upon the durability option set in the atomic block or commit statement. If it is set to *FORCED*, all transactions are delayed durable. Any durability options set in an atomic block or commit statement are ignored.

**Enable Change Data Capture**

Change data capture is designed to capture insert, update, and delete activity applied to SQL Server tables, and to make the details of the changes available in an easily consumed relational format. The change tables used by change data capture contain columns that mirror the column structure of a tracked source table, along with the metadata needed to understand the changes that have occurred.

**Page Verify***CHECKSUM*

A checksum is taken over the contents of the entire page and stored in the page header when a page is written to disk. When a page is read from disk, the checksum is recomputed and compared to the checksum value stored in the page header.

*TORN\_PAGE\_DETECTION*

A bit is reversed for each 512-byte sector in the 8-kilobyte (KB) database page when the page is written to disk. If a bit is in the wrong state when the page is later read, the page was written incorrectly; a torn page is detected., хранимые в заголовке страницы, сравниваются с действительными сведениями о секторах страницы.

*NONE*

*PAGE\_VERIFY* is set to OFF and future data page writes will not contain a *CHECKSUM* or *TORN\_PAGE\_DETECTION* bit and the page will not be verified at read time.

**Parameterization***Simple*

Queries are parameterized on the basis of the default behavior of the database. .

*Forced*

SQL Server parameterizes all queries in the database.

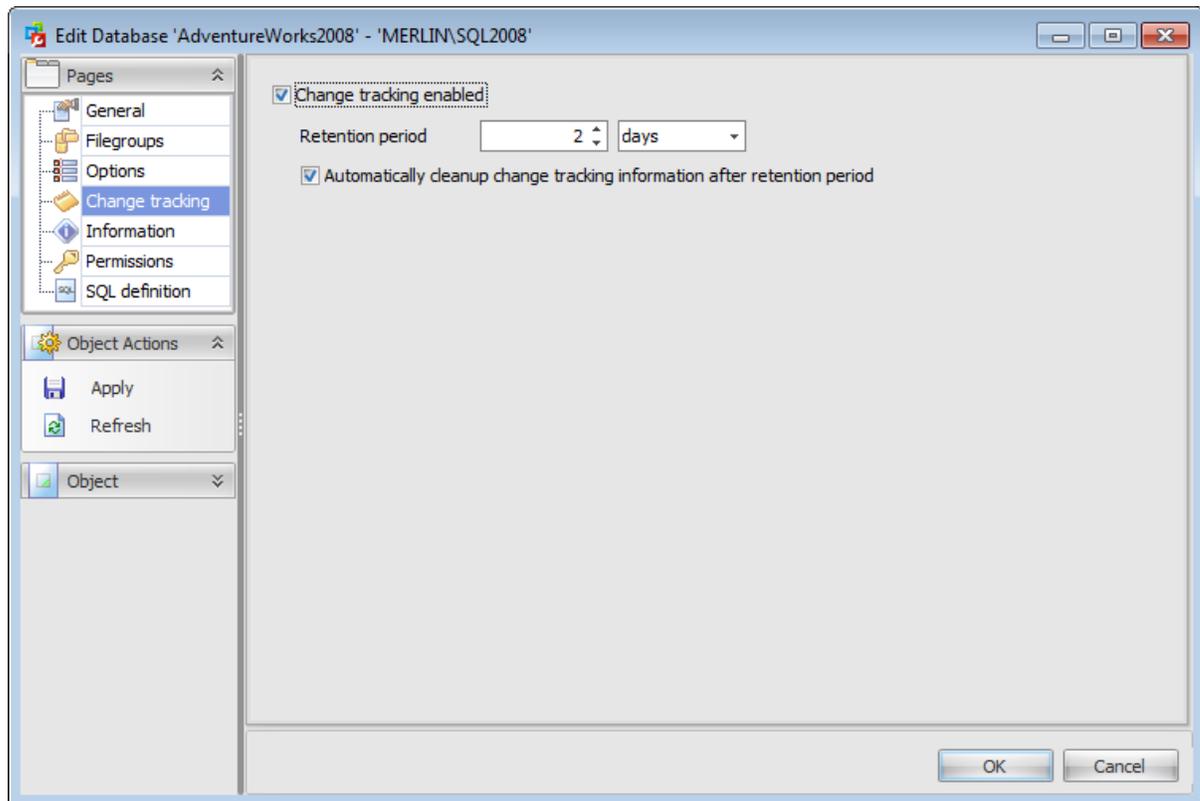
**Target Recovery Time, sec**

The maximum bound on the time (in seconds), to recover the specified database in the event of a crash.

## 6.1.5 Configuring change tracking

**Change tracking** in SQL Server 2008 and higher enables applications to obtain only changes that have been made to the user tables, along with the information about those changes.

Change tracking is an important building block for applications that synchronize and replicate data in scenarios in which end-to-end replication solutions do not work and a custom solution is required.



**Change tracking enabled**

Use this checkbox to enable or disable change tracking for the database.

**Note:** To enable change tracking, you must have permission to modify the database.

**Retention period**

Specify the minimum period (in days, hours or minutes) for keeping change tracking information in the database. Data is removed only when the **Automatically cleanup** is enabled.

The default retention period is 2 days. The minimum retention period is 1 minute.

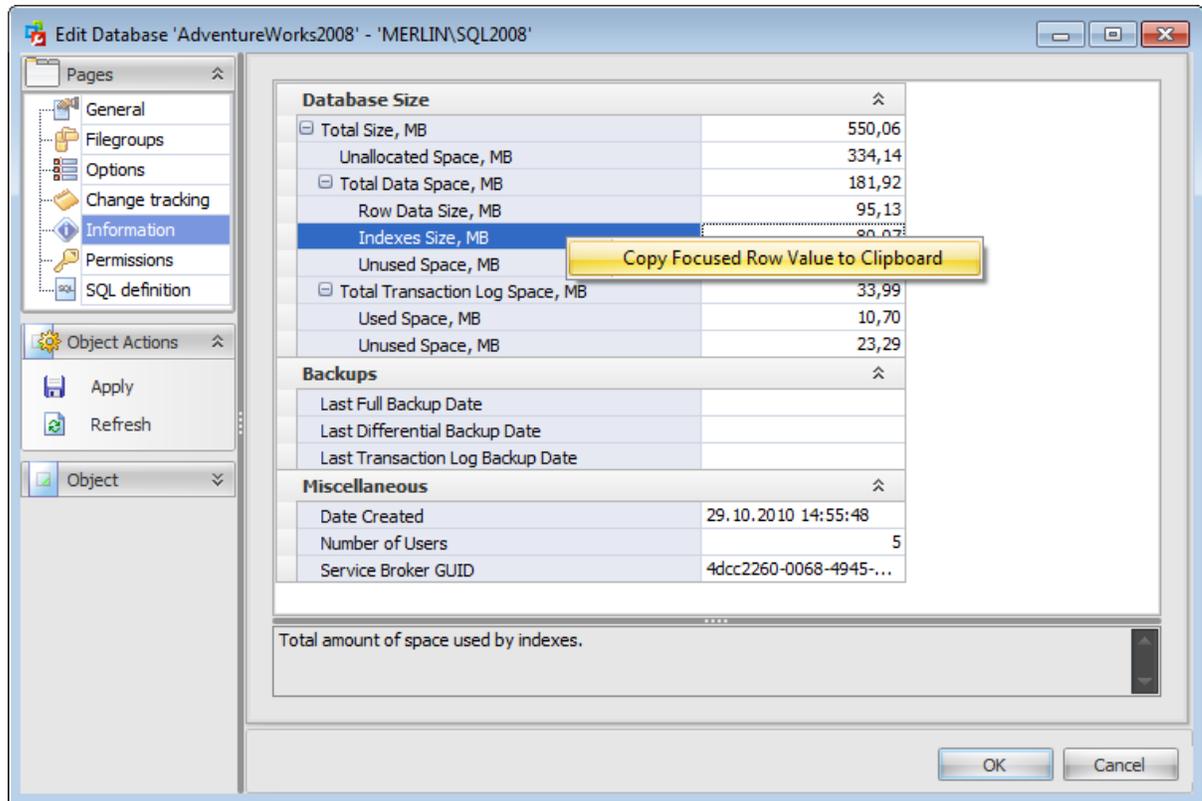
**Automatically cleanup change tracking information after retention period**

Indicates whether change tracking information is automatically removed after the specified retention period.

Enabling Auto Clean-Up resets any previous custom retention period to the default retention period of 2 days.

## 6.1.6 Viewing database properties

The **Information** section of the **Edit Database** window allows you to view additional database properties.



### Database size

This section contains information about hard disk space, used by different database parts.

### Backups

This section shows dates of database backups.

### Miscellaneous

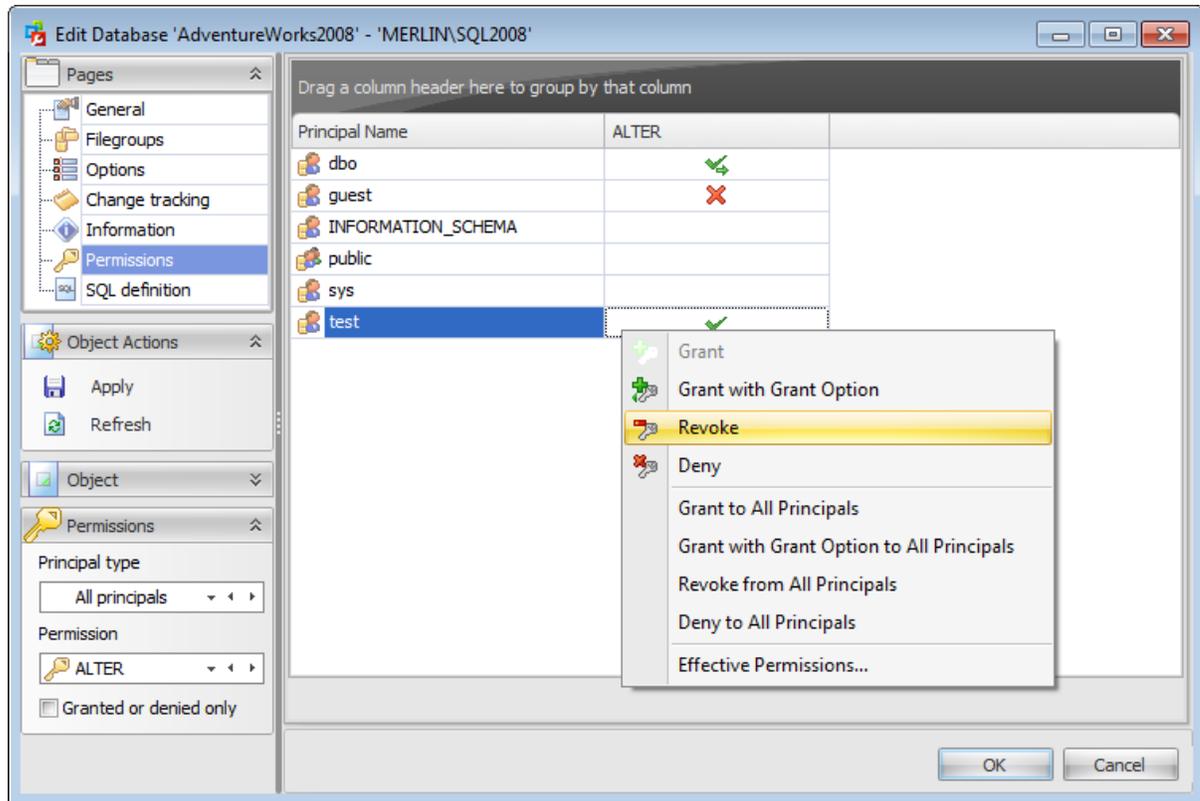
This displays other information about database.

When you choose a database property its meaning is displayed in the box below the grid.

Using the context menu of a row you can copy the parameter value to the Clipboard.

## 6.1.7 Setting permissions

The **Permissions** section of the **Edit Database** window allows you to set permissions to users.



### Principal name

This column contains the name of the database-level principals ([users](#) or [roles](#)).

**Note:** If you double-click the user / role name then the [User Editor](#) / [Role Editor](#) appears.

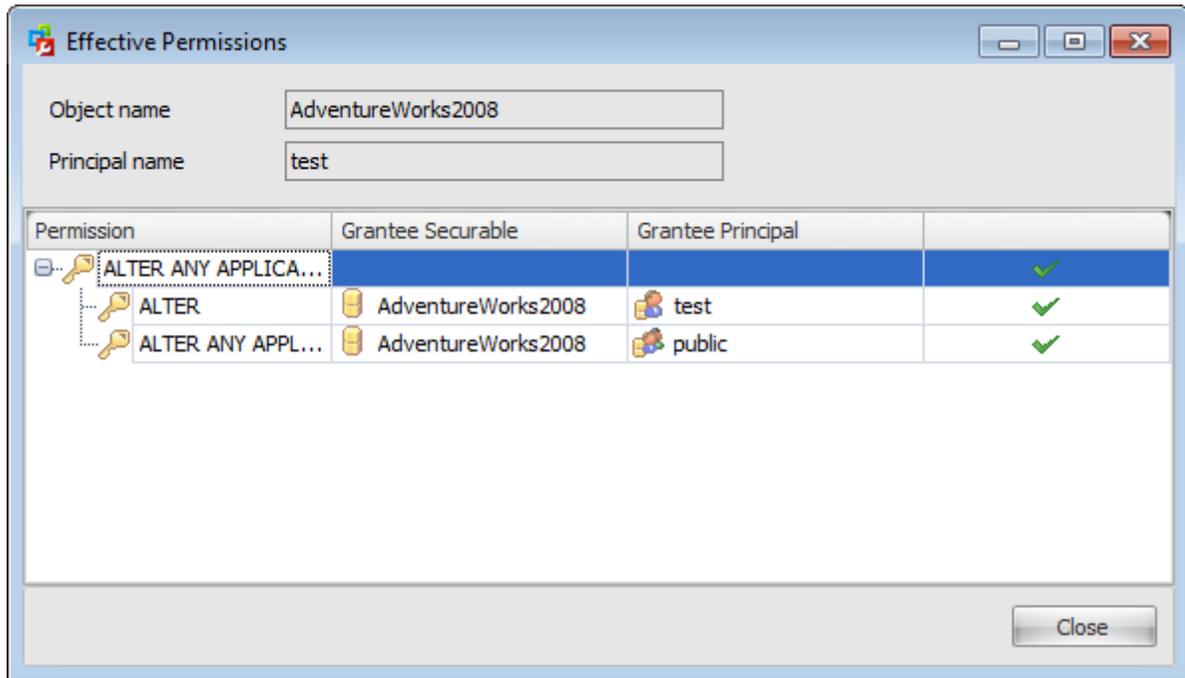
To change a **permission** to modify in the table select this permission from the drop-down list situated in the Permissions tab of the navigation bar.

Right-click a cell within the permission column to grant a permission to the principal. The context menu of a cell allows you to:

- grant a permission to the principal;
- grant a permission (with Grant Option) to the principal;
- deny a permission to the principal;
- revoke a previously granted or denied permission;
- grant all permissions to the principal;
- grant all permissions (with Grant Option) to the principal;
- deny all permissions to the principal;
- revoke all previously granted or denied permissions.

**Note:** Permissions can be granted to a principal by double-clicking a corresponding cell in of the permission column.

Also you can view effective permissions granted to the principal using the context menu.



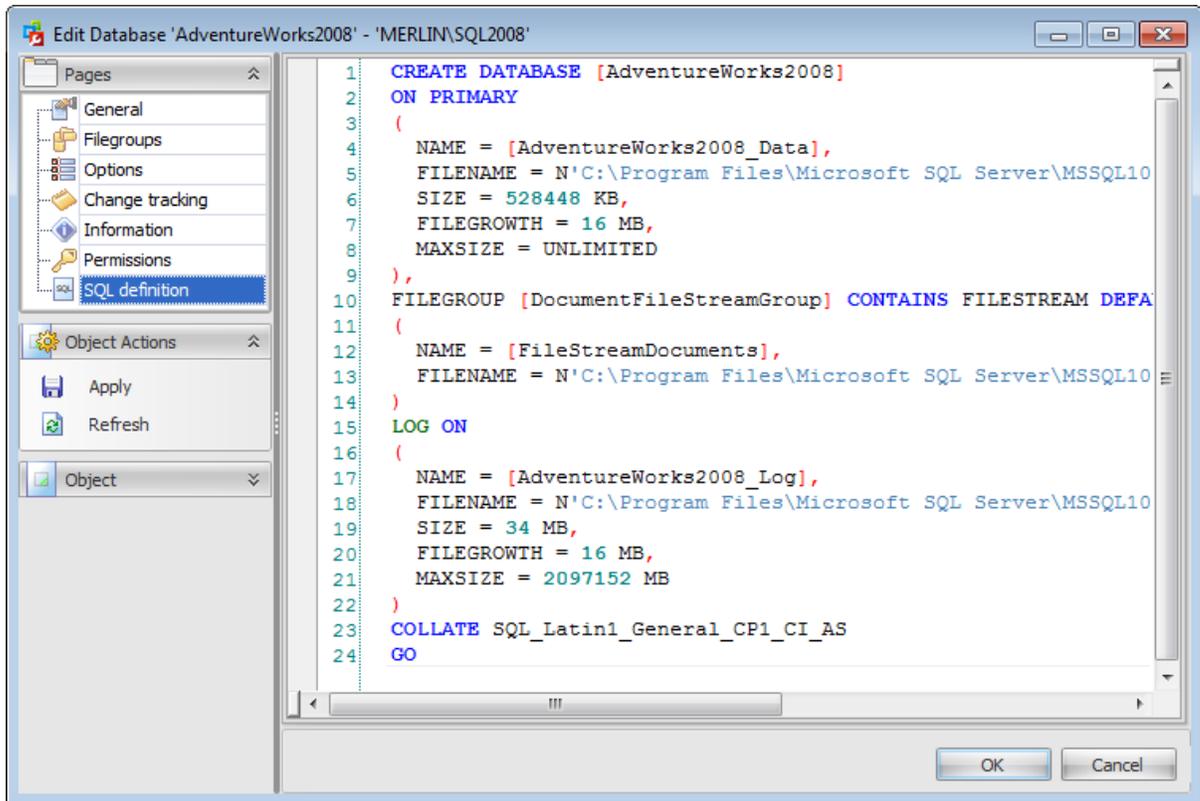
The opened window displays information about all effective permissions.

An **effective permission** is the one which is:

- granted directly to the principal, and not denied, or
- implied by a higher-level permission held by the principal and not denied, or
- granted to or held by, a [role](#) or group of which the principal is a member, and not denied.

## 6.1.8 Viewing SQL definition

The **SQL Definition** section of the **Edit Database** window allows you to view database DDL.



## 6.2 Databases

The **Databases** node of the **Storage** section is provided to manage the list of databases and perform common database tasks.

The toolbar and the context menu of the upper work area allow you to perform the following actions:

 [create](#) database

 [edit](#) selected database(s)

 drop selected database(s)

- create the object copy on the same server
- clone the object copy to another server
- rename selected database
- perform the following tasks on the selected databases:

 [backup database](#)

 [restore database](#)

[recover without restoring](#)

 [attach database](#)

 [detach database](#)

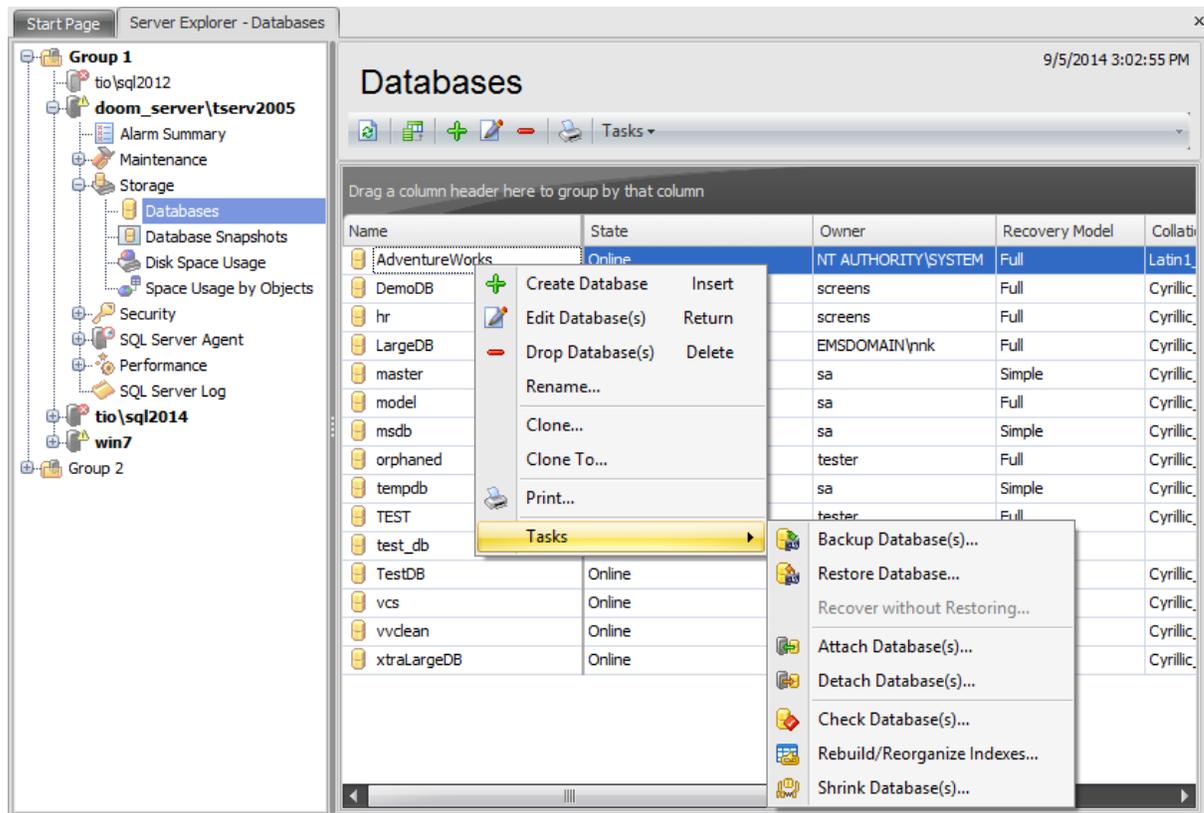
 [check database](#)

 [rebuild/reorganize indexes](#)

 [shrink database](#)

 create and [print](#) the report

Alternatively, you can use the *'Insert'* button to create a database, *'Enter'* button to edit database(s) and *'Delete'* button to drop database(s).



The list displays databases as a grid with the following columns: *Name, State, Owner, Recovery Model, Collation, Compatibility Level, Creation Date*. If necessary the following columns can be added: *ANSI Null Default, ANSI Nulls, ANSI Padding, ANSI Warnings, Arithmetic Abort, Auto Close, Auto Create Statistics, Auto Shrink, Auto Update Statistics, Auto Update Statistics Asynchronously, Change Data Capture Enabled, Close Cursor on Commit, Concatenate Null Yields Null, Cross-Database Ownership Changing, Date Correlation Optimization, Default Local Cursor, Distributor, Encrypted, Encrypted Master Key, Forced Parameterization, Fulltext Enabled, In Standby Mode, Is Cleanly Shut Down, Merge Published, Numeric Round-Abort, Page Verify, Published, Quoted Identifiers, Read Committed Snapshot, Read-Only, Recursive Triggers, Reuse Log Space Waiting For, Service Broker Enabled, Service Broker GUID, Service Broker Priority Honored, Snapshot Isolation State, Subscribed, Synchronization with Backup, Trustworthy, User Access*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a database in the [Database Editor](#), double-click it in the grid. Alternatively, you can right-click the database alias and select the **Edit Database** context menu item.

## 6.3 Database Snapshots

**Database snapshots** were implemented in Microsoft® SQL Server 2005, and they are available only in the Enterprise Edition of Microsoft® SQL Server 2005 and higher.

A **Database snapshot** is a read-only, static view of a database (the source database). Multiple snapshots can exist on a source database and always reside on the same server instance as the database. Each database snapshot is transactionally consistent with the source database as of the moment of the snapshot's creation.

Snapshots can be used for reporting purposes. Also, in the event of a user error on a source database, you can revert the source database to the state it was in when the snapshot was created. Data loss is confined to updates to the database since the snapshot's creation.

**Note:** The source database must be online.

Use the context menu to manage the list of snapshots:

- + create a new snapshot and specify its properties using the [Create New Database Snapshot](#);
- 🖨 view an existing snapshot properties using the [Edit Database Snapshot](#) dialog;
- drop the selected snapshot;
- 🖨 create and [print](#) the report;
- view and edit source [database properties](#);
- revert database to snapshot using the [Restore Database Wizard](#).

Alternatively, you can use the 'Insert' button to create a database snapshot, 'Enter' button to edit database snapshot(s) and 'Delete' button to drop database snapshot(s).

The screenshot shows the 'Database Snapshots' window in SQL Server Enterprise Manager. The window title is 'Server Explorer - Database Snapshots' and the timestamp is '17.05.2013 12:07:33'. The left pane shows a tree view with 'Database Snapshots' selected under 'doom\_server\TSERV2005'. The main pane displays a table with the following data:

Name	Source Database	Snapshot Owner	Snapshot Size, MB	Database Size...
AdventureWorks_20130517	AdventureWorks	screens	0,25	208,00
DemoDB_20130517	DemoDB	screens	0,25	13,38
LargeDB_20130517	LargeDB	screens	0,25	61,19

A context menu is open over the table, showing the following options:

- + Create Database Snapshot Insert
- 🖨 Edit Database Snapshot(s) Return
- Drop Database Snapshot(s) Delete
- 🖨 Print...
- Source Database(s) Properties
- Revert Database to Snapshot...

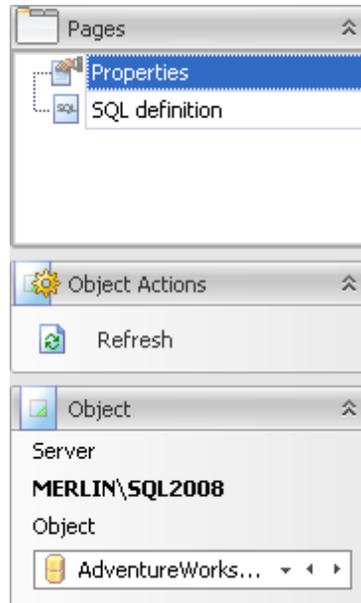
The list displays the existing database snapshots as a grid with the following columns: *Name*, *Source Database*, *Snapshot Owner*, *Snapshot Size (MB)*, *Database Size (MB)*, *Size (%)*. If more convenient, you can change the order of the columns by dragging their

headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a database snapshot in the [Database Snapshot Editor](#), double-click it in the grid. Alternatively, you can right-click the database snapshot alias and select the **Edit Database Snapshot** context menu item.

### 6.3.1 Using Navigation bar

The **Database Snapshot Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Database Snapshot Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes (available only on creating a database snapshot)

 refresh the content of the active tab

#### Object

##### Server

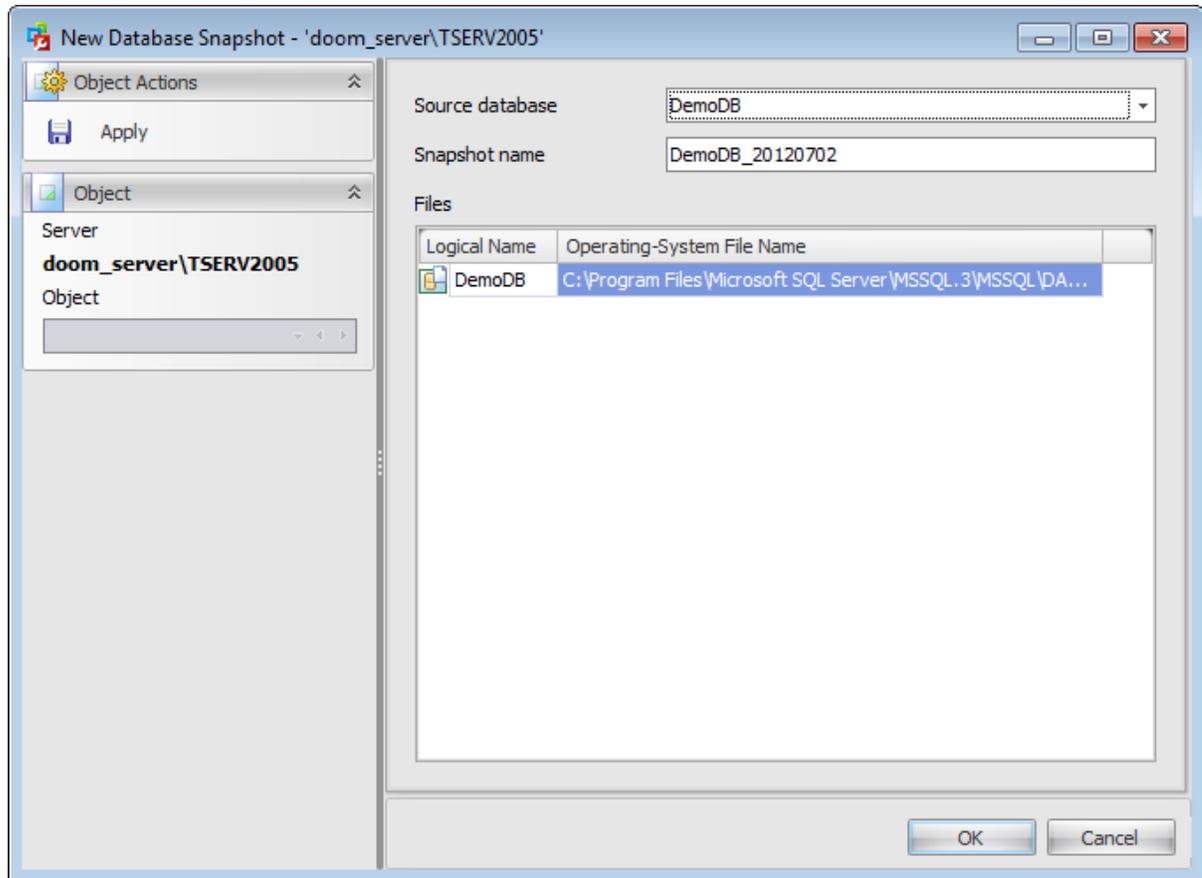
This field displays the name of the server where the edited object resides.

##### Object

This field displays the database which snapshot is currently being created or viewed. However, you can change it by using the drop-down list of databases or by clicking the arrow-left or arrow-right buttons.

### 6.3.2 Database snapshot editor

The **New Database Snapshot** dialog opens automatically when you create a new database snapshot and allows you to define the new snapshot parameters.



#### Source database

Use the drop-down list to select the source database for the new snapshot.

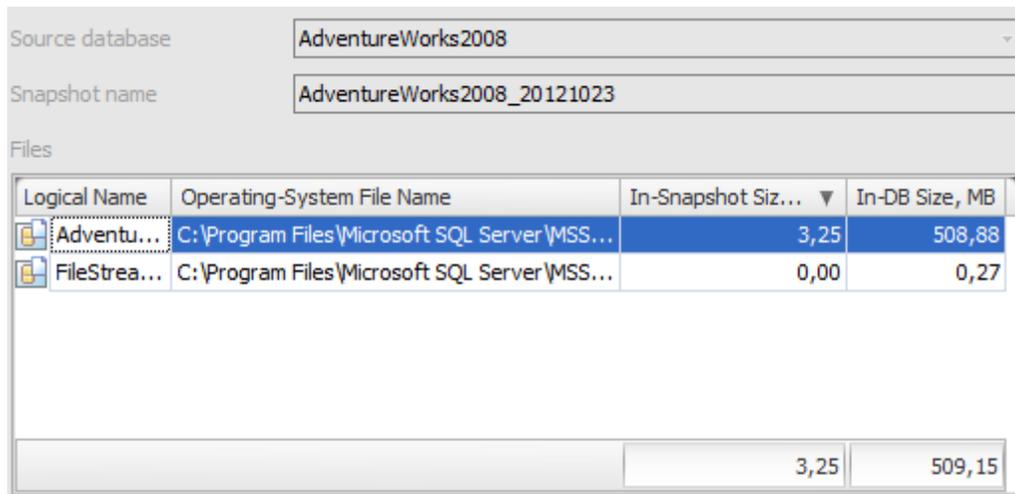
#### Snapshot name

By default, the snapshot name is generated by the application automatically on the basis of the source database name and the snapshot creation date. You can modify the name for the new snapshot, or leave the name unchanged. Note that each database snapshot requires a unique database name.

The **Files** area lists the database file(s) of the new database snapshot. The *Operating-System File Name* contains the name of the source database file and the snapshot creation date.

Click the  **Apply** button to create a snapshot.

As database snapshots are read-only, this dialog allows you just to view snapshot properties and [SQL definition](#).



The screenshot shows a window with the following fields and table:

Source database: AdventureWorks2008

Snapshot name: AdventureWorks2008\_20121023

Files

Logical Name	Operating-System File Name	In-Snapshot Siz...	In-DB Size, MB
Adventu...	C:\Program Files\Microsoft SQL Server\MSS...	3,25	508,88
FileStrea...	C:\Program Files\Microsoft SQL Server\MSS...	0,00	0,27

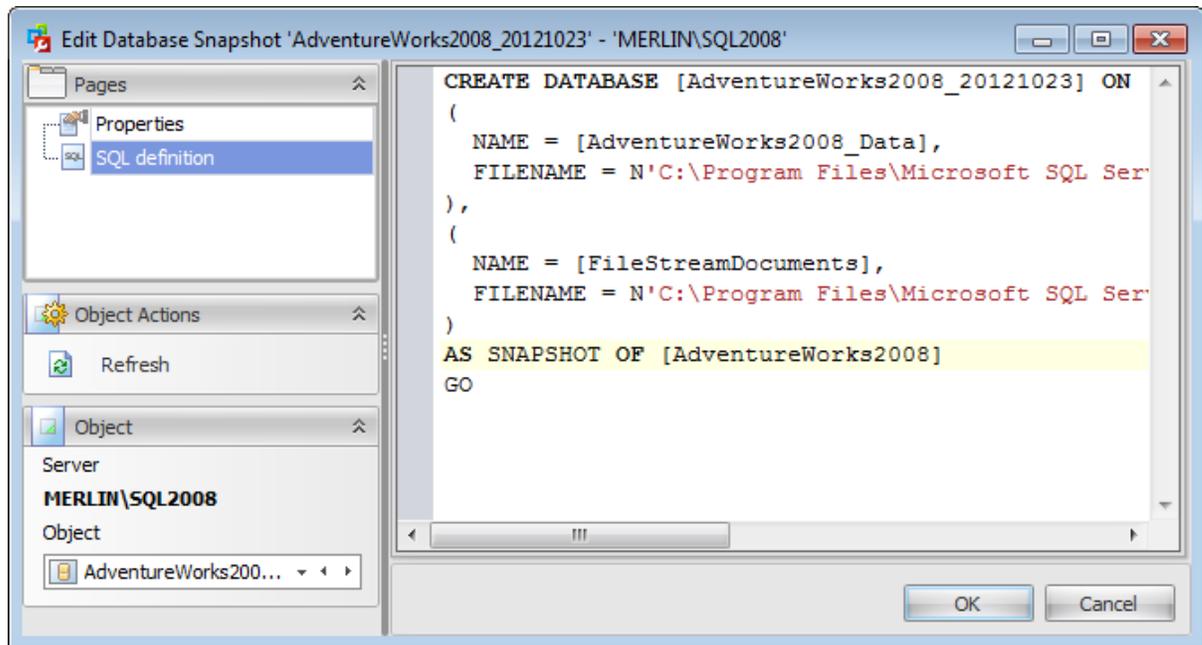
Summary row: 3,25 | 509,15

Here you can view the *Source database*, *Snapshot name* and *Database files*. The Files grid consists of the following fields: *Logical Name*, *Operating-System File-Name*, *In-Snapshot Size (MB)*, *In-DB Size(MB)*.

[Viewing SQL definition](#)

### 6.3.3 Viewing SQL definition

This tab allows you to view the object DDL.



## 6.4 Disk Space Usage

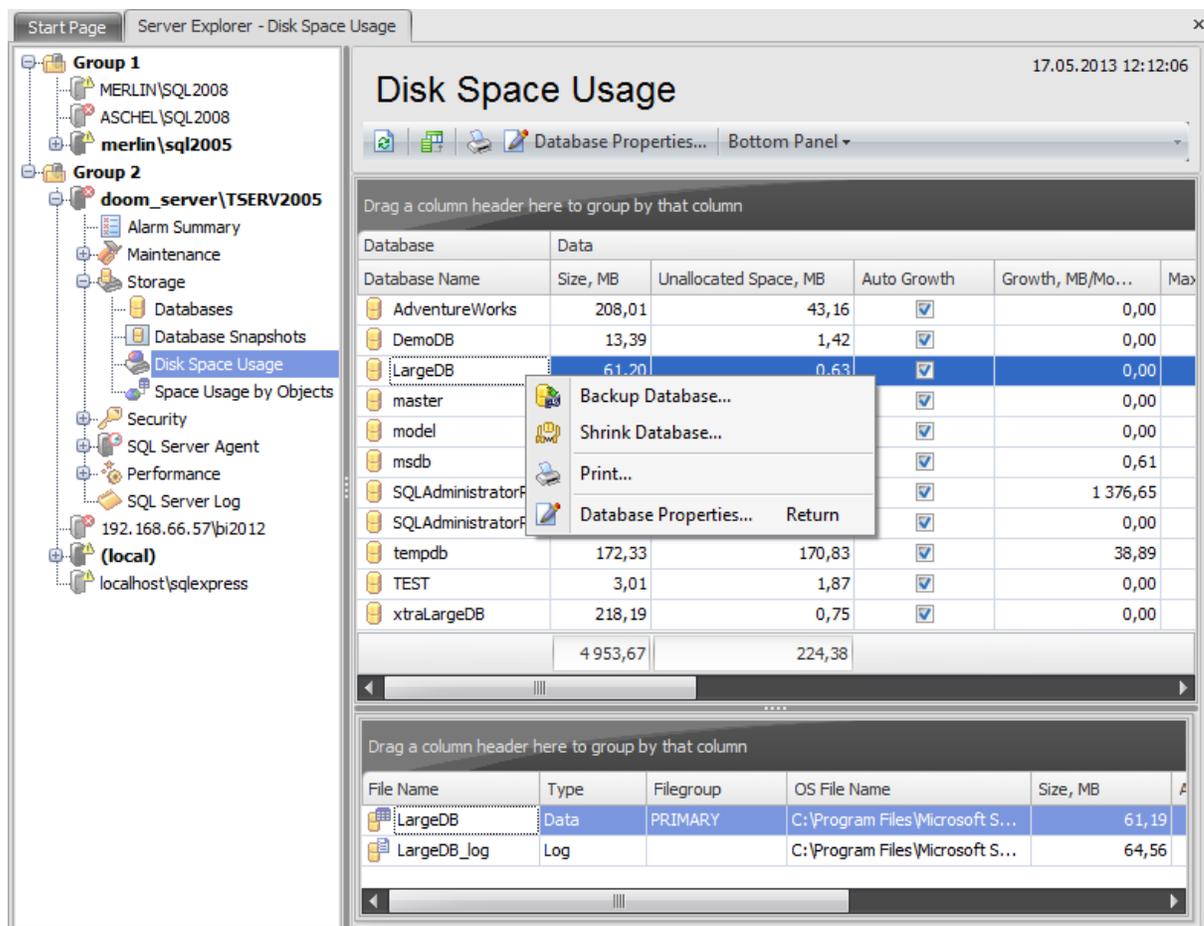
**Disk Space Usage** section of the [Server Explorer](#) is used to explore the amount of space occupied by databases, data and logs.

Use the context menu or buttons on the work area toolbar to manage the list of databases:

-  edit database properties
-  backup database
-  shrink database
-  create and [print](#) the report

Alternatively, you can use the 'Enter' button to edit database(s).

Click the **Bottom Panel** button to define what information should be displayed in the lower area of the window:  **Largest Databases** or  **Files**. This can be also done by clicking the corresponding buttons on the toolbar or by choosing the corresponding item of the bottom panel context menu.



The screenshot displays the 'Disk Space Usage' window in SQL Server Enterprise Manager. The left-hand tree view shows a hierarchy of servers, with 'doom\_server\TSERVER2005' expanded to show various components, including 'Disk Space Usage'. The main window area is titled 'Disk Space Usage' and contains two tables. The top table lists database disk space usage, and the bottom table lists file details for selected databases. A context menu is open over the 'LargeDB' row in the top table, showing options such as 'Backup Database...', 'Shrink Database...', 'Print...', and 'Database Properties...'. The bottom table shows details for 'LargeDB' (Data file, PRIMARY filegroup, OS File Name: C:\Program Files\Microsoft S..., Size: 61,19 MB) and 'LargeDB\_log' (Log file, OS File Name: C:\Program Files\Microsoft S..., Size: 64,56 MB).

Database Name	Size, MB	Unallocated Space, MB	Auto Growth	Growth, MB/Mo...	Max
AdventureWorks	208,01	43,16	<input checked="" type="checkbox"/>	0,00	
DemoDB	13,39	1,42	<input checked="" type="checkbox"/>	0,00	
LargeDB	61,20	0,63	<input checked="" type="checkbox"/>	0,00	
master			<input checked="" type="checkbox"/>	0,00	
model			<input checked="" type="checkbox"/>	0,00	
msdb			<input checked="" type="checkbox"/>	0,61	
SQLAdministratorF			<input checked="" type="checkbox"/>	1 376,65	
SQLAdministratorF			<input checked="" type="checkbox"/>	0,00	
tempdb	172,33	170,83	<input checked="" type="checkbox"/>	38,89	
TEST	3,01	1,87	<input checked="" type="checkbox"/>	0,00	
xtraLargeDB	218,19	0,75	<input checked="" type="checkbox"/>	0,00	
		4 953,67		224,38	

File Name	Type	Filegroup	OS File Name	Size, MB	A
LargeDB	Data	PRIMARY	C:\Program Files\Microsoft S...	61,19	
LargeDB_log	Log		C:\Program Files\Microsoft S...	64,56	

The upper area displays the following information about disk space usage:  
**Database:** Database Name, Database ID;

**Data:** Size (MB), Unallocated Space (MB), Auto Growth, Growth (MB/Month), Max Size (MB), Disk Free Space (MB), No Space Warned On;

**Log:** Disk Free Space (MB), Size (MB), Unused Space (MB), Auto Growth, Growth (MB/Month), Max Size (MB), Disk Free Space (MB), No Space Warned On.

To open a database in the [Database Editor](#), double-click it in the grid. Alternatively, you can right-click the database alias and select the **Database Properties...** context menu item.

The lower area shows the information about the database files or about largest databases disk space usage. These diagrams are graphed on the basis of log size and data size.

Availability:

**Full** version                    **Yes**

**Free** version                    **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 6.5 Space Usage by Objects

This node of the **Storage** section contains the list of the database objects along with the detailed information about the space they occupy.

### Database

Use the drop-down list to select the necessary database which objects you want to view.

The screenshot shows the 'Space Usage by Objects' window in SQL Server Enterprise Manager. The window title is 'Space Usage by Objects' and the database selected is 'AdventureWorks'. The window displays a table of database objects with the following columns: Object Name, Schema, Object Type, Number of Records, Reserved, Data, Indexes, Unused. The 'Contact' table is selected, and a context menu is open over it, showing options like 'Non-System Only' and 'Print...'. The 'Contact' table is expanded to show its indexes.

Object Name	Schema	Object Type	Number of Records	Reserved, KB	Data, KB	Indexes, KB	Unused, KB
Individual	Sales	Table	295 744				
SalesOrderDetail	Sales	Table	121 317				
TransactionHistory	Production	Table	113 443				
SalesOrderHeader	Sales	Table	31 465				
TransactionHistoryArchive	Production	Table	89 253				
Contact	Person	Table	195				
AK_Contact_rowguid		Index		19 972	584		
IX_Contact_EmailAddress		Index		19 972	1 480		
PK_Contact_ContactID		Index		19 972	4 616		
PXML_Contact_AddContac		Index		195	80		
WorkOrderRouting	Production	Table	67 131				
WorkOrder	Production	Table	72 591				
				978 240	168 792		

The list displays all database objects as a grid with the following columns: *Object Name*, *Schema*, *Object Type*, *System*, *Number of Records*, *Reserved (KB)*, *Data (KB)*, *Indexes (KB)*, *Unused (KB)*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

Use the **Non-system only** context menu item or the corresponding toolbar button to display only non-system objects.

Use the **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

If you click the '+' button situated at the beginning of the row then the table of the indexes appears. This grid contains the following fields: *Index Name*, *Partition*, *Number of Records*, *Total Reserved (KB)*, *Total Used (KB)*, *In-Row Data Reserved (KB)*, *In-Row Data Used (KB)*, *Leaf Level of In-Row Data*, *LOB Data Reserved (KB)*, *LOB Data Used (KB)*, *Row Overflow Data Reserved (KB)*, *Row Overflow Data Used (KB)*, *Internal Table*.

Availability:

**Full** version

**Yes**

**Free** version

**No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

**Part**



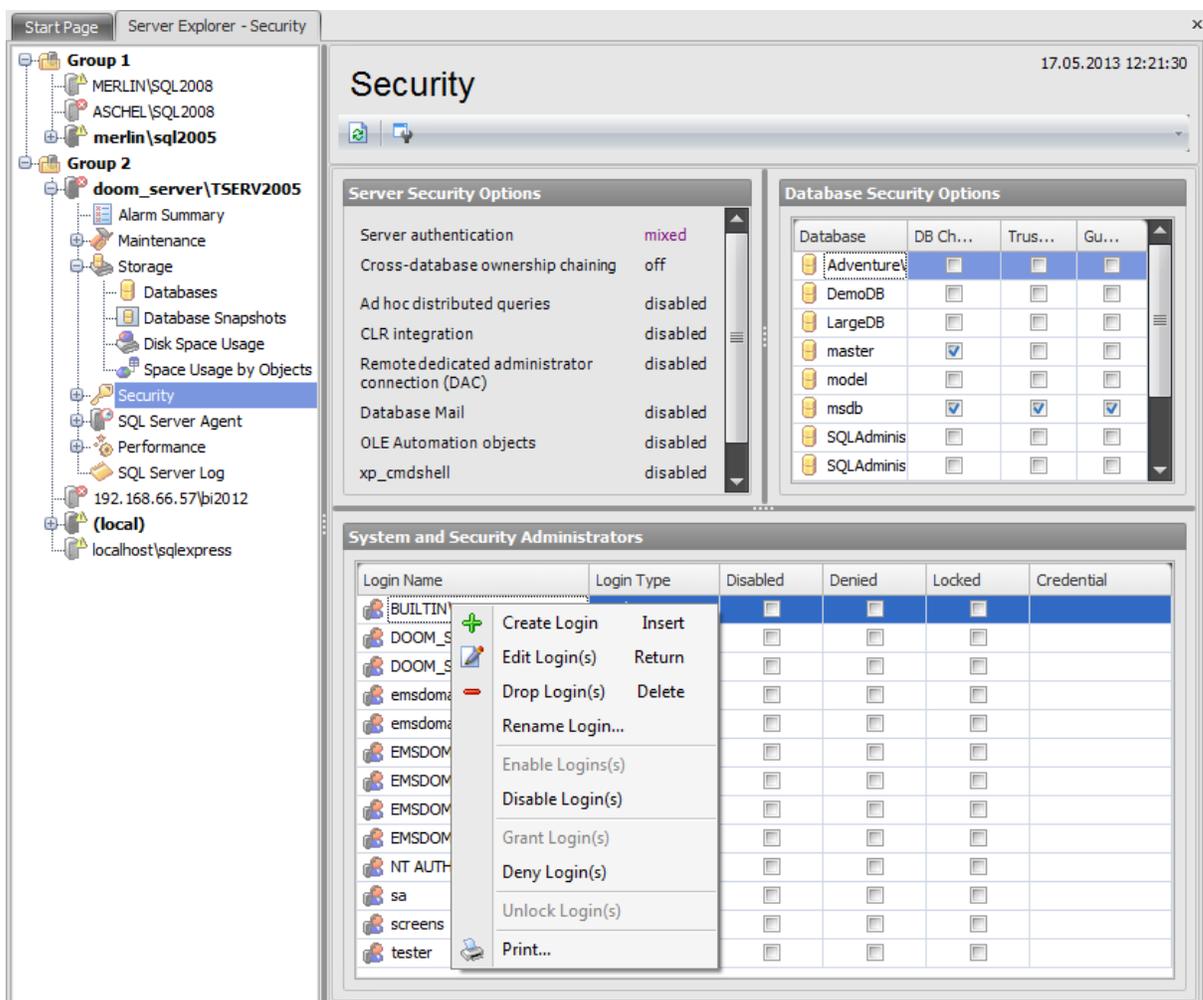
**VII**

## 7 Security

The **Security** section of the [Server Explorer](#) is provided to manage database-level and server-level permissions and principals.

The section has the following nodes for managing the server security:

- [Logins](#)
- [Database users](#)
- [Database roles](#)
- [Application Roles](#)
- [Server Roles](#)
- [Server Permissions](#)
- [Database Permissions](#)



### Server Security Options

This area shows some server options that define the server security. The context menu of the area allows you to launch the [Server Properties](#) dialog where you can view and configure server options. Also, you can double-click within the area to call this dialog.

### Database Security Options

This area shows some database options that define the database security. Use the context menu item of the area or the *Enter* button to launch the [Edit Database](#) dialog where you can view and configure database options.

### System and Security Administrators

This grid lists logins that are the members of the System Administrators and Security Administrators [server roles](#).

Use the context menu to:

 [create](#) a new login

 [edit](#) the selected login(s)

 drop selected login(s)

- enable/disable login
- grant/deny login(s)
- unlock login(s)

 create and [print](#) the report

Alternatively, you can use the *'Insert'* button to create login, *'Enter'* button to edit login (s) and *'Delete'* button to drop login(s).

---

#### See also:

[Getting Started](#)

[Servers](#)

[Database Maintenance](#)

[Storage](#)

[SQL Server Agent](#)

[Performance](#)

[SQL Server Log](#)

[Query Editor](#)

[Options](#)

## 7.1 Logins

Microsoft® SQL Server™ uses two ways to validate connections to SQL Server databases:

**Windows Authentication** which uses Windows-level principals and **SQL Server Authentication** which uses logins to validate the connection.

A **Login** is an indivisible principal which is used to connect to the server and use its resources.

Use the context menu to:

 [create](#) a new login

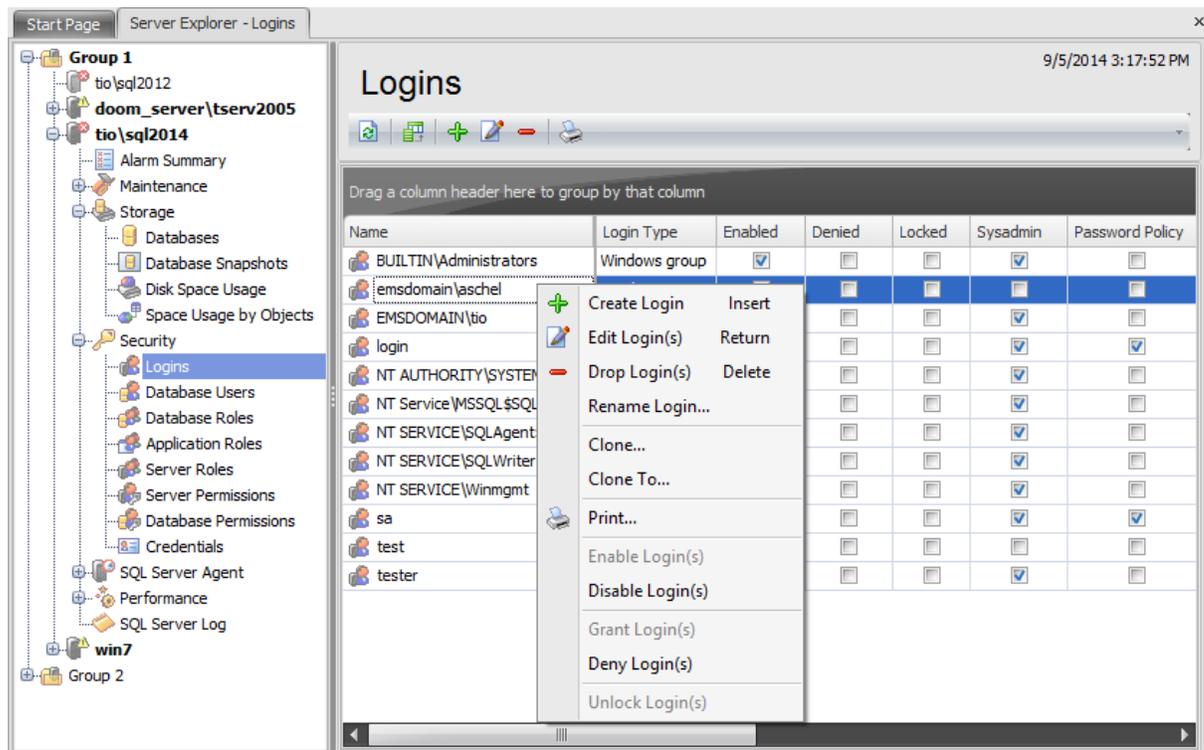
 [edit](#) the selected login(s)

 drop selected login(s)

- rename the selected login
- create the object copy on the same server
- clone the object copy to another server
- enable/disable login
- grant/deny login(s)
- unlock login(s)

 create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a login, 'Enter' button to edit login (s) and 'Delete' button to drop login(s).



The list displays the existing database logins as a grid with the following columns: *Name*, *Login type*, *Enabled*, *Denied*, *Locked*, *Sysadmin*, *Password Policy*, *Password Expiration*, *Credential*, *Certificate*, *Asymmetric Key*. If more convenient, you can change the order of

the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find operations that can be performed with the grid.

To open a Login in the [Login Editor](#), double-click it in the grid. Alternatively, you can right-click the Login alias and select the **Edit Login(s)** context menu item.

---

**See also:**

[Database users](#)

[Database roles](#)

[Application Roles](#)

[Server Roles](#)

[Server Permissions](#)

[Database Permissions](#)

### 7.1.1 Login Editor

**Login Editor** allows you to define login properties. It opens automatically when you create a new server login and is available on editing an existing one.

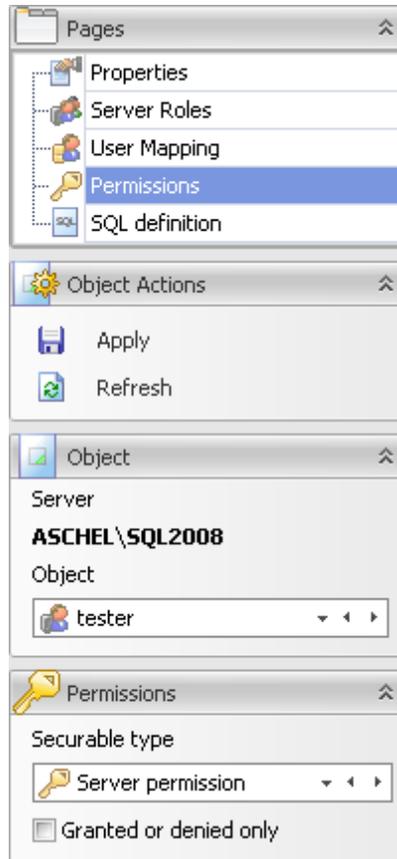
To open a login in **Login Editor**, double-click it in the login list. Alternatively, you can right-click the login alias and select the **Edit Login** context menu item.

The list of **Login Editor** tabs is displayed on the [navigation bar](#):

- [Setting login properties](#)
- [Assigning server roles](#)
- [User mapping](#)
- [Setting permissions](#)
- [Viewing SQL definition](#)

### 7.1.1.1 Using Navigation bar

The **Login Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Login Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Object

This field displays the login that is currently being edited. However, you can change the edited login by using the drop-down list of logins or by clicking the arrow-left or arrow-right buttons.

#### Permissions

This group of options appears on setting permissions.

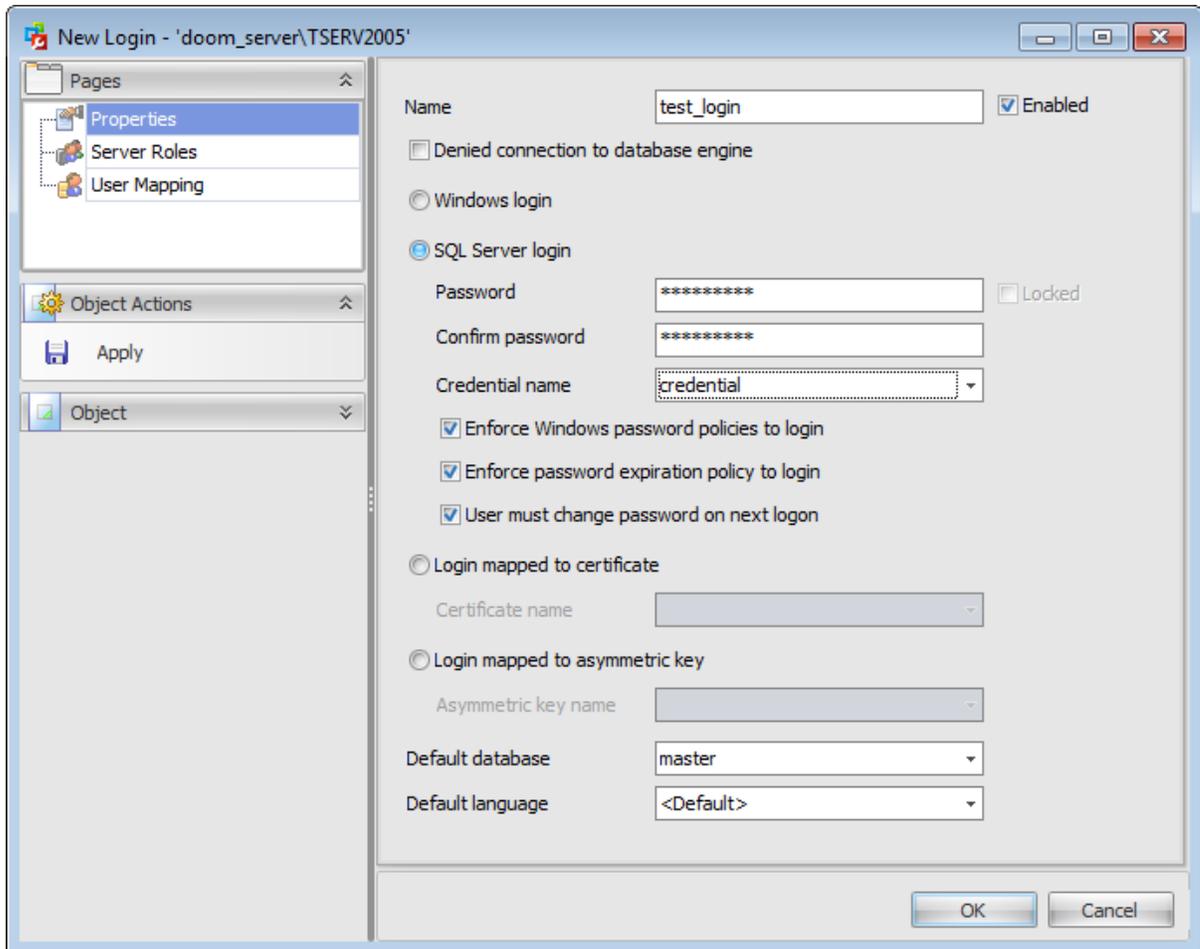
Select the server **Securable Type** from the drop-down list: *Server*, *Login* or *Endpoint*.

**Granted or denied only**

Check this option for displaying only granted permissions or those securables have been set granted or denied.

### 7.1.1.2 Setting login properties

Use the **Properties** section of the **Login Editor** to configure login authentication and default parameters.



#### Name

Enter a name for the new login, or modify the name of the login being edited.

#### Enabled

Enables/disables the login immediately after it is created. A disabled login still exists in the server, but it cannot be used to connect to the instance of SQL Server.

#### Denied connection to database engine

Select this option to forbid any connections to the server for the login.

Specify the way used by the login to connect to the instance of SQL Server:

- Windows login*
- SQL Server login*
- Login mapped to certificate*
- Login mapped to asymmetric key*

**Windows login**

When Windows Authentication is used to connect to SQL Server, Microsoft Windows is completely responsible for authenticating the client. In this case, the client is identified by its Windows account (Windows-level login).

**SQL Server login**

When SQL Server Authentication is used, SQL Server authenticates the client by comparing the client-supplied user name and password to the list of valid user names and passwords maintained within SQL Server.

**Credential name**

Use the drop-down list to select the credential that will be mapped to the new SQL Server login.

**Enforce Windows password policies to login**

This option specifies that the Windows password policies of the computer where SQL Server is running should be enforced on this login.

**Enforce password expiration policy to login**

This option specifies whether password expiration policy should be enforced on this login.

**User must change password next login**

Check this option to force a user to change their password at next logon.

**Login mapped to certificate**

Select this option to specify a certificate-mapped login.

**Certificate name**

Use the drop-down list to select the certificate that will be associated with this login.

**Login mapped to asymmetric key**

Select this option to specify an asymmetric key-mapped login.

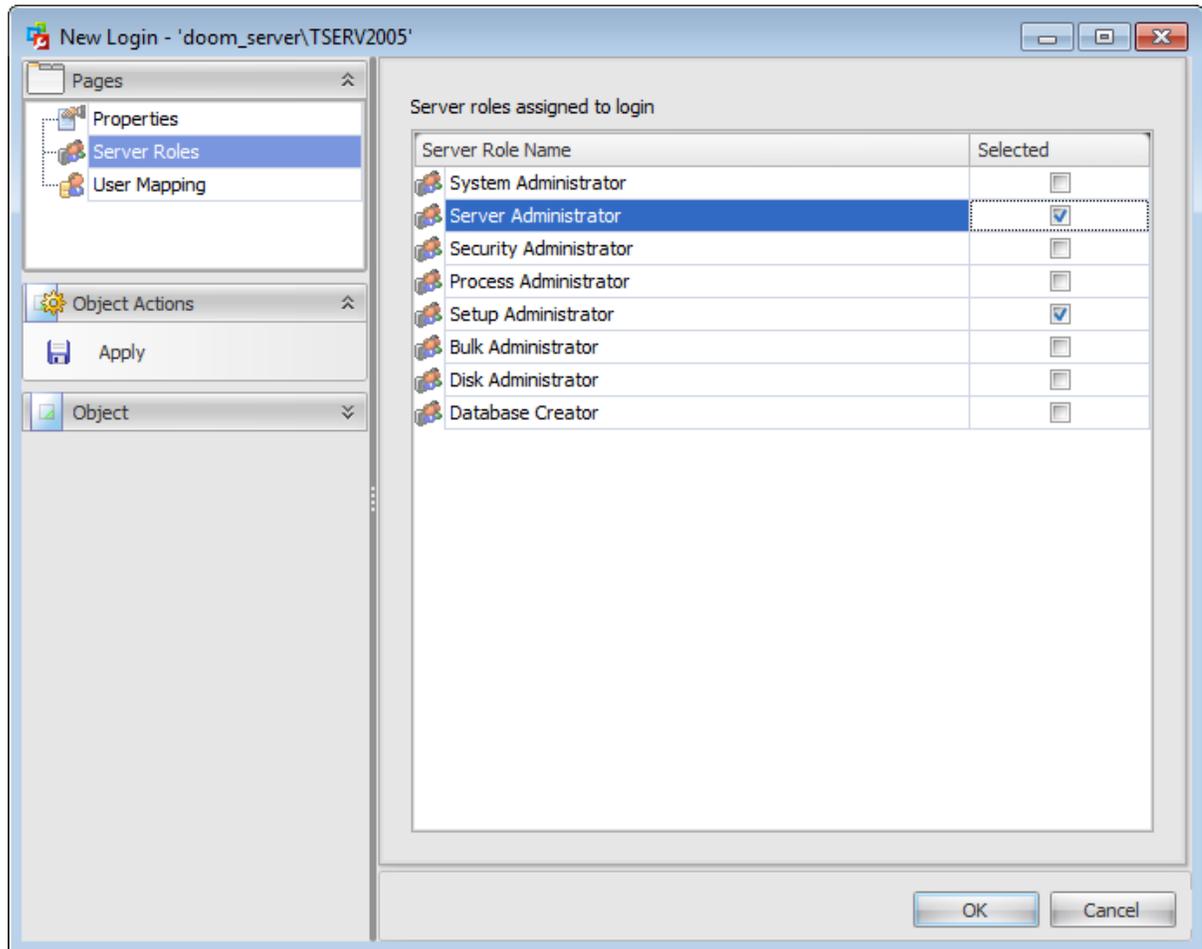
**Asymmetric key name**

Use the drop-down list to select the asymmetric key that will be associated with this login.

The lower group allows you to set the default parameters pertaining to the login: **Default database** and **Default language**. Use the corresponding drop-down lists to select the required values that will be assigned to the login.

### 7.1.1.3 Assigning server roles

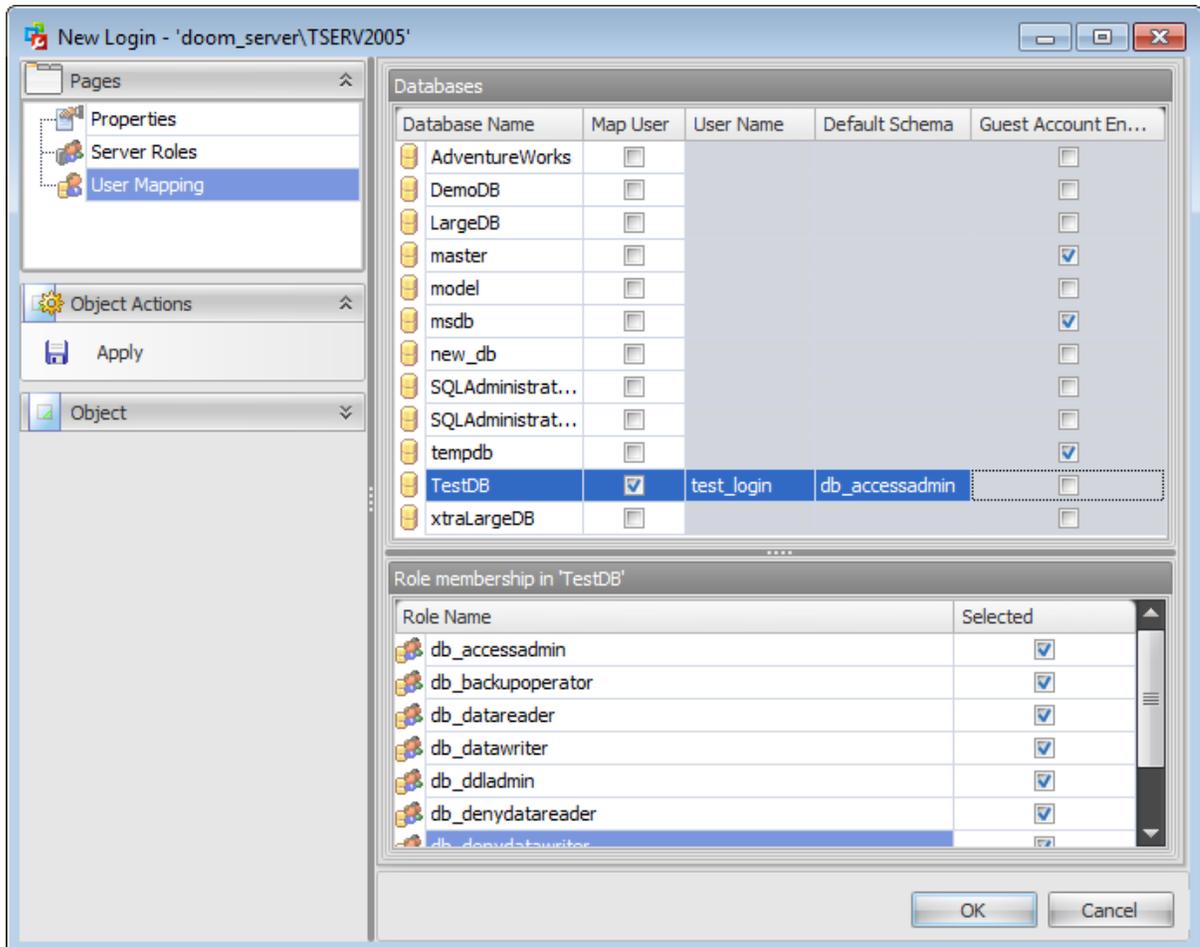
The **Server Roles** section of **Login Editor** allows you to define the login membership, i.e. to select the [server role\(s\)](#) the login will belong to. The selected server roles determine the tasks that can be performed through the login.



To select a server role, you need to choose it in the **Server Role Name** column and set the respective flag in the **Selected** column.

#### 7.1.1.4 User mapping

The **User Mapping** section allows you to browse the list of databases available on the specified server, and to map their [users](#) to the login.



Set the **Map user** flag next to a database name, type in the **User Name** and select the **Default Schema** of the database using the corresponding drop-down list.

#### Guest account enabled

This column determines whether the guest user has permission to access the database.

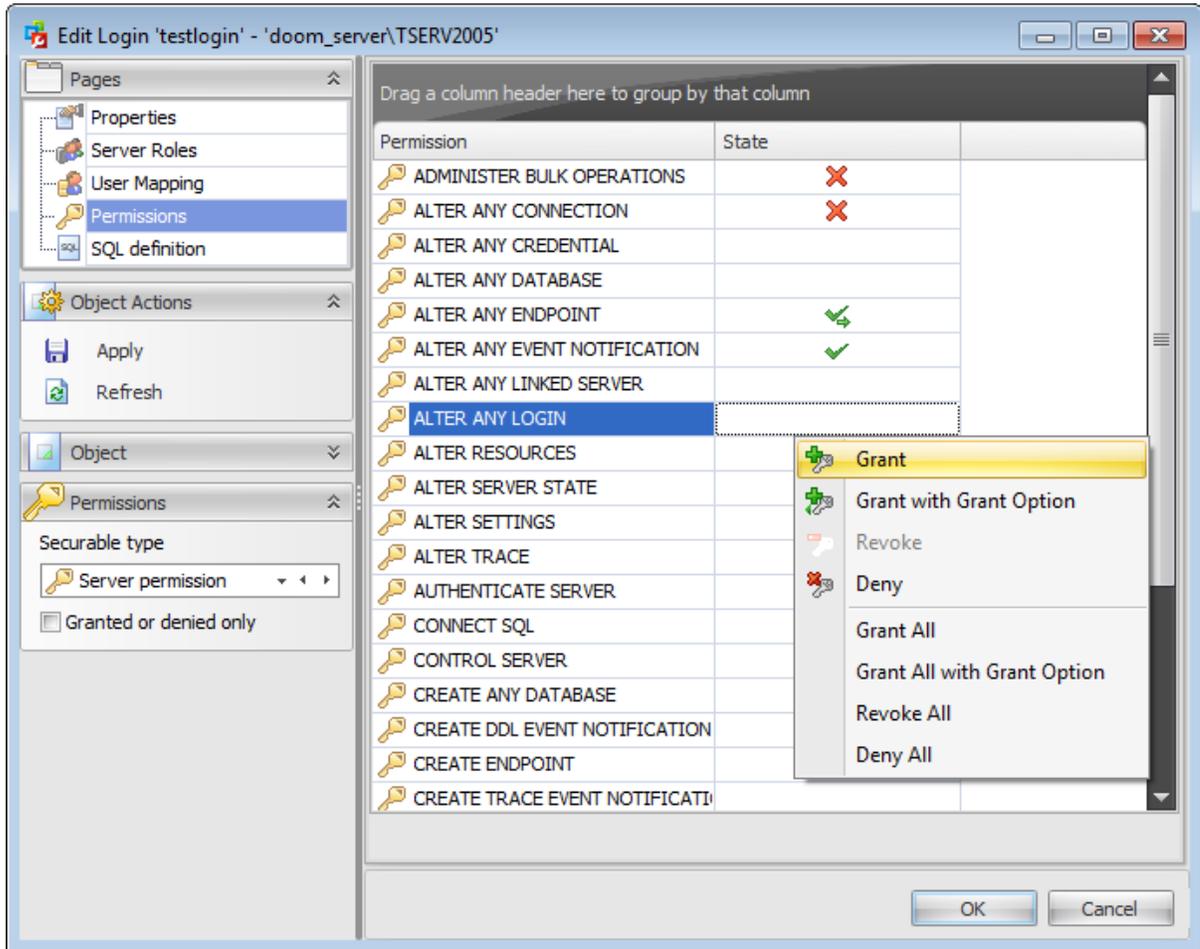
If necessary, use the **Role membership** area at the bottom to define [database role\(s\)](#) for each mapped user.

### 7.1.1.5 Setting permissions

This section allows you to grant **Server, Login, Endpoint** and **Server Roles** (for SQL Server 2012) permissions to the login.

This section is available only when editing an existing login.

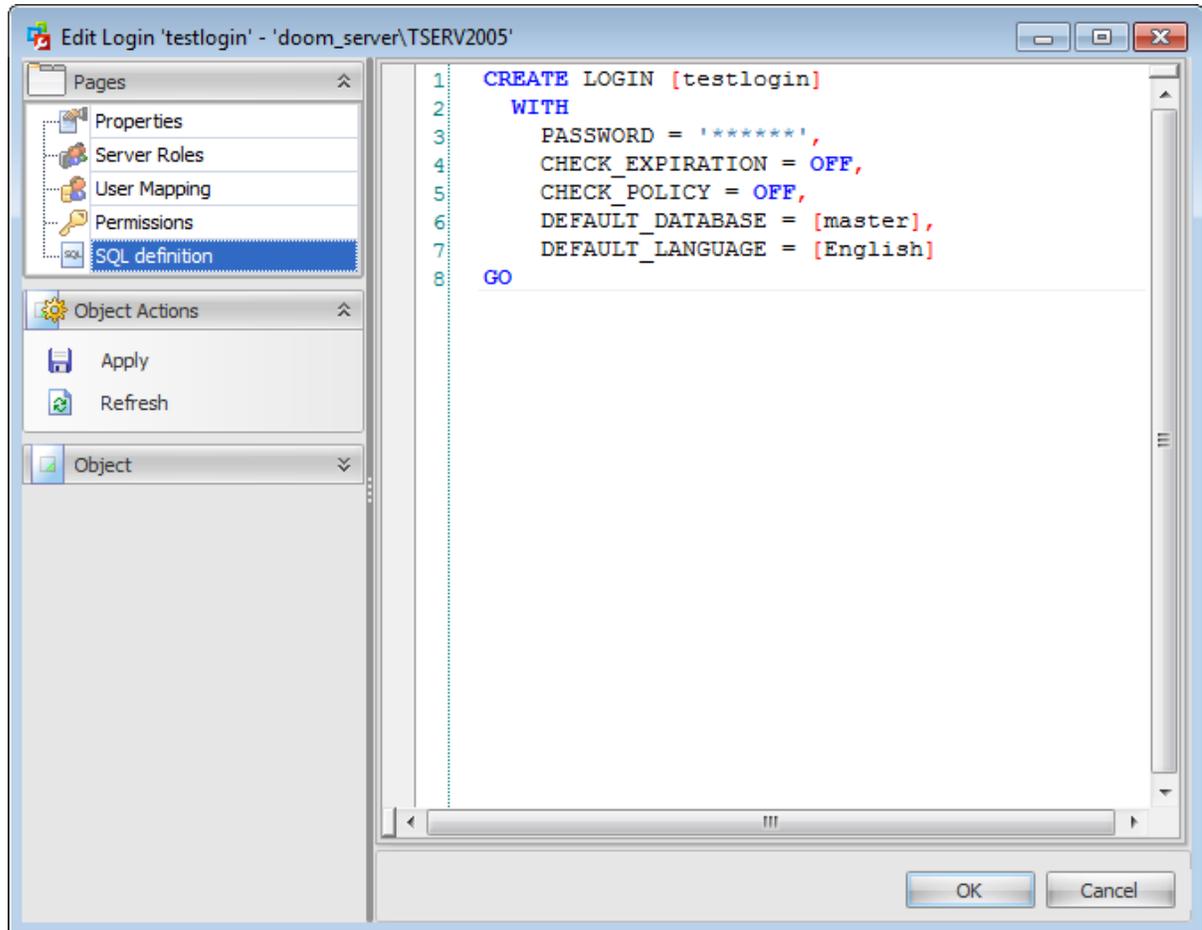
Select a **Securable type** in the *Permissions* tab on the left of the window to be displayed in the main working window.



To learn more about setting permissions for a login see the [Server Permissions](#) topic.

### 7.1.1.6 Viewing SQL definition

This tab allows you to view the object DDL.



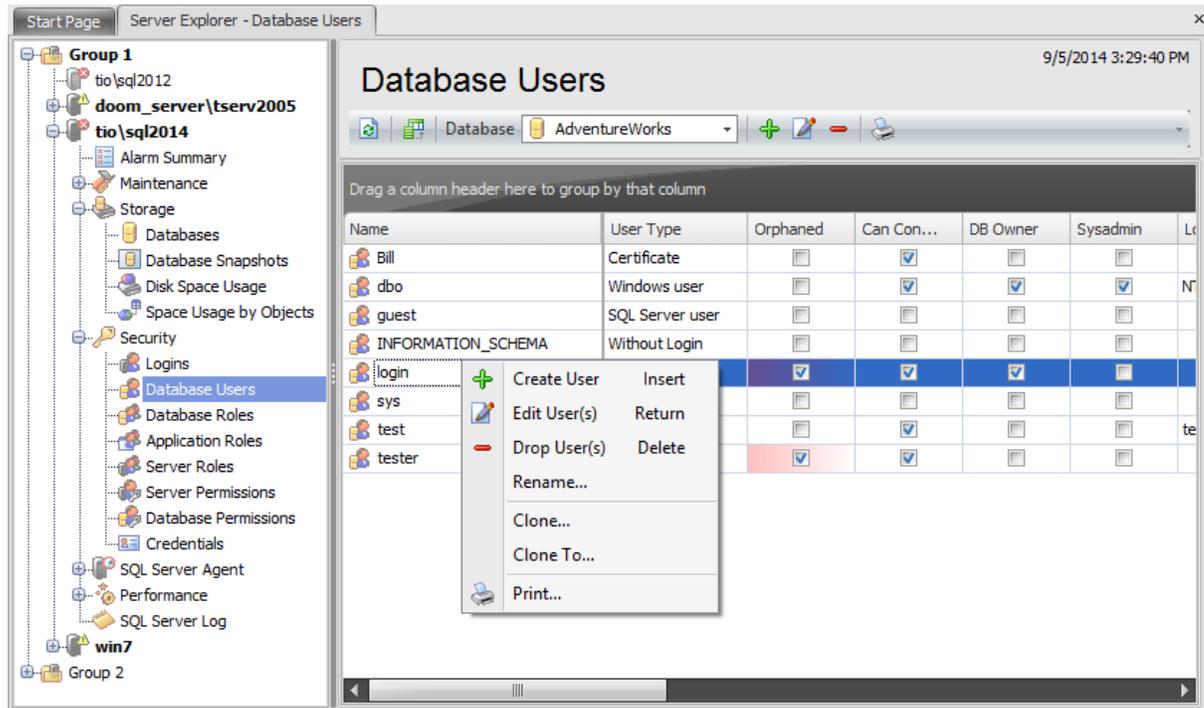
## 7.2 Database users

A **Database User** is a database-level security principal enabling object access permission control at the finest level of granularity. A user represents a single SQL Server login within the scope of the database in which the user is defined.

Use the context menu to:

-  [create](#) a new user
-  [edit](#) the selected user(s)
-  drop selected user(s)
- rename selected user
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a user, 'Enter' button to edit user (s) and 'Delete' button to drop user(s).



To view database users select a database from the **Database** drop-down list which is situated on the toolbar.

The list displays the existing database users as a grid with the following columns: *Name*, *User type*, *Orphaned*, *Can Connect*, *DB Owner*, *Sysadmin*, *Login*, *Certificate*, *Asymmetric Key*, *Default Schema*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Database User in the [User Editor](#), double-click it in the grid. Alternatively, you can right-click the Database User alias and select the **Edit User(s)** context menu item.

**See also:**[Logins](#)[Database roles](#)[Application Roles](#)[Server Roles](#)[Server Permissions](#)[Database Permissions](#)

## 7.2.1 User Editor

**User Editor** allows you to define user properties and membership. It opens automatically when you create a new user and is available on editing an existing one.

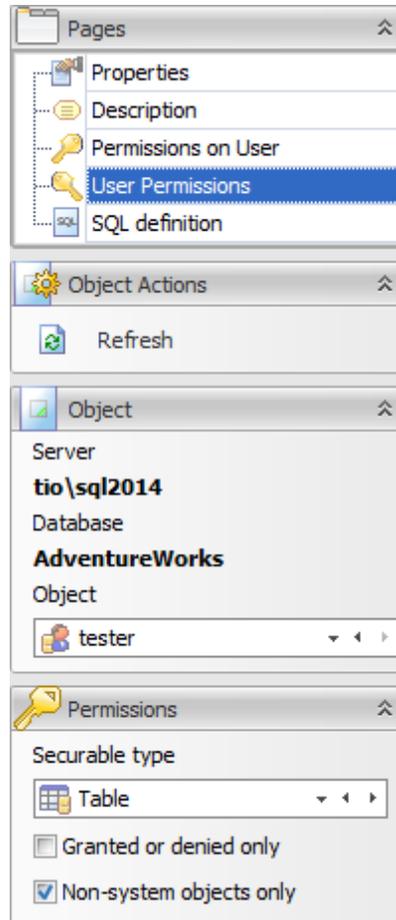
To open a user in **User Editor**, double-click it in the user list. Alternatively, you can right-click the user alias and select the **Edit User** context menu item.

The list of **User Editor** tabs is displayed on the [navigation bar](#):

- [Setting database user properties](#)
- [Editing description](#)
- [Setting permissions on user](#)
- [Setting user permissions](#)
- [Setting permissions](#)
- [Viewing SQL definition](#)

### 7.2.1.1 Using Navigation bar

The **User Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **User Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Database

In this field you can find out the name of the database which user you are editing.

##### Object

This field displays the database user that is currently being edited. However, you can

change the edited database user by using the drop-down list of users or by clicking the arrow-left or arrow-right buttons.

### Permissions

This group of options appears on setting permissions.

Select the database **Securable Type** from the drop-down list.

**Granted or denied only**

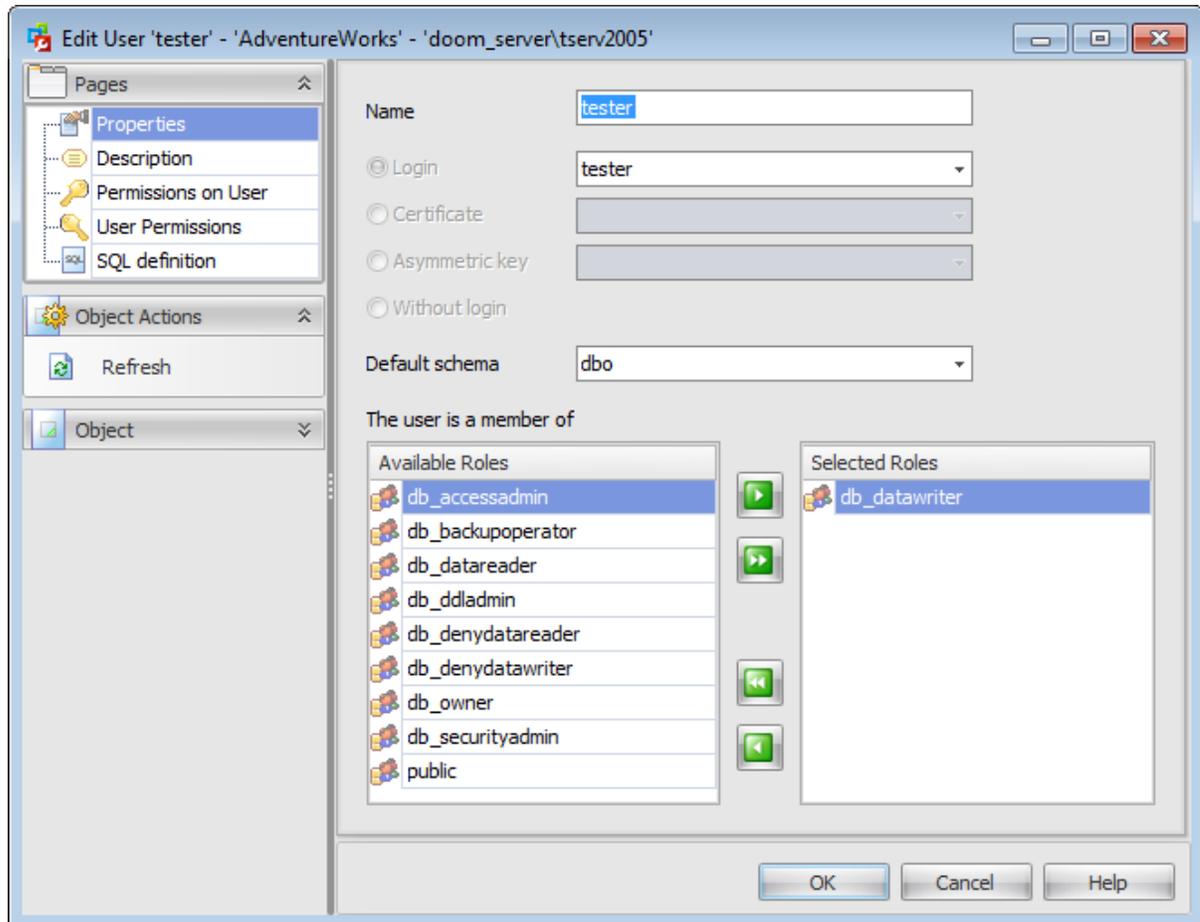
Check this option for displaying only granted permissions or those securables that have been set granted or denied.

**Non-system objects only**

Check this option for displaying only non-system objects in the list of securables.

### 7.2.1.2 Setting database user properties

Use the **Properties** section of the **User Editor** to create/edit a user and specify its attributes.



#### **Name**

Specifies the name by which the user is identified inside this database.

#### **Login**

If this option is selected, use the drop-down list to specify the SQL Server [login](#) for which the database user is being created.

#### **Certificate**

If this option is selected, use the drop-down list to specify the certificate for which the database user is being created.

#### **Asymmetric key**

If this option is selected, use the drop-down list to specify the asymmetric key for which the database user is being created.

#### **Without login**

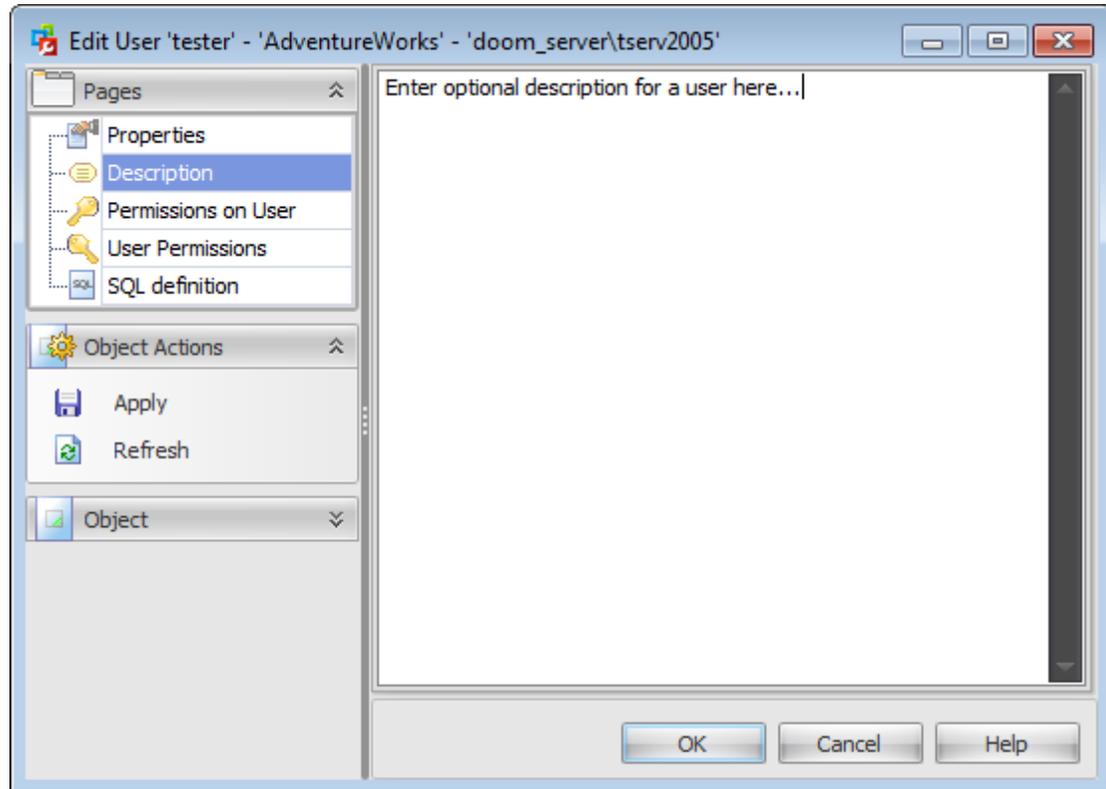
This option specifies that the user should not be mapped to any of existing [logins](#).

**Default schema**

Use the drop-down list to specify the first schema that will be searched by the server when it resolves the names of objects for this database user.

### 7.2.1.3 Editing description

The **Description** tab allows you to view and edit the comment for the user (optional).



You can save changes made in this area by clicking the **Apply** item on the [Navigation bar](#).

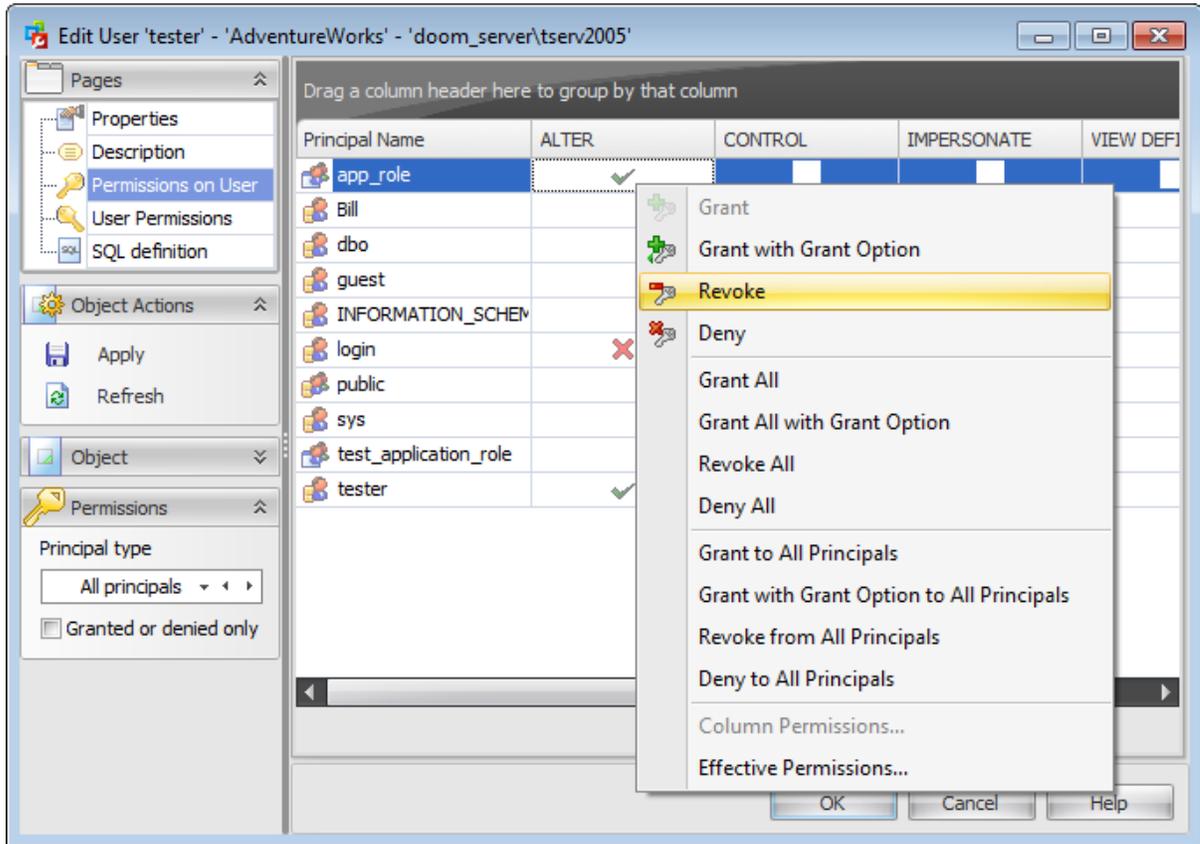
**Note:** Changing object description is performed by means of `sp_updateextendedproperty 'MS_Description'`.

#### 7.2.1.4 Setting permissions on user

This section allows you to grant permissions for another users and roles on the user being edited.

This section is available only when editing an existing user.

Use the [navigation bar](#) to select a **Principal type** so that only principals of the selected type are displayed in the grid.



The **Principal Name** column contains the list of [users](#), [database roles](#) or [application roles](#) according to the selection in the **Principal type** combo-box.

Each subsequent column corresponds to the permission which can be granted on the user being edited:

- *Alter*
- *Control*
- *Impersonate*
- *View definition*

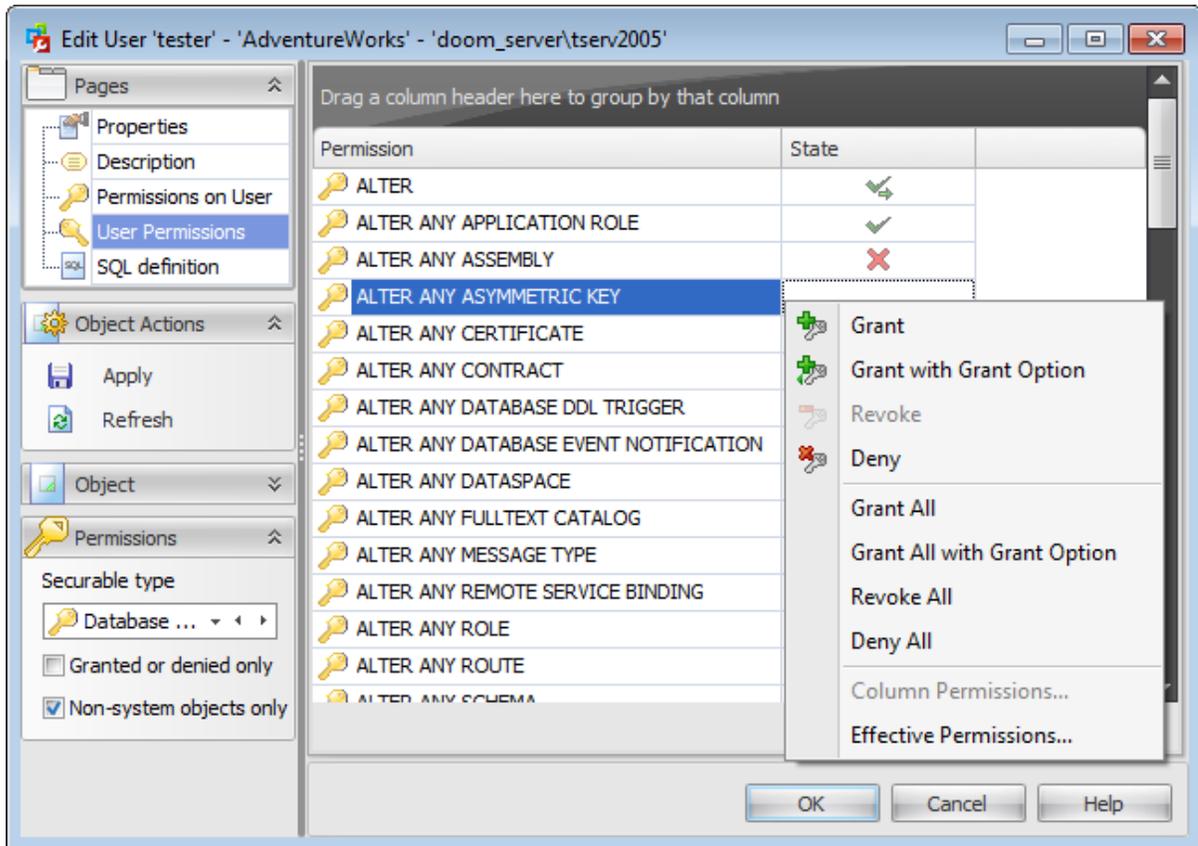
To learn more about setting permissions on a user see the [Database Permissions](#) topic.

### 7.2.1.5 Setting user permissions

This section allows you to grant different permissions to the user.

This section is available only when editing an existing user.

Use the [navigation bar](#) to select a **Securable type**.



The **Permission** column contains the list of objects of the selected type; each subsequent column corresponds to the permission which can be granted on the selected object:

*State* (for database permissions);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition, View change tracking* (for tables);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition* (for views, inline table-valued functions);

*Execute, Alter, Control, Take ownership, View definition* (for procedures, extended procedures);

*Select, References, Alter, Control, Take ownership, View definition* (for table-valued functions);

*Execute, References, Alter, Control, Take ownership, View definition* (for scalar functions, aggregate functions, XML schema collections);

*Select, Insert, Update, Delete, Execute, Control, Take ownership, View definition* (for synonyms);

*Execute, References, Control, Take ownership, View definition* (for data types);

*Select, Insert, Update, Delete, Execute, References, Alter, Control, Take ownership, View definition* (for schemas);

*Alter, Control, Take ownership, View definition* (for [roles](#), routes, remote service bindings);

*Alter, Control, Impersonate, View definition* (for [users](#));

*Alter, Control, View definition* (for application roles);

*References, Alter, Control, Take ownership, View definition* (for assemblies, full-text catalogs, certificates, asymmetric keys, symmetric keys, message types, contracts);

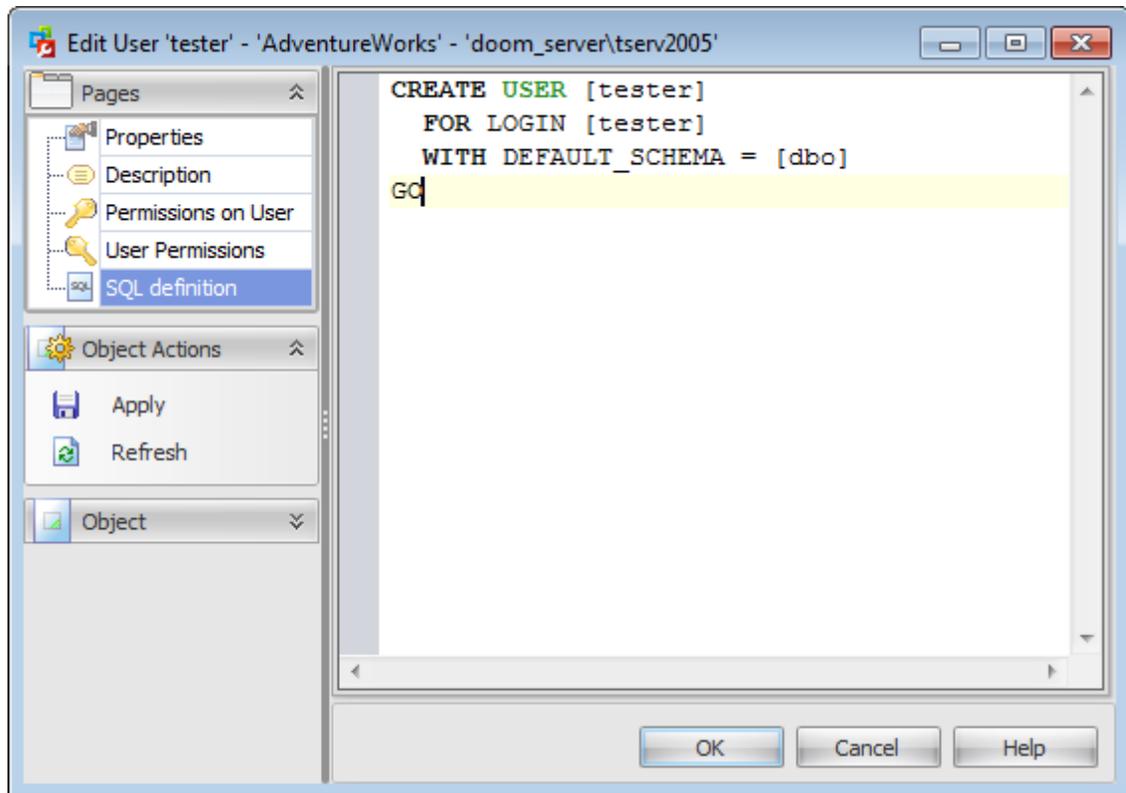
*Select, Receive, References, Alter, Control, Take ownership, View definition* (for queues);

*Send, Alter, Control, Take ownership, View definition* (for services).

To learn more about setting permissions for a user see the [Database Permissions](#) topic.

### 7.2.1.6 Viewing SQL definition

This tab allows you to view the object DDL.



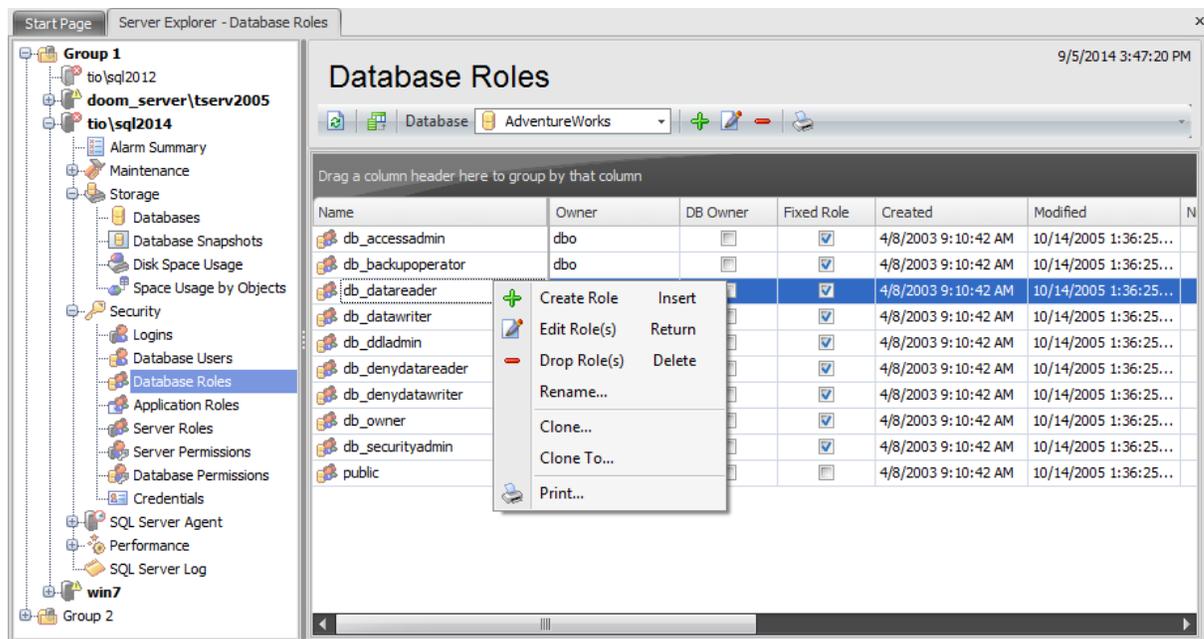
## 7.3 Database roles

All [users](#) interact with SQL Server within the context of a database-level principal - **Database Role**. A user can belong to multiple groups and have multiple roles, and the operations that are permitted by each role determine the actions that the user can perform. It is necessary to create SQL Server database roles when a group of users needs to perform a specified set of activities in the database.

Use the context menu or buttons on the work area toolbar to manage the list of database roles:

-  [create](#) a new database role
-  [edit](#) selected database role(s)
-  drop selected database role(s)
- rename selected database role
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a role, 'Enter' button to edit role (s) and 'Delete' button to drop role(s).



To view database roles select a database from the **Database** drop-down list which is situated on the toolbar.

The list displays the existing database roles as a grid with the following columns: *Name*, *Owner*, *DB Owner*, *Fixed Role*, *Created*, *Modified*, *Number of Assigned Roles*, *Number of Members*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a Database Role in the [Role Editor](#), double-click it in the grid. Alternatively, you can right-click the Database Role alias and select the **Edit Role(s)** context menu item.

**See also:**[Logins](#)[Database users](#)[Application Roles](#)[Server Roles](#)[Server Permissions](#)[Database Permissions](#)

### 7.3.1 Database Role Editor

**Database Role Editor** allows you to define role properties and membership. It opens automatically when you create a new role and is available on editing an existing one.

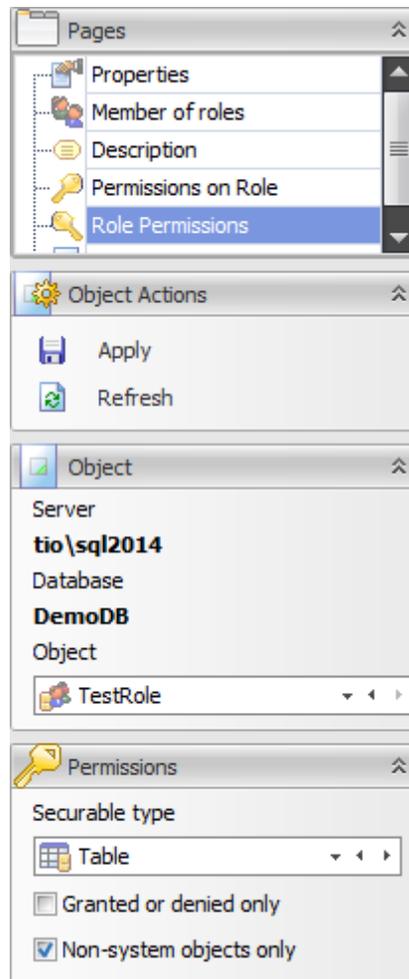
To open a role in the **Database Role Editor**, double-click it in the roles list. Alternatively, you can right-click the role alias and select the **Edit Role** context menu item.

The list of **Database Role Editor** tabs is displayed on the [navigation bar](#):

- [Setting database role properties](#)
- [Editing description](#)
- [Defining role membership](#)
- [Setting permissions on role](#)
- [Setting role permissions](#)
- [Viewing SQL definition](#)

### 7.3.1.1 Using Navigation bar

The **Database Role Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Database Role Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Database

In this field you can find out the name of the database which role you are editing.

**Object**

This field displays the database role that is currently being edited. However, you can change the edited database role by using the drop-down list of roles or by clicking the arrow-left or arrow-right buttons.

**Permissions**

This group of options appears on setting permissions.

Select the database **Securable Type** from the drop-down list.

 **Granted or denied only**

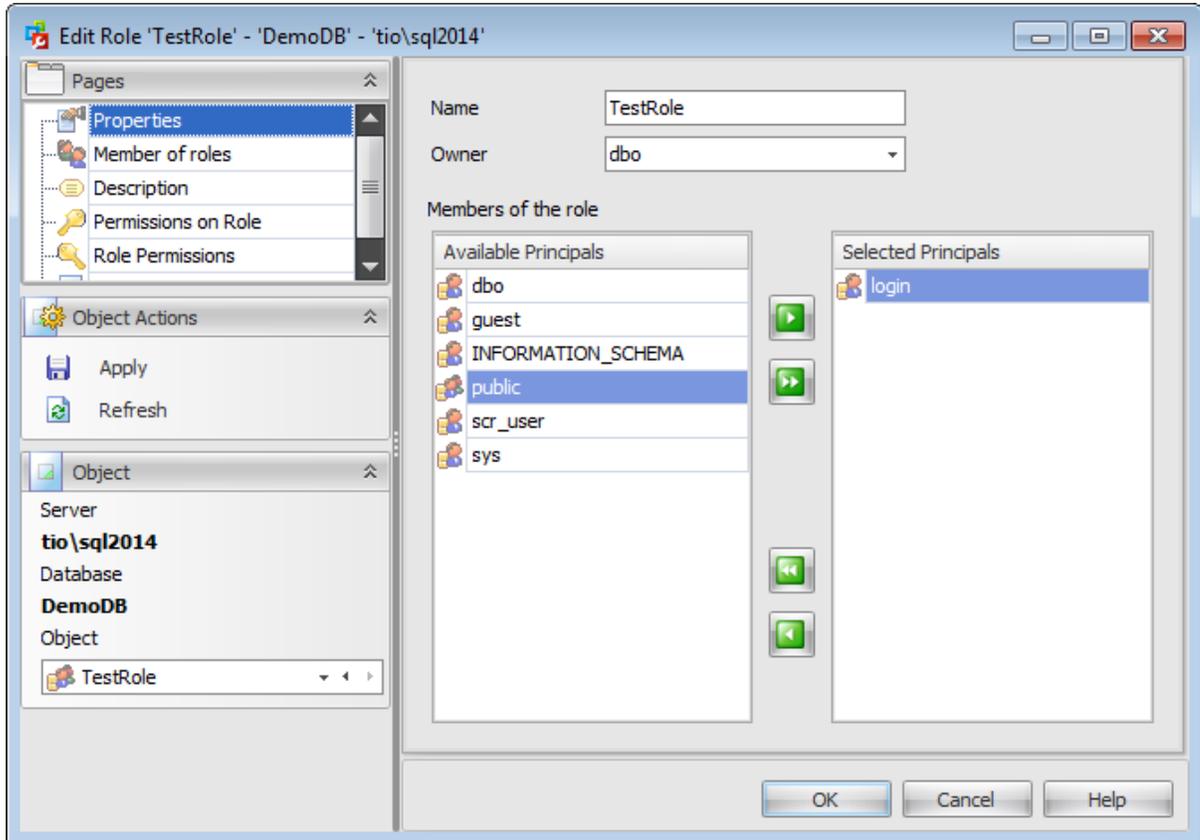
Check this option for displaying only granted permissions or those securables that have been set granted or denied.

 **Non-system objects only**

Check this option for displaying only non-system objects in the list of securables.

### 7.3.1.2 Setting database role properties

Use the **Properties** tab of **Role Editor** to create/edit a role and specify its properties and membership.



#### **Name**

Enter a name for the new role, or view the name of the role being edited.

#### **Owner**

Select a database owner from the drop-down list.

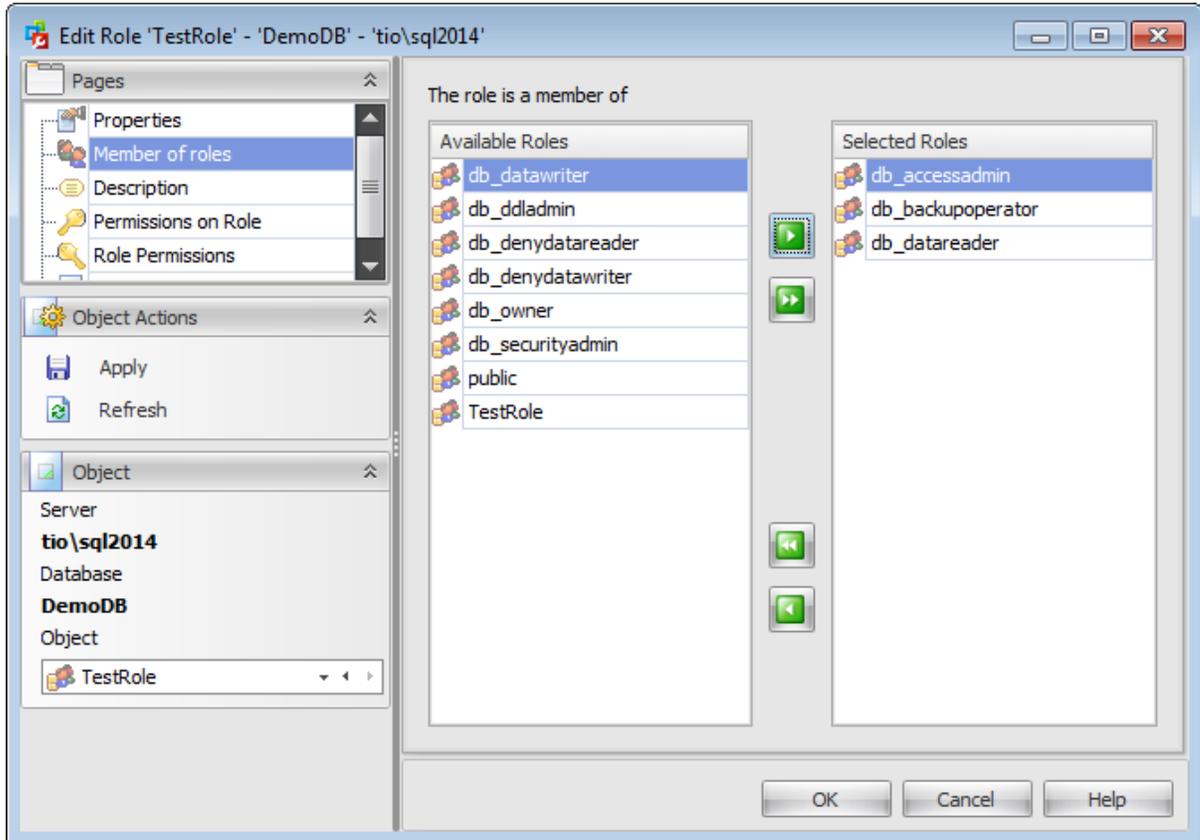
#### **Members of the role**

Add members to the role, i.e. add the [users](#) (or other roles) that will inherit the permissions of the role.

To add a user, you need to move it from the **Available Principals** list to the **Selected Principals** list. Use the   buttons or drag-and-drop operations to move the users from one list to another.

### 7.3.1.3 Defining role membership

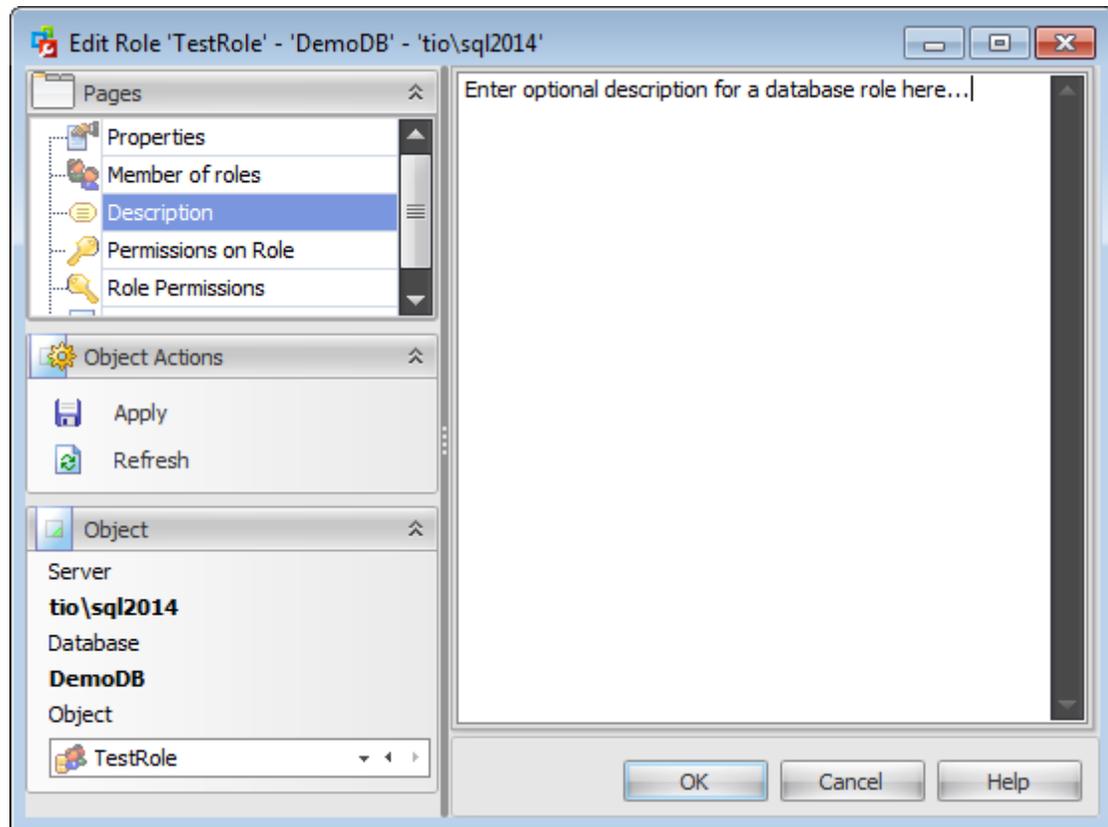
The **Member of roles** tab of **Role Editor** allows you to define role membership, i.e. to select the built-in database role(s) the current role will be a member of. The selected roles determine the tasks that can be performed through the current role.



To select a role, you need to move it from the **Available Roles** list to the **Selected Roles** list. Use the     buttons or drag-and-drop operations to move the roles from one list to another.

#### 7.3.1.4 Editing description

The **Description** tab allows you to view and edit the comment for the user (optional).



You can save changes made in this area by clicking the **Apply** item on the [Navigation bar](#).

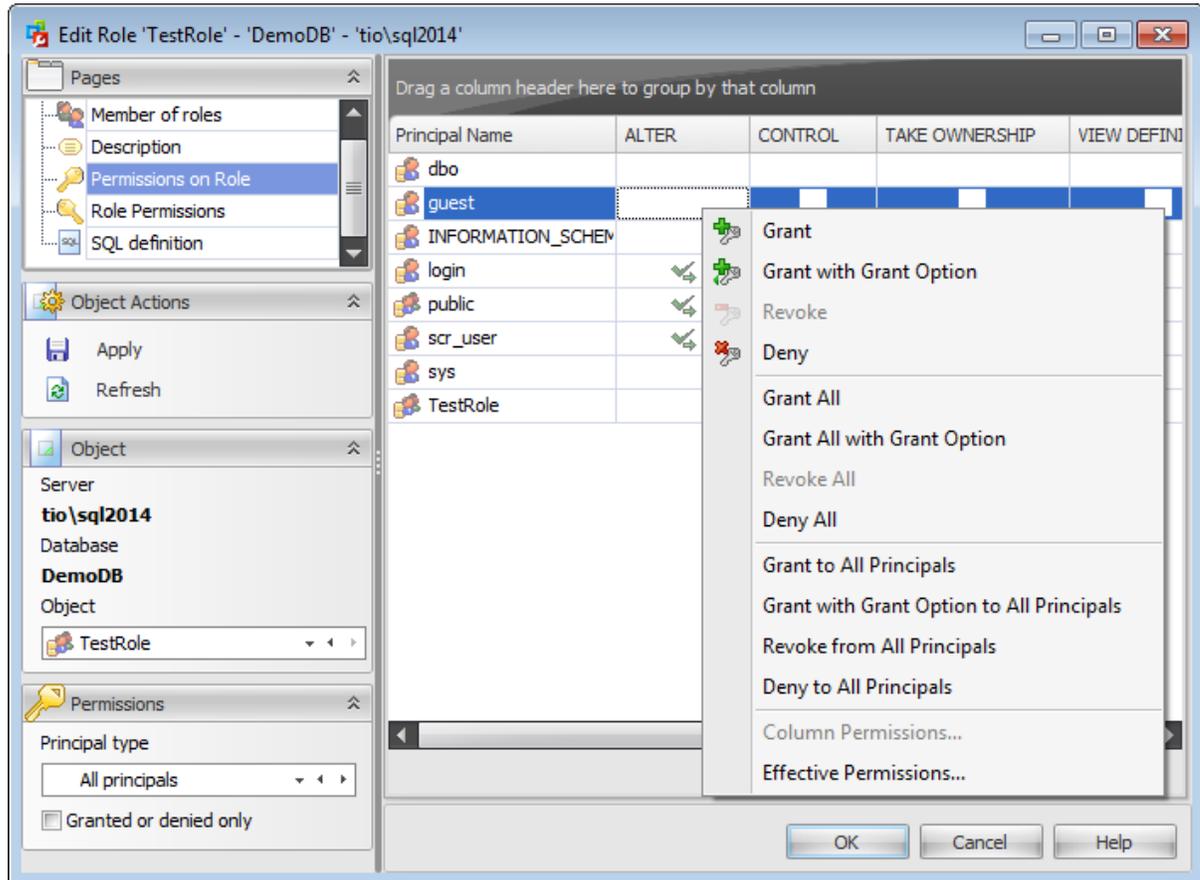
**Note:** Changing object description is performed by means of `sp_updateextendedproperty 'MS_Description'`.

### 7.3.1.5 Setting permissions on role

This section allows you to grant permissions for another users and roles on the role being edited.

This section is available only when editing an existing role.

Use the [navigation bar](#) to select a **Principal type** so that only principals of the selected type are displayed in the grid.



The **Principal Name** column contains the list of [users](#), [database roles](#) or [application roles](#) according to the selection in the **Principal type** combo-box.

Each subsequent column corresponds to the permission which can be granted on the role being edited:

- *Alter*
- *Control*
- *Impersonate*
- *View definition*

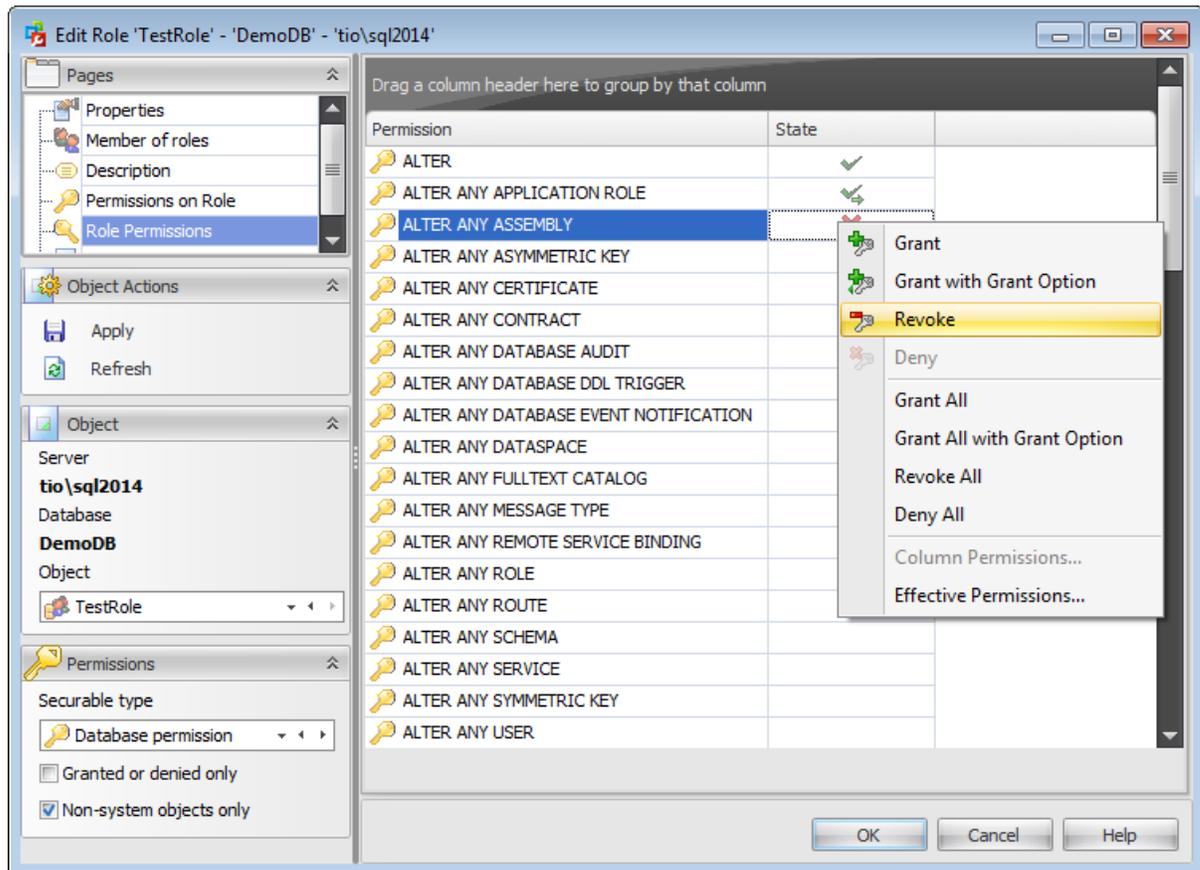
To learn more about setting permissions on a database role see the [Database Permissions](#) topic.

### 7.3.1.6 Setting role permissions

This section allows you to grant different permissions to the role.

This section is available only when editing an existing role.

Select a **Securable type** in the *Permissions* tab on the left of the window to be displayed in the main working window.



The **Securable object** column contains the list of objects of the selected type (or **Permission** column in case of *Database permission* securable type); each subsequent column corresponds to the permission which can be granted on the selected object: *State* (for database permissions);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition, View change tracking* (for tables);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition* (for views, inline table-valued functions);

*Execute, Alter, Control, Take ownership, View definition* (for procedures, extended procedures);

*Select, References, Alter, Control, Take ownership, View definition* (for table-valued functions)

*Execute, References, Alter, Control, Take ownership, View definition* (for scalar functions, aggregate functions, XML schema collections)

*Select, Insert, Update, Delete, Execute, Control, Take ownership, View definition* (for synonyms);

*Execute, References, Control, Take ownership, View definition* (for data types);

*Select, Insert, Update, Delete, Execute, References, Alter, Control, Take ownership, View definition (for schemas);*

*Alter, Control, Take ownership, View definition (for [roles](#), routes, remote service bindings);*

*Alter, Control, Impersonate, View definition (for [users](#));*

*Alter, Control, View definition (for application roles);*

*References, Alter, Control, Take ownership, View definition (for assemblies, full-text catalogs, certificates, asymmetric keys, symmetric keys, message types, contracts, );*

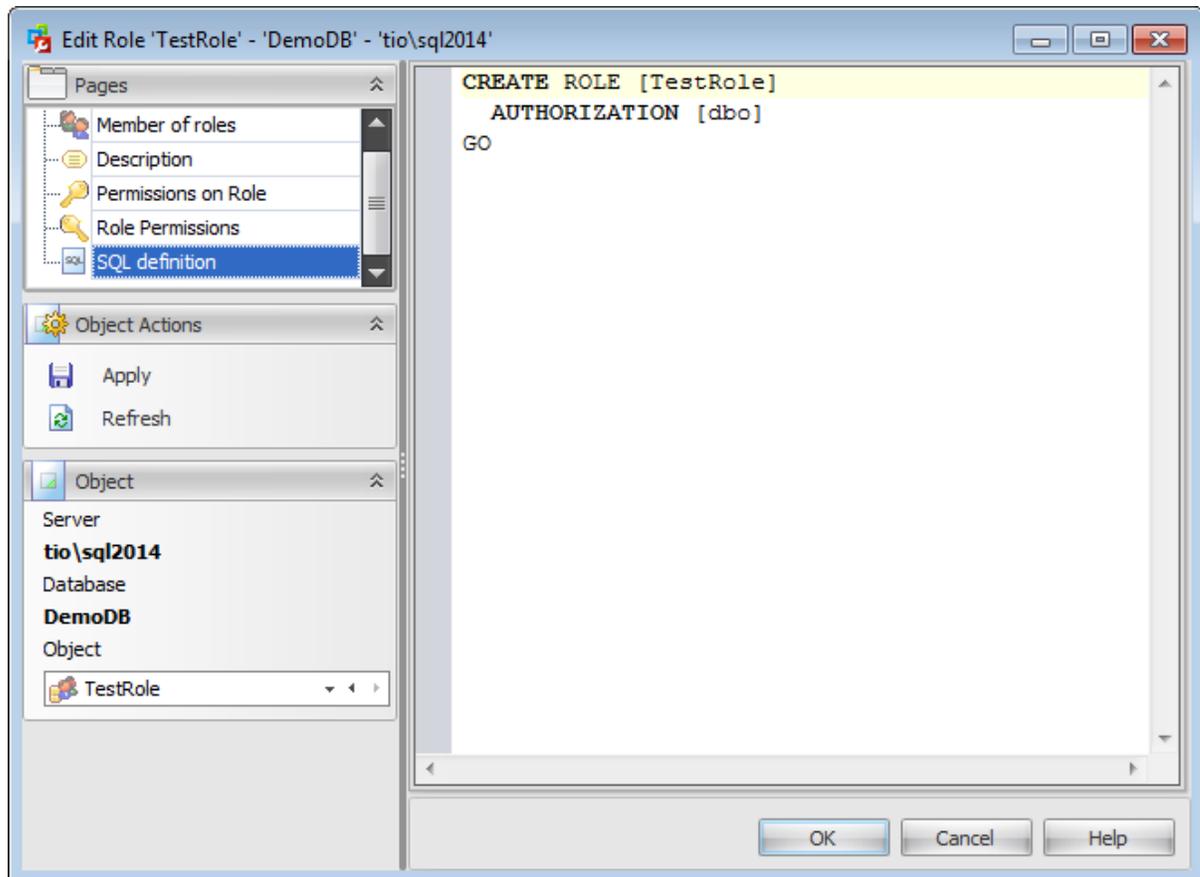
*Select, Receive, References, Alter, Control, Take ownership, View definition (for queues);*

*Send, Alter, Control, Take ownership, View definition (for services).*

To learn more about setting permissions for a login see the [Database Permissions](#) topic.

### 7.3.1.7 Viewing SQL definition

This tab allows you to view the object DDL.



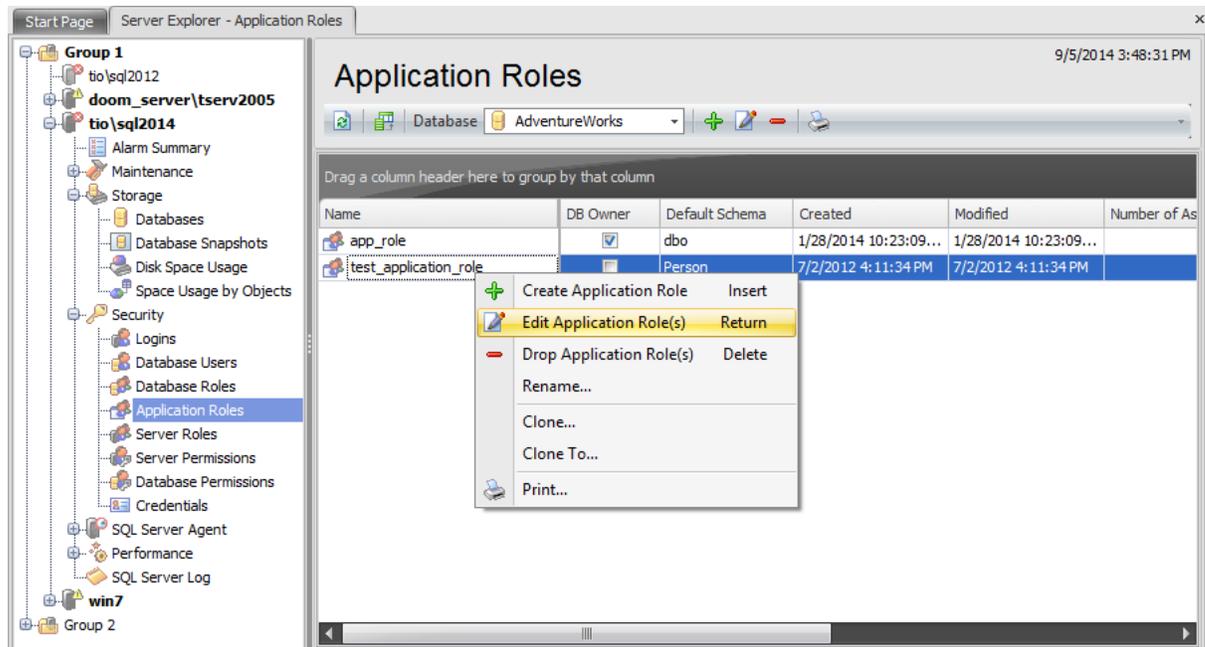
## 7.4 Application Roles

An **Application Role** is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application. Unlike database roles, application roles contain no members and are inactive by default. Because application roles are a database-level principal, they can access other databases only through permissions granted in those databases to guest. Therefore, any database in which guest has been disabled will be inaccessible to application roles in other databases.

Use the context menu or buttons on the work area toolbar to manage application roles:

-  [create](#) a new application role
-  [edit](#) selected application role(s)
-  drop selected application role(s)
- rename selected application role
- create the object copy on the same server
- clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a role, 'Enter' button to edit role (s) and 'Delete' button to drop role(s).



To view Application Roles select a database from the **Database** drop-down list which is situated on the toolbar.

The list displays the existing application roles as a grid with the following columns: *Name*, *DB Owner*, *Default Schema*, *Created*, *Modified*, *Number of Assigned Roles*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open an Application Role in the [Application role Editor](#), double-click it in the grid.

Alternatively, you can right-click the Application Role alias and select the **Edit Application Role(s)** context menu item.

---

**See also:**

[Logins](#)

[Database users](#)

[Database roles](#)

[Server Roles](#)

[Server Permissions](#)

[Database Permissions](#)

### 7.4.1 Application Role Editor

**Application role Editor** allows you to define role properties and membership. It opens automatically when you create a new application role and is available on editing an existing one.

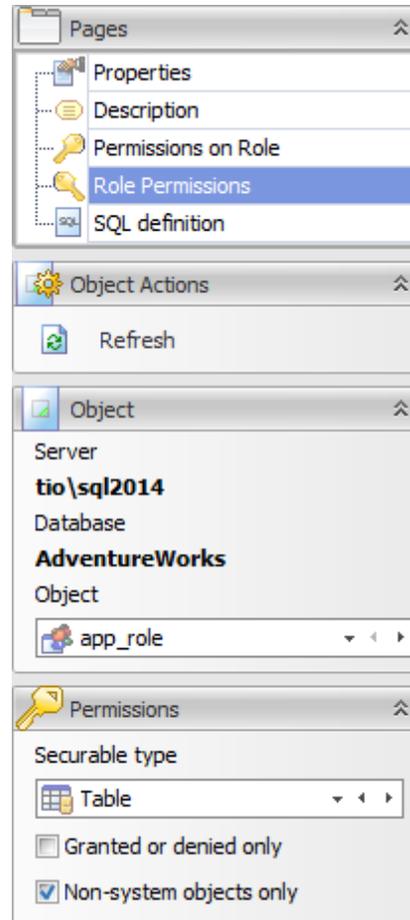
To open an application role in the **Application role Editor**, double-click it in the application roles list. Alternatively, you can right-click the application role alias and select the **Edit Application Role** context menu item.

The list of **Application Role Editor** tabs is displayed on the [navigation bar](#):

- [Setting application role properties](#)
- [Editing description](#)
- [Setting permissions on role](#)
- [Setting permissions](#)
- [Viewing SQL definition](#)

### 7.4.1.1 Using Navigation bar

The **Application Role Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Application Role Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Database

In this field you can find out the name of the database which application role you are editing.

##### Object

This field displays the application role that is currently being edited. However, you can change the edited application role by using the drop-down list of roles or by clicking the arrow-left or arrow-right buttons.

### Permissions

This group of options appears on setting permissions.

Select the database **Securable Type** from the drop-down list.

**Granted or denied only**

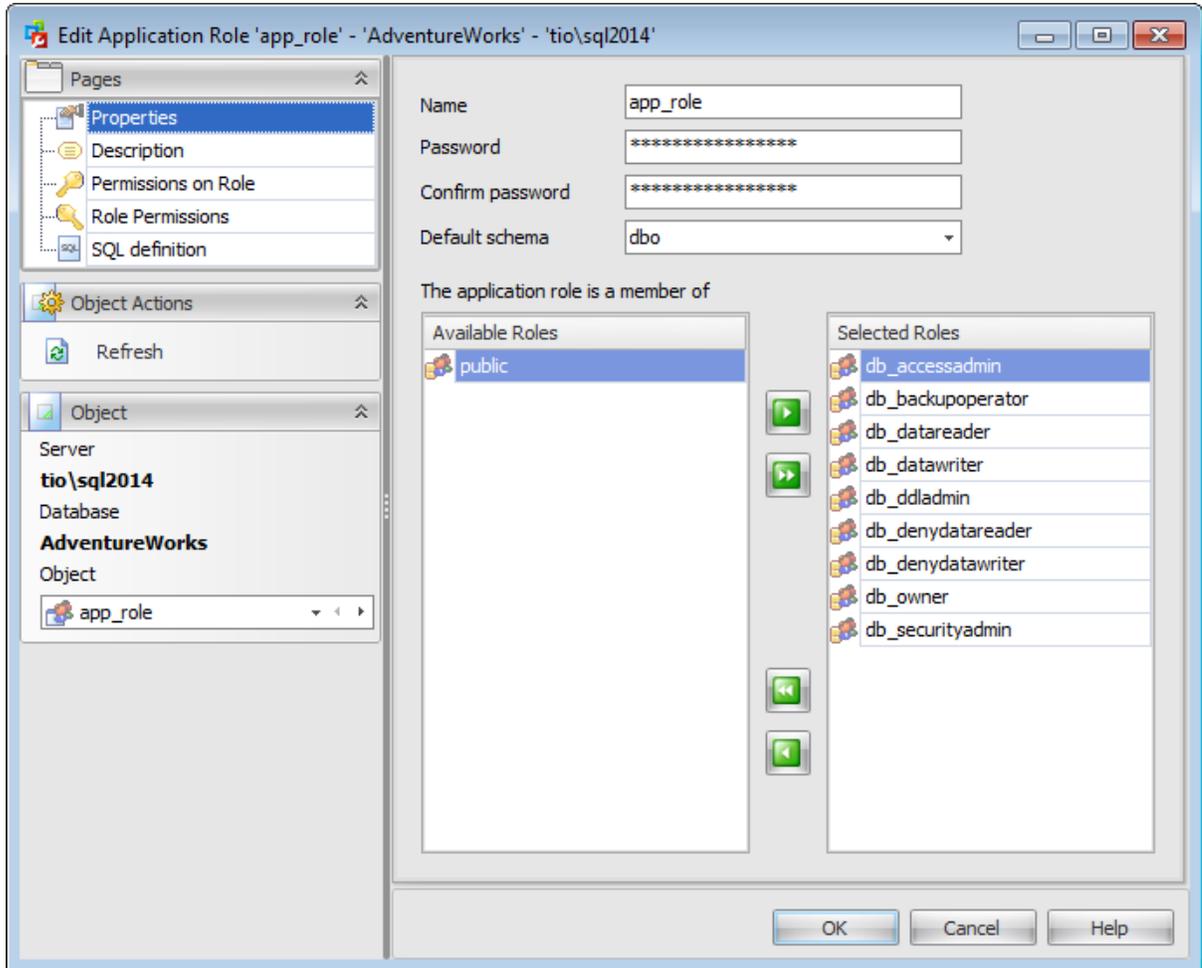
Check this option for displaying only granted permissions or those securables that have been set granted or denied.

**Non-system objects only**

Check this option for displaying only non-system objects in the list of securables.

### 7.4.1.2 Setting application role properties

Use the **Properties** tab of the **Application Role Editor** to create/edit an application role and specify its properties and membership.



#### **Name**

Enter a name for the new application role, or edit the name of the application role being edited.

#### **Password**

Specify a password for the new application role, or edit the password of the application role being edited and then confirm this password in the corresponding field.

#### **Default Schema**

Specify the first schema that will be searched by the server when it resolves the names of objects for this role.

**Note:** If this field remains empty, the application role will use DBO as its default schema. The schema defined may not exist in the database.

#### **The application role is member of**

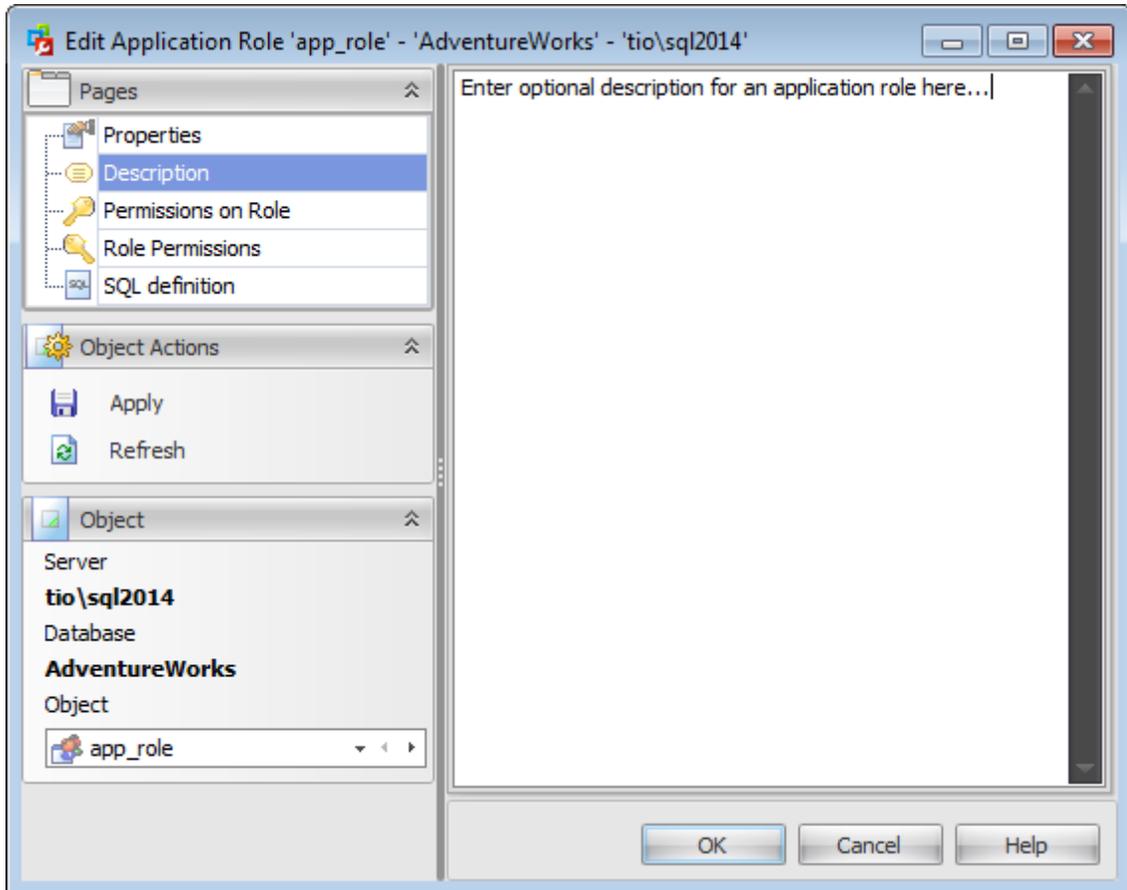
Define role membership, i.e. select the built-in database role(s) the current role will be a

member of. The selected roles determine the tasks that can be performed through the current role.

To select a role, you need to move it from the **Available Roles** list to the **Selected Roles** list. Use the     buttons or drag-and-drop operations to move the roles from one list to another.

### 7.4.1.3 Editing description

The **Description** tab allows you to view and edit the comment for the user (optional).



You can save changes made in this area by clicking the **Apply** item on the [Navigation bar](#).

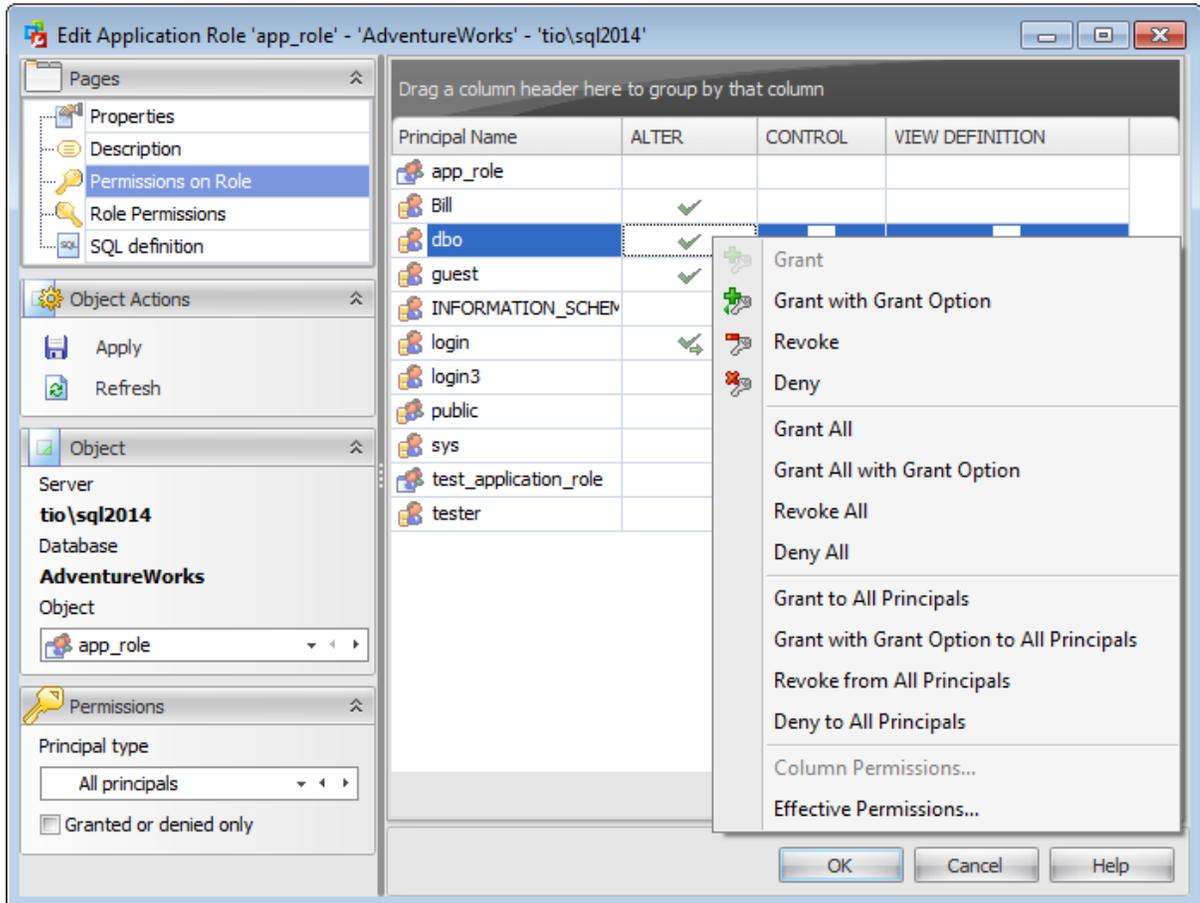
**Note:** Changing object description is performed by means of `sp_updateextendedproperty 'MS_Description'`.

#### 7.4.1.4 Setting permissions on role

This section allows you to grant permissions for another users and roles on the application role being edited.

This section is available only when editing an existing application role.

Use the [navigation bar](#) to select a **Principal type** so that only principals of the selected type are displayed in the grid.



The **Principal Name** column contains the list of [users](#), [database roles](#) or [application roles](#) according to the selection in the **Principal type** combo-box.

Each subsequent column corresponds to the permission which can be granted on the application role being edited:

- *Alter*
- *Control*
- *View definition*

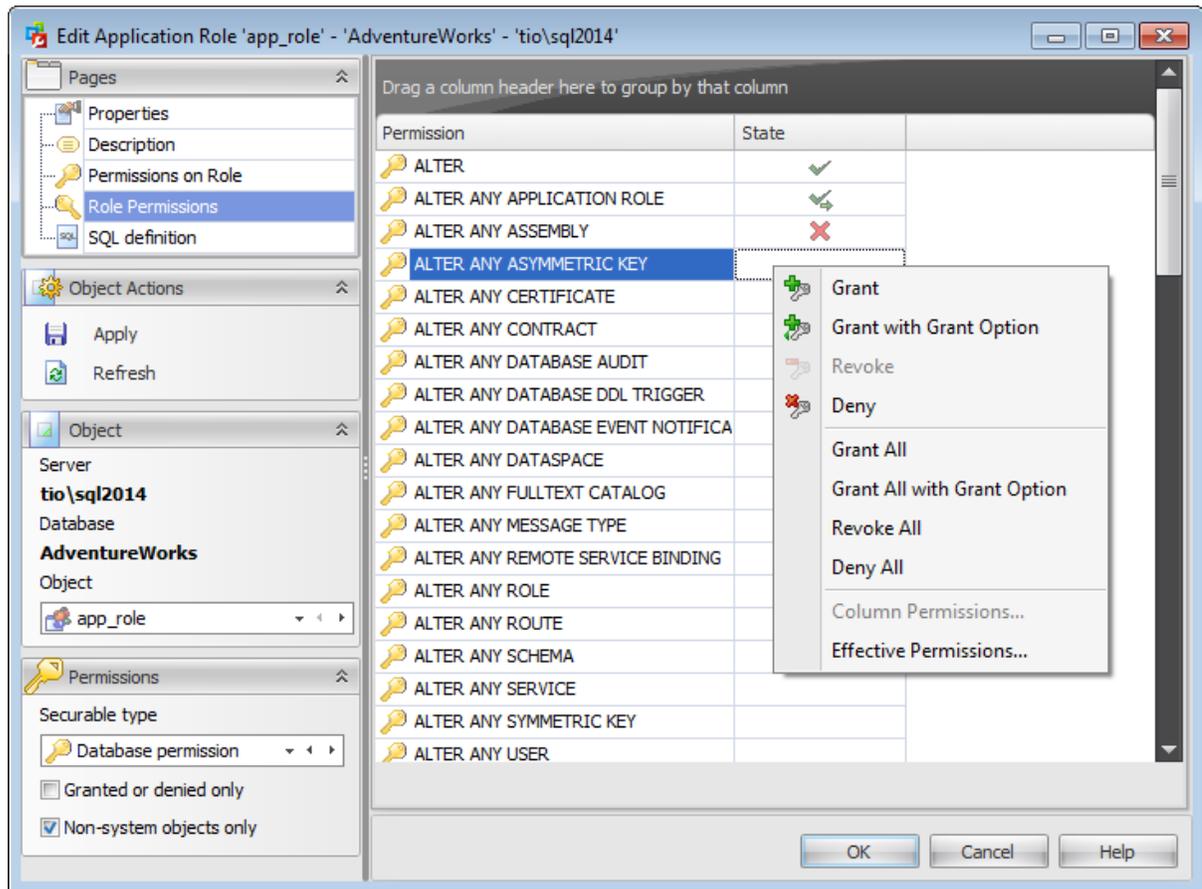
To learn more about setting permissions on an application role see the [Database Permissions](#) topic.

### 7.4.1.5 Setting permissions

This section allows you to grant different permissions to the application role.

This section is available only when editing an existing application role.

Select a **Securable type** in the *Permissions* tab on the left of the window to be displayed in the main working window.



The **Securable object** column contains the list of objects of the selected type (or **Permission** column in case of *Database permission* securable type); each subsequent column corresponds to the permission which can be granted on the selected object: *State* (for database permissions);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition, View change tracking* (for tables);

*Select, Insert, Update, Delete, References, Alter, Control, Take ownership, View definition* (for views, inline table-valued functions);

*Execute, Alter, Control, Take ownership, View definition* (for procedures, extended procedures);

*Select, References, Alter, Control, Take ownership, View definition* (for table-valued functions)

*Execute, References, Alter, Control, Take ownership, View definition* (for scalar functions, aggregate functions, XML schema collections)

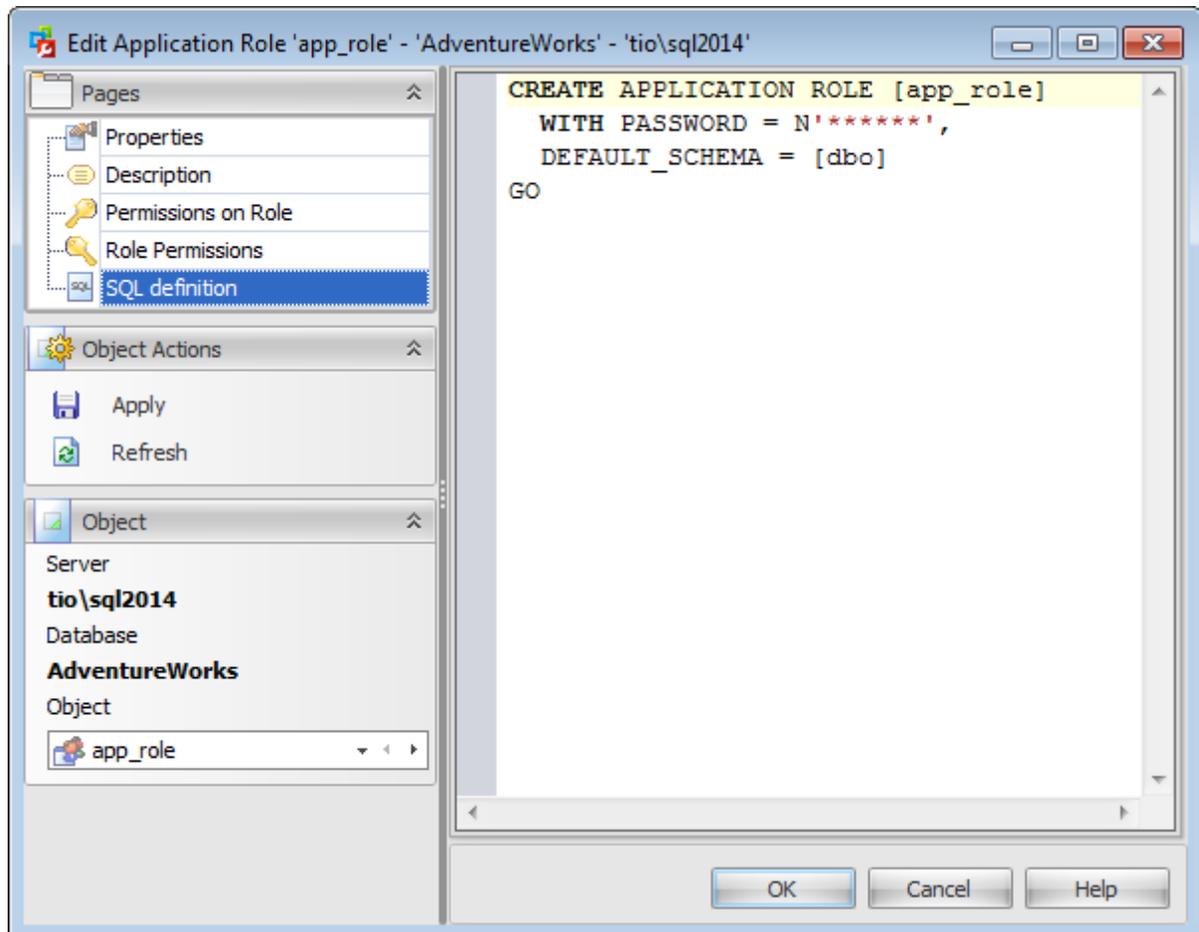
*Select, Insert, Update, Delete, Execute, Control, Take ownership, View definition* (for synonyms);

*Execute, References, Control, Take ownership, View definition (for data types);  
Select, Insert, Update, Delete, Execute, References, Alter, Control, Take ownership, View  
definition (for schemas);  
Alter, Control, Take ownership, View definition (for [roles](#), routes, remote service  
bindings);  
Alter, Control, Impersonate, View definition (for [users](#));  
Alter, Control, View definition (for application roles);  
References, Alter, Control, Take ownership, View definition (for assemblies, full-text  
catalogs, certificates, asymmetric keys, symmetric keys, message types, contracts, );  
Select, Receive, References, Alter, Control, Take ownership, View definition (for queues);  
Send, Alter, Control, Take ownership, View definition (for services).*

To learn more about setting permissions for a login see the [Database Permissions](#) topic.

#### 7.4.1.6 Viewing SQL definition

This tab allows you to view the object DDL.

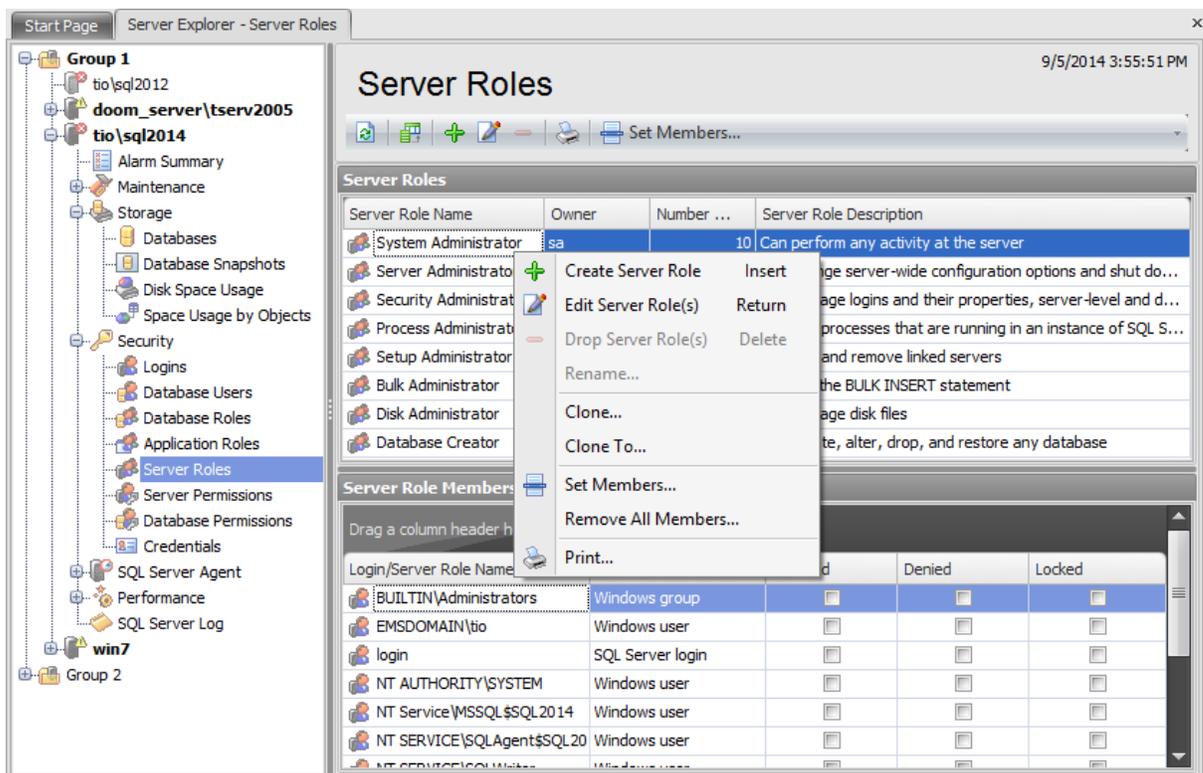


## 7.5 Server Roles

The **Server Roles** section allows you to browse and [edit](#) the list of SQL Server **Roles**, and to define [login\\_membership](#) for each role.

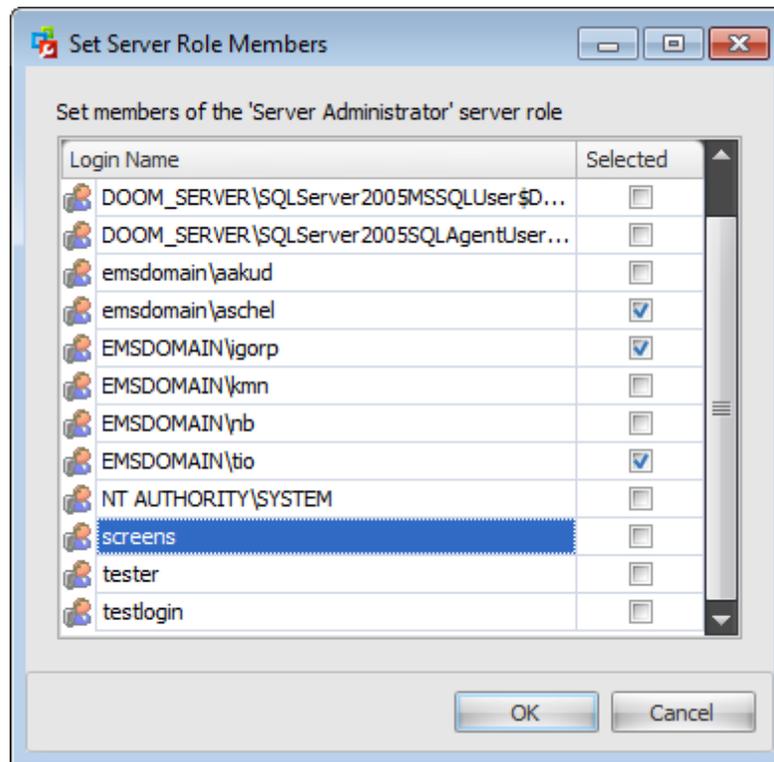
Use the context menu to:

- + [create](#) a new server role
-  [edit](#) the selected server role(s)
- drop selected server role(s)
- rename the selected server role
- create the object copy on the same server
- clone the object copy to another server
-  [set members](#) of the role
  - *remove all members* from the role
-  [create and print](#) the report



The upper list displays the server roles as a grid with the following columns: *Server Role Name*, *Number of Members*, *Server Role Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

If you choose a **Set Members...** item of the context menu the following window appears.



To set a member of the selected server role choose it among the list of [logins](#) and check the corresponding box.

For your convenience the *Select All* and *Unselect All* functions are implemented in the **context menu** of the members list.

The lower area allows you to define login membership, i.e. to select the [logins](#) that will belong to the selected server role. The Server Roles determine the tasks that can be performed through the selected [login](#). You can [edit](#) any login by double-clicking its alias.

---

**See also:**

[Logins](#)

[Database users](#)

[Database roles](#)

[Application Roles](#)

[Server Permissions](#)

[Database Permissions](#)

## 7.5.1 Server Roles Editor

**Server role Editor** allows you to define role properties and membership. It opens automatically when you create a new application role and is available on editing an existing one.

To open a server role in the **Server role Editor**, double-click it in the server roles list. Alternatively, you can right-click the server role alias and select the **Edit Server Role** context menu item.

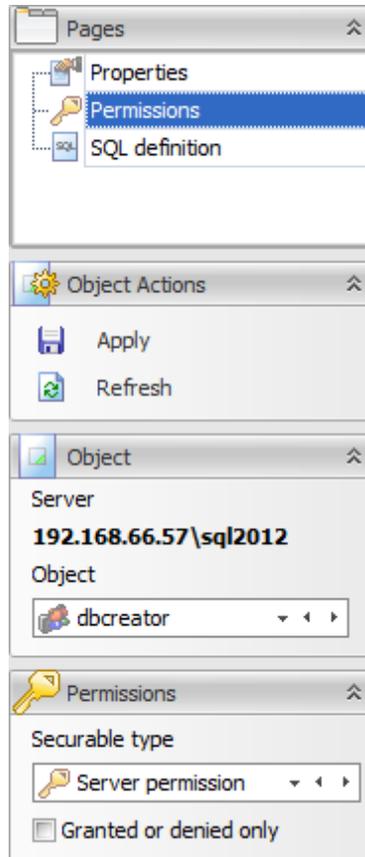
**Note:** The **Server Role Editor** is available only for SQL Server version 2012 and higher.

The list of **Server Role Editor** tabs is displayed on the [navigation bar](#):

- [Setting server role properties](#)
- [Defining server role membership](#)
- [Setting permissions](#)
- [Viewing SQL definition](#)

### 7.5.1.1 Using Navigation bar

The **Server Role Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Server Role Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Database

In this field you can find out the name of the database which application role you are editing.

##### Object

This field displays the server role that is currently being edited. However, you can change the edited server role by using the drop-down list of roles or by clicking the arrow-left or

arrow-right buttons.

### Permissions

This group of options appears on setting permissions.

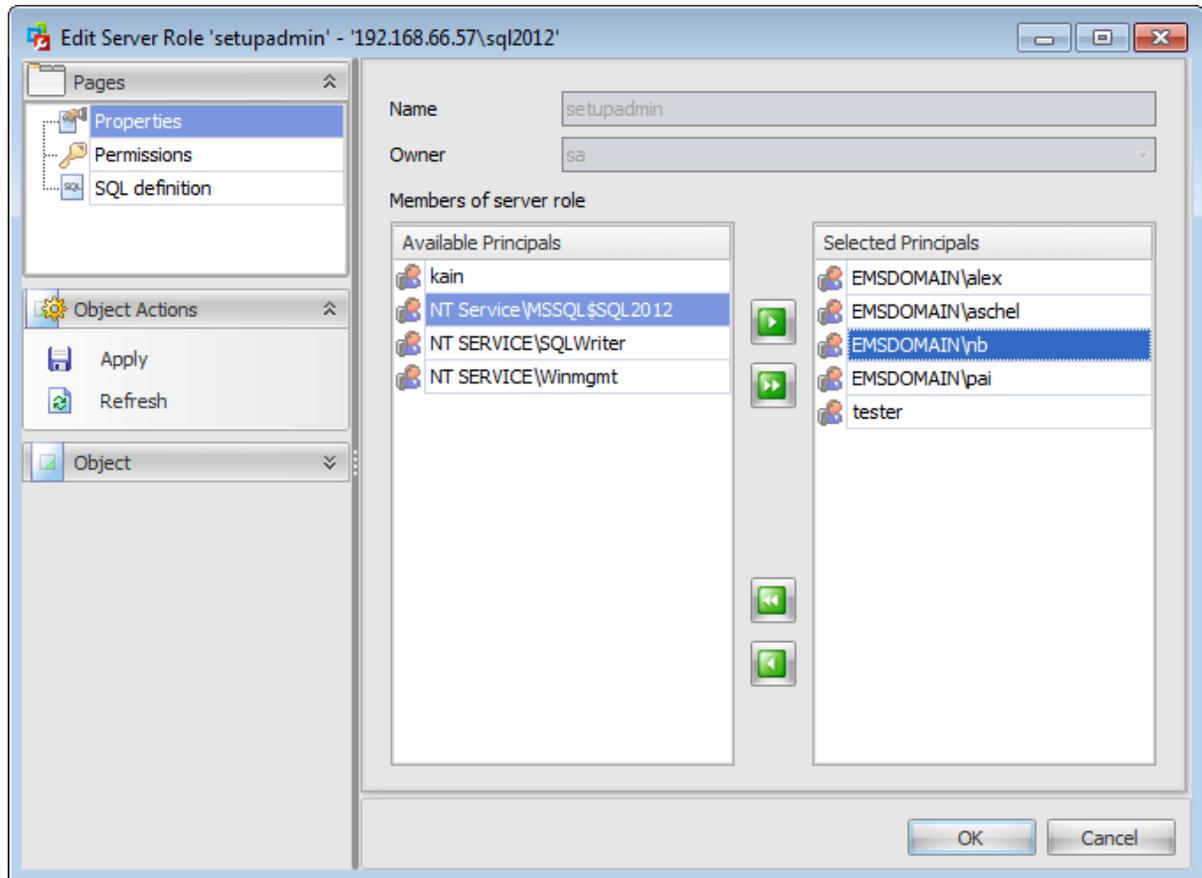
Select the database **Securable Type** from the drop-down list.

**Granted or denied only**

Check this option for displaying only granted permissions or those securables that have been set granted or denied.

### 7.5.1.2 Setting server role properties

Use the **Properties** tab of **Server Role Editor** to create/edit a role and specify its properties and membership.



#### Name

Enter a name for the new server role, or view the name of the role being edited.

#### Owner

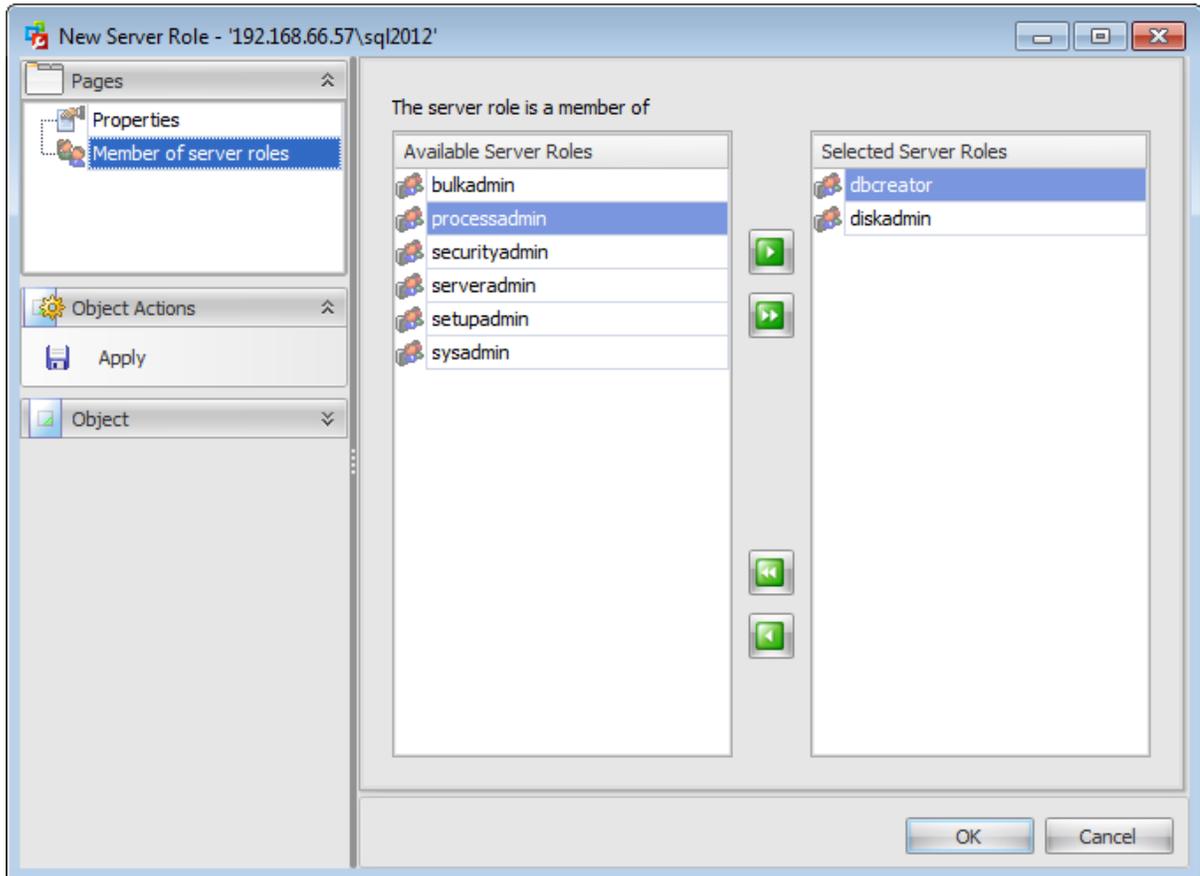
Select an owner from the drop-down list of [logins](#).

#### Members of server role

Add members to the role, i.e. add [logins](#) that will inherit the permissions of the role. To add a login, you need to move it from the **Available Principals** list to the **Selected Principals** list. Use the     buttons or drag-and-drop operations to move the users from one list to another.

### 7.5.1.3 Defining server role membership

The **Member of server roles** tab of **Server role Editor** allows you to define role membership, i.e. to select the built-in server role(s) the current role will be a member of. The selected roles determine the tasks that can be performed through the current role. This tab is available on creating a new server role only.



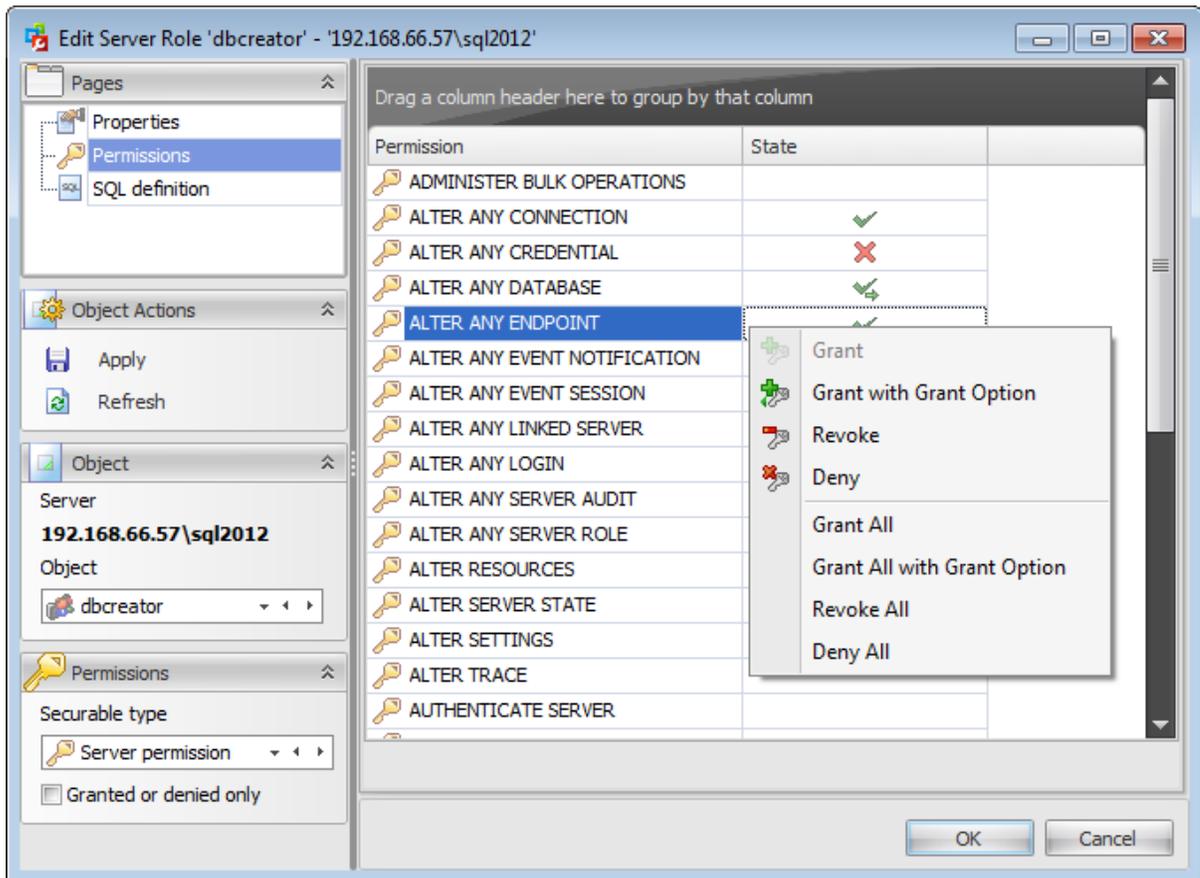
To select a role, you need to move it from the **Available Server Roles** list to the **Selected Server Roles** list. Use the     buttons or drag-and-drop operations to move the roles from one list to another.

#### 7.5.1.4 Setting permissions

This section allows you to grant different permissions to the server role.

This section is available only when editing an existing server role.

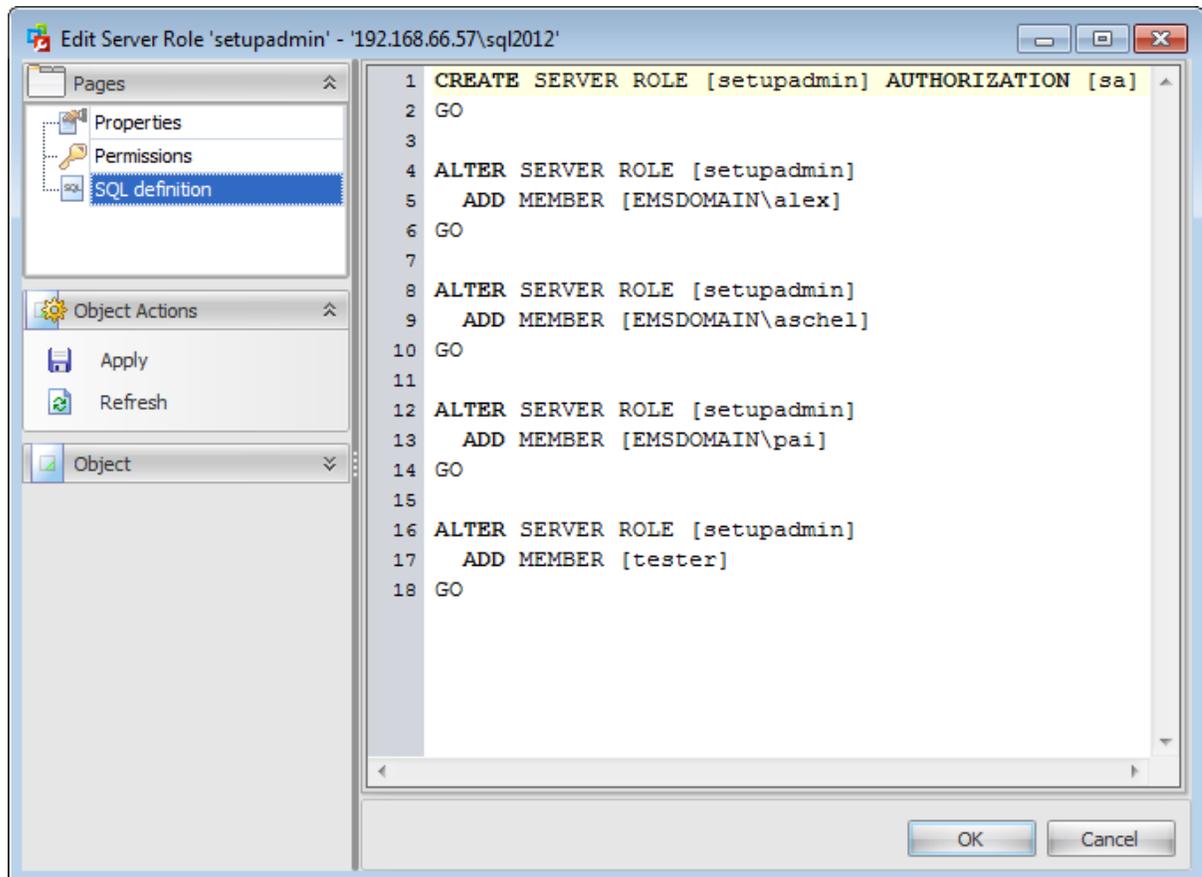
Select a **Securable type** in the *Permissions* tab on the left of the window to be displayed in the main working window.



To learn more about setting permissions for a server role see the [Server Permissions](#) topic.

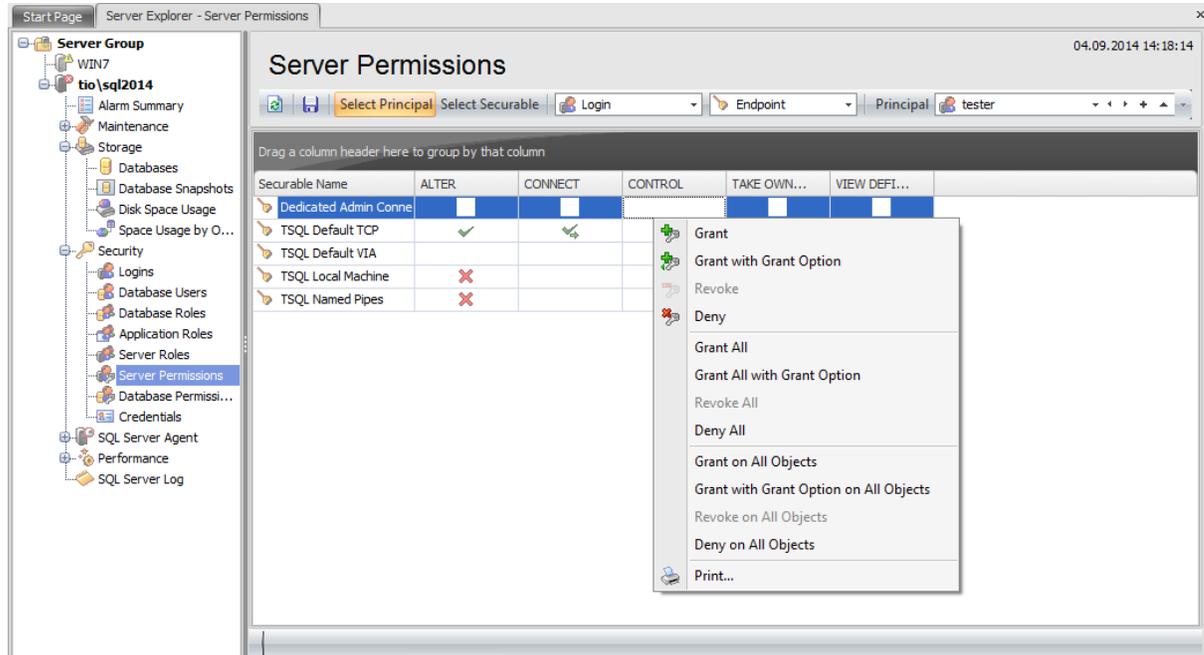
### 7.5.1.5 Viewing SQL definition

This tab allows you to view the object DDL.



## 7.6 Server Permissions

Use this page to manage server permissions.



### Toolbar

You can choose to **Select Principal** or **Select Securable** by clicking the corresponding buttons on the toolbar. In the first case you are to set *Server*, *Server Role* (for SQL Server 2012), *Login* or *Endpoint* permissions to the *Principal* ([login](#) or [server role](#) (for SQL Server 2012)) selected from the corresponding drop-down list.

If you click **Select Securable** then you are to select the server **Securable Type** from the drop-down list: *Server permission*, *Server Role*, *Login* or *Endpoint*. Then choose the securable name from the **Securable** drop-down list to set permissions for.

Click the  toolbar button if you want to view objects with granted or denied permissions only.

Right-click a cell next to the required permission to grant its execution privilege to the selected principal. The context menu of a cell contains possible permissions that can be granted:

-  *Grant*
-  *Grant with Grant Option*
-  *Revoke* (removes a previously granted permission)
-  *Deny*
  - *Grant All*
  - *Grant All with Grant Option*
  - *Revoke All*
  - *Deny All*
  - *Grant on All Objects / to All Principals*

- *Grant with Grant Option on All Objects / to All Principals*
- *Revoke on All Objects / from All Principals*
- *Deny on All Objects / to All Principals*

**Hint:** You can also assign privileges by double-clicking the respective cell - in this case the grant status is changed in the following order: *Grant -> Grant with Grant Option -> Deny -> Revoke*.

Use the  **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

When you grant permissions for principals the context menu allows you to:

-  [create](#) a new login
-  [edit](#) the selected login(s)
-  drop selected login(s)
-  [create](#) a new server role
-  [edit](#) the selected server role(s)
-  drop selected server role(s)

Availability:

**Full** version **Yes**

**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

---

**See also:**

[Logins](#)

[Database users](#)

[Database roles](#)

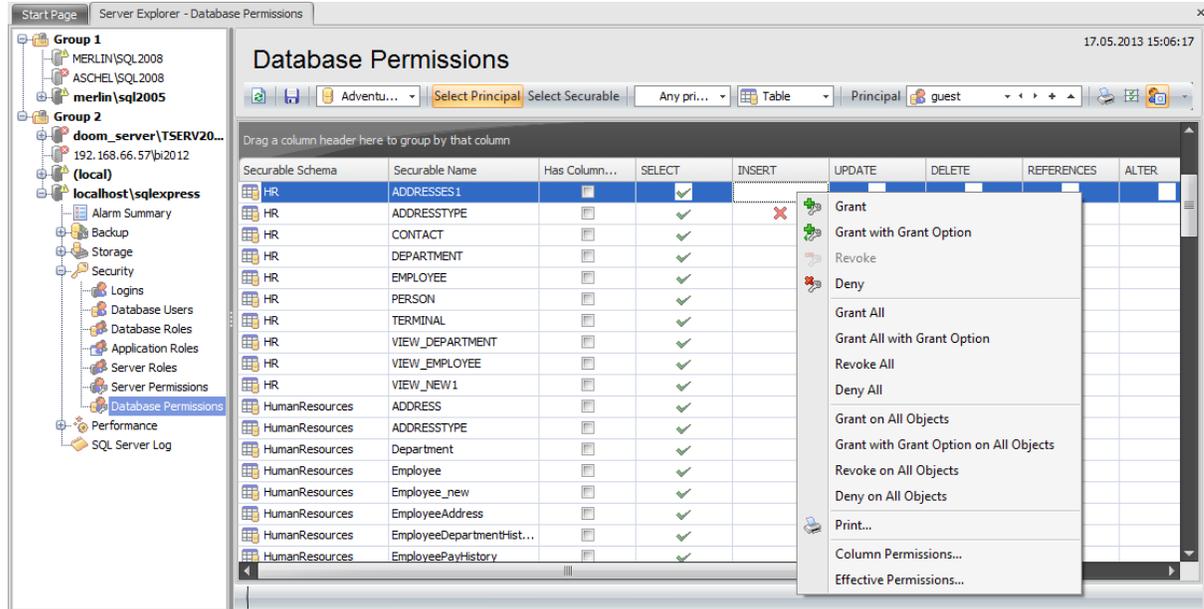
[Application Roles](#)

[Server Roles](#)

[Database Permissions](#)

## 7.7 Database Permissions

The **Database Permissions** node allows you to define privileges for the database principals and on execution of statements within the specified database.



### Toolbar



select a database which permissions you want to view.

Then select the **Securable Type** and **Principal Type** from the corresponding drop-down lists. After that choose to **Select Principal** or **Select Securable** by clicking the corresponding buttons on the toolbar. In the first case you are to select principals of the type set in the **Principal Type** field which you want to view database permissions for. The grid in this case will contain the list of securables to grant permissions on. The process of granting permissions is described in the [Permissions](#) section of the [User Editor](#).

If you click **Select Securable** then you are to select the **Securable** of the type defined in the **Securable Type** field from the drop-down list. The grid in this case will contain the list of principals ([roles](#), [users](#) or [application roles](#)) to grant permissions for.

Click the toolbar button if you want to view objects with granted or denied permissions only and the button if you want to view non-system objects only.

Right-click a cell next to the required login to grant it privileges to the selected securable. The context menu of a cell contains possible permissions that can be granted:

- Grant
- Grant with Grant Option
- Revoke (removes a previously granted permission)
- Deny
  - Grant All
  - Grant All with Grant Option
  - Revoke All

- *Deny All*
- *Grant on All Objects / to All Principals*
- *Grant with Grant Option on All Objects / to All Principals*
- *Revoke on All Objects / from All Principals*
- *Deny on All Objects / to All Principals*

**Hint:** You can also assign privileges by double-clicking the respective cell - in this case the grant status is changed in the following order: *Grant* -> *Grant with Grant Option* -> *Deny* -> *Revoke*.

Use the  **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

Using the context menu you can access the dialog for managing [column permissions](#) and viewing [effective permissions](#).

Availability:

**Full** version                      **Yes**

**Free** version                      **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

---

**See also:**

[Logins](#)

[Database users](#)

[Database roles](#)

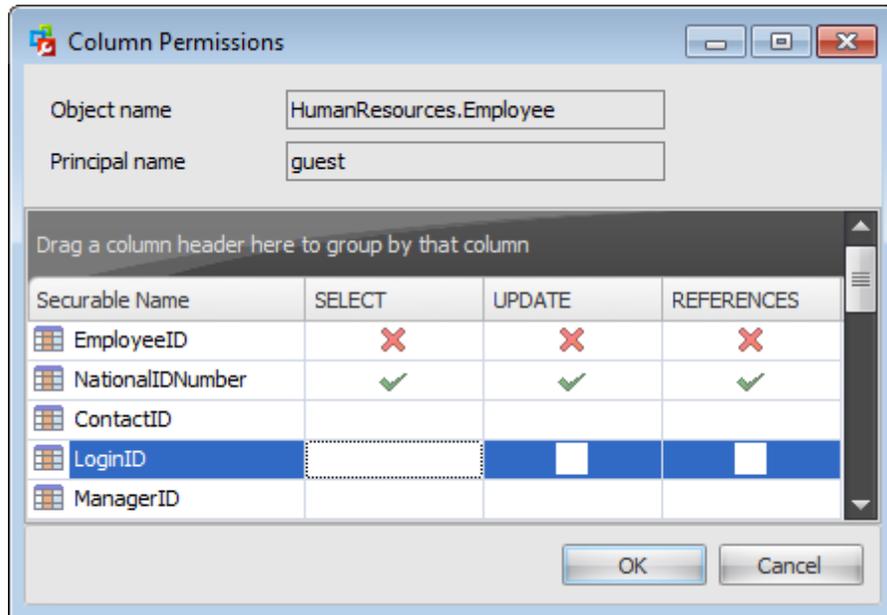
[Application Roles](#)

[Server Roles](#)

[Server Permissions](#)

### 7.7.1 Column permissions

Some objects allow granting permissions on columns. Right-click within [the list of permissions](#) and select *Column permission* from the context menu in order to grant permissions (*Select*, *Update* and *References*) on columns.



The **Column permissions** dialog displays the grid with the selected object columns and the privileges that can be granted to the selected principal.

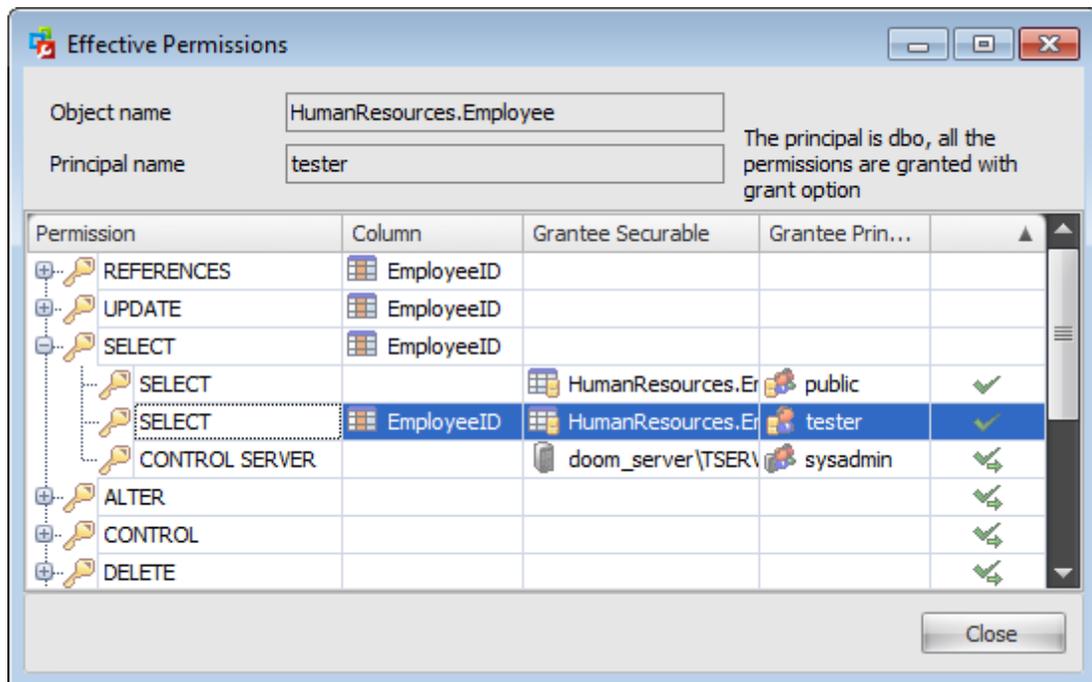
Use items of the context menu to grant/deny/revoke permissions on columns.

## 7.7.2 Effective permissions

The effective permissions that a principal has on a securable are the result of the explicit permissions defined for that principal on that securable, the permissions defined on the parents of the securable, and the permissions that the principal inherits through role or group membership.

Use the Effective Permissions dialog box to view the permissions that result when all the permissions of a principal on a securable are combined. The **Effective Permissions** dialog is read-only.

The grid in the dialog displays the entire list of permissions granted to the principal on a definite object in the form of a tree. For your convenience the **Expand / Collapse** buttons are available in the context menu of the grid.



## 7.8 Credentials

A **Credential** is a record containing the authentication information required to connect to a resource outside of SQL Server. Most credentials consist of a Windows login and password.

Credentials allow [users](#) that connect to SQL Server using SQL Authentication to connect to Windows or other resources outside of SQL Server.

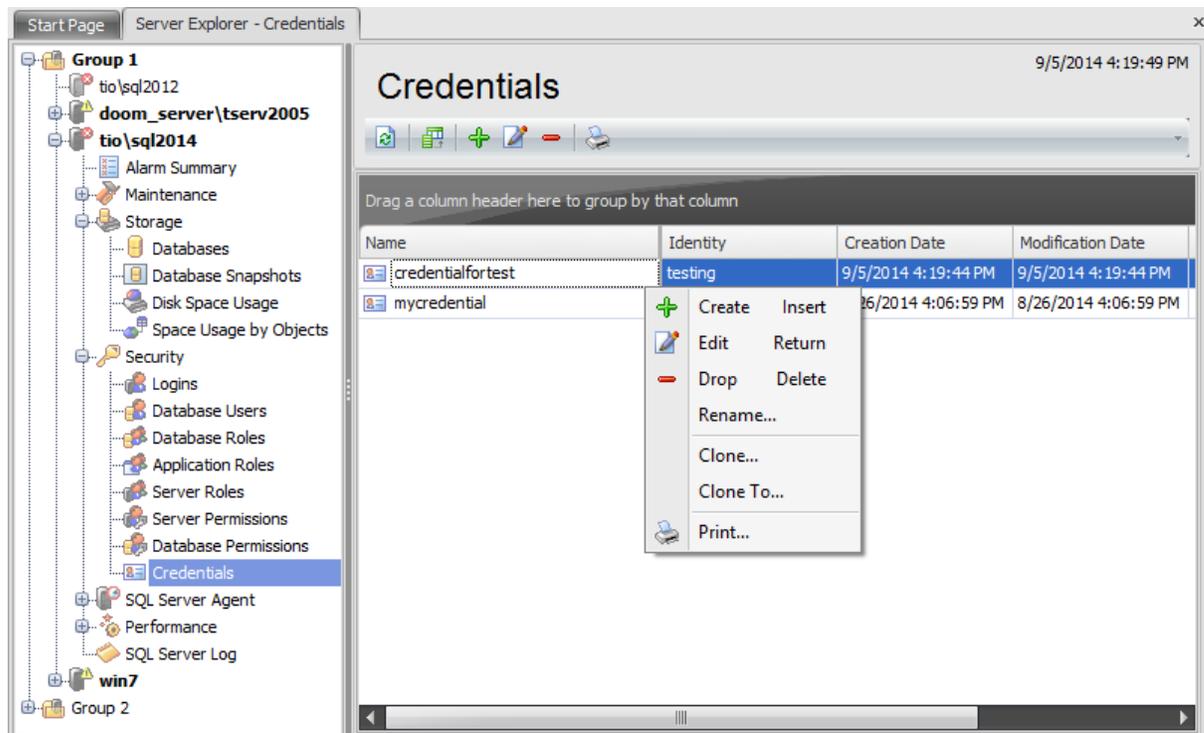
After creating a credential, you can map it to a [login](#). A single credential can be mapped to multiple SQL Server [logins](#). But a SQL Server [login](#) can be mapped to only one credential.

Credentials are used for accessing the Windows Azure storage for [uploading backup](#) files.

Use the context menu to:

- + [create](#) a new credential
-  [edit](#) the selected credential(s)
- drop the selected credential(s)
  - rename the selected login
  - create the object copy on the same server
  - clone the object copy to another server
-  create and [print](#) the report

Alternatively, you can use the 'Insert' button to create a credential, 'Enter' button to edit credential(s) and 'Delete' button to drop credential(s).



The list displays the existing server credentials as a grid with the following columns: *Name*, *Identity*, *Creation Date*, *Modification Date*, *Target Type*. If more convenient, you can

change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find operations that can be performed with the grid.

To open a Credential in the [Credential Editor](#), double-click it in the grid. Alternatively, you can right-click the Credential alias and select the **Edit** context menu item.

### 7.8.1 Credential Editor

**Credential Editor** allows you to define credential properties. It opens automatically when you create a new server credential and is available on editing an existing one.

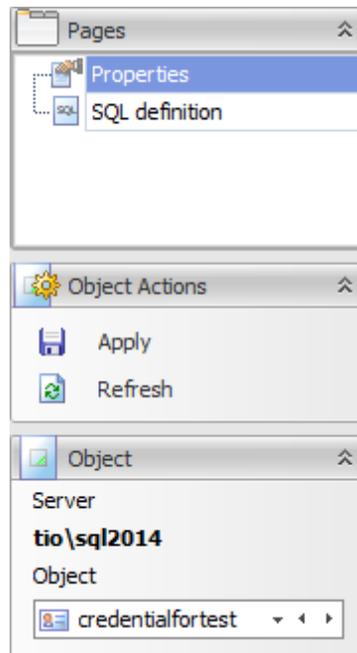
To open a credential in **Credential Editor**, double-click it in the login list. Alternatively, you can right-click the credential alias and select the **Edit** context menu item.

The list of **Credential Editor** tabs is displayed on the [navigation bar](#):

- [Setting credential properties](#)
- [Viewing SQL definition](#)

### 7.8.1.1 Using Navigation bar

The **Credential Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Credential Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

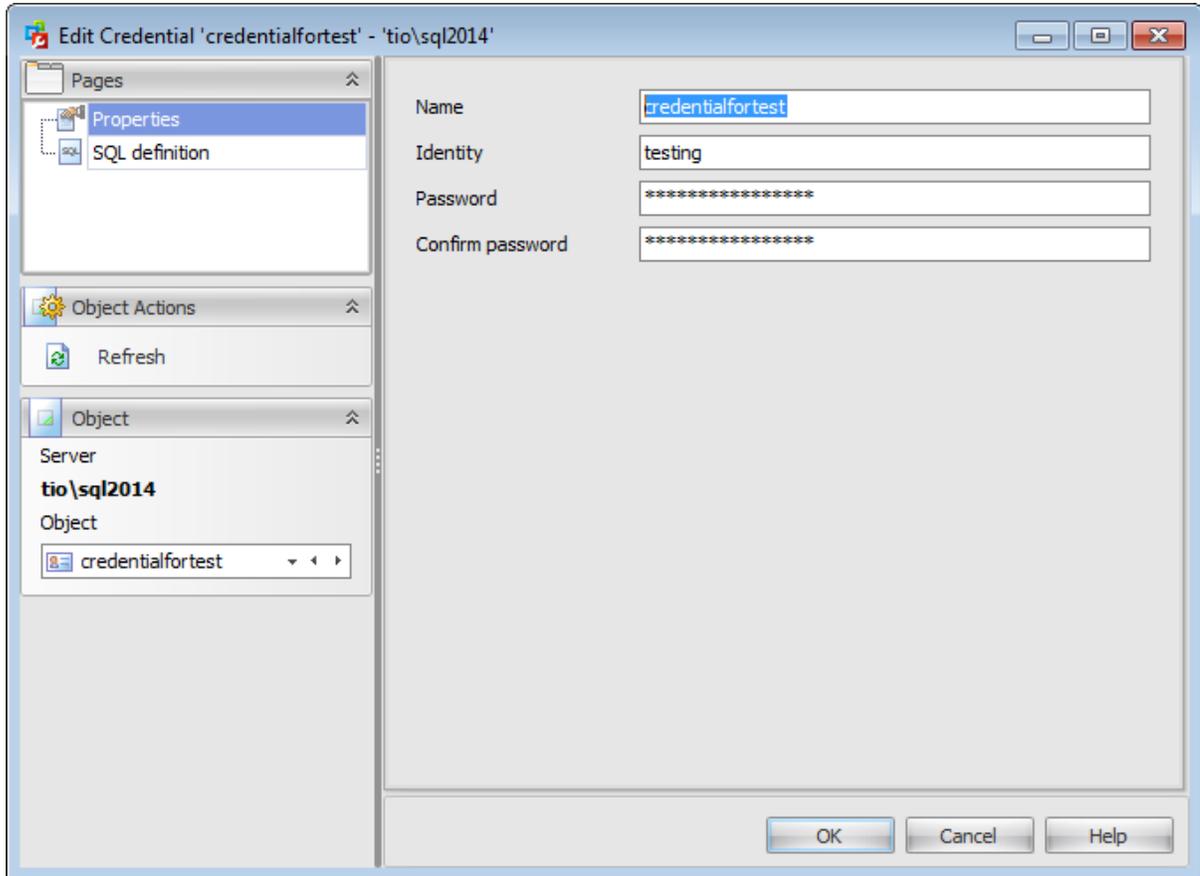
This field displays the name of the server where the edited object resides.

##### Object

This field displays the login that is currently being edited. However, you can change the edited login by using the drop-down list of logins or by clicking the arrow-left or arrow-right buttons.

### 7.8.1.2 Setting credential properties

Use the **Properties** section of the **Credential Editor** to configure credential default parameters.

**Name**

Enter a name for the new credential, or view the name of the credential being edited.

**Identity**

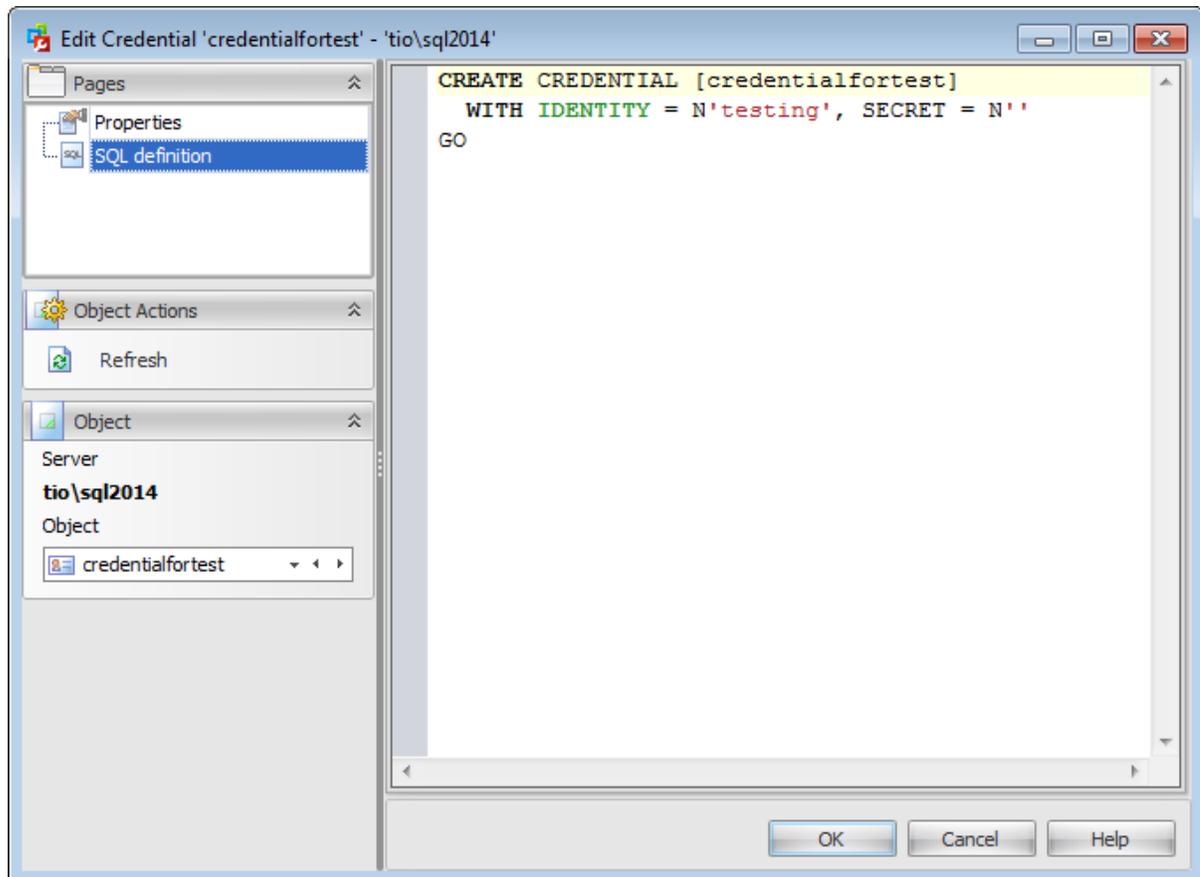
Specify the name of the account to be used when connecting outside the server.

**Password**

Specify the password required for outgoing authentication, and **confirm the password** in the corresponding box.

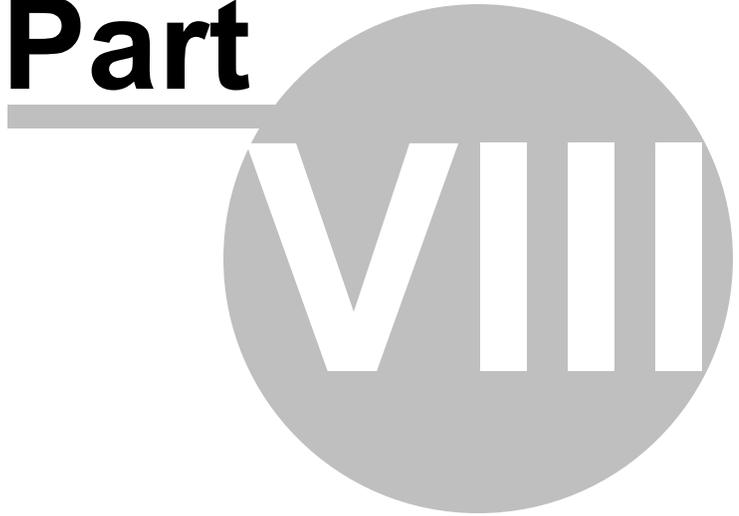
### 7.8.1.3 Viewing SQL definition

This tab allows you to view the object DDL.





**Part**



## 8 SQL Server Agent

**SQL Server Agent** is a Microsoft Windows service that executes scheduled administrative tasks, which are called jobs.

SQL Server Agent uses the following components to define the tasks to be performed, when to perform the tasks, and how to report the success or failure of the tasks:

- [Jobs](#)
- [Schedules](#)
- [Alerts](#)
- [Operators](#)
- [Proxies](#)

**Note:** SQL Server Agent objects are available only for System Administrator role members.

The screenshot shows the SQL Server Agent interface in SQL Server Enterprise Manager. The main window is titled "SQL Server Agent" and displays a "Date Interval" of "Unlimited". The interface is divided into three main sections:

- Last Run Jobs:** A table showing the last successful runs of jobs.
 

Job	Last Run ...	Dur...
AdventureWorks	17.05.2013 ...	00:00:11
AdventureWorks	17.05.2013 ...	00:00:05
check Adventure	17.05.2013 ...	00:00:01
shrink Adventure	17.05.2013 ...	00:00:00
DemoDB_shrink	17.05.2013 ...	00:00:00
AW_Maintenance	17.05.2013 ...	00:00:22
AdventureWorks	17.05.2013 ...	00:00:07
- Next Running Jobs:** A table showing the next scheduled runs of jobs.
 

Job	Next Run...	Schedule
AdventureWo	19.05.2013 ...	sche
avg_stats_SQ	18.05.2013 ...	avg_stats...
AdventureWo	24.05.2013 ...	weekly
cleanup_stats	18.05.2013 ...	cleanup_st...
Stats_upd	22.05.2013 ...	sched
DemoDB_shrin	21.05.2013 ...	sc
cleanup_alarm	18.05.2013 ...	cleanup_al...
- Failed Jobs:** A table showing jobs that have failed.
 

Job	Run Date	Run Status	Duration	Server	Retries
Stats_upd	17.05.2013 11:0...	Failed	00:00:00	DOOM_SERVE...	0
cleanup_stats_SQLAdmin	17.05.2013 0:30...	Failed	00:00:00	DOOM_SERVE...	0
avg_stats_SQLAdministra	17.05.2013 0:00...	Failed	00:00:03	DOOM_SERVE...	0
DemoDB_shrink	14.05.2013 11:2...	Failed	00:00:00	DOOM_SERVE...	0
AdventureWorks_check	12.05.2013 11:3...	Failed	00:00:04	DOOM_SERVE...	0
AdventureWorks_mainten	10.05.2013 10:5...	Failed	00:00:00	DOOM_SERVE...	0

The SQL Server Agent section provides information about all planned jobs that have been created on the server in the three groups: *Last Run Jobs*, *Next Running Jobs* and *Failed Jobs*.

Double-click the job to open it in the [Job Editor](#).

Using the context menu of any area you can:

- + [create](#) a new job
- [edit](#) selected job(s)
- [drop](#) selected job(s)
- rename job

enable job(s)

disable job(s)



start job(s)

stop job(s)



view [job history](#)



create and [print](#) the report

---

**See also:**

[Getting Started](#)

[Servers](#)

[Database Maintenance](#)

[Storage](#)

[Security](#)

[Performance](#)

[SQL Server Log](#)

[Query Editor](#)

[Options](#)

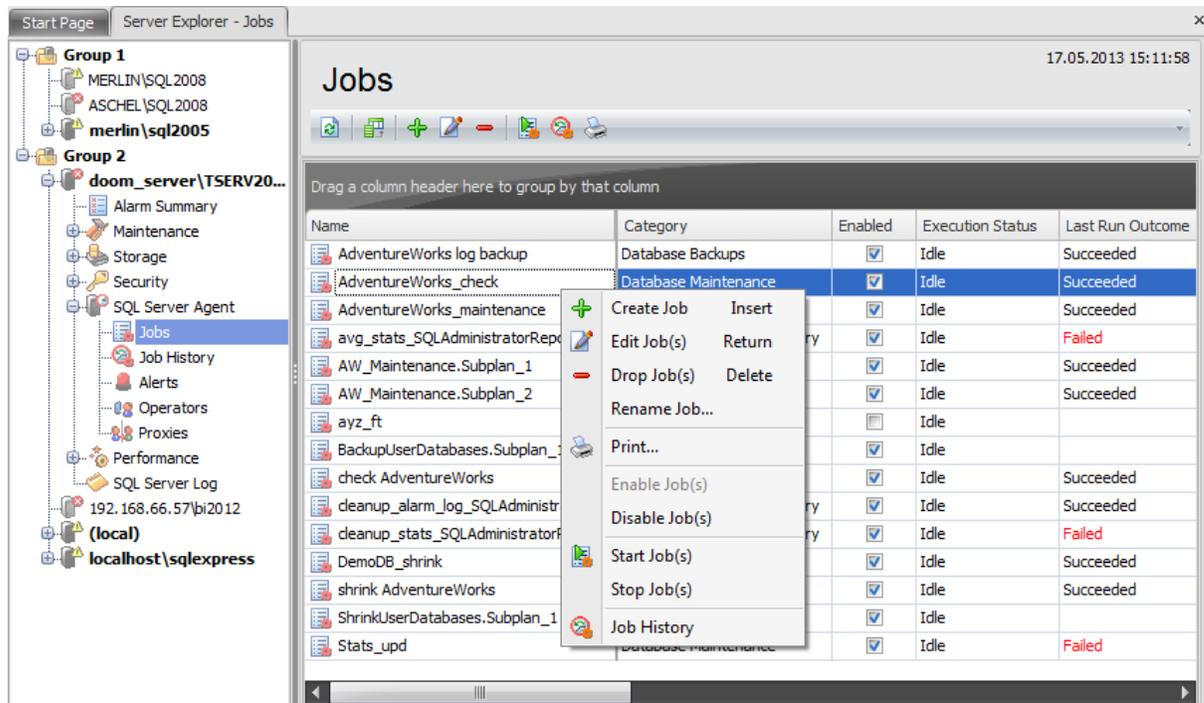
## 8.1 Jobs

A **job** is a specified set of operations performed sequentially by SQL Server Agent. A job can perform a wide range of activities, including running Transact-SQL scripts, command-line applications, Microsoft ActiveX scripts, Integration Services packages, Analysis Services commands and queries, or Replication tasks. Jobs can run repetitive or scheduled tasks, and they can automatically notify operators of job status by generating alerts. SQL Server Agent **jobs** can be used to simplify SQL Server administration and make it more efficient by automating routine administrative tasks.

**Note:** Before using SQL Server jobs, make sure that the **SQL Server Agent** service is running.

Use the object context menu or toolbar buttons to:

-  [create](#) a new job
-  [edit](#) selected job(s)
-  drop selected job(s)
- rename job
- enable job(s)
- disable job(s)
-  start job(s)
- stop job(s)
-  view [job history](#)
-  create and [print](#) the report



The list displays all created jobs as a grid with the following columns: *Name, Category, Enabled, Execution Status, Last Run Outcome, Last Run Date, Next Run Date, Schedules, Creation Date, Owner, Description*. If more convenient, you can change the order of the

columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open a job in the [Job Editor](#), double-click it in the grid. Alternatively, you can right-click the job alias and select the **Edit Job** context menu item.

---

**See also:**

[Job history](#)

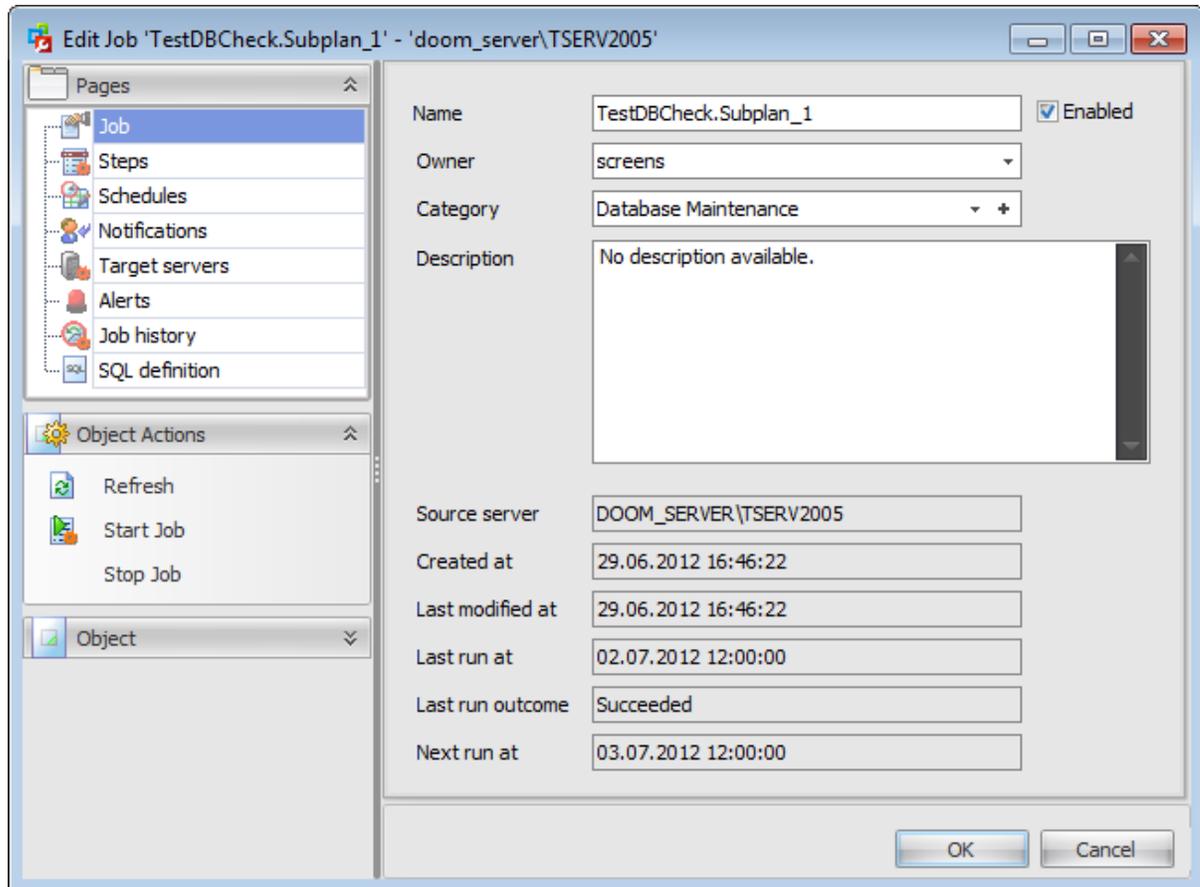
[Alerts](#)

[Operators](#)

[Proxies](#)

### 8.1.1 Job Editor

**Job Editor** allows you to define job properties. It opens automatically when you create a new job and is available on editing an existing one.

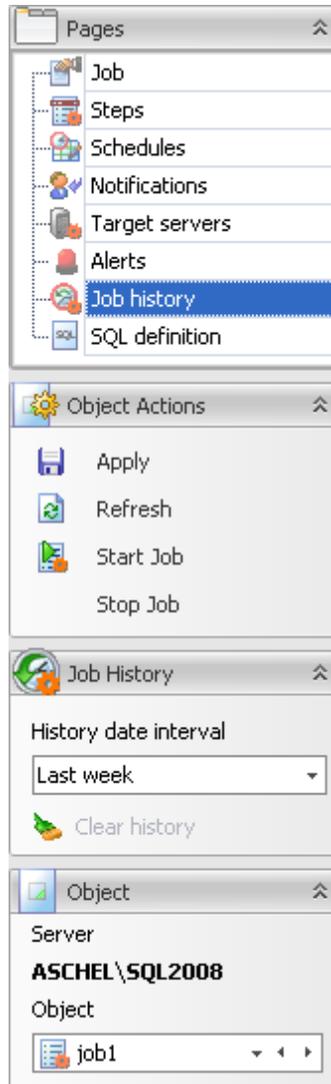


The list of **Job Editor** tabs is displayed on the [navigation bar](#):

- [Setting job properties](#)
- [Managing job steps](#)
- [Managing job schedules](#)
- [Managing notifications](#)
- [Managing target servers](#)
- [Managing alerts](#)
- [Viewing job history](#)
- [Viewing SQL definition](#)

### 8.1.1.1 Using Navigation bar

The **Job Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Job Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

-  apply changes
-  refresh the content of the active tab
-  start job executing
- stop job executing

#### Job History

##### History date interval

Use the drop-down menu to define the period at which to display the job history.

 clear job history

### **Object**

#### **Server**

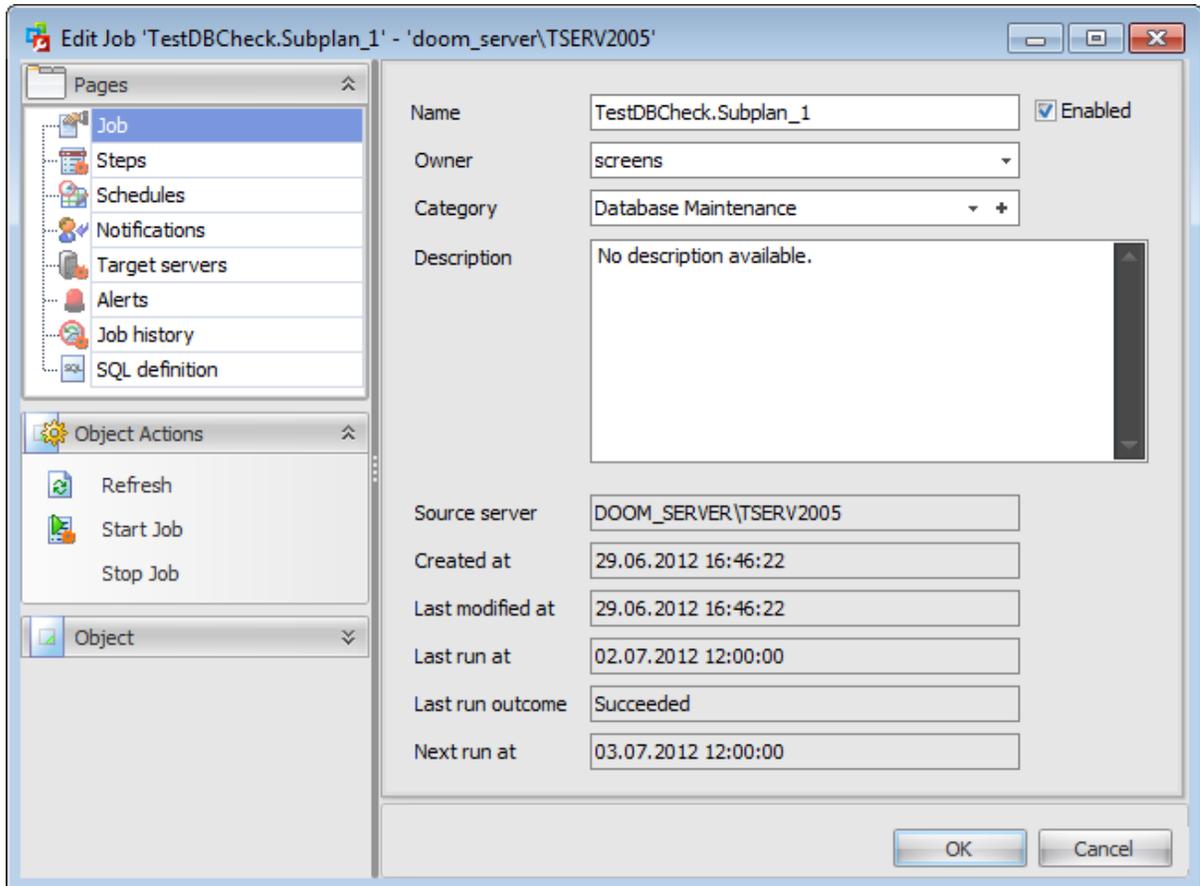
This field displays the name of the server where the edited object resides.

#### **Object**

This field displays the job that is currently being edited. However, you can change the edited job by using the drop-down list of jobs or by clicking the arrow-left or arrow-right buttons.

### 8.1.1.2 Setting job properties

Use the **Job** tab of the **Job Editor** to configure/view job properties.



#### Name

Enter a name for the new job, or modify the name of the job being edited. Enable/disable the job using the  **Enabled** option.

#### Owner

Use the drop-down list to select the job owner.

**Note:** To create a job, a user must be a member of one of the SQL Server Agent fixed database roles or the sysadmin fixed server role. A job can be edited only by its owner or members of the sysadmin role.

#### Category

Use the drop-down list to select the job category.

Job categories help you organize your jobs for easy filtering and grouping. For example, you can organize all your database backup jobs in the *Database Maintenance* category. You can add a new job category by pressing the + **Add** button.

#### Description

In this memo field you can supply an optional description for the job.

The lower area displays the following job events and properties:

*Source server*

*Created at*

*Last modified at*

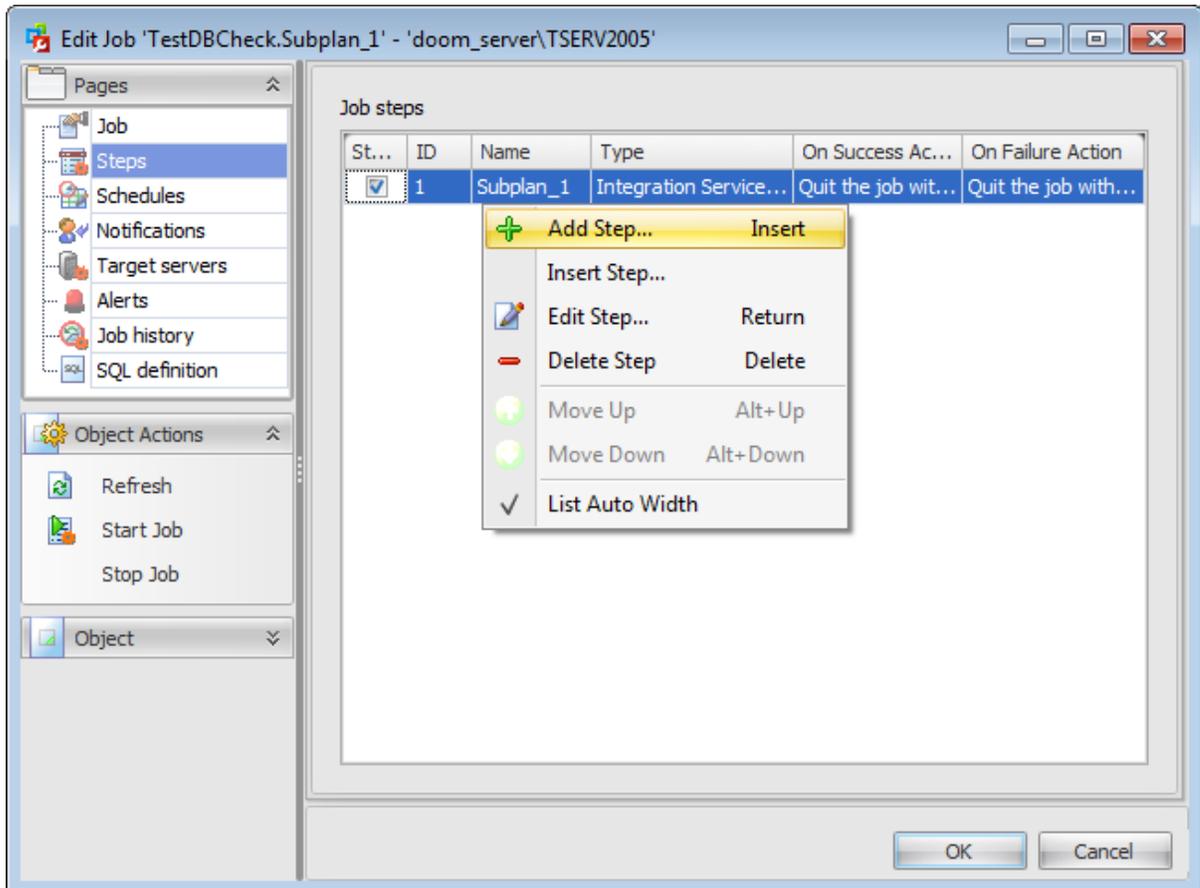
*Last run at*

*Last run outcome*

*Next run at*

### 8.1.1.3 Managing job steps

Use the **Steps** tab of **Job Editor** to manage job steps (operations). A **job step** is an action that the job takes on a database or a server. Every job must have at least one job step.



The current job steps are displayed as a grid with the following columns: *Start step, ID, Name, Type, On success Action, On failure Action*.

Right-click an item within the list to call the **context menu** allowing you to *create* a new job step and specify its properties using the **Edit Step** dialog, *insert* a step, *edit* or *delete* the selected step.

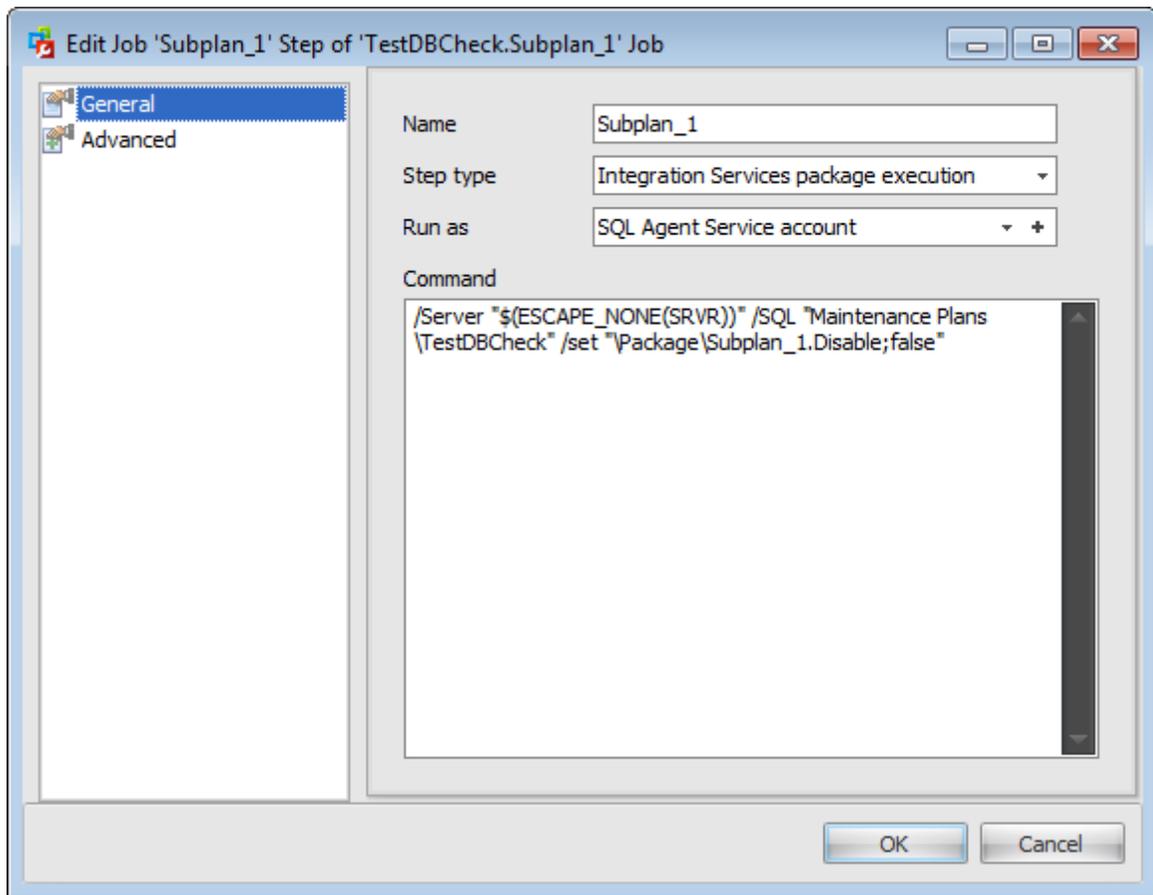
The **Add New Job Step / Edit Step** dialog allows you to define job step parameters.

Steps ID start with 1 and increment in numerical order. If the step is inserted into an existing sequence then order numbers change automatically.

#### Step Editor

##### General

The **General** tab allows you to define basic parameters of the step.



### Name

Enter a name for the new step, or modify the name of the step being edited.

Specify the **Step Type** of the step indicating the subsystem used by Microsoft SQL Server Agent service to execute the job command:

*Transact-SQL statement*

*Operating-system command or executable program*

*ActiveX Script*

*Replication Distribution Agent job*

*Replication Snapshot Agent job*

*Replication Log Reader Agent job*

*Replication Merge Agent job*

*Replication Queue Reader Agent job*

*Analysis Services query (MDX, DMX)*

*Analysis Services command (XMLA)*

*Integration Services package execution*

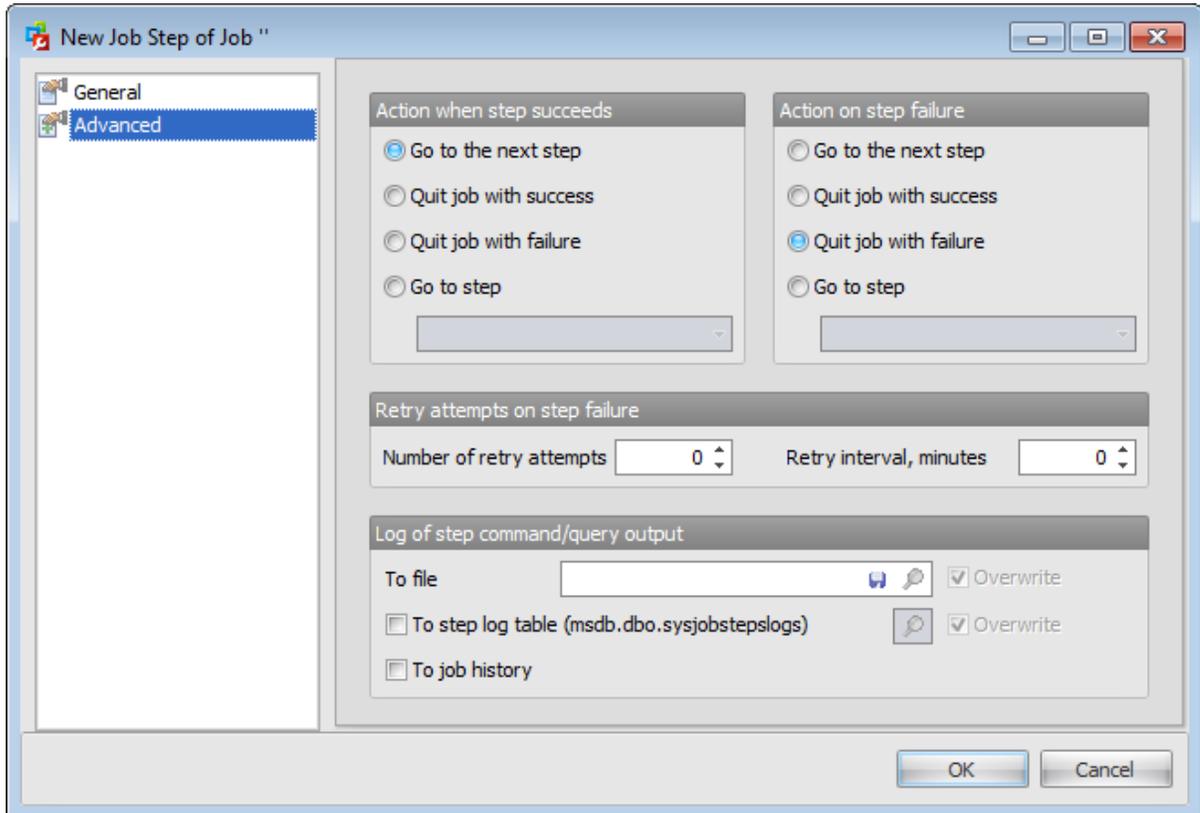
Use the **Database** drop-down list to select the database for which the current step will be executed (for *Transact-SQL* command type step; for an *ActiveX* job step, you are to specify the name of the scripting language that the step uses), and specify the security context under which the command is executed using the **Run as user** drop-down list.

Use the **Command text** memo field to enter the job step command code: *SQL* code (for

**Transact-SQL** step), *Visual Basic Script / Java Script / other script* code (for **ActiveX Script** step), or *OS command / executable program* (for **Operating-system command** step).

### Advanced

The **Advanced** tab provides flexible job step control of flow for success and failure.



Specify the **Action when step succeeds**, i.e. the action the job will take if the step succeeds:

- Go to next step*
- Quit job with success*
- Quit job with failure*
- Go to step ...* (select the step from the drop-down list)

Specify the **Action when step fails**, i.e. the action the job will take if the step fails:

- Go to next step*
- Quit job with success*
- Quit job with failure*
- Go to step ...* (select the step from the drop-down list)

The **Retries if step fails** group allows you to set optional parameters for the job step: the *Number of retry attempts* and the *Retry interval* between attempts (in minutes).

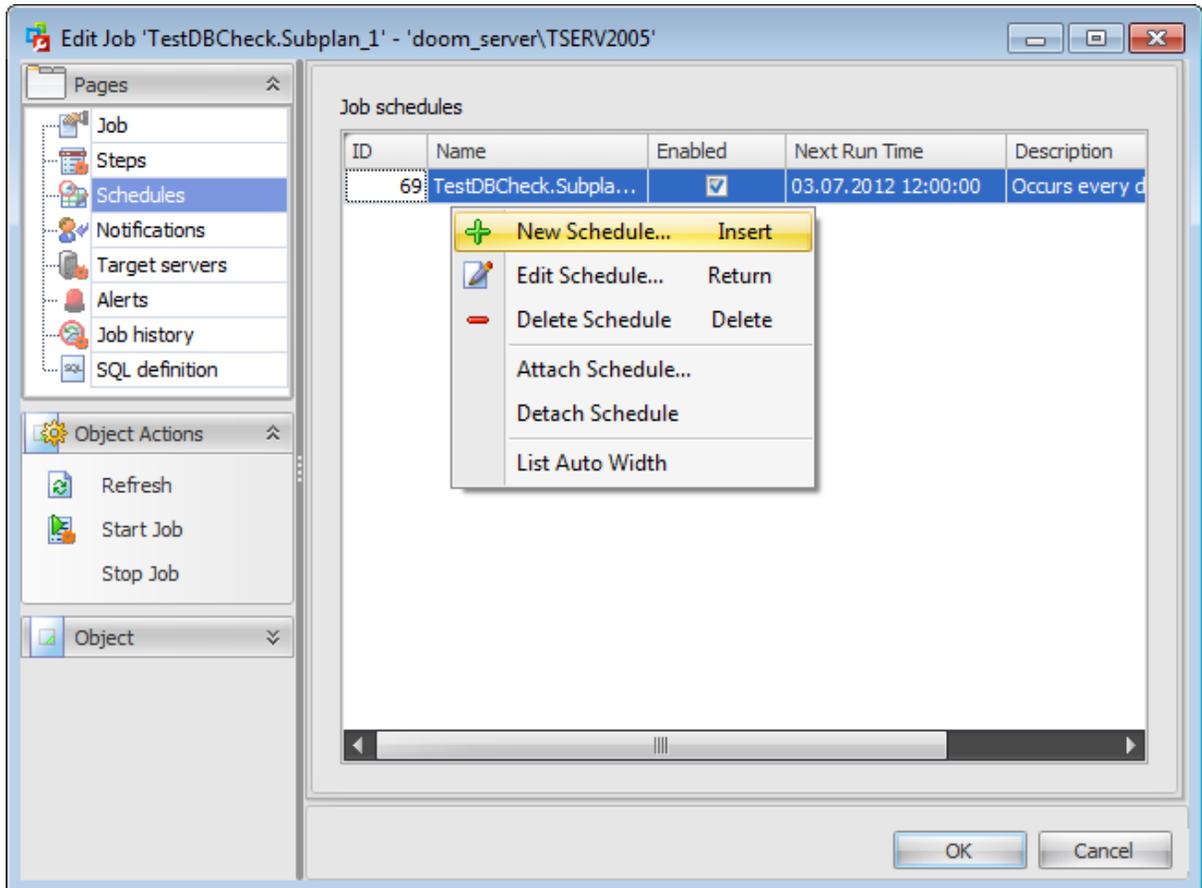
### Log of step command/query output

SQL Server Agent can write output from some job steps either to the *sysjobsteplogs* table in the *msdb* database or to an operating system file.

To use this feature, set the path and the file name in the **To file** field, and/or check  **To step log table (msdb.dbo.sysjobsteplogs)** and/or  **To job history**.

#### 8.1.1.4 Managing job schedules

The **Schedules** tab allows you to manage schedules assigned to the job.

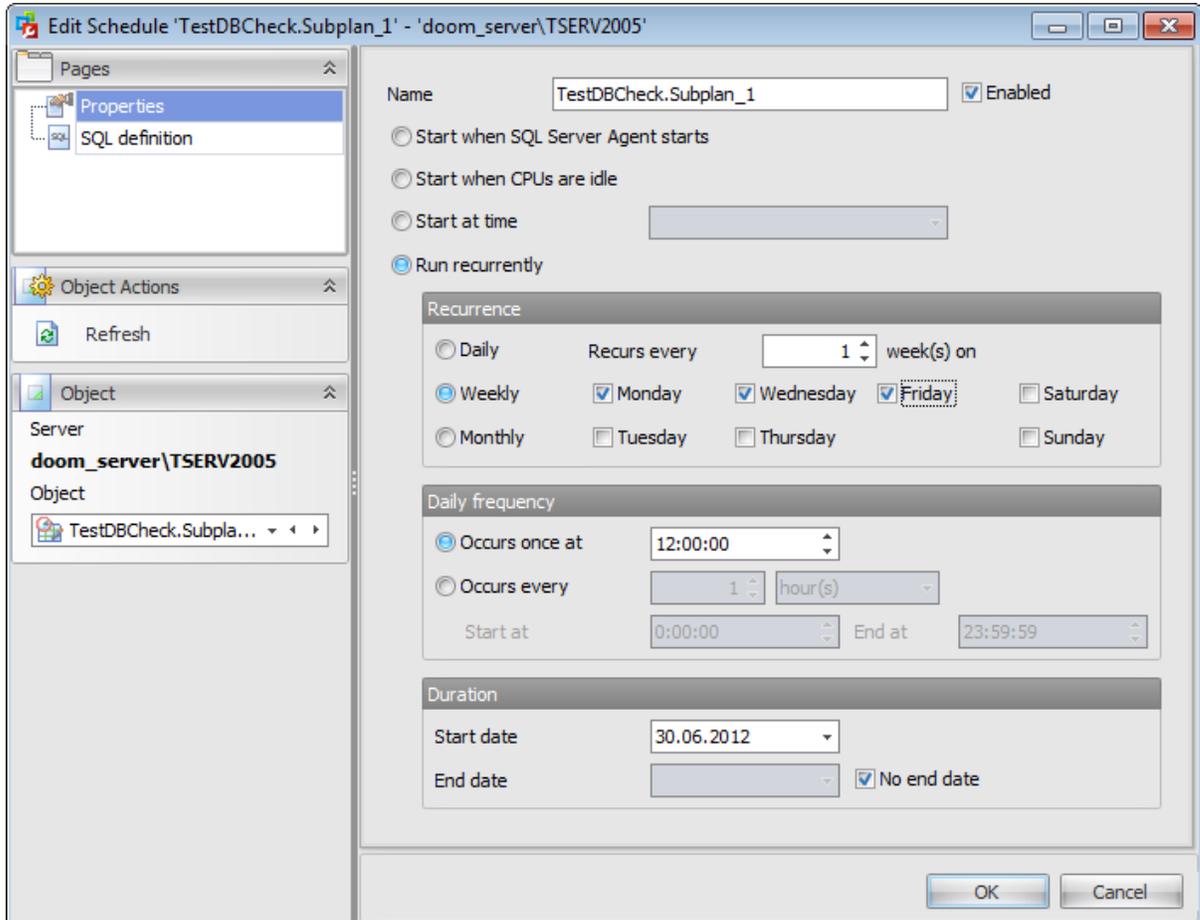


Scheduling administrative jobs consists in defining the condition(s) that cause the job to begin running. It is possible to schedule any type of job. More than one job can use the same job schedule.

Right-click an item within the list to call the **context menu** allowing you to *create* a new job schedule and specify its properties using the **Add New Job Schedule** dialog, *add an existing schedule*, *edit* or *delete* the selected schedule.

## 8.1.1.4.1 Schedule Editor

The **Add New Job Schedule / Edit Schedule** dialog allows you to define schedule parameters.



Use the **Schedule** tab of **Schedule Editor** to configure/view schedule properties.

**Name**

Enter a name for the new schedule, or modify the name of the schedule being edited. Enable/disable the schedule using the  **Enabled** option.

Specify the *Execution Type* of the schedule indicating when the scheduled job is to be executed:

- Start when SQL Server Agent starts*
- Start when CPUs are idle*
- Start at time...* (set the date using the **date editor** which is activated when you click the Arrow-Down element of the combo-box; specify the time value using the spinner control)
- Run recurrently*

If the **Recurrent execution** type is selected, you should also set the **Recurrence**, **Daily Frequency** and **Duration** properties of the current schedule.

The **Recurrence** area allows you to specify the **daily/weekly/monthly** recurrence rule, define the **Recurs Every ... days/weeks** value, and select the **days of the week/month** that can be set for the schedule.

The **Daily Frequency** tab allows you to select whether the scheduled job will be run **once at ...** or **every ... hours/minutes start at ... end at ...** (specify the day/time values using the drop-down menu and spinner controls).

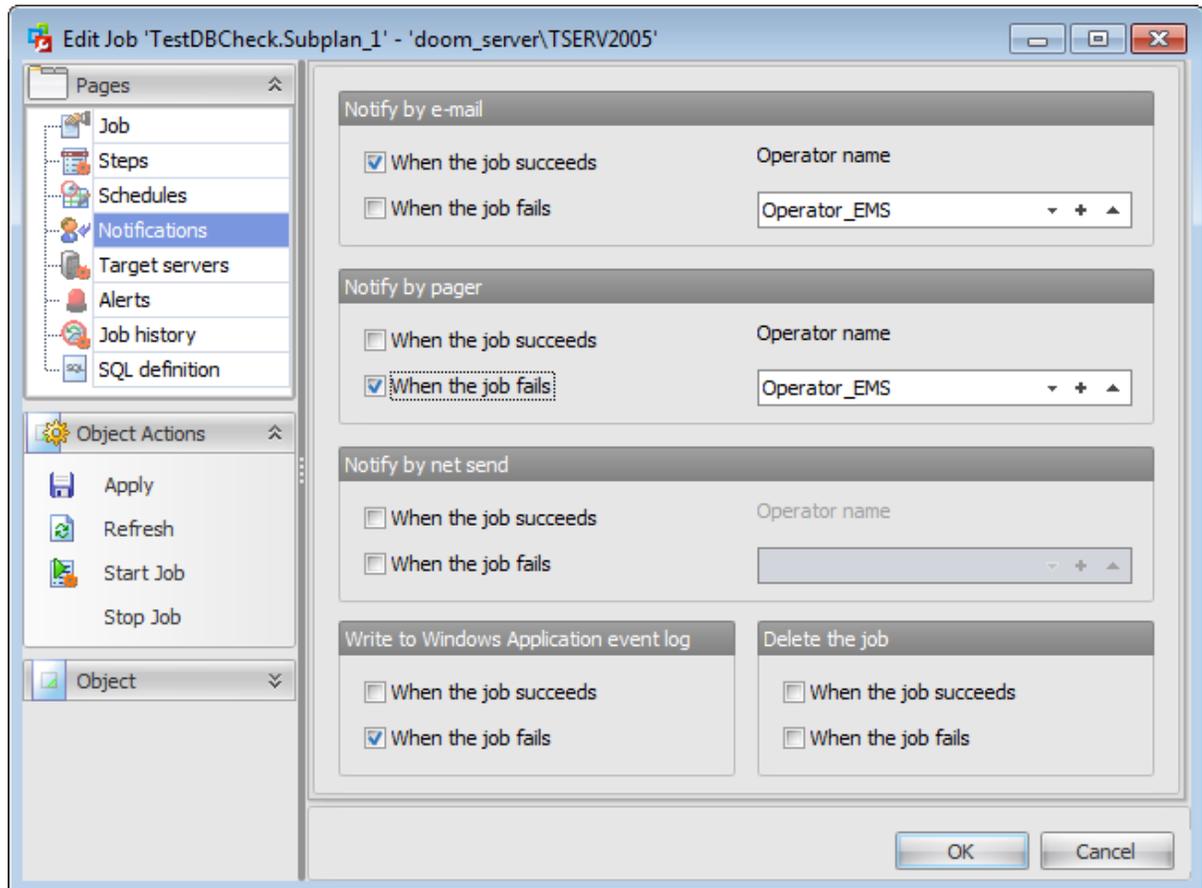
The **Duration** tab allows you to set the **Start date** and the **End date** threshold values for the interval using the **date editor** which is activated when you click the Arrow-Down element of the combo-box.

**No end date**

Check this option if you are uncertain when the scheduled job should be finished, or if no limitations on the job duration are required.

### 8.1.1.5 Managing notifications

The **Notifications** tab allows you to specify actions that the SQL Server Agent service will take after the job completes (succeeds or fails). Job responses ensure that database administrators know when jobs complete and how frequently they run.



#### Notify by e-mail

Use the drop-down list to select the operator that will be notified via *e-mail*, and specify the job status for the notification to be performed:

- When the job succeeded and/or*
- When the job failed*

#### Notify by pager

Use the drop-down list to select the operator that will be notified using *electronic paging*, and specify the job status for the notification to be performed:

- When the job succeeded and/or*
- When the job failed*

#### Notify by net send

Use the drop-down list to select the operator that will be notified using a *net send* message, and specify the job status for the notification to be performed:

- When the job succeeded and/or*
- When the job failed*

**Write to Windows Application event log**

If necessary, specify the job status for writing to Microsoft Windows Application event log:

*When the job succeeded* and/or

*When the job failed*

**Delete the job**

If you are certain that you do not need to rerun this job later, you can specify the job status deleting the job:

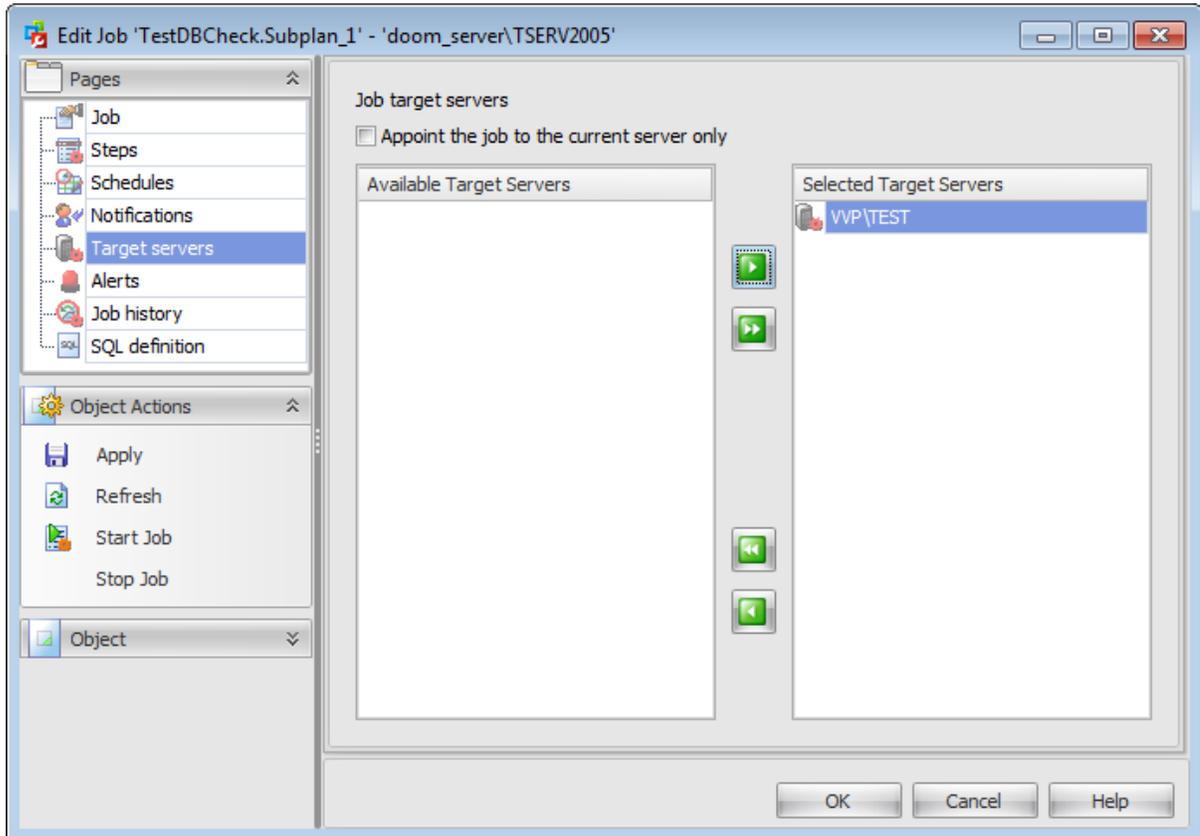
*When the job succeeded* and/or

*When the job failed*

### 8.1.1.6 Managing target servers

The **Target servers** tab allows you to target the job at the specified target server in case you want a job to run on a single or multiple target servers.

A target server is a server that gets jobs from the master server and sends events to it. Target server connects periodically to the master server to update its schedule.



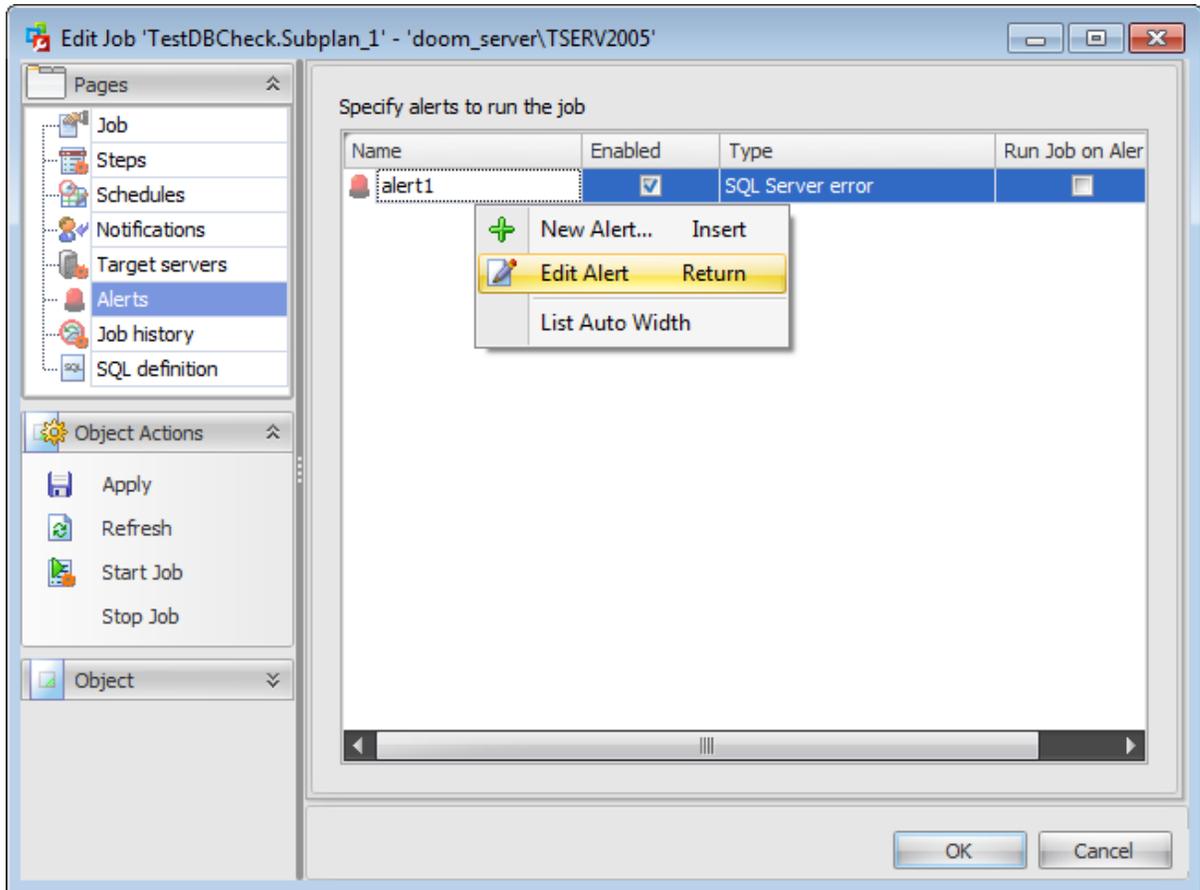
**Appoint the job to the current server only**

Check this option to run the job on the master server only.

To select a target server, you need to move it from the **Available Target Servers** list to the **Selected Target Servers** list. Use the     buttons or drag-and-drop operations to move the target server from one list to another.

### 8.1.1.7 Managing alerts

The **Alerts** section allows you to browse the list of **alerts** and manage them efficiently.



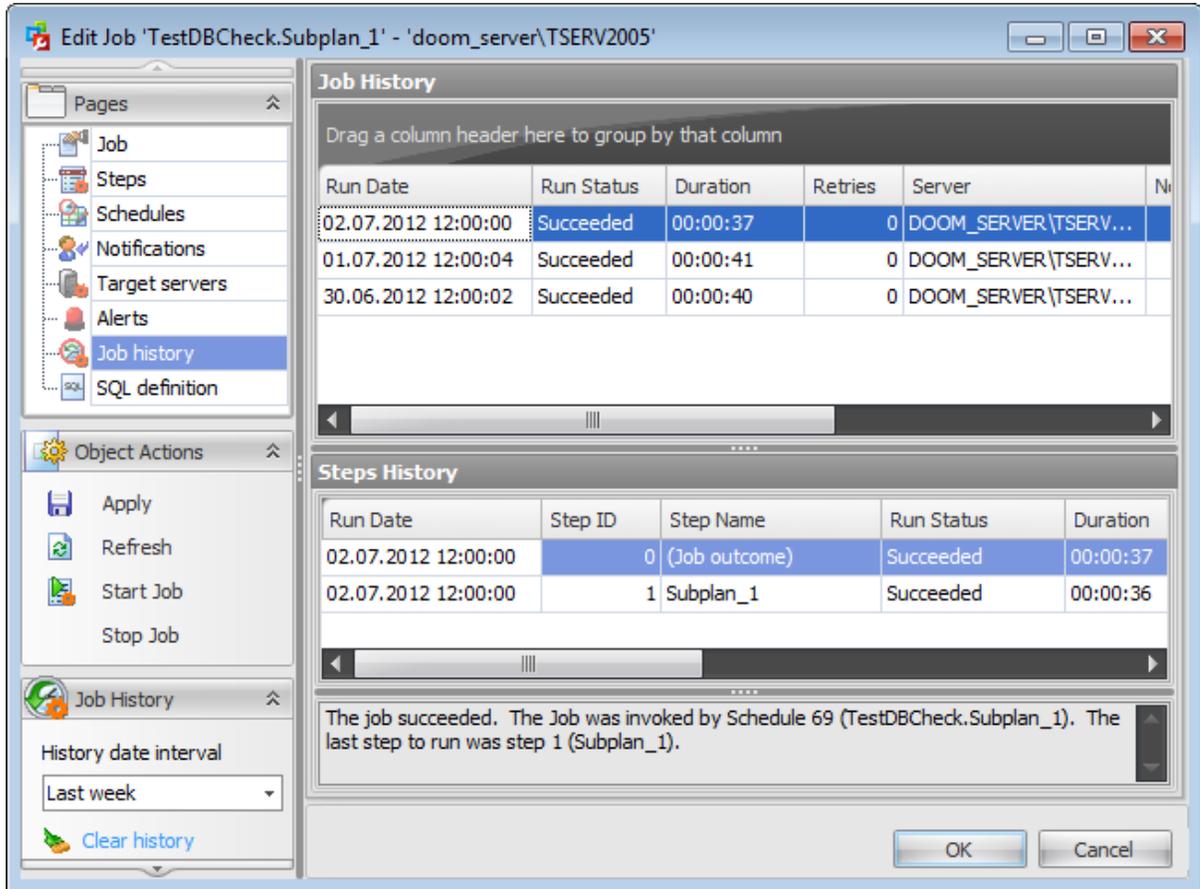
The list displays the alerts as a grid with the following columns: *Name*, *Enabled*, *Type*, *Run Job on Alert*.

Click a column caption to **sort** items by values of this column in the ascending or the descending mode.

Right-click an item within the list to call the **context menu** allowing you to *create* a new alert and specify its properties using [Alert Editor](#), *edit*, *delete* the selected alert, *refresh* the list, or show/hide columns of the list.

### 8.1.1.8 Viewing job history

This tab allows you to view the whole history of performed [jobs](#) and execution of every job step.

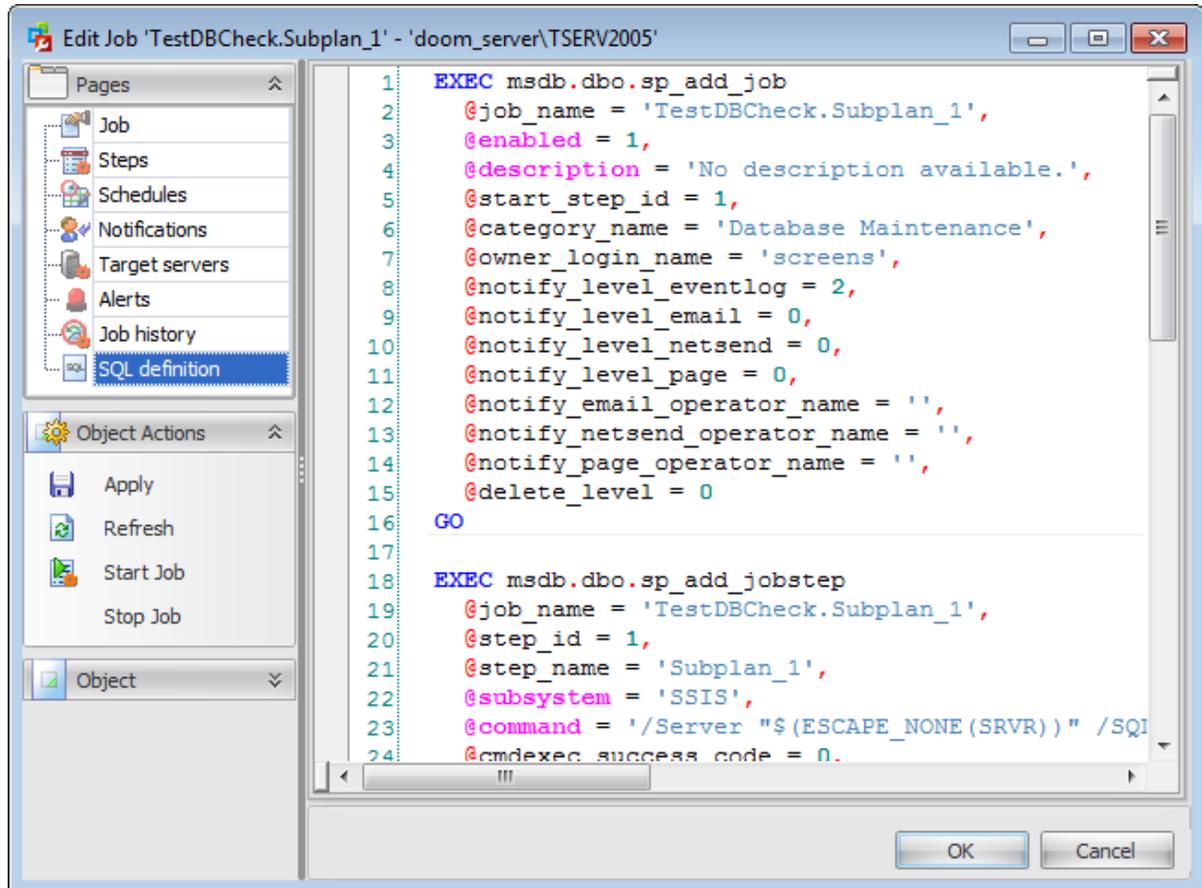


In the **Job History** section you can view the following information about the current job: *Run Date, Job Name, Run Status, Duration, Retries, Server, Notified by Email, Notified by Pager, Notified by Net Send.*

In the **Step History** section you can view the following information about all steps of the job: *Run Date, Step ID, Step Name, Run Status, Duration, Retries, Error ID, Severity, Message.*

### 8.1.1.9 Viewing SQL definition

This tab allows you to view the object DDL.



## 8.2 Job history

This section allows you to view the whole history of performed [jobs](#) and execution of every job step.

Use the context menu or buttons on the work area toolbar to manage the list of job history:

 refresh the content of the job list

 choose columns to display

 start job execution

 [edit job](#)

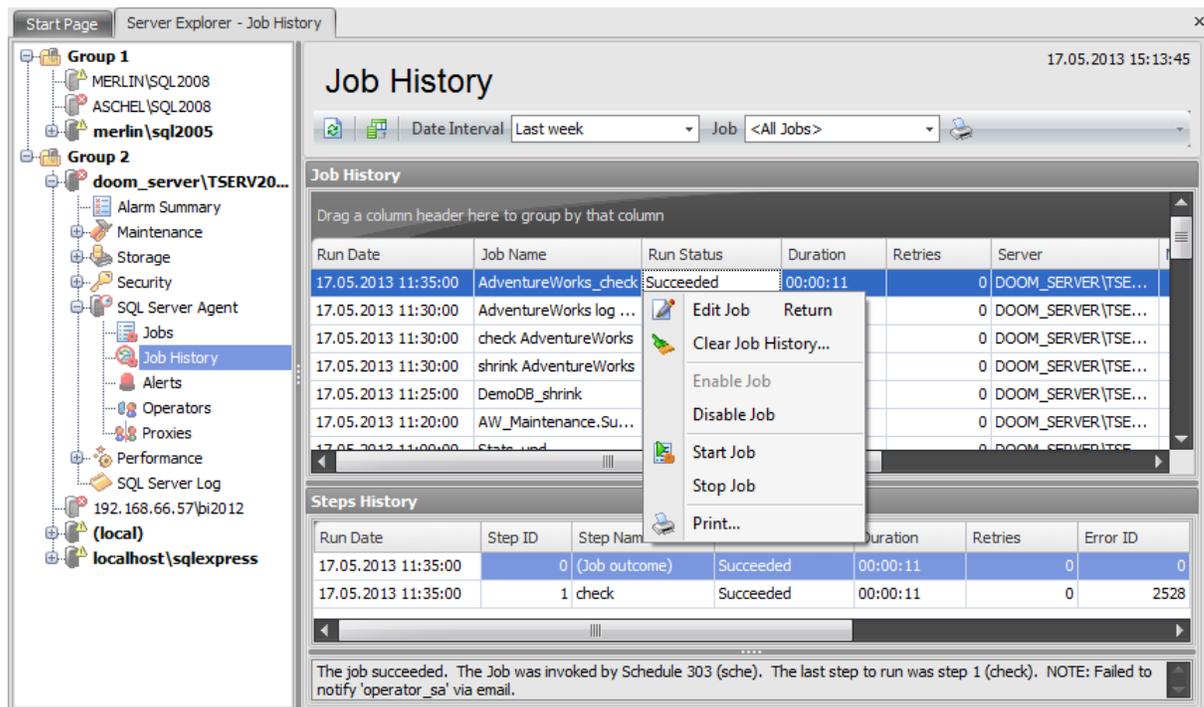
 [edit job step](#)

 clear job history

 create and [print](#) the report

**Date Interval** - select an interval to display jobs within

**Job** - select a job from the drop-down list to display information about



**Job History**

Run Date	Job Name	Run Status	Duration	Retries	Server
17.05.2013 11:35:00	AdventureWorks_check	Succeeded	00:00:11	0	DOOM_SERVER\TSE...
17.05.2013 11:30:00	AdventureWorks log ...			0	DOOM_SERVER\TSE...
17.05.2013 11:30:00	check AdventureWorks			0	DOOM_SERVER\TSE...
17.05.2013 11:30:00	shrink AdventureWorks			0	DOOM_SERVER\TSE...
17.05.2013 11:25:00	DemoDB_shrink			0	DOOM_SERVER\TSE...
17.05.2013 11:20:00	AW_Maintenance.Su...			0	DOOM_SERVER\TSE...

**Steps History**

Run Date	Step ID	Step Name	Duration	Retries	Error ID
17.05.2013 11:35:00	0	(Job outcome)	Succeeded	00:00:11	0
17.05.2013 11:35:00	1	check	Succeeded	00:00:11	2528

The job succeeded. The Job was invoked by Schedule 303 (sche). The last step to run was step 1 (check). NOTE: Failed to notify 'operator\_sa' via email.

In the **Job History** section you can view the following information about the current job: *Run Date, Job Name, Run Status, Duration, Retries, Server, Notified by Email, Notified by Pager, Notified by Net Send.*

To open a job in the [Job Editor](#), double-click it in the grid. Alternatively, you can right-click the job alias and select the **Edit Job** context menu item.

In the **Step History** section you can view the following information about all steps of the job: *Run Date, Step ID, Step Name, Run Status, Duration, Retries, Error ID, Severity, Message.*

To open a job step in the [Step Editor](#), double-click it in the grid. Alternatively, you can right-click the job step alias and select the **Edit Step** context menu item.

---

**See also:**

[Jobs](#)

[Alerts](#)

[Operators](#)

[Proxies](#)

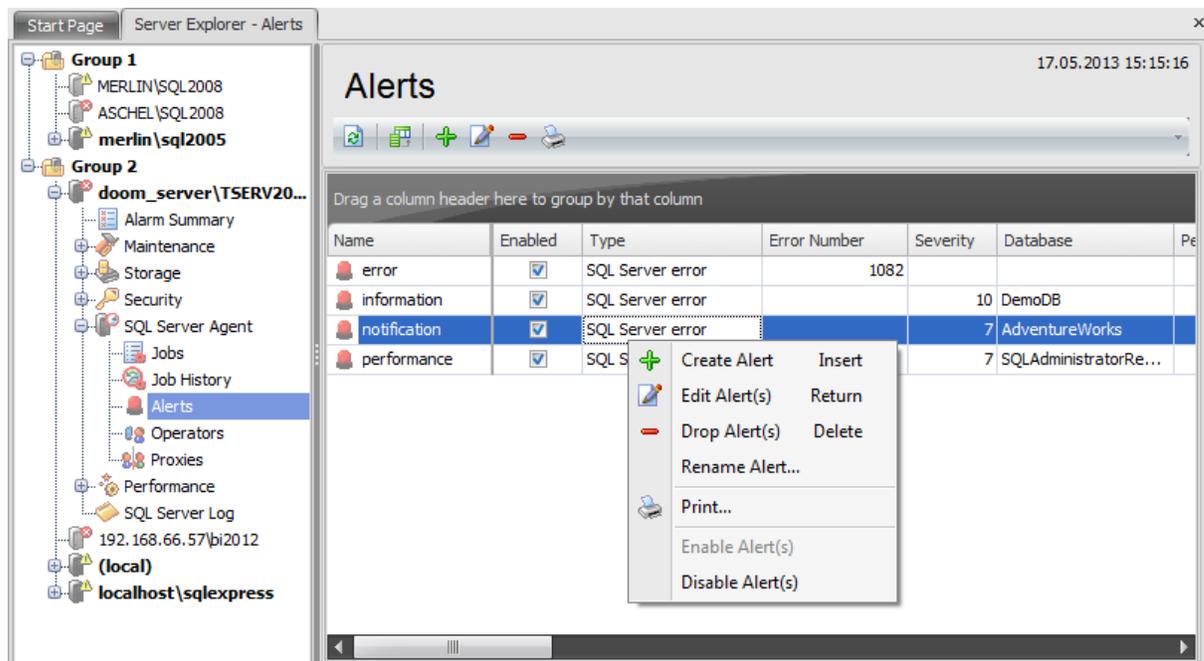
## 8.3 Alerts

An **alert** is an automated response to one or more events. You can define an alert to specify how SQL Server Agent should respond to their occurrence. An alert can respond to an event by notifying an administrator or running a [job](#), or both. An alert can also forward an event to the Microsoft Windows application log on a different computer. By defining alerts, database administrators can monitor and manage SQL Server more effectively.

**Note:** Before using SQL Server alerts, make sure that the **SQL Server Agent** service is running.

Use the object context menu or toolbar buttons to:

- + [create](#) a new alert
-  [edit](#) selected alert(s)
-  drop selected alert(s)
-  create and [print](#) the report
- rename alert
- enable alert(s)
- disable alert(s)



The list displays all created alerts as a grid with the following columns: *Name*, *Enabled*, *Type*, *Error Number*, *Severity*, *Database*, *Performance Object*, *Performance Counter*, *Performance Instance*, *Performance Condition*, *WMI Namespace*, *WMI Query*, *Run Job*, *Has E-mail Notification*, *Has Net Send Notification*, *Last Occurrence Date*, *Last Response Date*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open an alert in **Alert Editor**, double-click it in the list of alerts. Alternatively, you can right-click the alert alias and select the **Edit Alert** context menu item.

---

**See also:**

[Jobs](#)

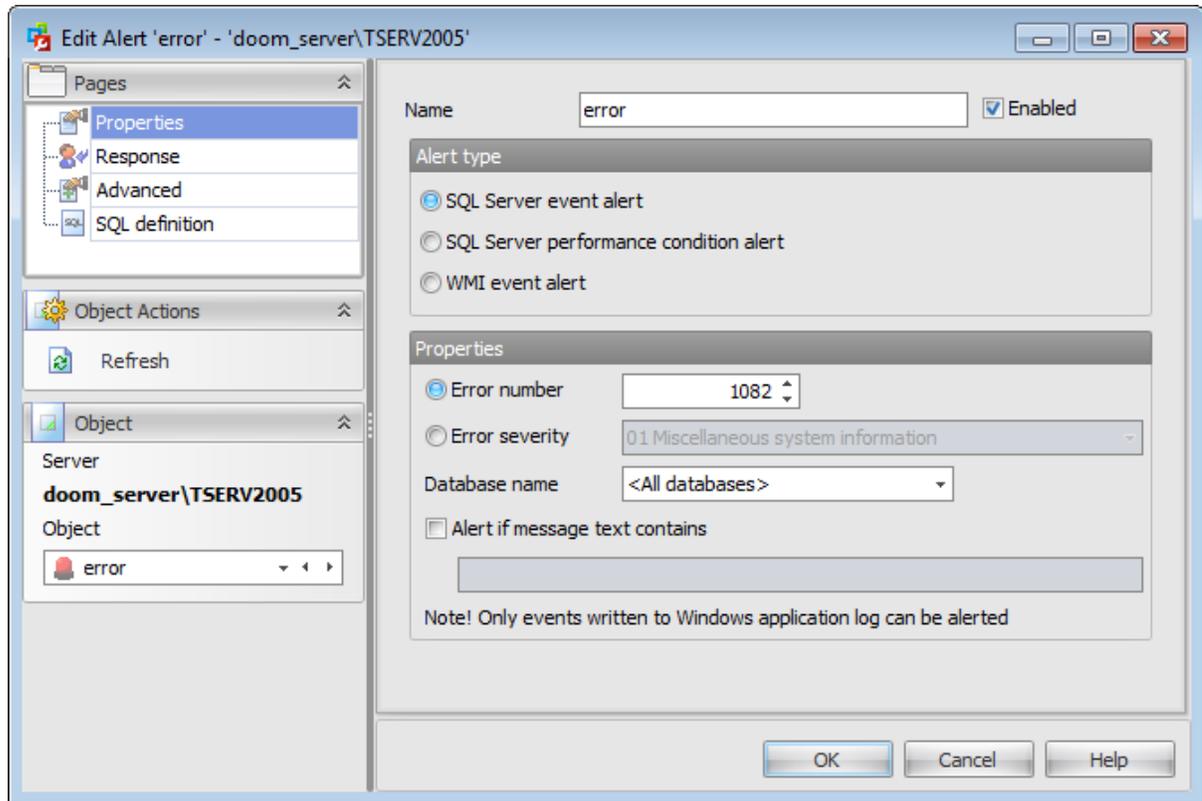
[Job history](#)

[Operators](#)

[Proxies](#)

### 8.3.1 Alert Editor

**Alert Editor** allows you to define alert properties. It opens automatically when you create a new alert and is available on editing an existing one.

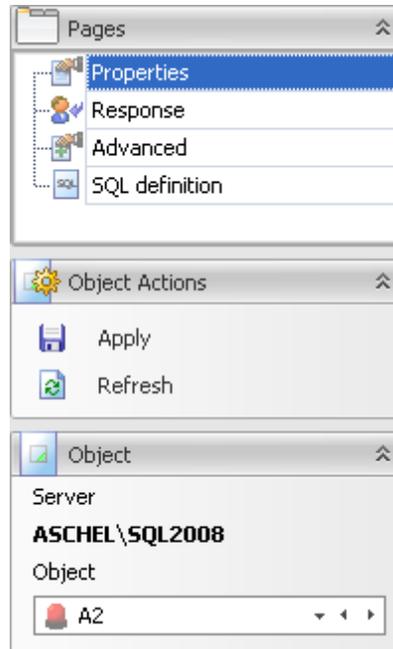


The list of **Alert Editor** tabs is displayed on the [navigation bar](#):

- [Setting alert properties](#)
- [Setting response](#)
- [Setting advanced properties](#)
- [Viewing SQL definition](#)

### 8.3.1.1 Using Navigation bar

The **Alert Editor** Navigation bar provides the number of panels that contain items which are necessary for editing operation.



#### Pages

This group contains the list of **Alert Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Object

This field displays the alert that is currently being edited. However, you can change the edited alert by using the drop-down list of alerts or by clicking the arrow-left or arrow-right buttons.

### 8.3.1.2 Setting alert properties

Use the **Alert** tab of **Alert Editor** to configure/view alert properties.

#### Name

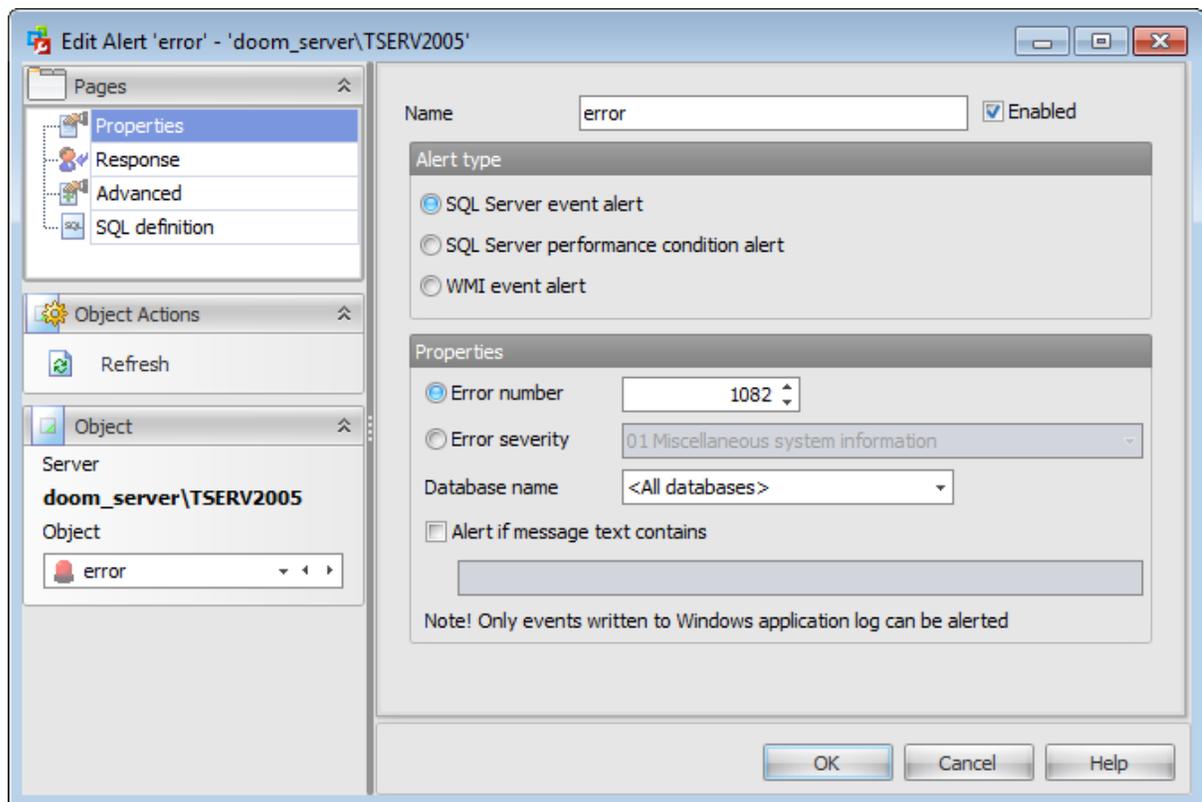
Enter a name for the new alert, or modify the name of the alert being edited. Enable/disable the alert using the  **Enabled** option.

#### Alert type

An alert responds to an event of a specific type: *SQL server event*, *SQL server performance condition* or *WMI event*.

#### SQL Server event alert

Specify properties for the alert that responds to a *server event*.



#### Properties:

Use the following parameters to specify the events that will trigger the alert:

#### Error number

Select this option to specify that SQL Server Agent will fire the alert when a specific error occurs. Use the spinner control to specify the *Error ID*.

#### Error severity

Select this option to specify that SQL Server Agent will fire the alert when any error of the specific severity occurs. Use the drop-down list to select the *Severity level*: 01-25. For example, you can specify a severity level of 15 to respond to syntax errors in

Transact-SQL statements.

### Database name

SQL Server Agent fires an alert only when the event occurs in a particular database. Select a database from the list of databases that reside on the specified SQL Server instance.

### Alert only when message text contain

SQL Server Agent fires an alert when the specified event contains a particular text string in the event message.

### SQL Server performance condition alert

Specify properties for the alert that responds to a *SQL server performance condition*.

<input checked="" type="radio"/> SQL Server performance condition alert	
<input type="radio"/> WMI event alert	
Properties	
Object	MSSQL\$TSERV2005:Databases
Counter	Active Transactions
Instance	master
Alert if value is	more than 10

Use the following parameters to specify the server performance conditions that will trigger the alert:

### Object

Use the drop-down list to select the area of performance to be monitored.

### Counter

Use the drop-down list to select the attribute of the area to be monitored.

### Instance

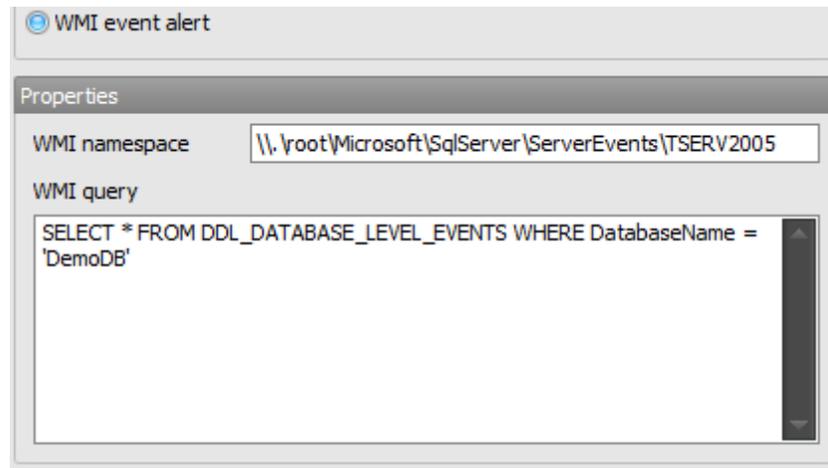
Use the drop-down list to define the specific instance (if any) of the attribute to be monitored.

### Alert if value is

Using these boxes you should specify the threshold for the alert and the behavior that produces the alert. Use the drop-down list to select the behavior value: less than (falls below), equal to (becomes equal to), more than (rises above). Use the **Value** edit-box to specify the threshold number that describes the performance condition counter.

### WMI event alert

Specify properties for the alert that responds to a *WMI event*.



**WMI** is an acronym for *Windows Management Instrumentation*. WMI is the Microsoft's implementation of Web-Based Enterprise Management (WBEM) - a new management technology that allows software to monitor and control managed resources throughout the network.

One common way to use the WMI Event Provider is to create SQL Server Agent alerts that respond to specific events. SQL Server Agent submits a WQL request, receives WMI events, and runs a [job](#) in response to the event.

#### **WMI namespace**

Specify the WMI namespace on the machine where SQL Server is running.

#### **WMI Query**

Use this area to specify the WMI query text for the alert.

The WMI Query Language (WQL) is a subset of standard American National Standards Institute Structured Query Language (ANSI SQL) with minor semantic changes to support WMI.

#### Examples:

```
SELECT * FROM Win32_OperatingSystem WHERE Caption="Microsoft Windows XP Professional" AND CSDVersion="Service Pack 2"
```

If one needs to monitor the number of processes which use more than 10 threads, the following query can be used:

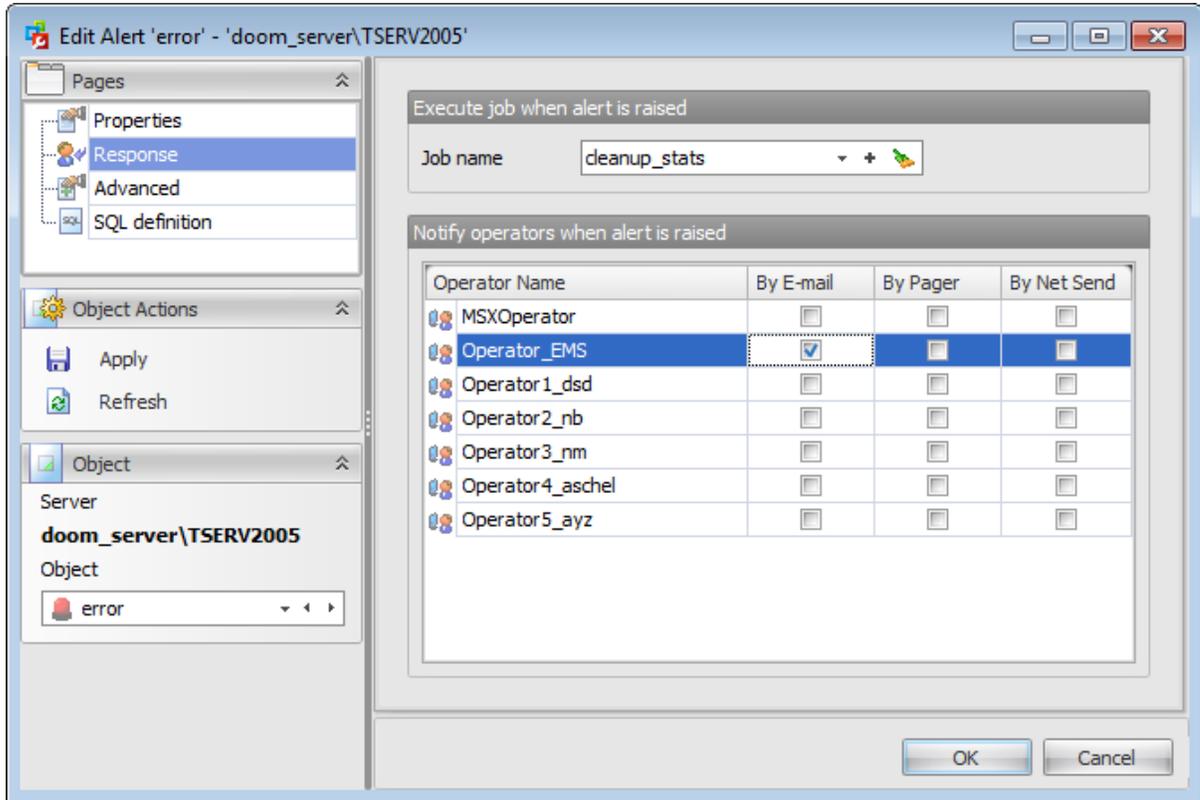
```
SELECT ThreadCount FROM Win32_Process WHERE ThreadCount>10
```

The following WQL query retrieves specific event properties for any event that occurs in the AdventureWorks database and exists under the DDL\_DATABASE\_LEVEL\_EVENTS event group:

```
SELECT SPID, SQLInstance, DatabaseName FROM DDL_DATABASE_LEVEL_EVENTS WHERE DatabaseName = 'AdventureWorks'
```

### 8.3.1.3 Setting response

The **Response** tab allows you to specify actions that the SQL Server Agent service will take after the alert fires.



#### Job name

Use the drop-down list to select the name of the [job](#) to be executed in response to this alert. You can also create a new job by clicking the **+ Add** button. To clear the field use the  button.

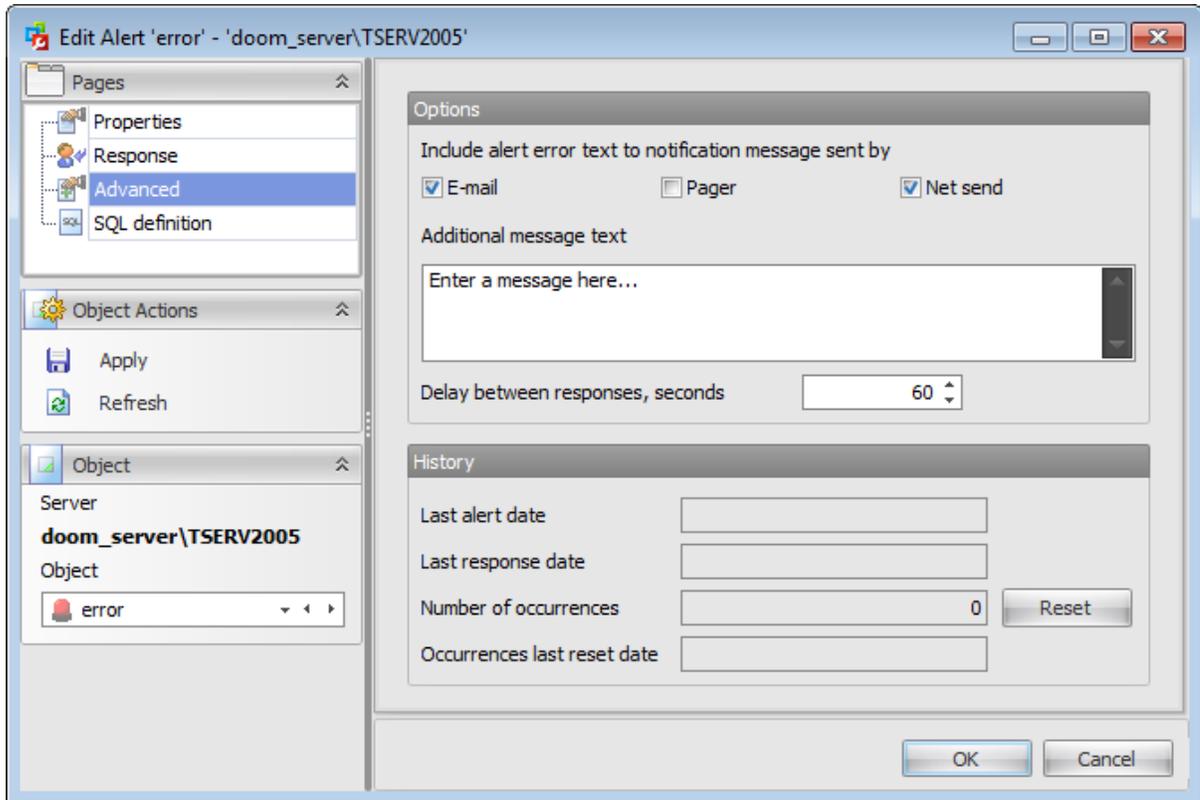
#### Notify operators when alert is raised

Alert notifications ensure that database administrators know when events/errors occur and which measures are taken in response. The main area lists all currently available [operators](#). Specify the method by which an operator is to be notified: *E-mail*, *Pager*, *Net Send*. Set the corresponding flag next to the operator name.

**Note:** You can add an operator by right-clicking the area and selecting **New operator...** item.

### 8.3.1.4 Setting advanced properties

The **Advanced** tab provides additional settings used for the specified alert.



#### Options

##### Include alert error text to notification message sent by

Specify in which way the description of the server error should be included as part of the notification message:

E-mail,  Pager,  Net send.

##### Additional message text

Use this box to specify an optional additional message sent to the [operator](#) as part of the e-mail, pager, or net send notification.

##### Delay between responses, seconds

Specify the wait period, in seconds, between responses to the alert.

#### History

The lower area displays the following alert events:

*Last alert date*

*Last response date*

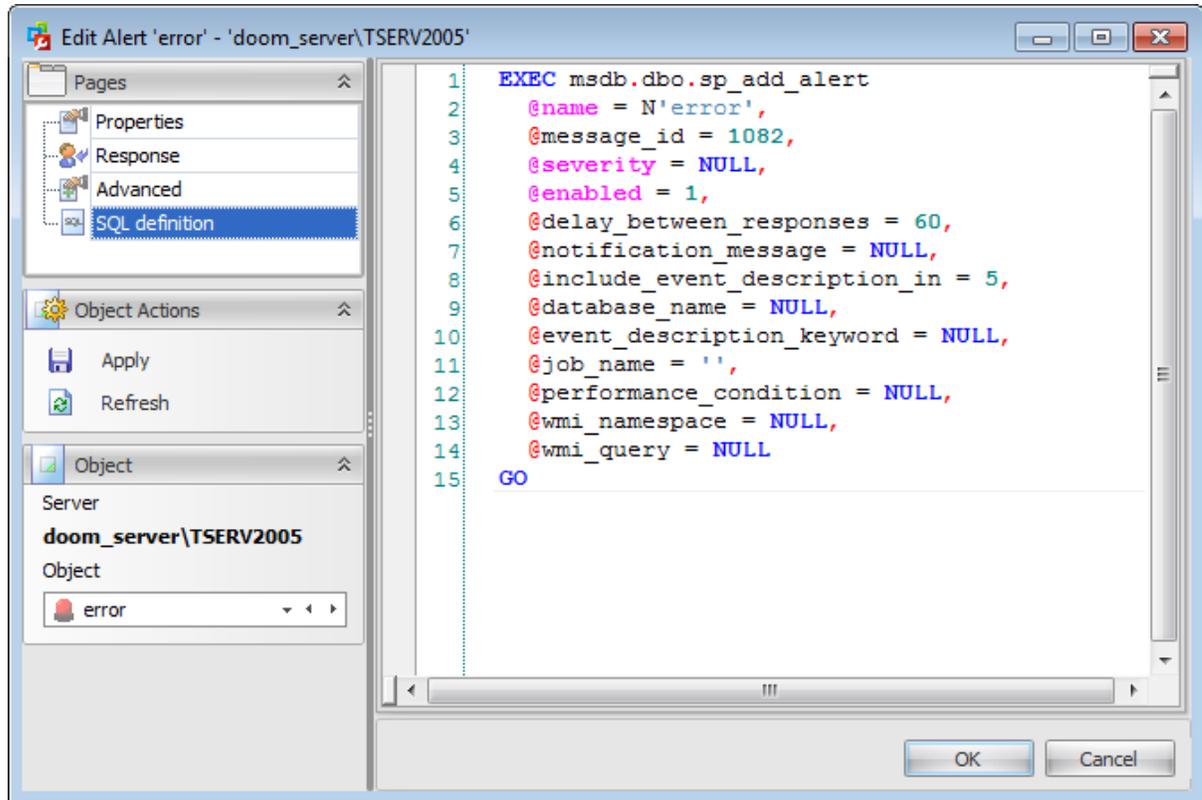
*Number of occurrences*

*Occurrences last reset date*

If necessary, you can **reset** the occurrence history by clicking the corresponding button.

### 8.3.1.5 Viewing SQL definition

This tab allows you to view the object DDL.



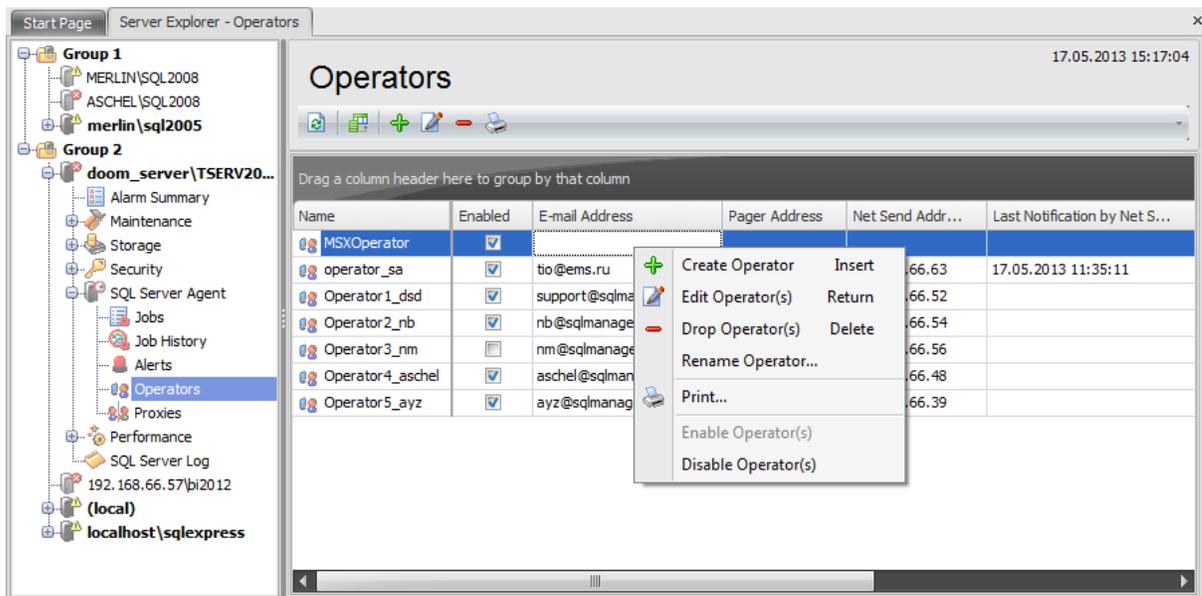
## 8.4 Operators

An **operator** is the object that represents a single SQL Server operator. SQL Server operators receive [alert](#) and [job](#) status notification in response to events generated by the server. You can assign [alert notifications](#) and [job notifications](#) for any operator defined for the instance of SQL Server.

**Note:** Before using SQL Server operators, make sure that the **SQL Server Agent** service is running.

Use the object context menu or toolbar buttons to:

- + [create](#) a new operator
-  [edit](#) selected operator(s)
-  drop selected operator(s)
-  create and [print](#) the report
- rename operator
- enable operator(s)
- disable operator(s)



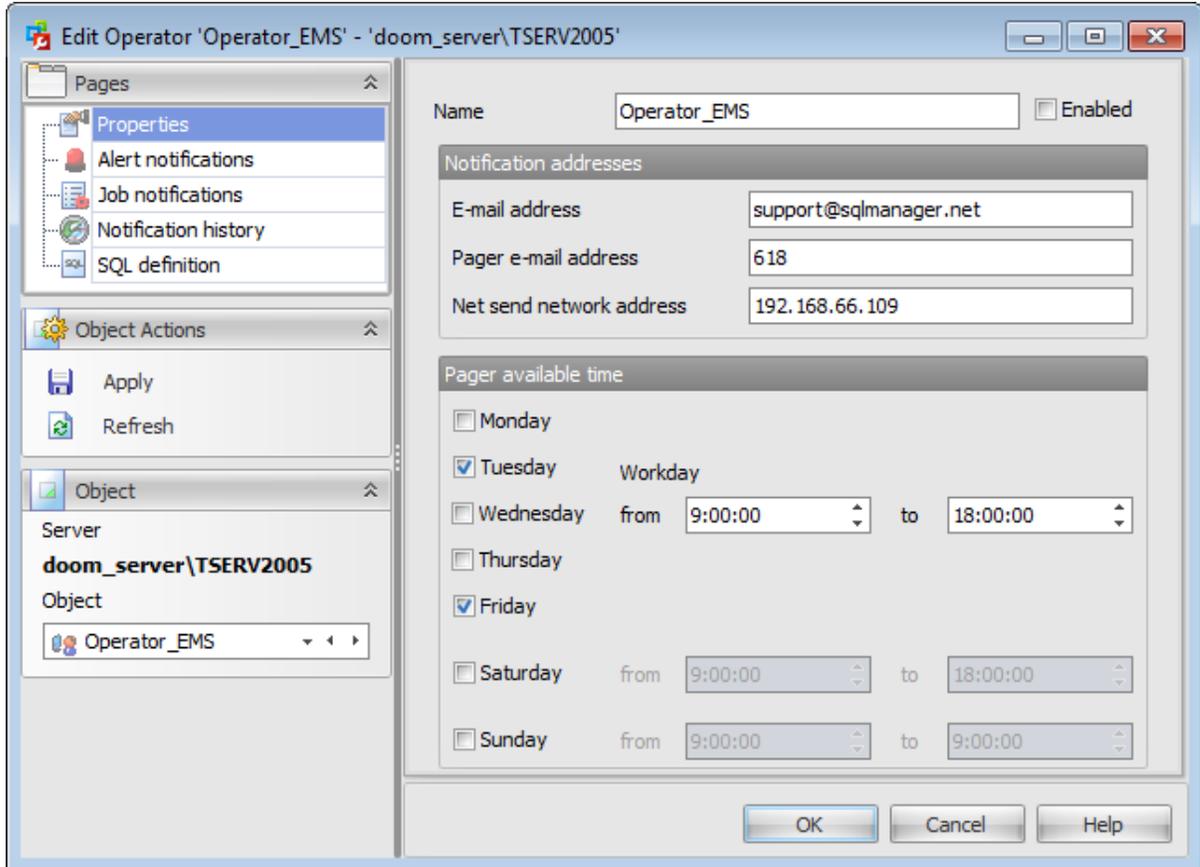
The list displays all existing operators as a grid with the following columns: *Name*, *Enabled*, *E-mail Address*, *Pager Address*, *Net Send Address*, *Last Notification by E-mail*, *Last Notification by Pager*, *Last Notification by Net Send*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To open an operator in **Operator Editor**, double-click it in the list of operators. Alternatively, you can right-click the operator alias and select the **Edit Operator** context menu item.

**See also:**[Jobs](#)[Job history](#)[Alerts](#)[Proxies](#)

## 8.4.1 Operator Editor

**Operator Editor** allows you to define operator properties. It opens automatically when you create a new operator and is available on editing an existing one.

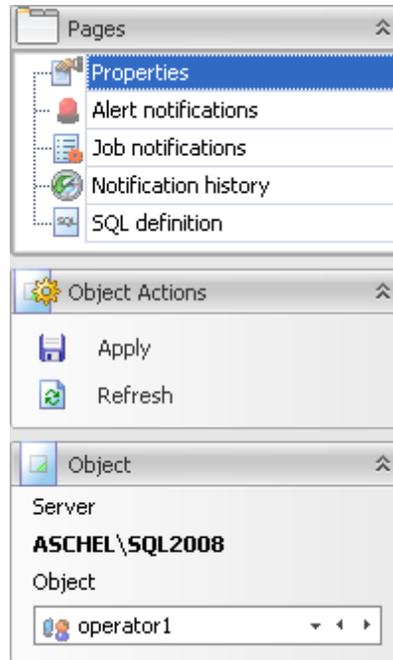


The list of **Operator Editor** tabs is displayed on the [navigation bar](#):

- [Setting operator properties](#)
- [Managing alert notifications](#)
- [Managing job notifications](#)
- [Viewing notification history](#)
- [Viewing SQL definition](#)

### 8.4.1.1 Using Navigation bar

The **Operator Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Operator Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Object

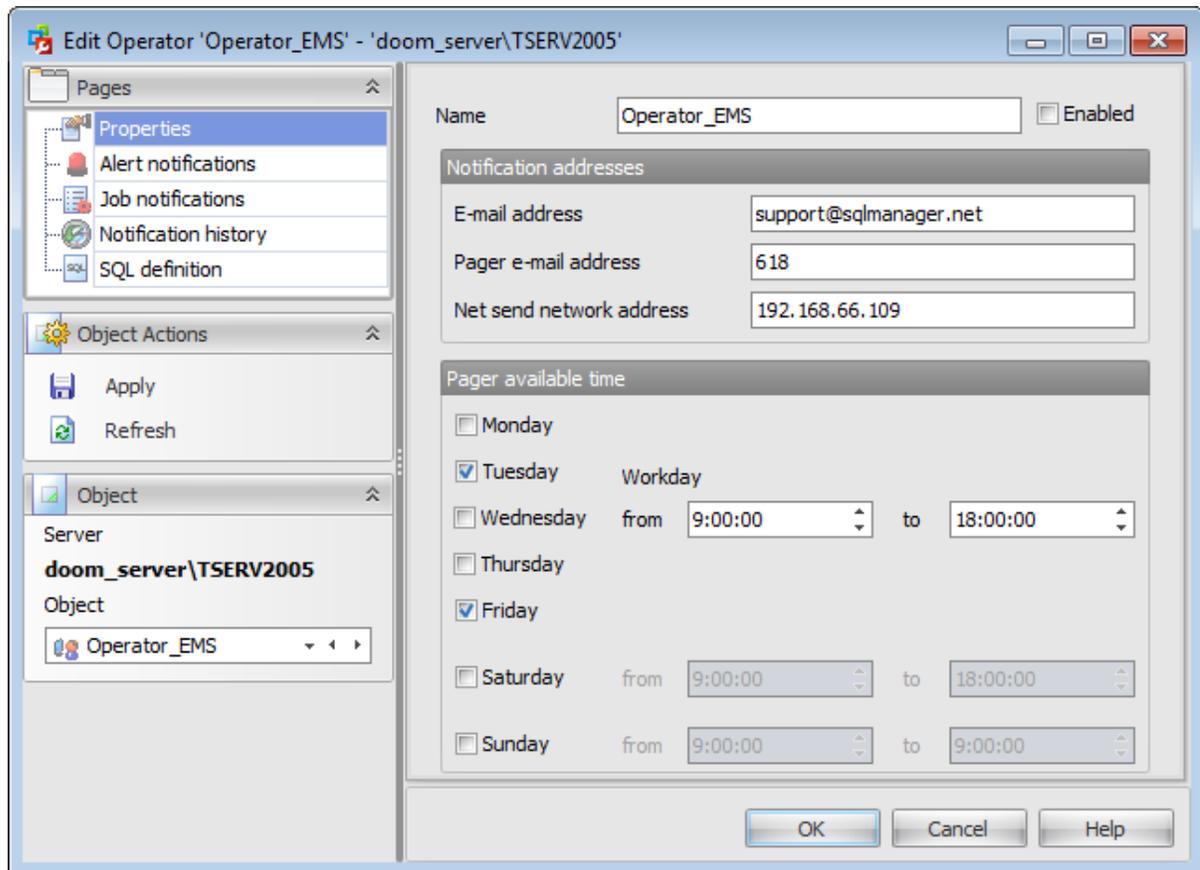
This field displays the operator that is currently being edited. However, you can change the edited operator by using the drop-down list of operators or by clicking the arrow-left or arrow-right buttons.

### 8.4.1.2 Setting operator properties

Use the **Properties** tab of the **Operator Editor** to configure/view operator (notification recipient) properties.

#### Name

Enter a name for the new operator, or modify the name of the operator being edited. Enable/disable the operator using the  **Enabled** option.



#### Notification addresses

##### E-mail address

Specify the e-mail address of the operator.

##### Pager e-mail address

Specify the pager address of the operator.

##### Net send address

Specify the network address of the operator to whom the network message will be sent.

#### Pager available time

Specify when SQL Server Agent service can send pager notification to the operator:

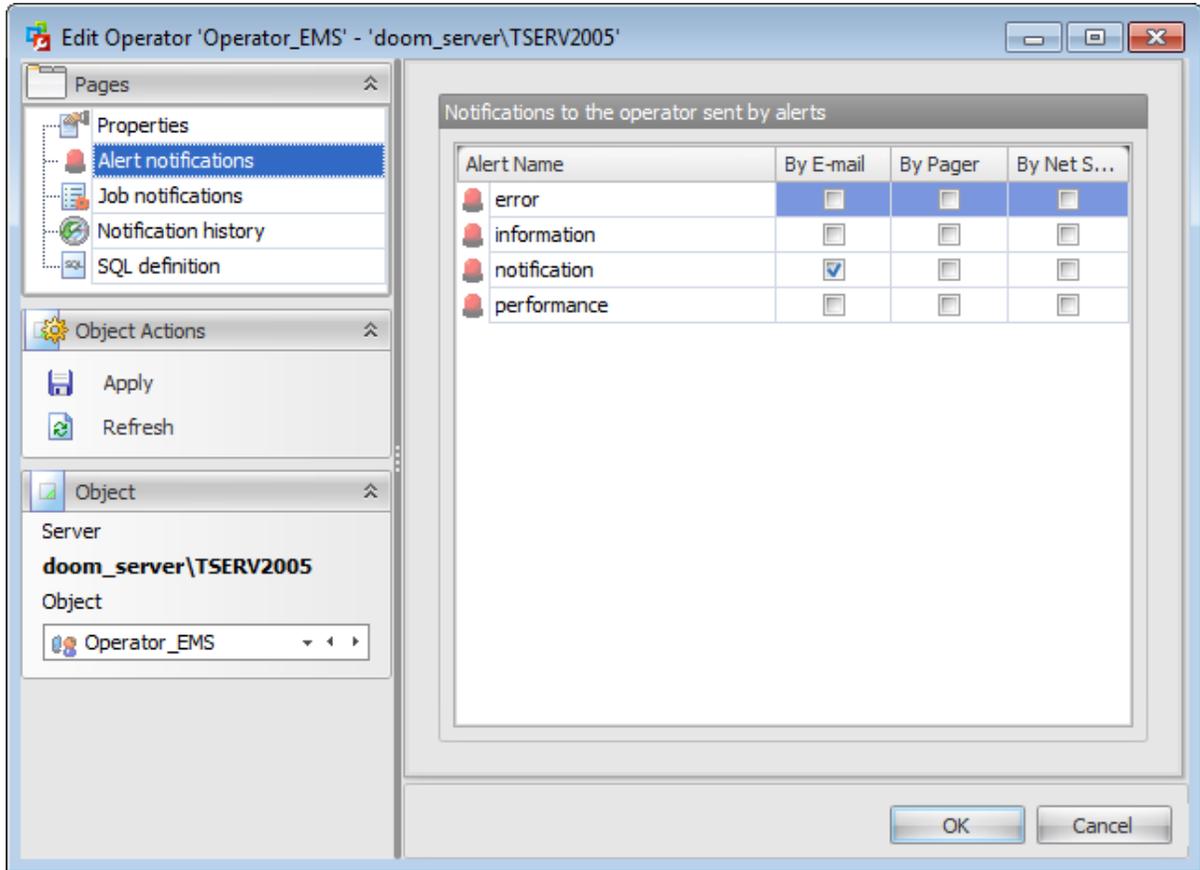
Monday,  Tuesday,  Wednesday,  Thursday,  Friday,  Saturday,  Sunday

From ... To ...

Set the day time interval values using the spinner controls.

### 8.4.1.3 Managing alert notifications

The **Alert Notifications** tab allows you to specify methods that the SQL Server Agent service uses to notify the operator when each of existing [alerts](#) fires.

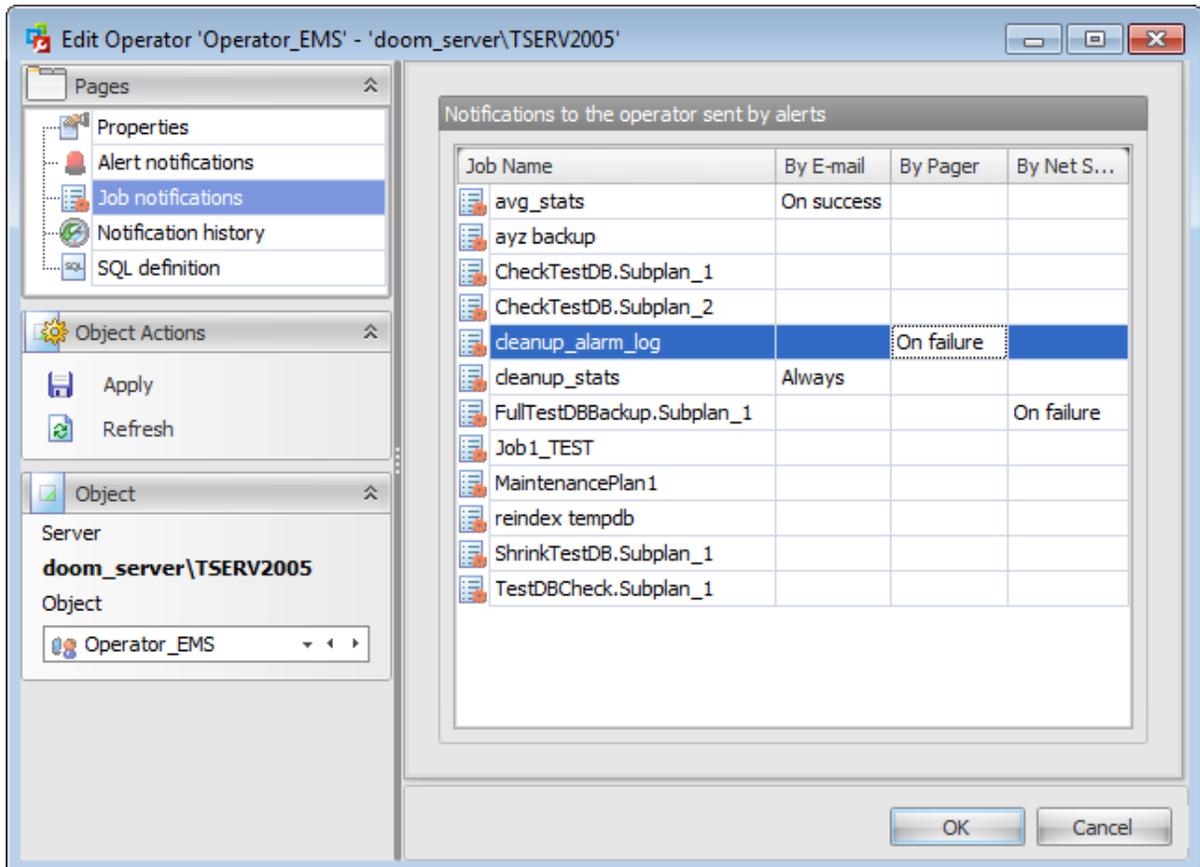


The main area lists all currently available [alerts](#). Specify the method by which the operator is to be notified: *E-mail*, *Pager*, *Net Send*. Set the corresponding flag next to the alert name.

**Note:** You can create a new [alert](#) by right-clicking the area and selecting **New alert...** item.

#### 8.4.1.4 Managing job notifications

The **Job Notifications** tab allows you to specify methods that the SQL Server Agent service uses to notify the operator when each of existing [jobs](#) is tried (succeeded or failed).



The grid lists all currently available [jobs](#). Specify the method by which the operator is to be notified: *E-mail*, *Pager*, *Net Send*. To set the notification type right-click a job, proceed to the [notifications](#) tab and choose preferable notification type.

**Note:** You can create a new [job](#) by right-clicking the area and selecting **New job...** item.

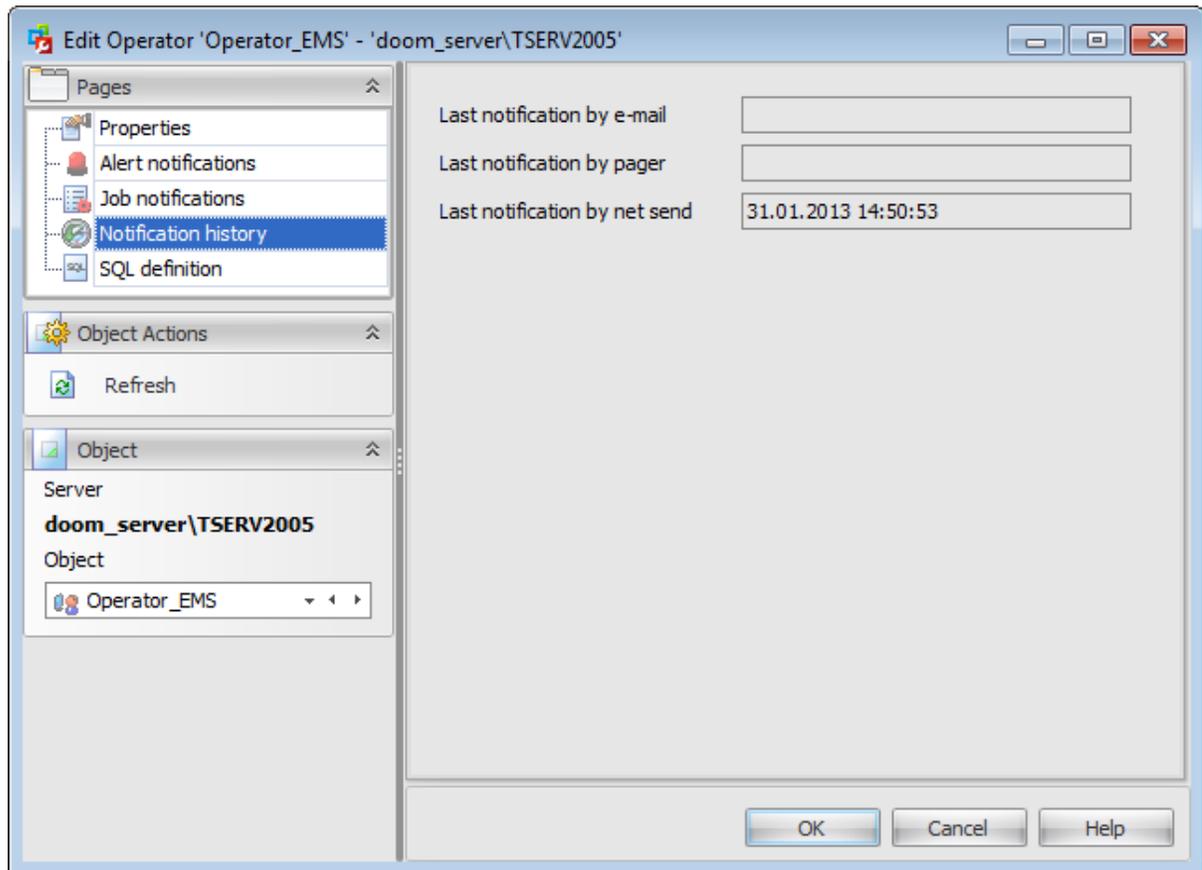
#### 8.4.1.5 Viewing notification history

The **Notification history** tab allows you to view the latest operator notifications:

*Last notification by e-mail*

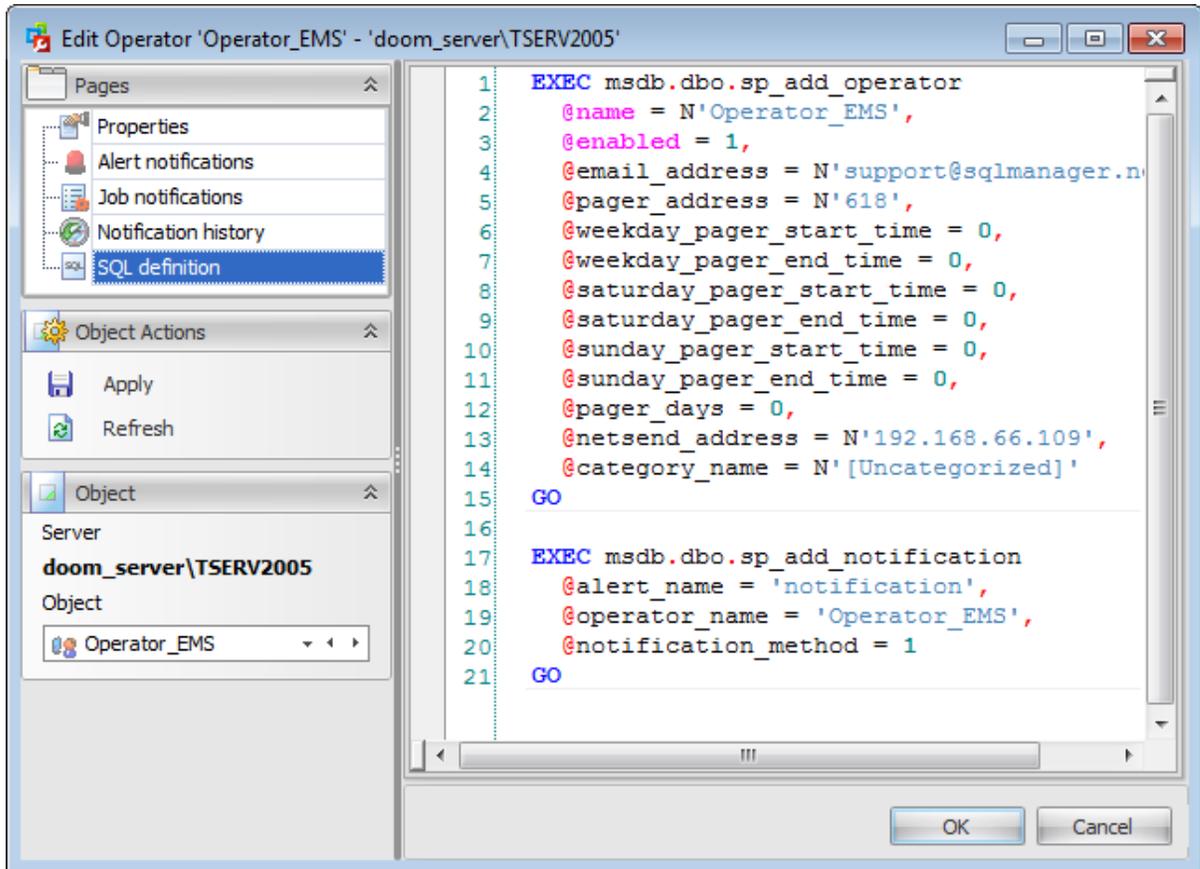
*Last notification by pager*

*Last notification by net send*



#### 8.4.1.6 Viewing SQL definition

This tab allows you to view the object DDL.



## 8.5 Proxies

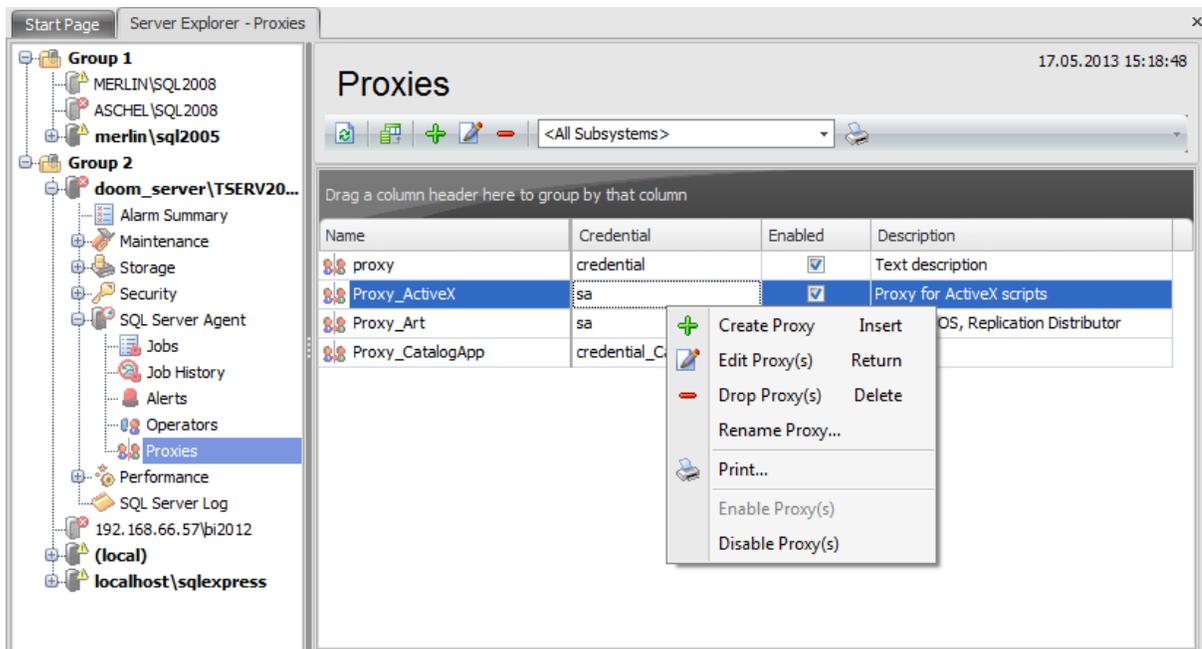
A Microsoft® SQL Server Agent **proxy** defines the security context for a [job step](#). A proxy provides SQL Server Agent with access to the security credentials for a Microsoft Windows user. Each proxy can be associated with one or more subsystems. A [job step](#) that uses the proxy can access the specified subsystems by using the security context of the Windows user. Before SQL Server Agent runs a [job step](#) that uses a proxy, SQL Server Agent impersonates the credentials defined in the proxy, and then runs the job step by using that security context.

SQL Server **proxies** can be used to help database administrators ensure that each job step runs with the minimum permissions required to perform its task.

**Note:** Before using SQL Server proxies, make sure that the **SQL Server Agent** service is running.

Use the object context menu or toolbar buttons to:

- + [create](#) a new proxy
-  [edit](#) selected proxy(ies)
-  drop selected proxy(ies)
-  create and [print](#) the report
- rename proxy
- enable proxy(ies)
- disable proxy(ies)



The list displays all existing proxies as a grid with the following columns: *Name*, *Credential*, *Enabled*, *Description*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

The drop-down menu on the toolbar allows you to browse among the subsystems to display only those proxies which are granted to the selected subsystem.

To open a proxy in **Proxy Editor**, double-click it in the list of proxies. Alternatively, you can right-click the proxy alias and select the **Edit Proxy** context menu item.

---

**See also:**

[Jobs](#)

[Job history](#)

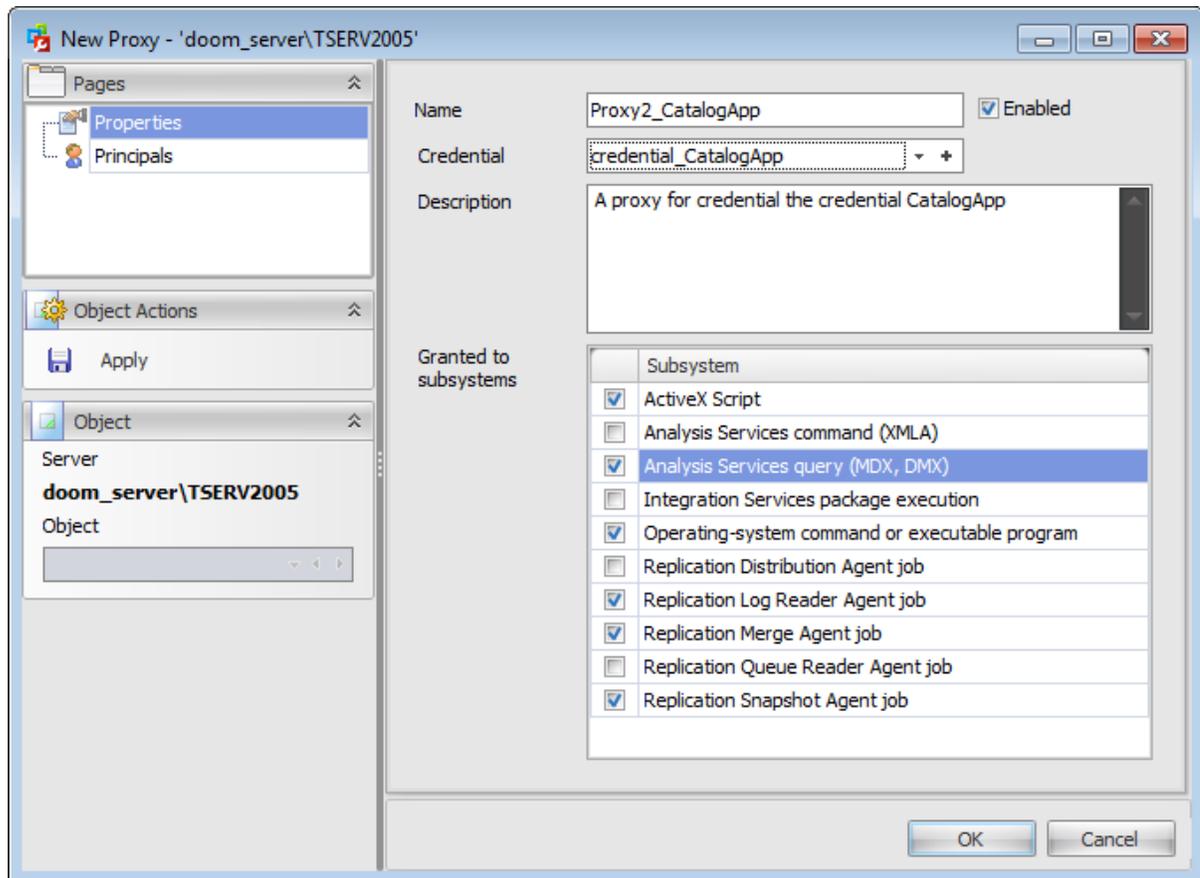
[Alerts](#)

[Operators](#)

## 8.5.1 Proxy Editor

**Proxy Editor** allows you to define proxy properties. It opens automatically when you create a new proxy and is available on editing an existing one.

A SQL Server Agent proxy manages security for [job steps](#) that involve subsystems other than the Transact-SQL subsystem. Each proxy corresponds to a security credential. A proxy may have access to any number of subsystems.

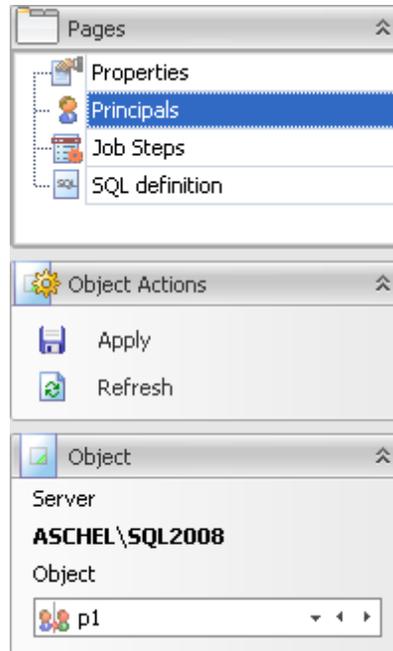


The list of **Proxy Editor** tabs is displayed on the [navigation bar](#):

- [Setting proxy properties](#)
- [Managing principals](#)
- [Managing job steps](#)
- [Viewing SQL definition](#)

### 8.5.1.1 Using Navigation bar

The **Proxy Editor** Navigation bar provides the number of panels that contain items which are necessary for the editing operation.



#### Pages

This group contains the list of **Proxy Editor** tabs which are clusters of properties that can be configured.

#### Object Actions

 apply changes

 refresh the content of the active tab

#### Object

##### Server

This field displays the name of the server where the edited object resides.

##### Object

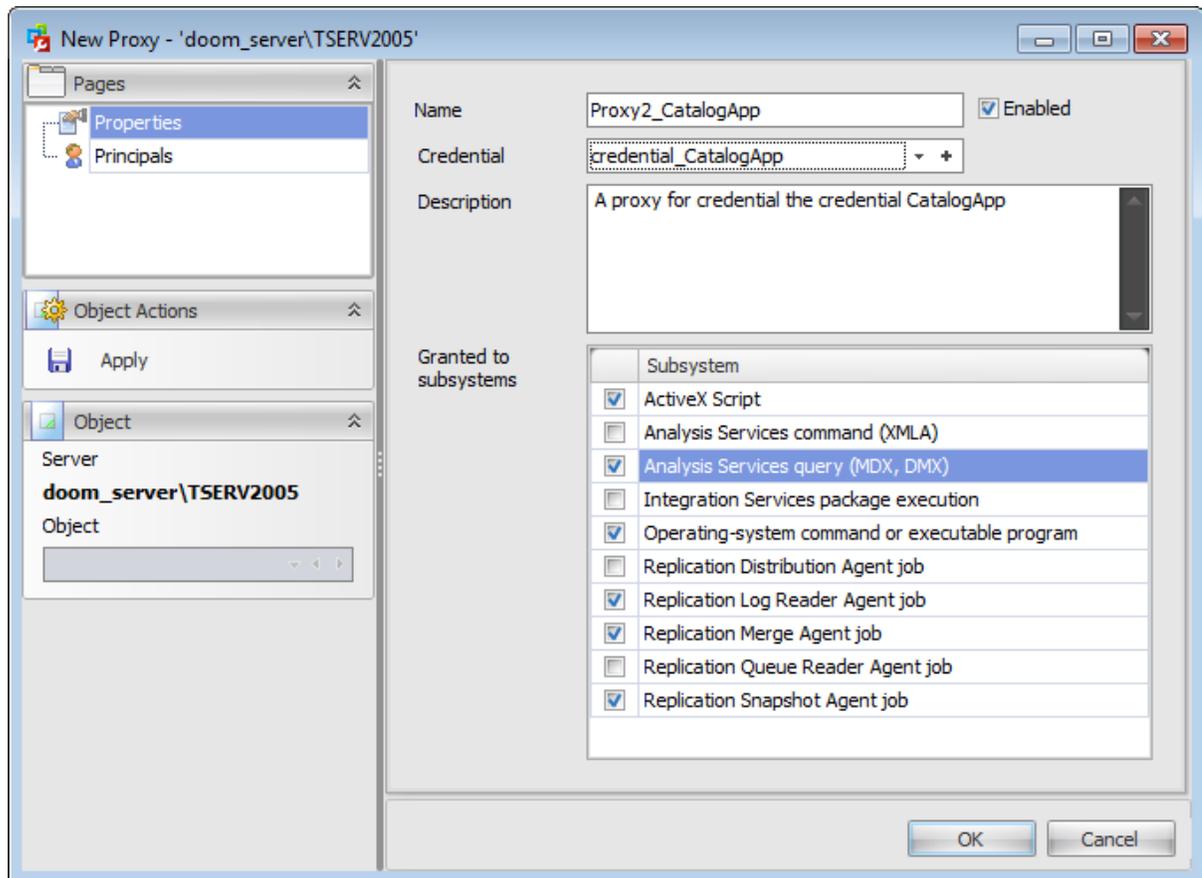
This field displays the proxy that is currently being edited. However, you can change the edited proxy by using the drop-down list of proxies or by clicking the arrow-left or arrow-right buttons.

### 8.5.1.2 Setting proxy properties

Use the **Properties** tab of **Proxy Editor** to configure/view proxy properties. SQL Server Agent uses **proxies** to manage security contexts.

#### Name

Enter a name for the new proxy, or modify the name of the proxy being edited. Enable/disable the proxy using the  **Enabled** option.



#### Credential

Each proxy corresponds to a security credential. Use the drop-down list to select a credential for the proxy. By clicking the **+ Add** button you can create a new credential using the [Credential Editor](#). Double-click the selected credential to edit its properties.

#### Description

In this memo field you can supply an optional description for the proxy.

#### Granted to subsystems

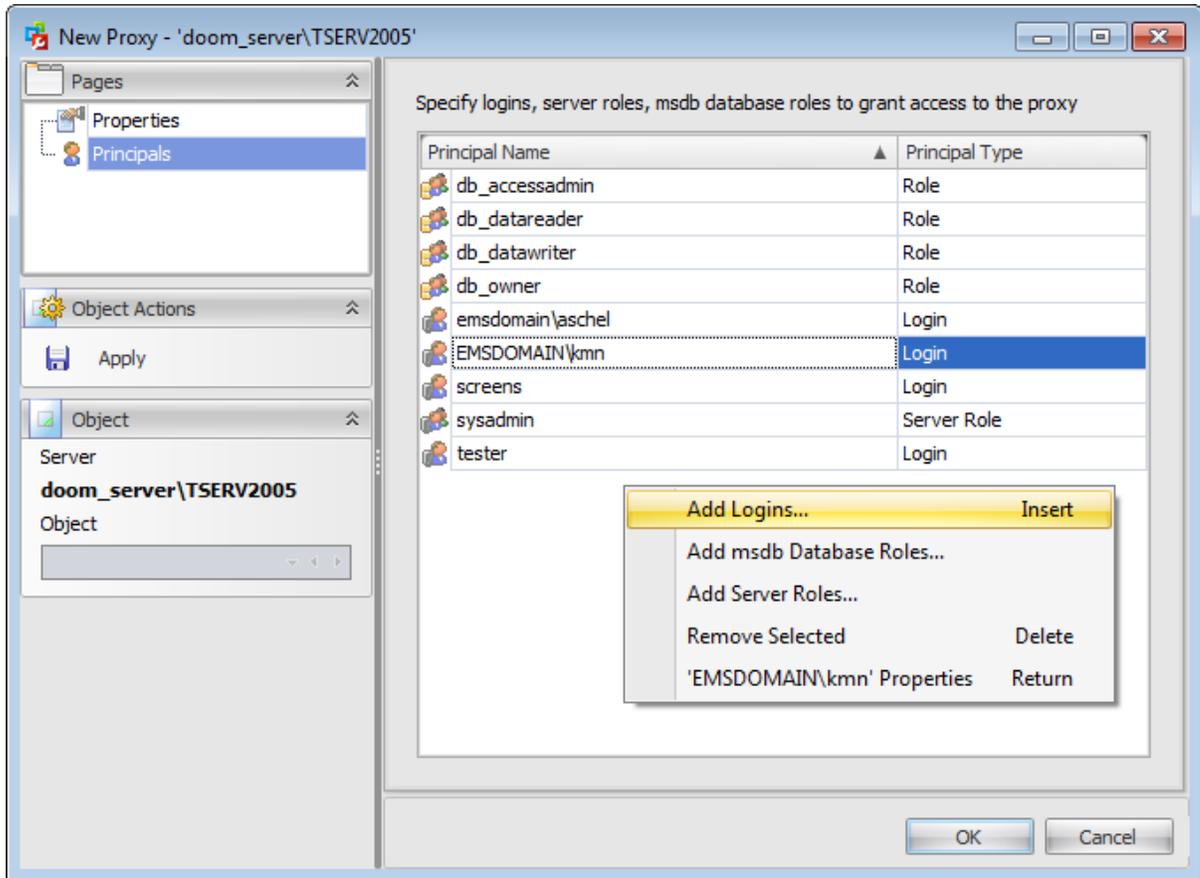
Tick off the subsystem(s) to grant access to:

- ActiveX Script*
- Analysis Services command (XMLA)*
- Analysis Services query (MDX, DMX)*
- Integration Services package execution*
- Operating-system command or executable program*

- PowerShell Script*
- Replication Distribution Agent job*
- Replication Log Reader Agent job*
- Replication Merge Agent job*
- Replication Queue Reader Agent job*
- Replication Snapshot Agent job*

### 8.5.1.3 Managing principals

The **Principals** tab allows you to grant security principal access to the proxy.

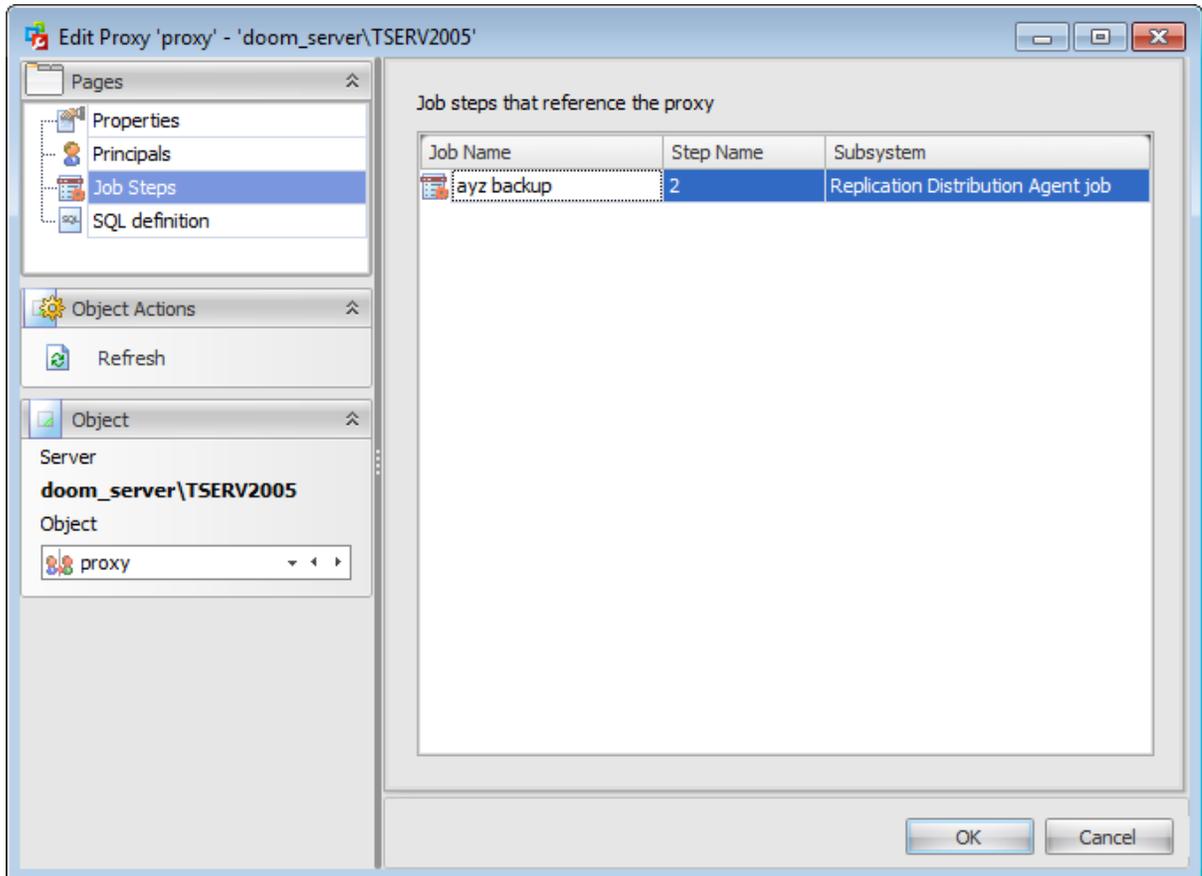


The grid displays the list of principals with a granted access to the proxy. To add a principal to this list right-click the grid and select **Add Logins...**, **Add msdb Database Roles** or **Add Server Roles...** context menu item.

#### 8.5.1.4 Managing job steps

The **Proxy Jobs** tab allows you to browse the list of [job steps](#) that use the proxy.

**Note:** The proxy can be used only for [job steps](#) that use a subsystem associated with the proxy. To create a job step that uses a specific proxy, the job owner must either belong to a login associated with that proxy or be a member of a role with unrestricted access to proxies.

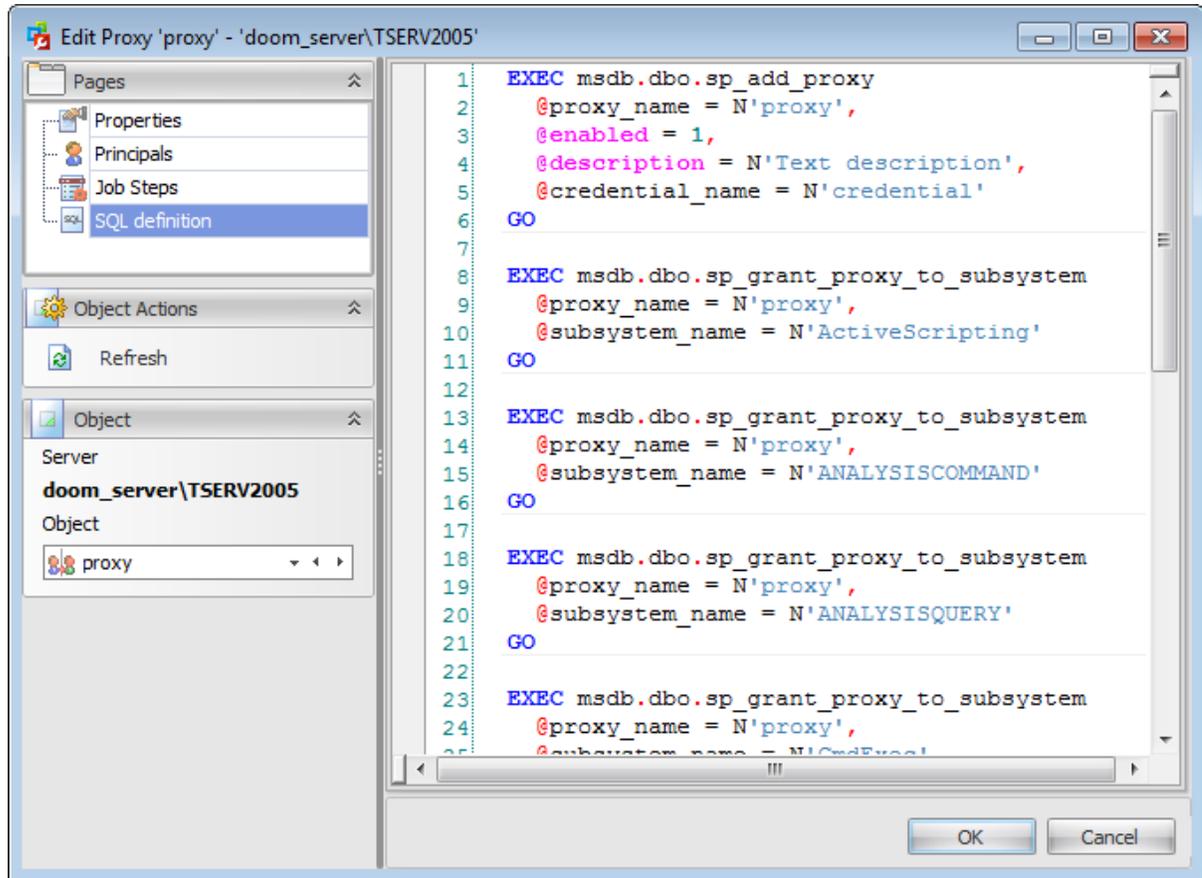


A proxy can be used in more than one [job step](#).

The list displays the job steps as a grid with the following columns: *Job Name*, *Step Name*, *Subsystem*.

### 8.5.1.5 Viewing SQL definition

This tab allows you to view the object DDL.

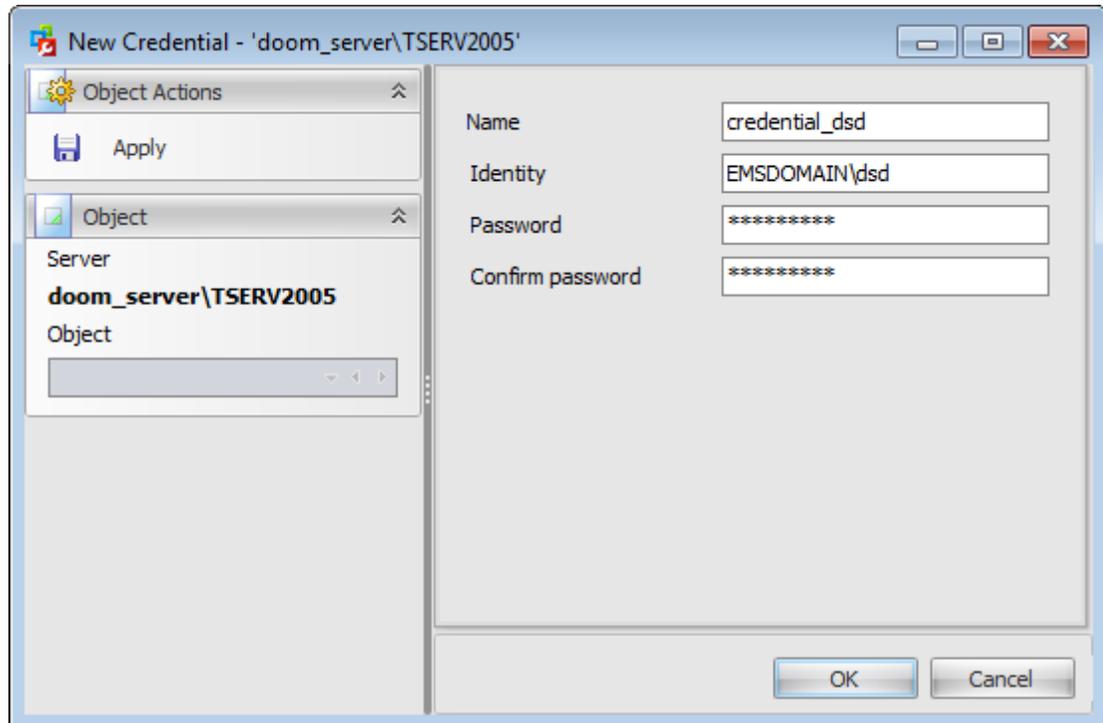


### 8.5.1.6 Credential Editor

A **Credential** is a record containing the authentication information required to connect to a resource outside of SQL Server. Most credentials consist of a Windows login and password.

Credentials allow [users](#) that connect to SQL Server using SQL Authentication to connect to Windows or other resources outside of SQL Server.

After creating a credential, you can map it to a [login](#). A single credential can be mapped to multiple SQL Server [logins](#). But a SQL Server [login](#) can be mapped to only one credential.



#### **Name**

Enter a name for the new credential, or view the name of the credential being edited.

#### **Identity with**

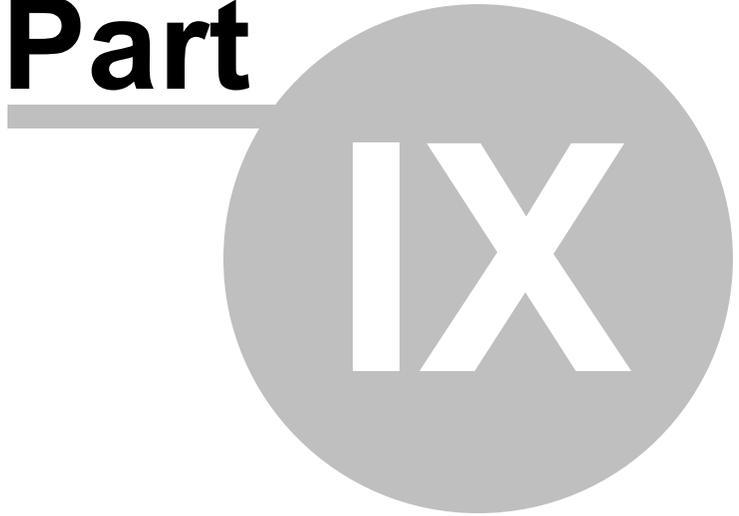
Use the drop-down list to select the name of the [account](#) to be used when connecting outside the server.

#### **Password**

Specify the password required for outgoing authentication, and **confirm the password** in the corresponding box.



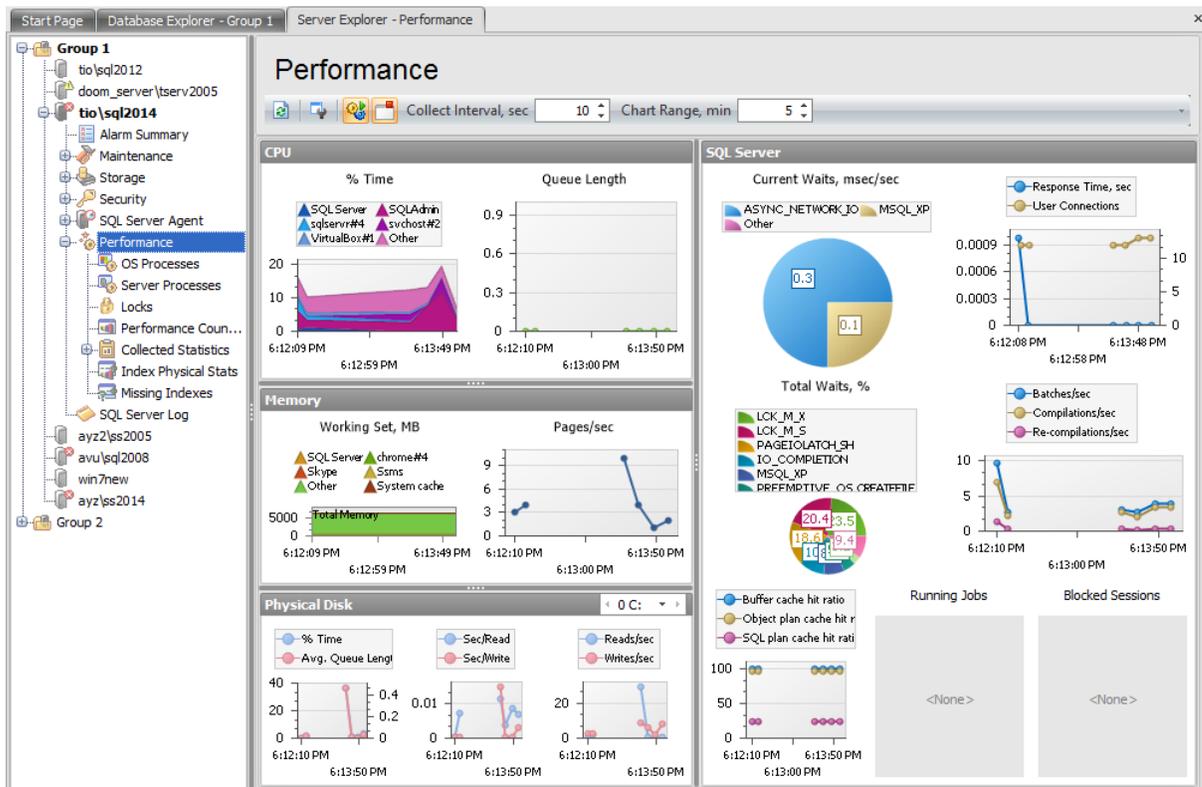
**Part**



## 9 Performance

The **Performance** section of the [Server Explorer](#) allows you to manage:

- [OS Processes](#)
- [Server Processes](#)
- [Locks](#)
- [Performance Counters](#)
- [Collected Statistics](#)
- [Collected Statistics Trend](#)
- [Index Physical Stats](#)
- [Missing Indexes](#)



Use the toolbar items to manage the displayed statistics:

- collect statistics automatically
- don't collect statistics when the view is not active
- **Collect Interval, sec** - when the statistics is collected automatically this value determines the frequency of the statistics collection
- **Chart Range, min** - this value determines the range of every chart

The work area of the **Performance** tab contains several sections with graphics and diagrams which are the visual representation of several of the *CPU*, *Memory*, *Physical Disk* and *SQL Server* counters.

### CPU % Time

This graphic provides a measure of how much time the processor actually spends working on productive threads and how often it was busy servicing requests. This counter actually provides a measurement of how often the system is doing nothing subtracted from 100%. Above the graphic you can see which processes are shown on the graph.

### **Queue Length**

This diagram shows how many threads (vertically) are ready in the processor queue, but not currently able to use the processor.

### **Memory**

#### **Working set, MB**

This graphic shows which part (in MB) of the total memory is taken by the processes with the highest memory consumption. These processes are listed above the graphic.

### **Pages/sec**

This diagram is a general indicator of how often the system is using the hard drive to store or retrieve memory associated data.

### **Physical Disk**

This section displays several diagrams of the selected physical disk performance.

#### **% Time**

Shows the percentage of elapsed time that the selected disk drive was busy servicing read or write requests.

#### **Avg. Queue Length**

Presents the average number of both read and write requests that were queued for the selected disk during the sample interval.

#### **Sec/Read**

Shows the average time (in seconds) of a read of data from the disk.

#### **Sec/Write**

Displays the average time (in seconds) of a write of data to the disk.

#### **Reads/sec**

Presents the rate of read operations on the disk.

#### **Reads/sec**

Presents the rate of write operations on the disk.

### **SQL Server**

Two circular diagrams show **Current** (msec/sec) and **Total** (%) waits. The list of waits is shown above the diagram.

#### **Response Time, sec**

This counter shows server response time (in seconds) when executing an SQL statement (the left vertical axis).

#### **User Connections**

Shows the number of users currently connected to the system (the right vertical axis).

#### **Batches/sec**

Presents the number of Transact-SQL command batches received per second. This statistic is affected by all constraints (such as I/O, number of users, cache size, complexity of requests, and so on). High batch requests mean good throughput.

**Compilations/sec**

Displays the number of SQL compilations.

**Re-compilations/sec**

Shows the number of SQL re-compiles.

**Buffer cache hit ratio**

Presents the percent of pages that were found in the buffer pool without having to incur a read from disk.

**Object plan cache hit ratio**

Displays the ratio between procedures, functions and triggers plans cache hits and lookups.

**SQL plan cache hit ratio**

Shows the ratio between SQL statements or batches plans cache hits and lookups.

**Running Jobs**

This list displays [jobs](#) that are currently running on the server.

**Blocked Sessions**

When a process is *blocked* it waits on the *blocking* process to finish before it continues execution. This can result in long running. In other words, SQL Server often intentionally prevents one process from accessing an object if another process has a lock on it. This means the second process is blocked from continuing until the first process releases the lock on the object to which the second process needs access.

Availability:

**Full** version                      **Yes**

**Free** version                      **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

---

**See also:**

[Getting Started](#)

[Servers](#)

[Database Maintenance](#)

[Storage](#)

[Security](#)

[SQL Server Agent](#)

[SQL Server Log](#)

[Query Editor](#)

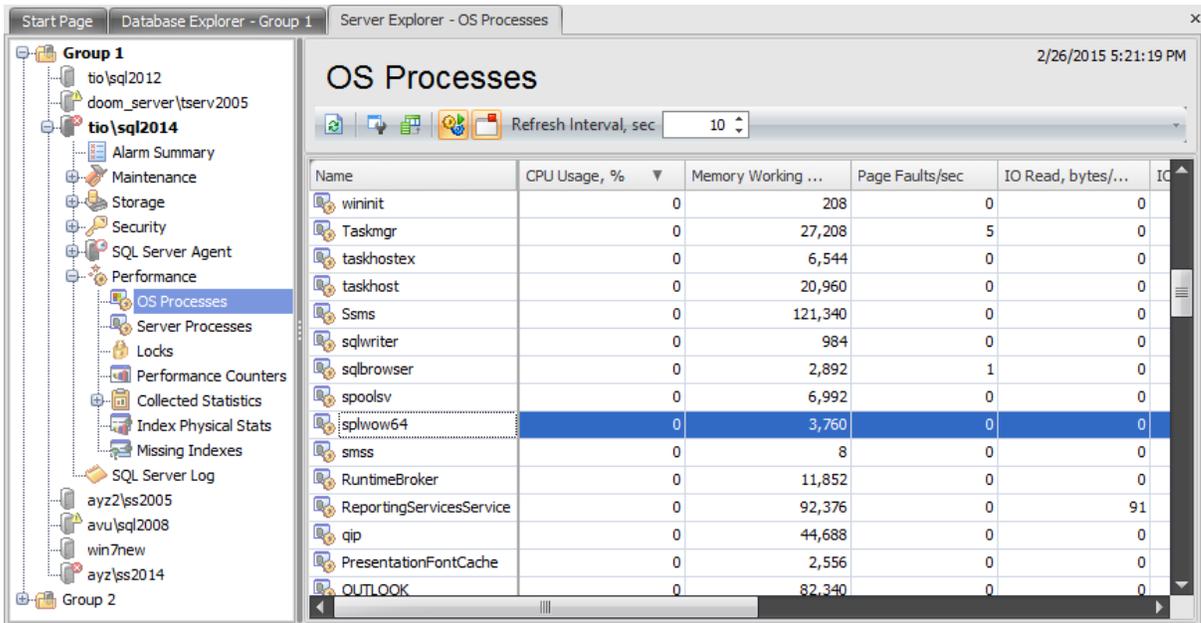
[Options](#)

## 9.1 OS Processes

This node of the **Performance** section displays all processes on the machine which is running the SQL Server. Here you can view properties of the current processes.

Use the toolbar items to manage the displayed processes:

-  refresh processes automatically
- don't refresh processes when the view is not active
- **Refresh Interval, sec** - when the processes are refreshed automatically this value determines the frequency of the refresh



Name	CPU Usage, %	Memory Working ...	Page Faults/sec	IO Read, bytes/...	IO Write, bytes/...	IO Other, bytes/...
wininit	0	208	0	0	0	0
Taskmgr	0	27,208	5	0	0	0
taskhostx	0	6,544	0	0	0	0
taskhost	0	20,960	0	0	0	0
Sms	0	121,340	0	0	0	0
sqlwriter	0	984	0	0	0	0
sqlbrowser	0	2,892	1	0	0	0
spoolsv	0	6,992	0	0	0	0
splwow64	0	3,760	0	0	0	0
sms	0	8	0	0	0	0
RuntimeBroker	0	11,852	0	0	0	0
ReportingServicesService	0	92,376	0	0	91	0
qip	0	44,688	0	0	0	0
PresentationFontCache	0	2,556	0	0	0	0
OUTLOOK	0	82,340	0	0	0	0

The work area displays all processes running on the server operating system as a grid with the following columns: *Name*, *CPU Usage (%)*, *Memory Working Set (KB)*, *Page Faults/sec*, *I/O Read (bytes/sec)*, *I/O Write (bytes/sec)*, *I/O Other (bytes/sec)*, *Process Description*.

Availability:

**Full** version **Yes**

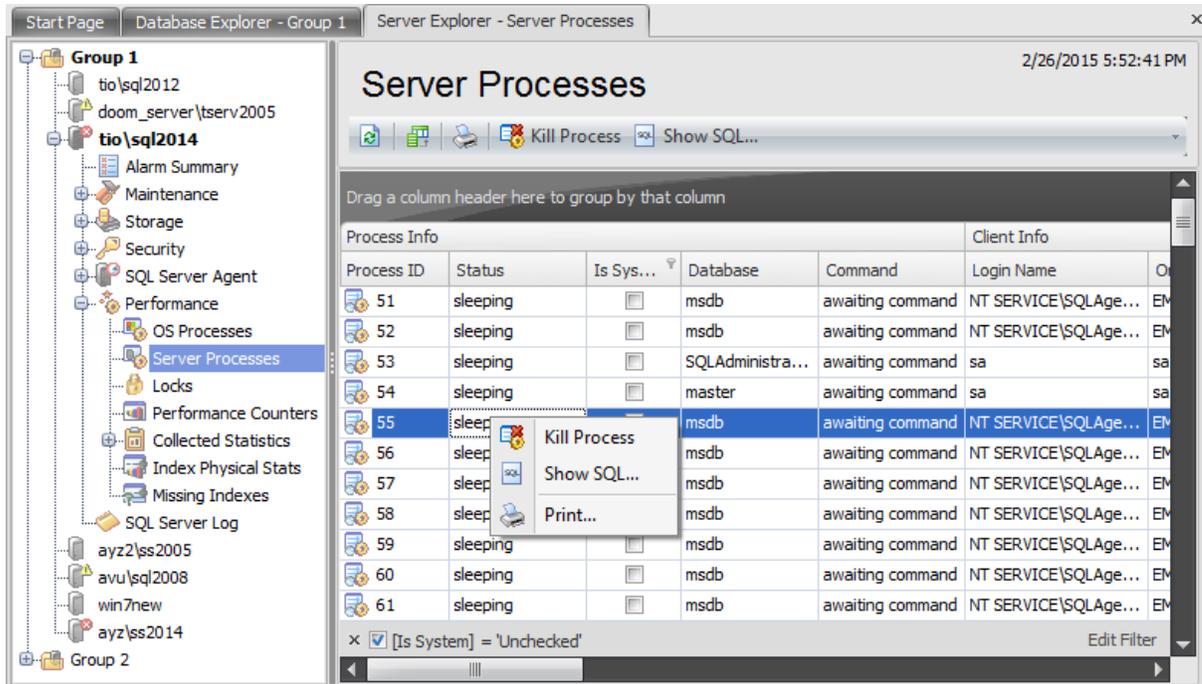
**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.2 Server Processes

This node of the **Performance** section displays all system and non-system processes running on the server. Here you can view properties of current processes. Using the context menu of the selected process or toolbar buttons you can:

-  kill the process
-  show command text of the process
-  create and [print](#) the report



The list displays processes as a grid with the following columns: **Process Info:** Process ID, Status, Is System, Database, Command; **Client Info:** Login Name, Original Login Name, Host Name, Program Name; **Locks:** Blocked By, Number of Blocking Processes, Wait Time, Last Wait Type, Wait Resource; **Process Statistics:** CPU Time, Physical IO, Reads, Writes, Logical Reads, Memory Usage (KB), Open Transaction, Transaction Isolation Level, Row Count. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

**Availability:**

**Full** version **Yes**

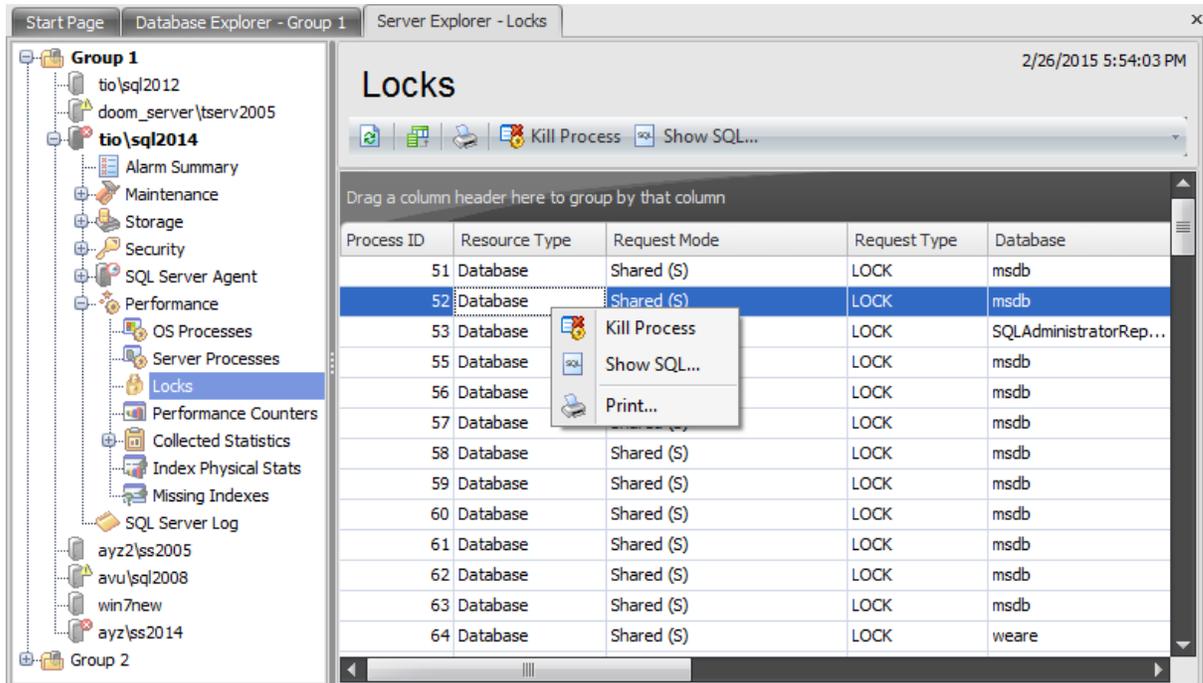
**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.3 Locks

This node of the **Performance** section displays locks belonging to specific processes. Using the context menu of the selected process or toolbar buttons you can:

-  kill the process
-  show command text of the process
-  create and [print](#) the report



The list displays locks as a grid with the following columns: *Process ID*, *Request Mode*, *Resource Type*, *Resource Subtype*, *Database*, *Schema*, *Object*, *Resource Description*, *Request Mode*, *Request Type*, *Request Owner GUID*, *Request Owner ID*, *Request Status*, *Request Reference Count*, *Request Owner Type*.

*Shared (S)* locks allow concurrent transactions to read (SELECT) a resource. No other transactions can modify the data while shared (S) locks exist on the resource.

*Update (U)* locks are used on resources that can be updated. They locks prevent a common form of deadlock.

*Exclusive (X)* locks prevent access to a resource by concurrent transactions.

*Intent shared (IS)* locks indicate the intention of a transaction to read some (but not all) resources lower in the hierarchy by placing S locks on those individual resources.

*Intent exclusive (IX)* locks indicate the intention of a transaction to modify some (but not all) resources lower in the hierarchy by placing X locks on those individual resources.

*Shared with intent exclusive (SIX)* locks indicates the intention of the transaction to read all of the resources lower in the hierarchy and modify some (but not all) resources lower in

the hierarchy by placing IX locks on those individual resources. Concurrent IS locks at the top-level resource are allowed.

*Schema modification (Sch-M)* locks are used when a table data definition language (DDL) operation (such as adding a column or dropping a table) is being performed.

*Schema stability (Sch-S)* locks are used when compiling queries. Schema stability (Sch-S) locks do not block any transactional locks, including exclusive (X) locks. Therefore, other transactions can continue to run while a query is being compiled, including transactions with exclusive (X) locks on a table. However, DDL operations cannot be performed on the table.

*Bulk update (BU)* locks allow processes to bulk copy data concurrently into the same table while preventing other processes that are not bulk copying data from accessing the table.

Availability:

**Full** version                      **Yes**

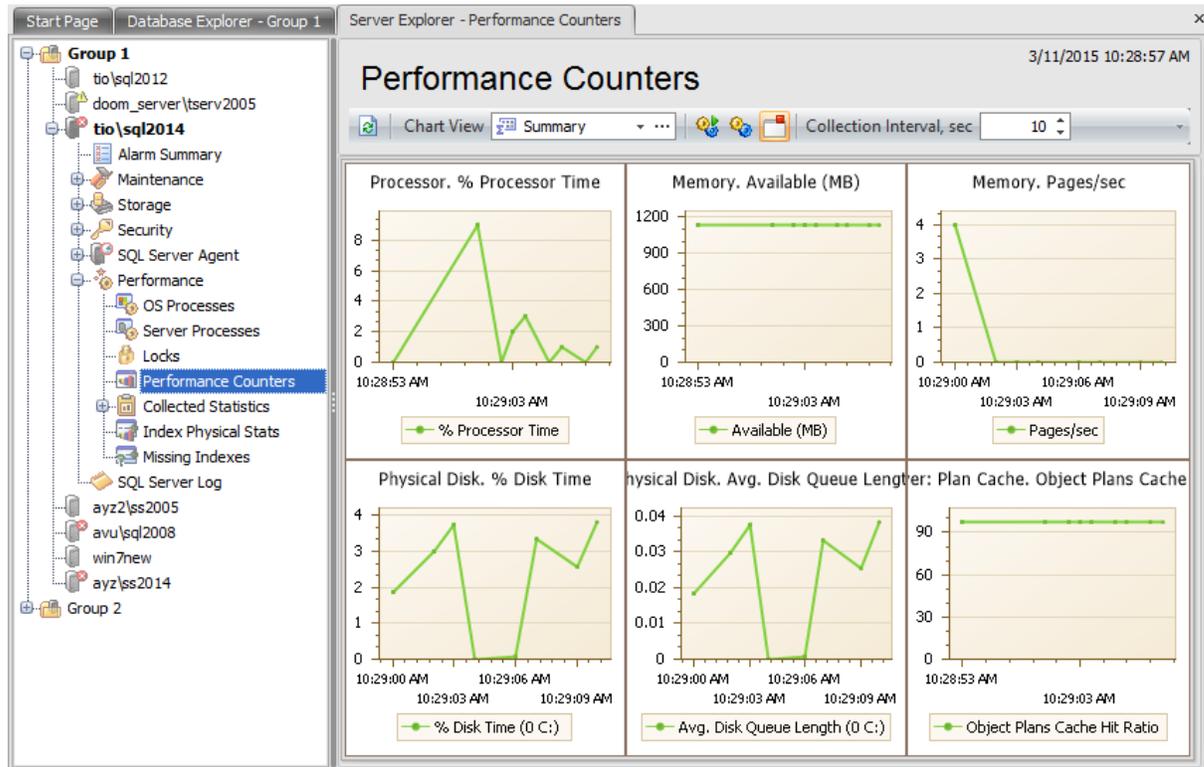
**Free** version                      **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.4 Performance Counters

This node of the **Performance** section shows you time diagrams of the statistics collected by SQL Administrator. These diagrams are graphed on the basis of the SQL Server productivity counters values.

Statistics help you to learn about the current SQL Server productivity.



Initially the work area is separated into six sections:

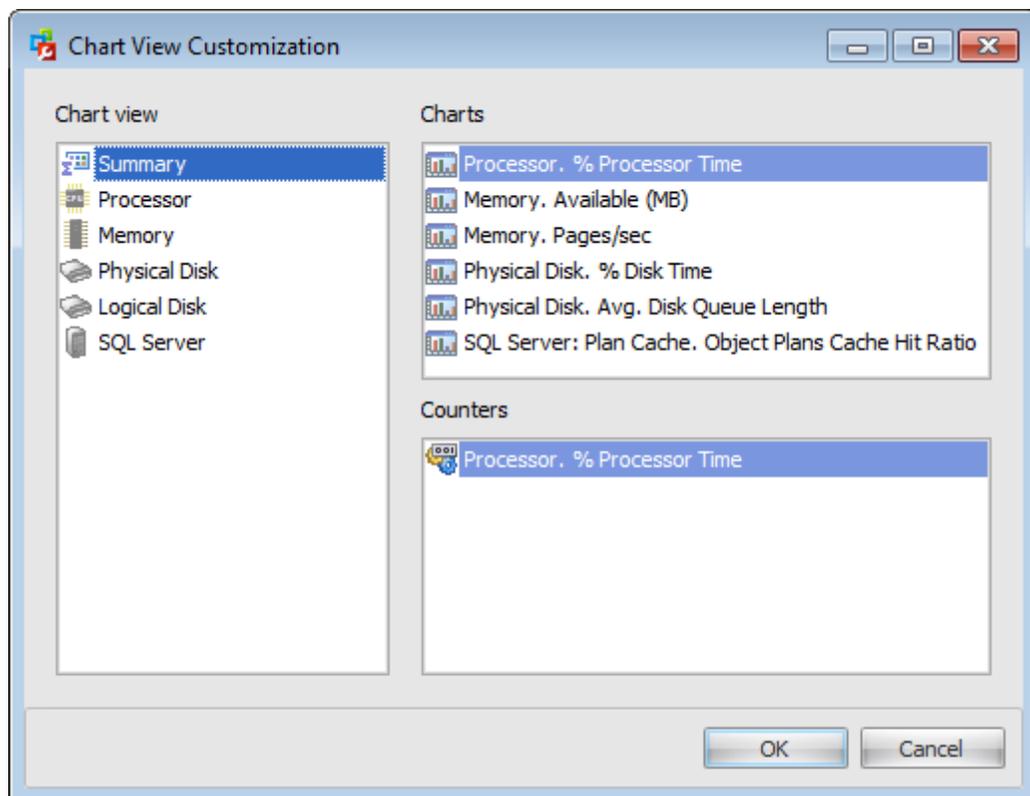
- *Processor. % Processor Time*. It displays a diagram of the processor time, that is the percentage of time the processor is busy by measuring the percentage of time the thread of the Idle process is running and then subtracting that from 100 percent. This measurement is the amount of processor utilization.
- *Memory. Available (MB)*. This diagram shows the amount of memory that is left after nonpaged pool allocations, paged pool allocations, process' working sets, and the file system cache have all taken their piece.
- *Memory. Pages/sec*. The Pages/sec diagram shows the rate at which pages are read from or written to disk to resolve hard page faults.
- *Physical Disk. % Disk Time*. This diagram is a general mark of how busy the disk is. It indicates a disk problem, but must be observed in conjunction with the Disk Queue Length counter to be truly informative.
- *Physical Disk. Avg. Disk Queue Length*. This counter converts the % Disk Time to a decimal value and displays it.
- *SQL Server: Plan Cache. Object Plans Cache Hit Ratio*. Ratio between cache hits and lookups.

You can enlarge the particular section by double-clicking it.

Use the toolbar items to manage the displayed statistics:

- **Diagram Set** - select the necessary set of the diagrams
  -  collect statistics automatically
  -  collect statistics now
  - don't collect statistics when the view is not active
- **Collection Interval, sec** - when the statistics is collected automatically this value determines the frequency of the statistics collection
- **Chart Range, min** - this value determines the range of every chart
  -  set counter instance filter, i.e. set disks for some counters
  -  create and [print](#) the report

Click the  button (in the **Diagram Set** field) to customize the chart view. In the window appeared you can set displayed charts and used counters.



### Chart view

Using this section you can set the groups of charts that then could be selected in the **Diagram Set** field.

The context menu of the section allows you to:

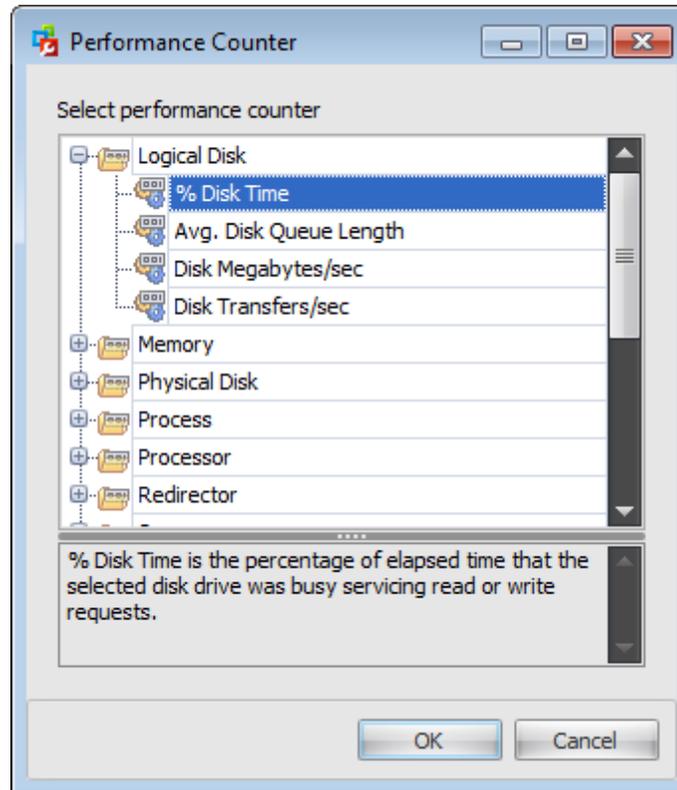
-  create chart view
-  rename chart view
-  delete chart view
-  move up
-  move down

## Charts

Use this section to manage charts in the chart view:

- + create chart
- ✎ rename chart
- delete chart
- ↑ move up
- ↓ move down

When creating a new chart the **Performance Counter** window appears:



It lists groups of *performance counters* that help you to determine the performance of the processor, logical and physical disks, memory, server etc. To create a new chart select the necessary performance counter from the list and click the 'OK' button.

## Counters

Use this section to manage performance counters of the selected chart:

- + add counter
- remove counter
- ↑ move up
- ↓ move down

The **Performance Counter** window appears on the adding a counter action.

**Note:** Some statistics are not collected if the [WMI Connection](#) to the server is not set.

Availability:

**Full** version **Yes**

**Free** version **No**

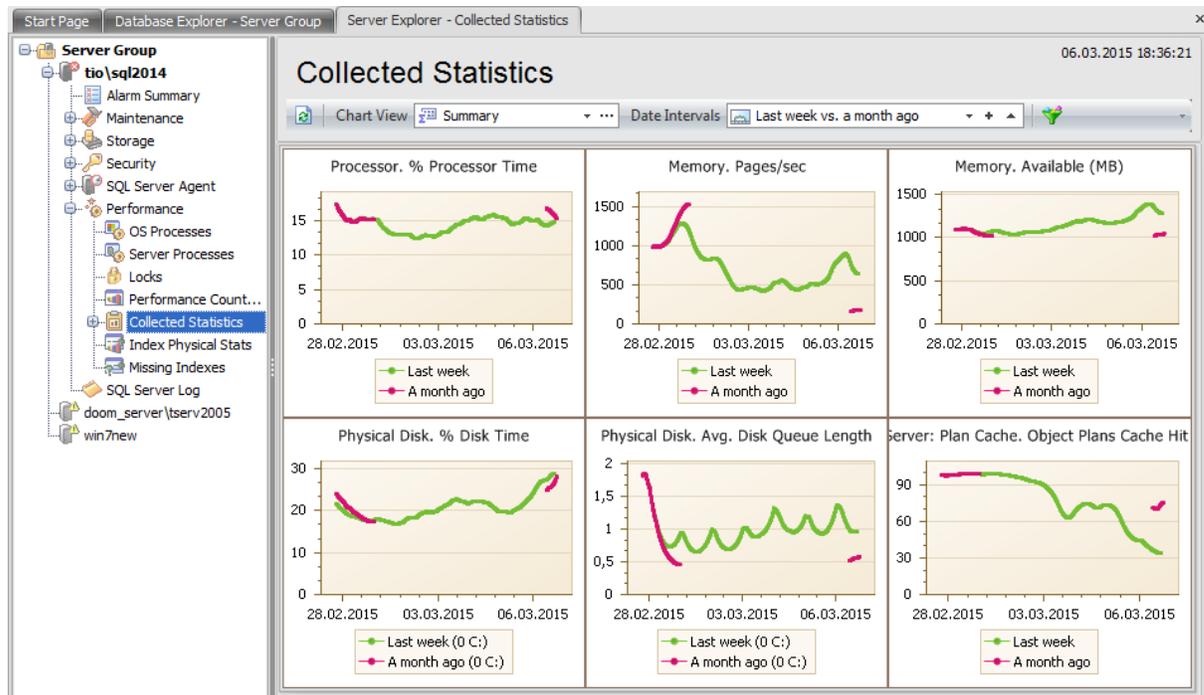
**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.5 Collected Statistics

This node of the **Performance** section shows you time diagrams of the collected statistics. But unlike the [Current Statistics](#) it is collected via [Statistics Collector](#) service. It records the data (performance counters values) about SQL server productivity to the defined repository database. Diagrams are graphed on the basis of the data stored in that database.

These statistics help you to learn about the current SQL Server productivity as well as the [Current Statistics](#) do, but also allow you to compare the server productivity performance within the defined time intervals.

**Note:** Gaps on diagrams indicate periods of time when the [Statistics Collector](#) service was not running.



Initially the work area is separated into six sections:

- *Processor. % Processor Time.* It displays a diagram of the processor time, that is the percentage of time the processor is busy by measuring the percentage of time the thread of the Idle process is running and then subtracting that from 100 percent. This measurement is the amount of processor utilization.
- *Memory. Available (MB).* This diagram shows the amount of memory that is left after nonpaged pool allocations, paged pool allocations, process' working sets, and the file system cache have all taken their piece.
- *Memory. Pages/sec.* The Pages/sec diagram shows the rate at which pages are read from or written to disk to resolve hard page faults.
- *Physical Disk. % Disk Time.* This diagram is a general mark of how busy the disk is. It indicates a disk problem, but must be observed in conjunction with the Disk Queue Length counter to be truly informative.
- *Physical Disk. Avg. Disk Queue Length.* This counter converts the % Disk Time to a decimal value and displays it.

- *SQL Server: Plan Cache. Object Plans Cache Hit Ratio.* Ratio between cache hits and lookups.

You can enlarge the particular section by double-clicking it.

Use the toolbar items to manage the displayed statistics:

- **Chart View** - select the necessary set of the diagrams
- **Date Intervals** - select the period during which you want to view collected statistics
-  set counter instance filter, i.e. set disks for some counters
- **Smoothing** - use the '+' / '-' buttons or just move the runner of this control to set the preferable graph smoothing
-  create and [print](#) the report

### Setting chart view

Using this section you can set the groups of charts that then could be selected in the **Chart View** field.

The context menu of the section allows you to:

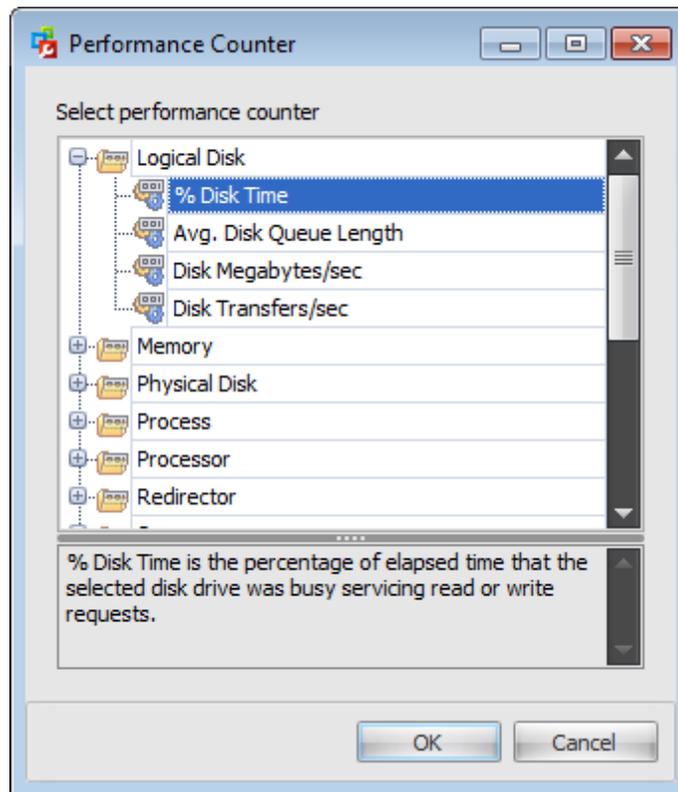
-  create chart view
-  rename chart view
-  delete chart view
-  move up
-  move down

### Charts

Use this section to manage charts in the chart view:

-  create chart
-  rename chart
-  delete chart
-  move up
-  move down

When creating a new chart the **Performance Counter** window appears:



It lists groups of *performance counters* that help you to determine the performance of the processor, logical and physical disks, memory, server etc. To create a new chart select the necessary performance counter from the list and click the 'OK' button.

### Counters

Use this section to manage performance counters of the selected chart:

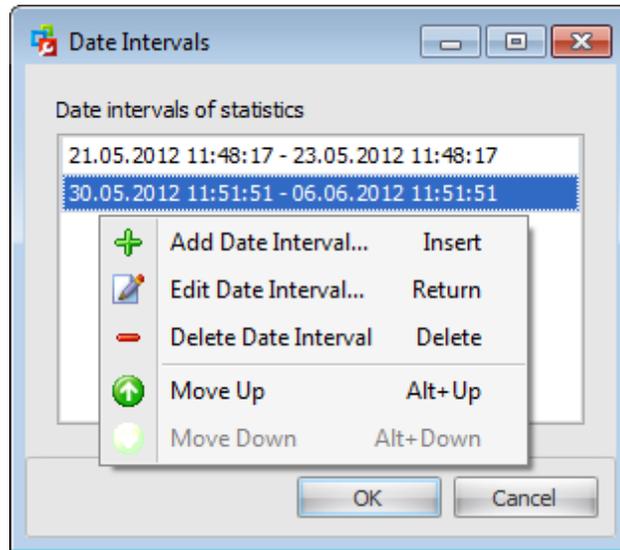
- + add counter
- remove counter
- ↑ move up
- ↓ move down

The **Performance Counter** window appears on the adding a counter action.

**Note:** Some statistics are not collected if the [WMI Connection](#) to the server is not set.

### Setting date interval

When configuring the date interval you can add the preferable period manually by clicking the + button or edit an existing one by clicking the ^ button. These buttons open the **Date Intervals** dialog.

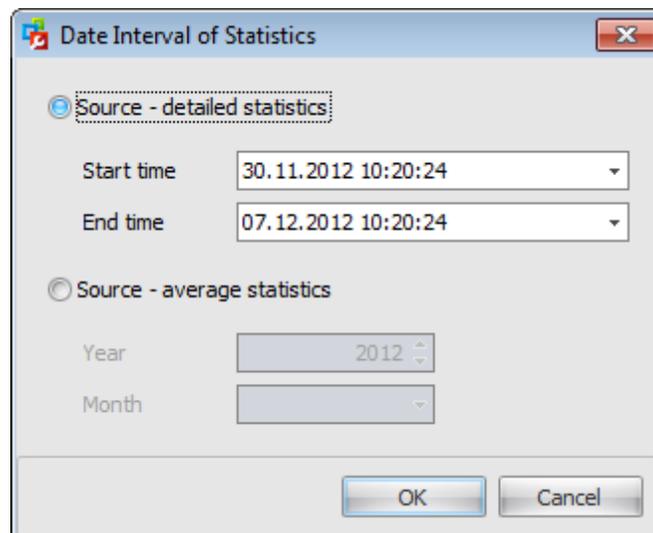


This window shows date intervals of the time domain. So, here you can define at which periods the collected statistics should be displayed.

The context menu of the list allows you to:

- + create date interval
- edit date interval
- delete date interval
- move up
- move down

The following window appears on adding or editing a date interval:



First, select the source:  **Source - detailed statistics** or  **Source - average statistics**.

In the first case use the **Start time** and **End time** fields to set the time interval. You can

enter it manually or pick up a date from the drop-down calendar.

In the second case set the **Year** and **Month** in the corresponding fields to display average statistics collected in this month.

**Note:** If you set several date intervals then the first one will be displayed on the horizontal axis. If there are longer intervals in the set then a difference between its length and the first interval length will be added to the axis. So the length is equal to the duration of the longest interval but the start point is the start time of the first interval of the set.

Also, if you have set an average statistics interval among the current statistics intervals then the average statistics is displayed in compliance with the longest interval.

Availability:

**Full** version                      **Yes**

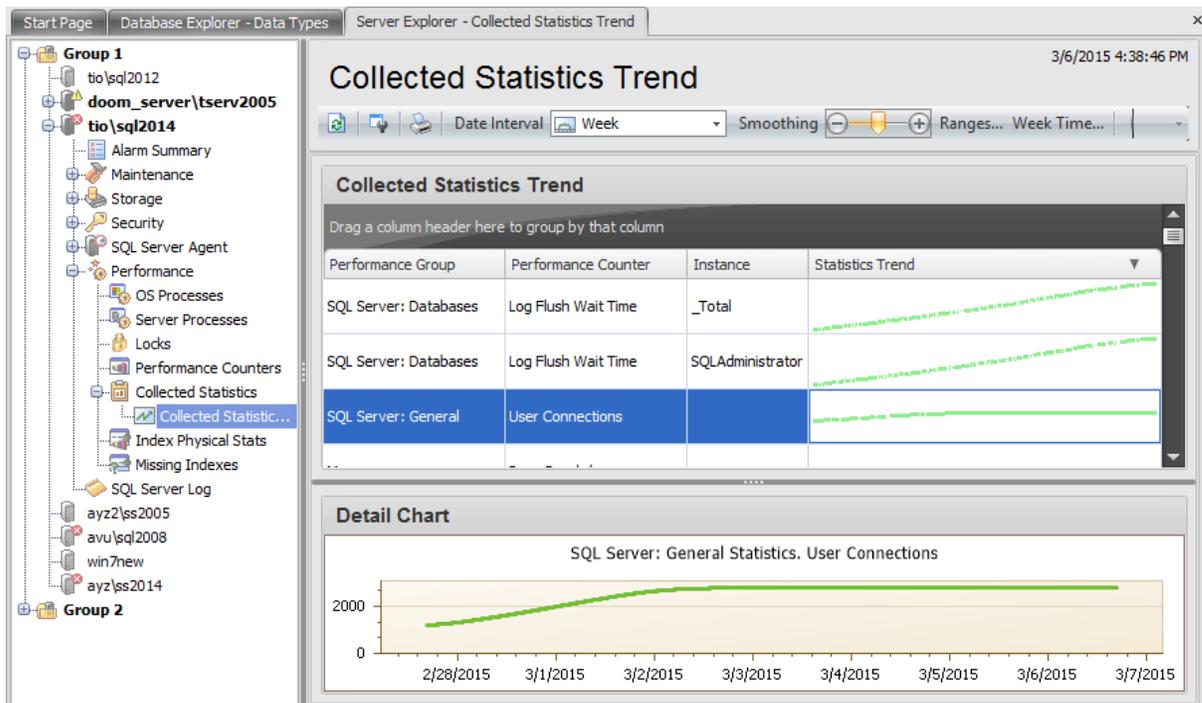
**Free** version                      **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.5.1 Collected Statistics Trend

This subnode is intended to show you diagrams which reflect the tendency of different performance counters values collected by the [Statistics Collector](#) service. This tool allows you to find counters which values have changed greatly in the defined period of time. The work area is separated into two sections. The upper one contains the list of available counters described by the following information: *Performance Group, Performance Counter, Instance, Statistics Trend*. By default, the grid is sorted descending the value of a counter change (Statistics Trend).

Select the necessary counter and double-click it or use the **Show Details** context menu item for the lower area to display a more detailed diagram of the statistics trend.

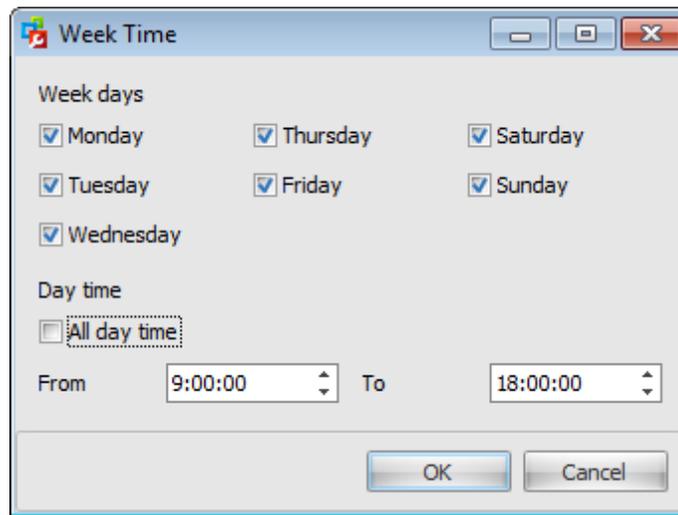


Use the toolbar items to manage the displayed statistics:

- **Date Interval** - select the period during which you want to view collected statistics: *Day, Week, Month, Quarter or Year*
- **Smoothing** - use the '+' / '-' buttons or just move the runner of this control to set the preferable graph smoothing
- **Ranges** - set the equal counter value ranges
- **Week Time** - set week day(s) and day time at which you want to watch the statistics trend
- **Expand/Collapse all rows** - use this buttons when the grouping is applied to show/hide all records

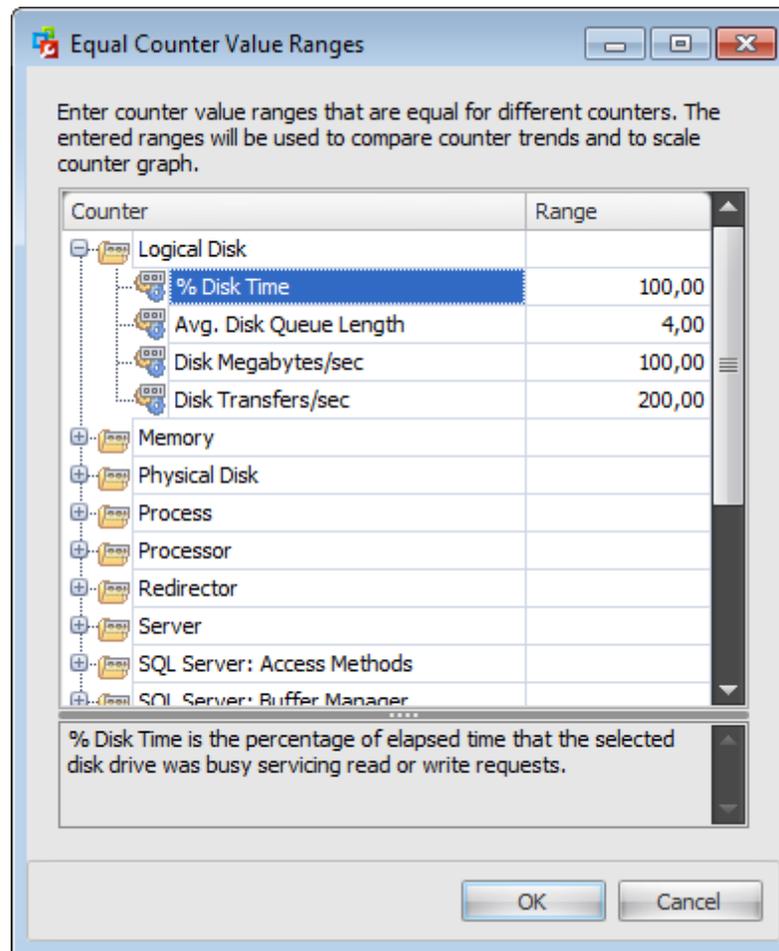
Use the **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

### Setting Week Time



Check the days at which to show the statistics from the **Week days** list. Then set the **Day time**. You can choose to display the statistics for the whole day or set the interval.

### Setting Ranges



**Collected Statistics Trend** tool displays counter values that have different units of measure, for instance, the *Avg Disk Queue Length* counter usually takes on a little value whereas the *Disk Transfers/sec* reaches the value of 100 and more. So values of different counters (and their changes) cannot be compared as they are. That's why you must set the ranges of the counter changes that will be considered equal. This is done in the **Equal Counter Value Ranges** dialog. For example, the data on the picture show that the interval of the *% Disk Time* counter change from 0 to 100 is considered equivalent to the interval of the of the *Avg Disk Queue Length* counter change from 0 to 4 and so on. It means that the counter values are scaled with a glance to the values provided in the **Equal Counter Value Ranges** dialog. This is done according to the following scheme:

1. The obtained value for every counter is divided by the value set in the *Range* column.
2. The counter with the maximum change is searched for and is set on the first position in the grid.
3. Other counter values are divided on the value of that maximum counter change.
4. All diagrams in the grid are replotted according to the scaled values, then they are smoothed.

Availability:  
Full version

Yes

**Free** version

**No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 9.6 Index Physical Stats

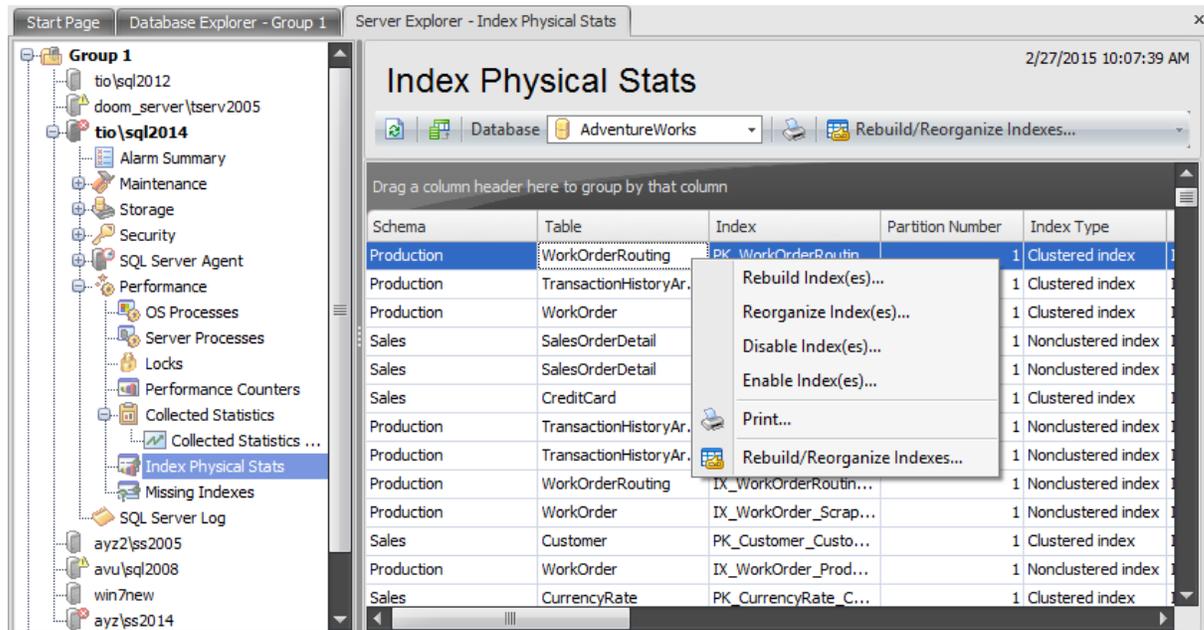
In the **Index Physical Stats** section you can find out information about the size and fragmentation for the data and indexes of the specified table or view. The context menu or buttons on the work area toolbar allows you to:

 rebuild/reorganise indexes

- rebuild indexes
- reorganise indexes
- disable indexes
- enable indexes

 create and [print](#) the report

The last operations are performed by means of the [Rebuild/Reorganize Indexes Wizard](#).



Schema	Table	Index	Partition Number	Index Type
Production	WorkOrderRouting	PK_WorkOrderRouting	1	Clustered index
Production	TransactionHistoryAr...		1	Clustered index
Production	WorkOrder		1	Clustered index
Sales	SalesOrderDetail		1	Nonclustered index
Sales	SalesOrderDetail		1	Nonclustered index
Sales	CreditCard		1	Clustered index
Production	TransactionHistoryAr...		1	Nonclustered index
Production	TransactionHistoryAr...		1	Nonclustered index
Production	WorkOrderRouting	IX_WorkOrderRoutin...	1	Nonclustered index
Production	WorkOrder	IX_WorkOrder_Scrap...	1	Nonclustered index
Sales	Customer	PK_Customer_Custo...	1	Clustered index
Production	WorkOrder	IX_WorkOrder_Prod...	1	Nonclustered index
Sales	CurrencyRate	PK_CurrencyRate_C...	1	Clustered index

To view Index Physical Stats select a database from the **Database** drop-down list which is situated on the toolbar.

The list displays indexes as a grid with the following columns: *Schema, Table, Index, Partition Number, Index Type, Allocation Unit, Fragmentation (%), Index Depth, Number of Fragments, Avg Fragment Size (Pages), Number of Pages, Avg. Used Page Space (%), Number of Records, Min. Record Size (Bytes), Max. Record Size (Bytes), Avg. Record Size (Bytes), Number of Compressed Pages*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

**Availability:**

**Full** version **Yes**

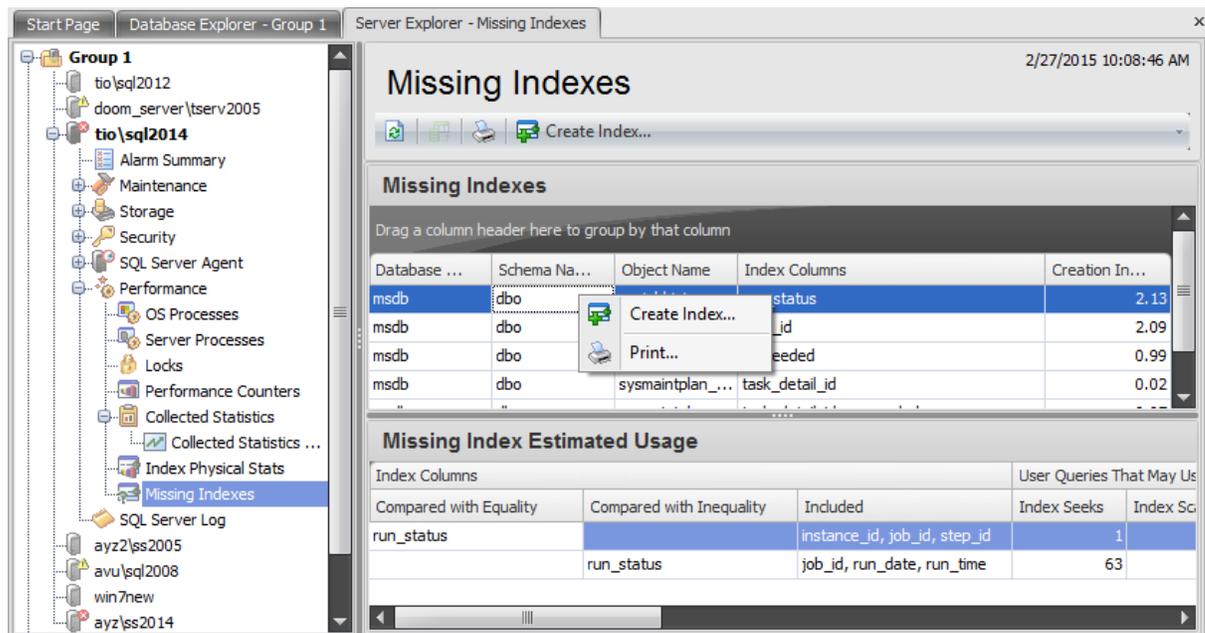
**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.



## 9.7 Missing Indexes

When the query optimizer generates a query plan, it analyzes what are the best indexes for a particular filter condition. If the best indices do not exist, the query optimizer generates a suboptimal query plan, but still stores information about these indexes. The **Missing Indexes** section enables you to access information about these indexes so you can decide whether they should be implemented.



The list displays missed indexes as a grid with the following columns: *Database Name*, *Schema Name*, *Object Name*, *Index Columns*, *Creation Index Benefit*. If more convenient, you can change the order of the columns by dragging their headers horizontally. See the [Working with grid](#) section of the SQL Administrator documentation to find out operations that can be performed with the grid.

To restore a missed index use the **Create Index...** context menu item or click the corresponding button on the toolbar. In the window appeared edit the script if needed and press OK button to create the index.

Use the **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

When you choose a particular index the lower area displays the following detailed information about the Missing Index Estimated Usage: **Index Columns:** *Compared with Equality*, *Included*, *Compared with Inequality*; **User Queries That May Use Index:** *Index Seeks*, *Index Scans*, *Last Seek*, *Last Scan*, *Avg Total Query Cost*, *Query Cost Benefit (%)*; **System Queries That May Use Indexes:** *Index Seeks*, *Index Scans*, *Last Seek*, *Last Scan*, *Avg Total Query Cost*, *Query Cost Benefit (%)*; *Total Benefit*.

Availability:

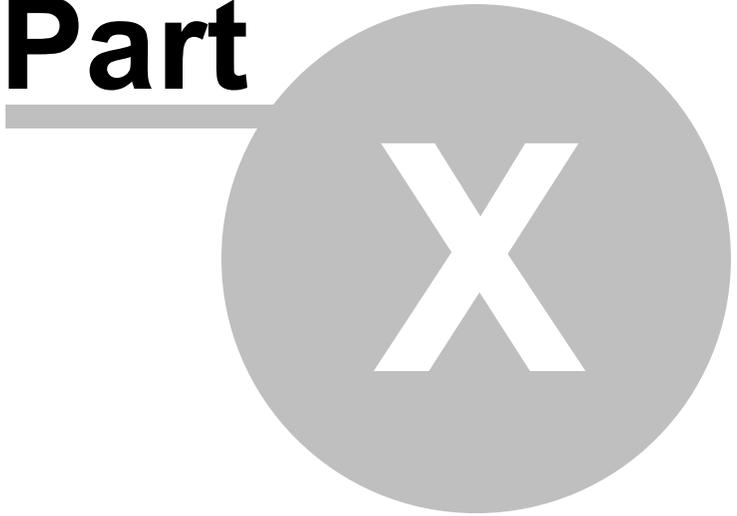
**Full** version **Yes**

**Free** version **No**

**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

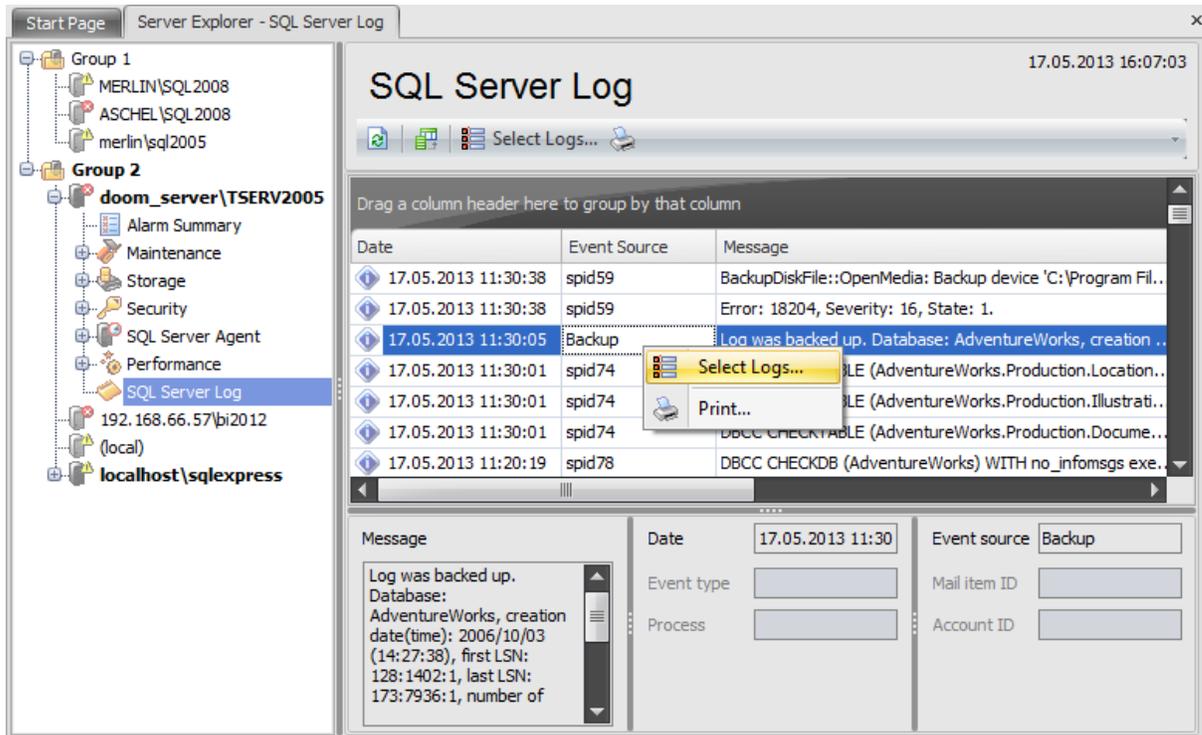


**Part**



## 10 SQL Server Log

The **SQL Server Log** section of the [Server Explorer](#) allows you to analyze SQL Server server logs in a simple way, providing the ability to sort, filter and group log statements.



All logs have two common fields: *Date* and *Message*.

The grid of the **SQL Server Log** contains the following fields: *Event Source*, *Log Type*, *Log Name*.

The **SQL Server Agent Log** has the following fields: *Error Level*, *Log Type*, *Log Name*.

The **Database Mail Log** contains the following information: *Event Type*, *Process*, *Mail Item ID*, *Account ID*.

The **Windows Log** provides information about: *Event Source*, *Category*, *Event Code*, *User*, *Event Type*, *Log Type*, *Log Name*.

When you choose a row in the grid the fields below it are filled up with the following information (depending on the log type):

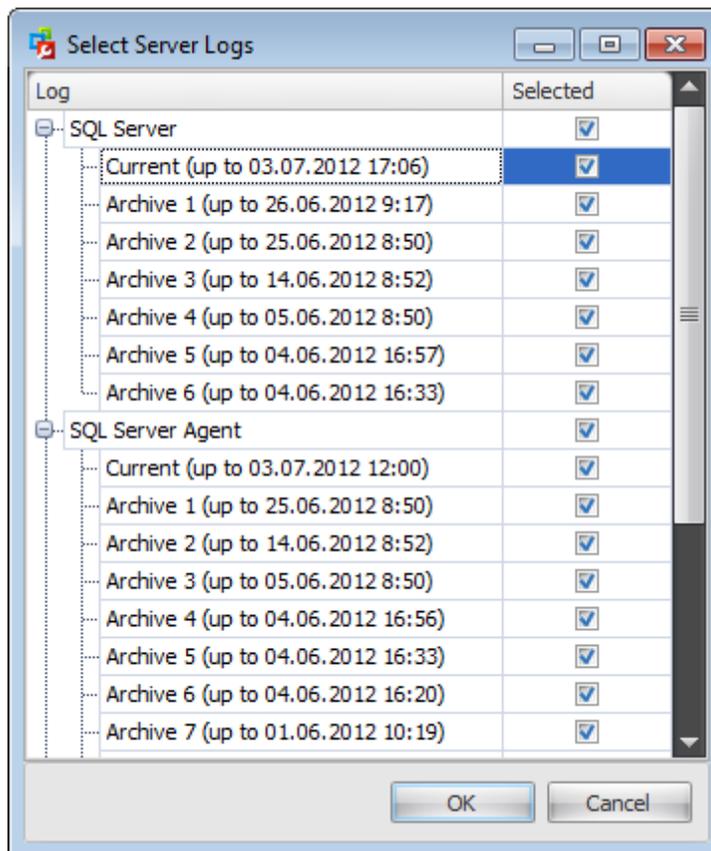
- **Message (for all logs)** - displays any messages associated with the event;
- **Date (for all logs)** - displays the date of the event;
- **Category (for Windows logs only)** - the category of the event, which is sometimes used to further describe the related action;
- **Event code (for Windows logs only)** - an identifier for the specific event;
- **Event type (for Database Mail and Windows logs)** - the type of notice inserted in the log. Possible values are errors, warnings, informational messages, success messages, and additional internal messages;
- **Process (for Database Mail logs only)** - the process id of the Database Mail external program. This typically changes each time the Database Mail external program starts;
- **Event Source (for SQL Server and Windows logs)** - displays the source feature from

which the event is created, such as the name of the service (MSSQLSERVER, for example);

- **User (for Windows logs only)** - the user account that was logged on when the event occurred;
- **Error level (for SQL Server Agent log only)** - indicates the type of problem encountered by SQL Server;
- **Mail item ID (for Database Mail only)** - identifier of the mail item in the mail queue. NULL if the message is not related to a specific e-mail item;
- **Account ID (for Database Mail only)** - the account ID of the account related to the event. NULL if the message is not related to a specific account.

Use the  **Print** context menu item or the corresponding toolbar button to create and [print](#) the report.

Use the  **Select Logs** toolbar button or the corresponding context menu item to launch the **Select Server Logs** dialog.



Use this dialog to set the logs to be displayed in the **SQL Server Logs** grid. The following logs are available:

- *SQL Server Log*

View this log to ensure that processes have completed successfully (for example, backup and restore operations, batch commands, or other scripts and processes). This can be helpful to detect any current or potential problem areas, including automatic recovery messages (particularly if an instance of SQL Server has been stopped and restarted),

kernel messages, or other server-level error messages. SQL Server retains backups of the previous six logs and gives the most recent log backup the extension *.1*, the second most recent the extension *.2*, and so on.

- *SQL Server Agent Log*

SQL Server Agent creates an error log that records warnings and errors by default. The following warnings and errors are displayed in the log:

- ✓ Warning messages that provide information about potential problems, such as "Job <job\_name> was deleted while it was running."
- ✓ Error messages that usually require intervention by a system administrator, such as "Unable to start mail session."

SQL Server maintains up to nine SQL Server Agent error logs. Each archived log has an extension that indicates the relative age of the log. For example, an extension of *.1* indicates the newest archived error log and an extension of *.9* indicates the oldest archived error log.

- *Database Mail Log*

Database Mail stores the log information in the *msdb* database. Information about Database Mail e-mail content, status of e-mails, and any messages received, such as errors are logged by Database Mail and can be used for troubleshooting and auditing purposes.

- *Windows Log*

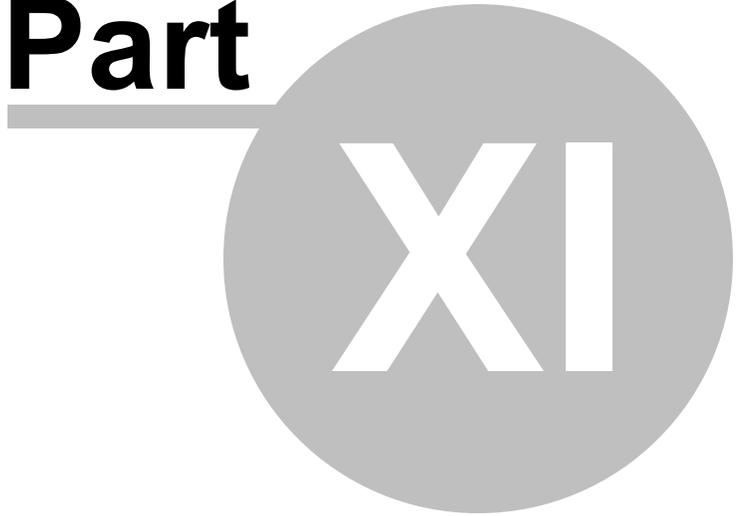
An important source of information for troubleshooting SQL Server errors, the Windows Event log contains four useful logs. The application log records events in SQL Server and SQL Server Agent and can be used by SQL Server Integration Services (SSIS) packages. The security log records authentication information, and the system log records service startup and shutdown information. The PowerShell Script log contains events logged by Windows PowerShell.

---

**See also:**

[Getting Started](#)  
[Servers](#)  
[Database Maintenance](#)  
[Storage](#)  
[Security](#)  
[SQL Server Agent](#)  
[Performance](#)  
[Query Editor](#)  
[Options](#)

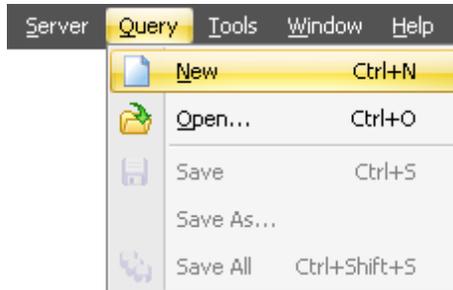
**Part**



## 11 Query Editor

**Query Editor** is a tool used for creating and executing queries. The tool allows you to create and edit the SQL text of a query, execute queries, view and export the results of query execution.

To open the Query Editor, select the **Query | New / Query | Open...** [main menu](#) items or use the *Ctrl+N / Ctrl+O* shortcuts for the same purpose.



- [Using toolbar](#)
- [Working with the SQL Editor area](#)
- [Executing queries](#)
- [Viewing messages](#)

---

**See also:**

[Getting Started](#)  
[Servers](#)  
[Database Maintenance](#)  
[Storage](#)  
[Security](#)  
[SQL Server Agent](#)  
[Performance](#)  
[SQL Server Log](#)  
[Options](#)

## 11.1 Using toolbar

The toolbar of the **Query Editor** tool provides quick access to query management tools.

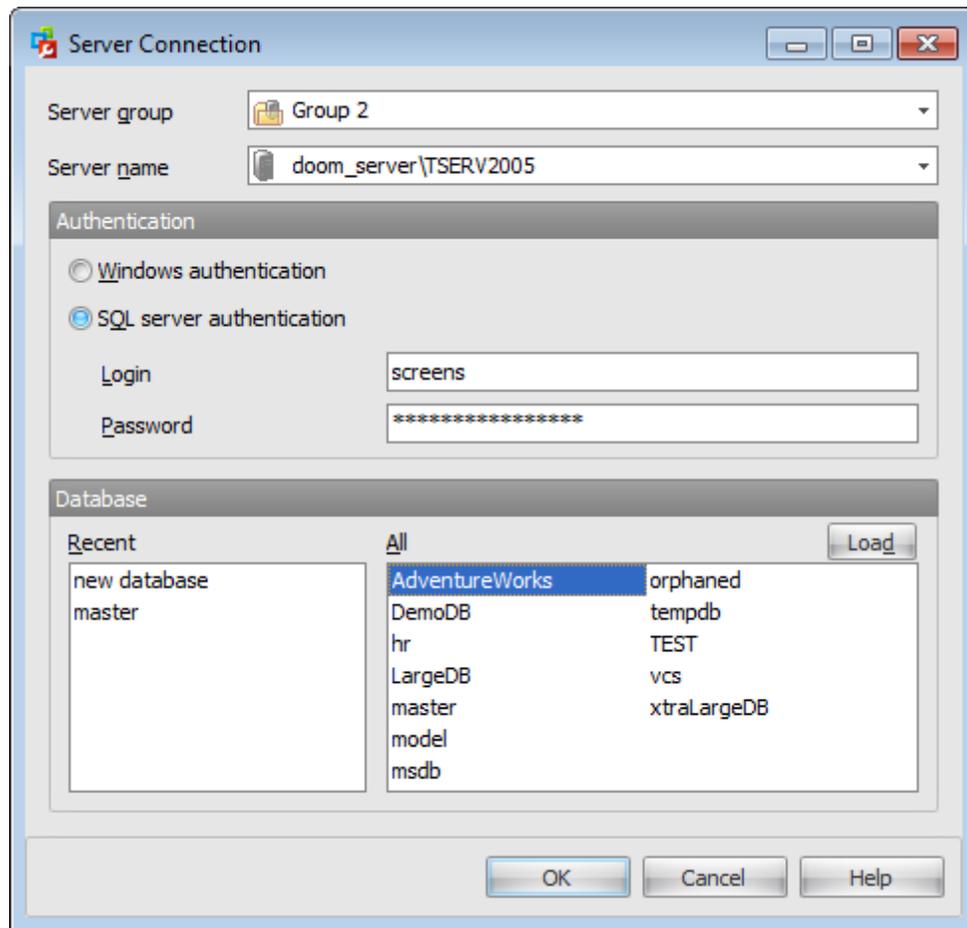


-  connect to server
-  disconnect from server
-  .. select a database for the query
-  create new query
-  load a query from an \*.sql file
-  save the query to an \*.sql file
-  execute the current query
-  stop the current query execution
-  find text

To execute a query you need to choose a connected database.

When you click the  **Connect to server button** the Server Connection dialog appears.

**Note:** This dialog appears every time you create a new query or open an existing one as well.



### Server Group

Select the group that the necessary server belongs to.

### Server Name

Select the server to connect from the drop-down list of all registered servers.

### Authentication

Specify the type of Microsoft® SQL Server™ authentication to be used for the connection:  *Windows authentication* or  *SQL Server authentication*.

#### Windows Authentication

SQL Administrator identifies user by his domain name and password which are defined automatically according to his Windows domain user account.

#### SQL Server authentication

SQL Administrator identifies user by his server login and password.

If  *Server authentication* is selected as the *authentication type*, you should provide *authorization* settings: **Login** and **Password**. Analyzing them a system determines information and therefore object access permissions.

### Database

Select the database for query execution.

**Recent**

This list contains the set of databases that have been recently used for query execution.

**All**

In this area all databases located on the selected server are listed. If a connection to the selected server is not established in the SQL Administrator then click the **Load** button to get the list of databases on this server.

---

**See also:**

[Using toolbar](#)

[Working with the SQL Editor area](#)

[Executing queries](#)

[Viewing messages](#)

## 11.2 Working with the Query Editor area

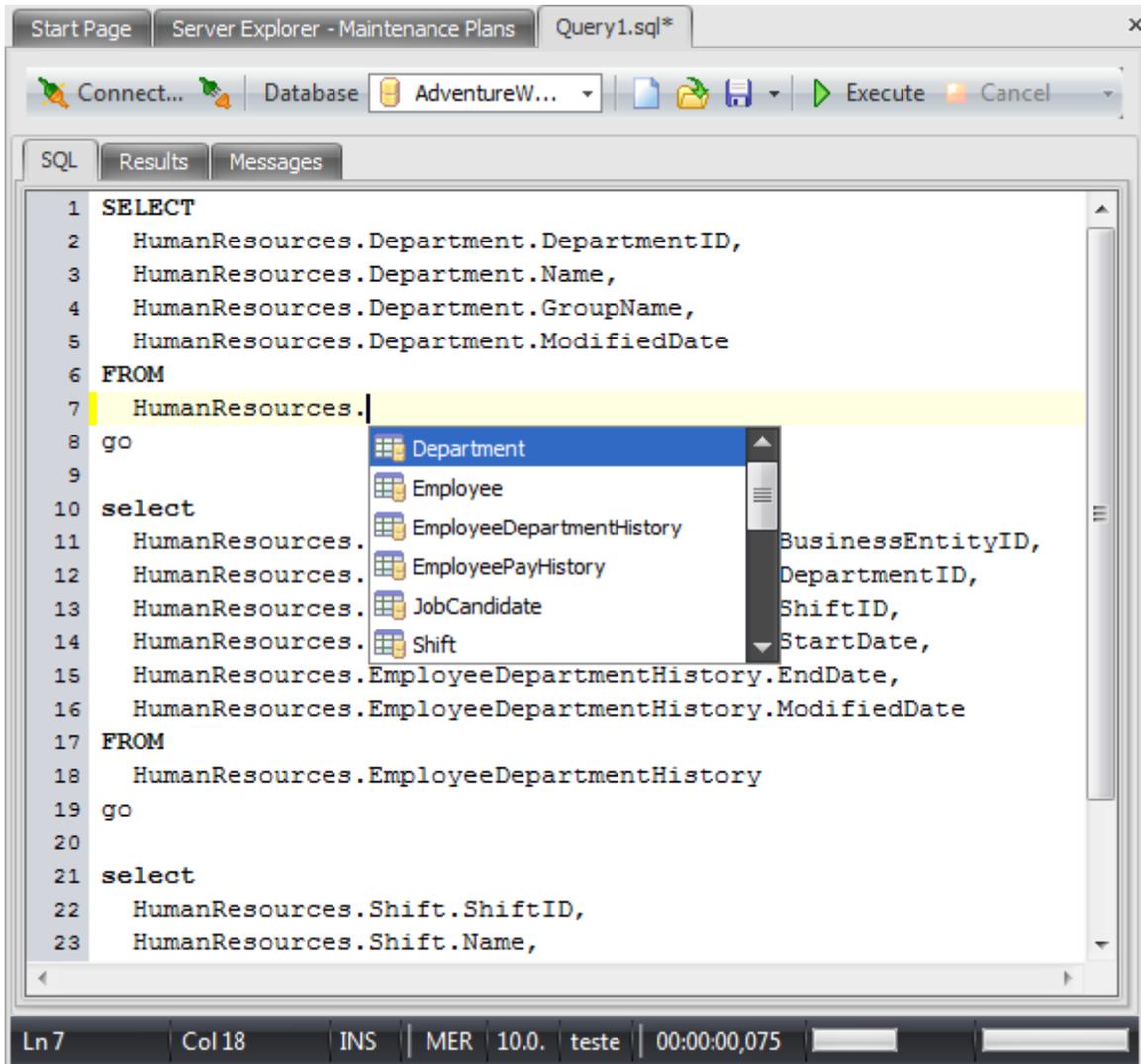
The **Query Editor** work area is provided for editing query statement text. For your convenience a number of features for efficient SQL editing is implemented:

- the syntax highlight;
- ability to display line numbers;
- customizable margins and gutters;

and more.

If necessary, you can enable/disable or customize some of **Query Editor** features using the [Query Tool Options](#) dialog.

The example of code completion is illustrated in the picture below. You can set the delay within the [Quick code](#) section of the [SQL Editor Options](#) dialog or activate the completion list manually by pressing the *Ctrl+Space* [shortcut](#).



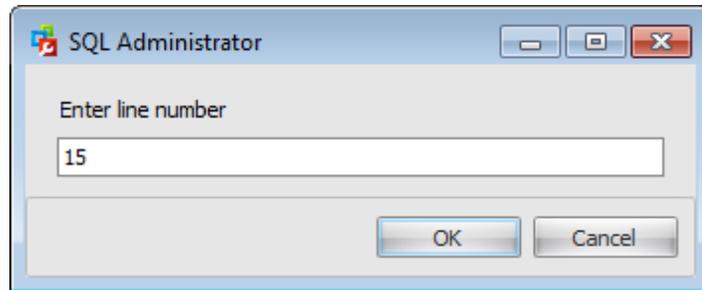
The status bar situated at the bottom of the Query window displays the following

information:

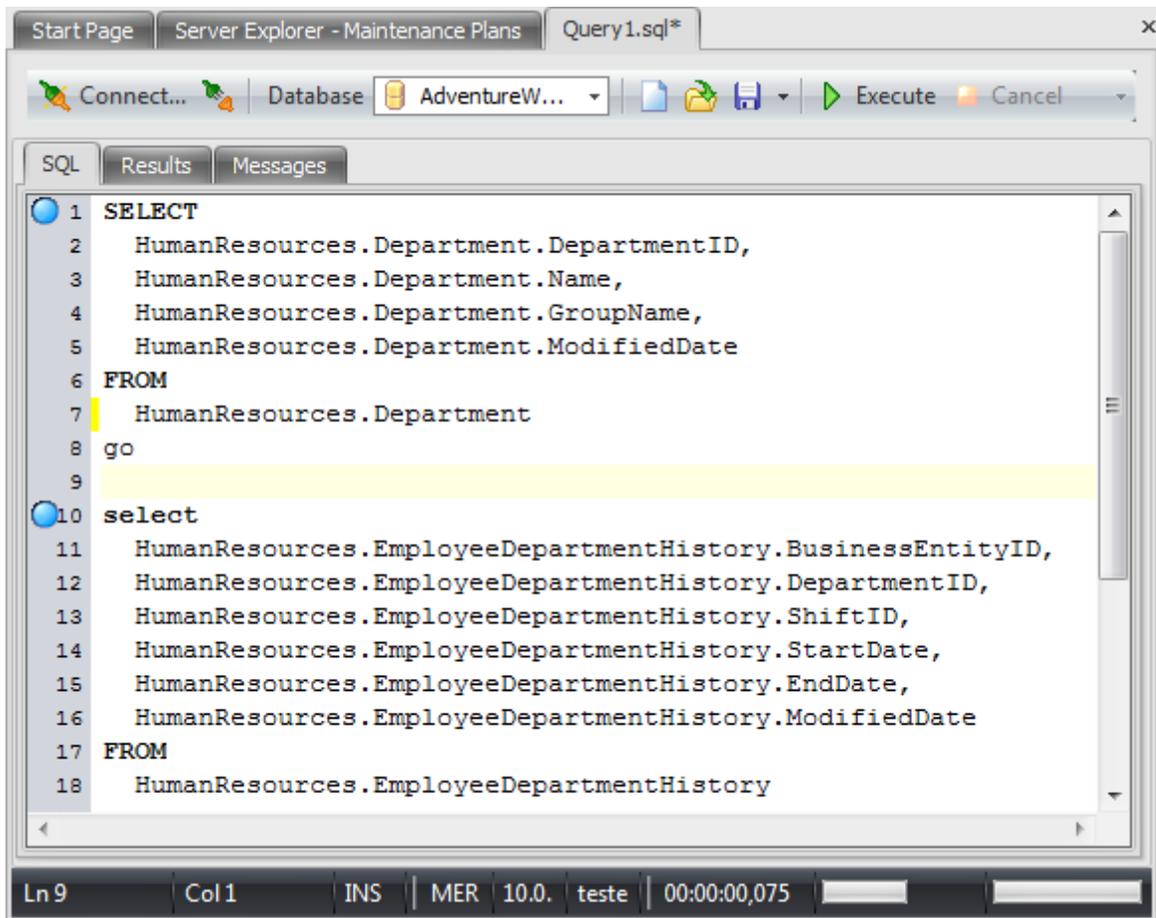
- *Ln, Col* - the current position (line and column) of a cursor;
- server on which the query is executed;
- server version;
- current server login;
- query execution time;
- progress bars: the first one indicates the execution of the query, the second shows the percent of script execution.

Here are some features available via the **Query Editor** context menu:

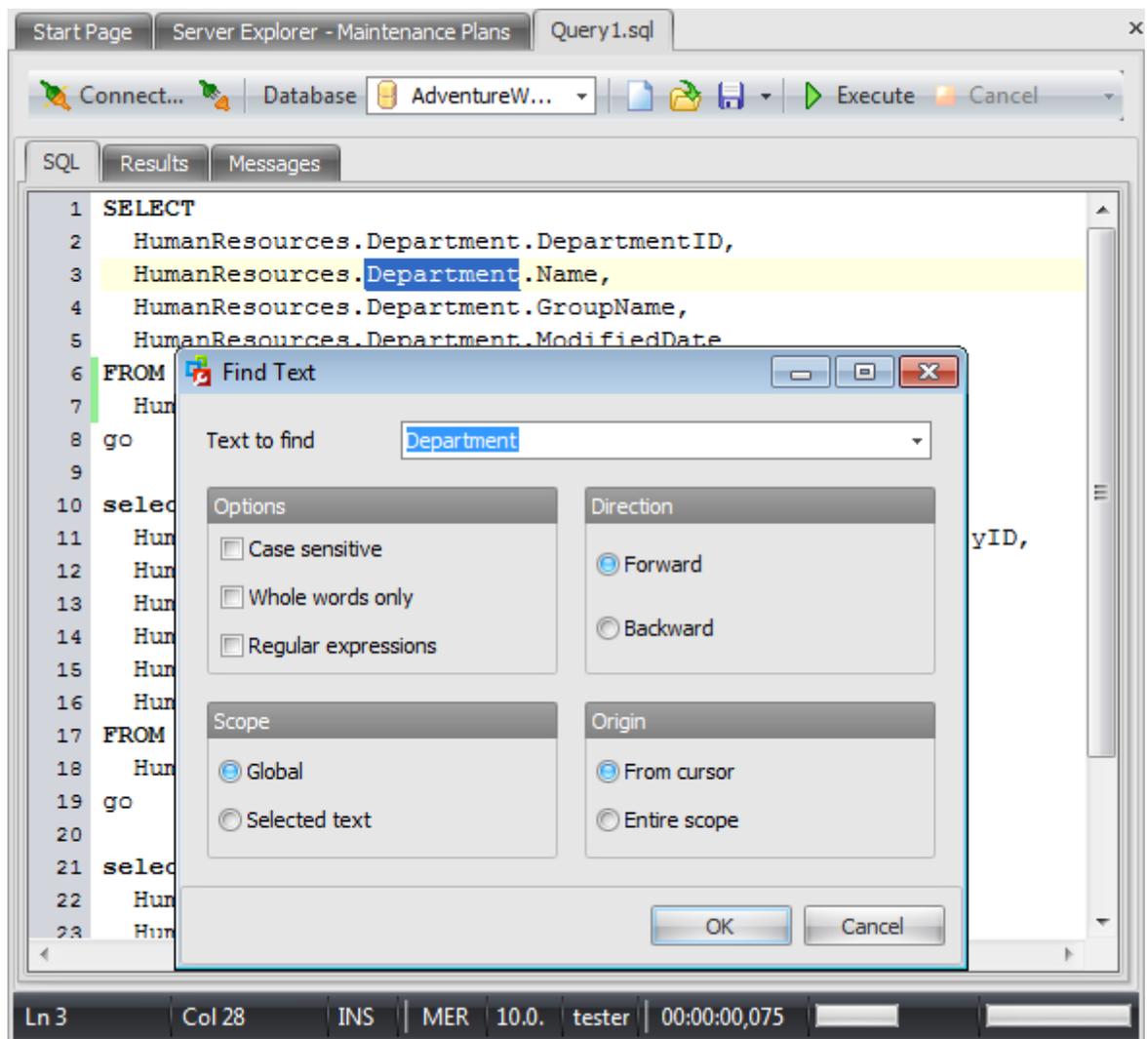
- Move the cursor to a particular line.  
Use the **Go to Line...** context menu item to evoke the dialog where you can set a number of the line you want to move the cursor to.



- Toggle bookmarks allowing you to navigate through the query text and jump to a line with a particular number.  
Bookmarks are blue marks on the margin. After you have toggled several bookmarks you can go to next/previous bookmark using the *F2 / Shift+F2* button.



- Perform search and replace operations using the [Find&Replace Text dialog](#).



The complete list of context menu items can be found [here](#).

**See also:**

[Using toolbar](#)

[Executing queries](#)

[Viewing messages](#)

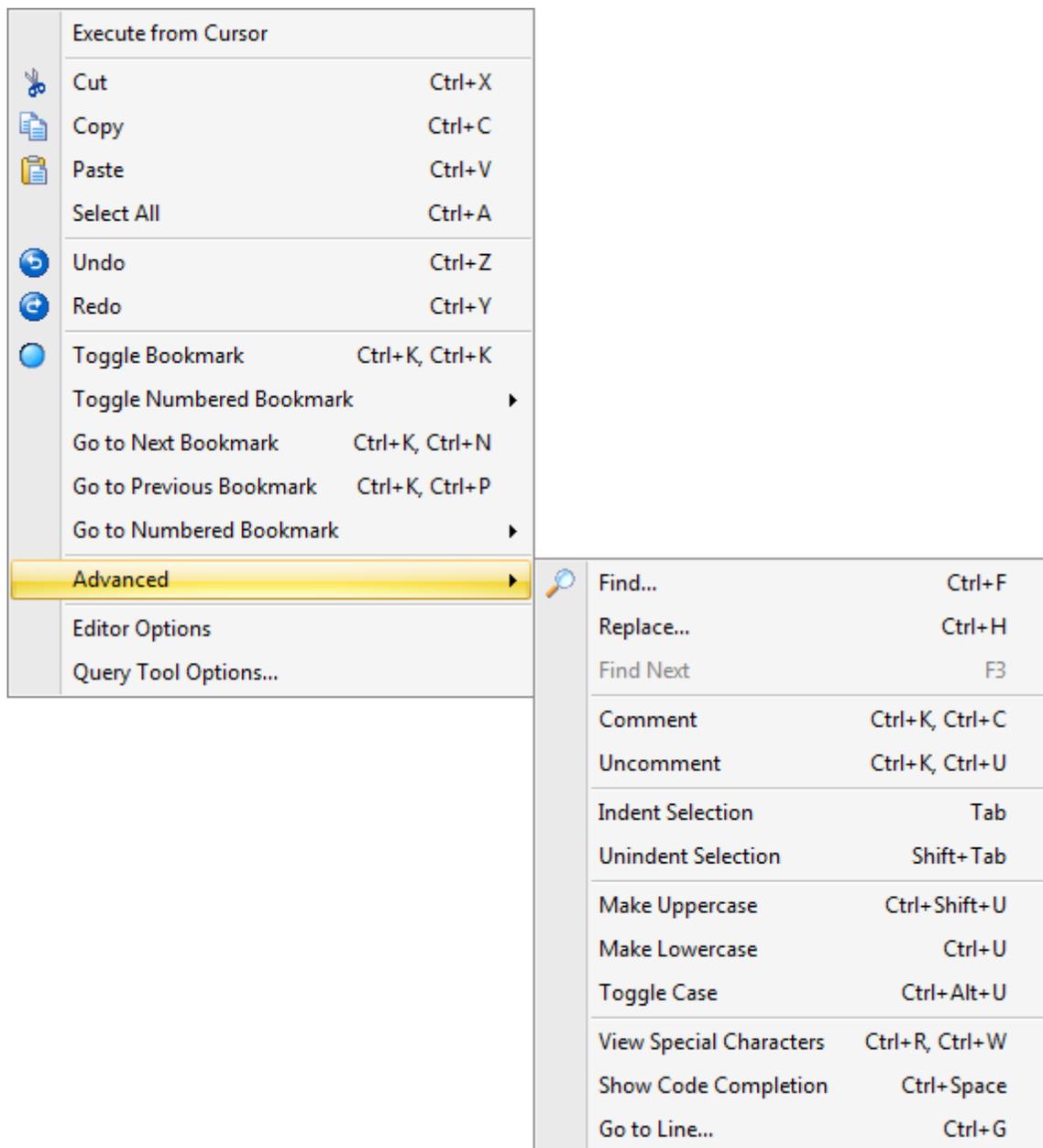
## 11.3 Using the context menu

The **context menu** of SQL Editor area contains execution commands, most of the standard text-processing functions (*Cut, Copy, Paste, Select All*) and functions for working with the query as a whole, e.g. you can *move the cursor to a particular line, change the case of selected text, view the query properties* or *print the text of the query*. Each of these operations can be also performed with the corresponding hot keys used.

Implementation of the [Find Text](#) / [Replace Text](#) dialogs contributes to more efficient work with the SQL code.

Find the complete list of **SQL Editor** context menu items below. The context menu allows you to:

- execute the query/selected text/text under cursor;
- perform editing operations: *Undo/Redo, Cut, Copy, Paste, Select All*;
- toggle bookmarks allowing you to navigate through the query text and jump to a line with a particular number;
- perform [search](#) and [replace](#) operations;
- comment/uncomment the selected line;
- indent/unindent code lines;
- change case of the selected text;
- view special characters;
- show code completion;
- open the [Editor Options](#) dialog;
- open [Query Tool Options](#) dialog.

**See also:**[Using toolbar](#)[Working with the SQL Editor area](#)[Executing queries](#)[Viewing messages](#)

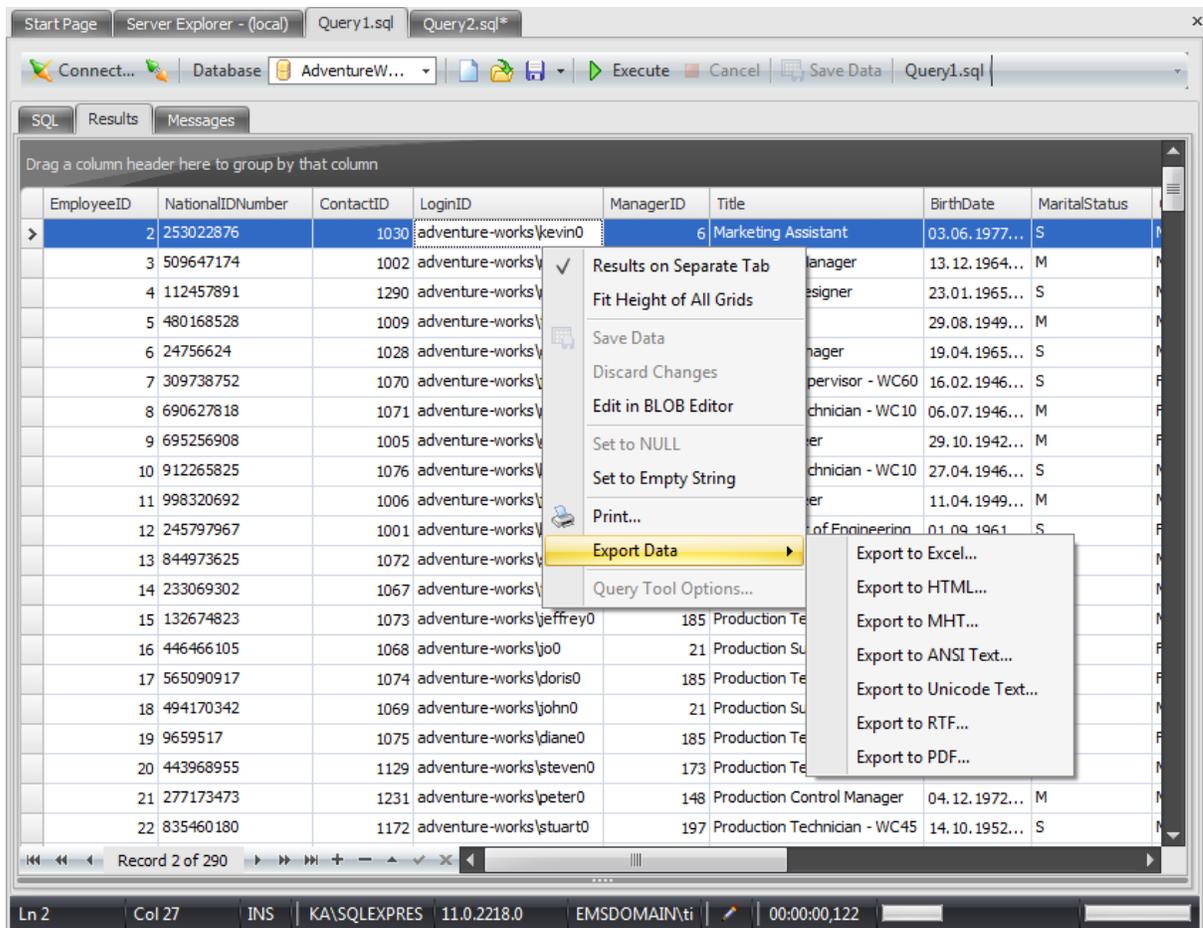
## 11.4 Executing queries

When all the query parameters are set, you can immediately **execute the query** in **SQL Editor**.

To execute a query, click the **Execute** item of the [toolbar](#). You can also use the [context menu](#) item to execute the query from the cursor

If the SQL syntax is correct, the query is executed and, in case the query statement is supposed to return data (e.g. as SELECT statement), the returned dataset appears within the **Results** tab. The position of the tab can be set when configuring the [Query tool options](#).

If SQL syntax of the query contains any errors, the query execution is stopped and the corresponding error message appears within the **Messages** tab.



Data returned by a query are displayed as a grid. The **context menu** of the grid allows you to

- set whether the query results should be displayed on the separate tab or under the editor area;
- fit height of all grids;
- save changed data;

- discard changes;
- edit field in [BLOB Editor](#) (for text and BLOB fields);
- set field value to NULL (if allowed);
- make the selected string empty;
- [print](#) query results;
- export data to some popular formats;
- launch the [Query Tool Options](#) dialog.

If the table has the primary key then the results of the query execution can be edited. The possibility of editing queries is indicated by the  icon. To edit a value just click the corresponding cell.

The toolbar located in the left bottom corner allows you to:

- go to the first record of the dataset;
- go to the previous page;
- go to the previous record;
- go to the next record;
- go to the next page;
- go to the last record of the dataset;
- insert a new record (if editing is enabled);
- delete the selected record (if editing is enabled);
- edit the selected record (if editing is enabled);
- post edit (in *Table Editor* only);
- cancel edit (in *Table Editor* only).

To learn more about managing the grid, see the [Working with grid](#) section.

---

**See also:**

[Using toolbar](#)

[Working with the SQL Editor area](#)

[Viewing messages](#)

## 11.5 Using BLOB Editor

SQL Administrator provides BLOB Editor to view and edit BLOB (Binary Large Object) fields content.

The BLOB Editor can be invoked by double-clicking the cell of the corresponding BLOB field. You can use the **Edit in BLOB Editor** context menu item as well.

Fields of the following types can be edited in the BLOB Editor:

- *binary*;
- *char*;
- *image*;
- *nchar*;
- *ntext*;
- *nvarchar*;
- *sysname*;
- *text*;
- *varbinary*;
- *varchar*;
- *xml*.

The BLOB Editor consists of four tabs:

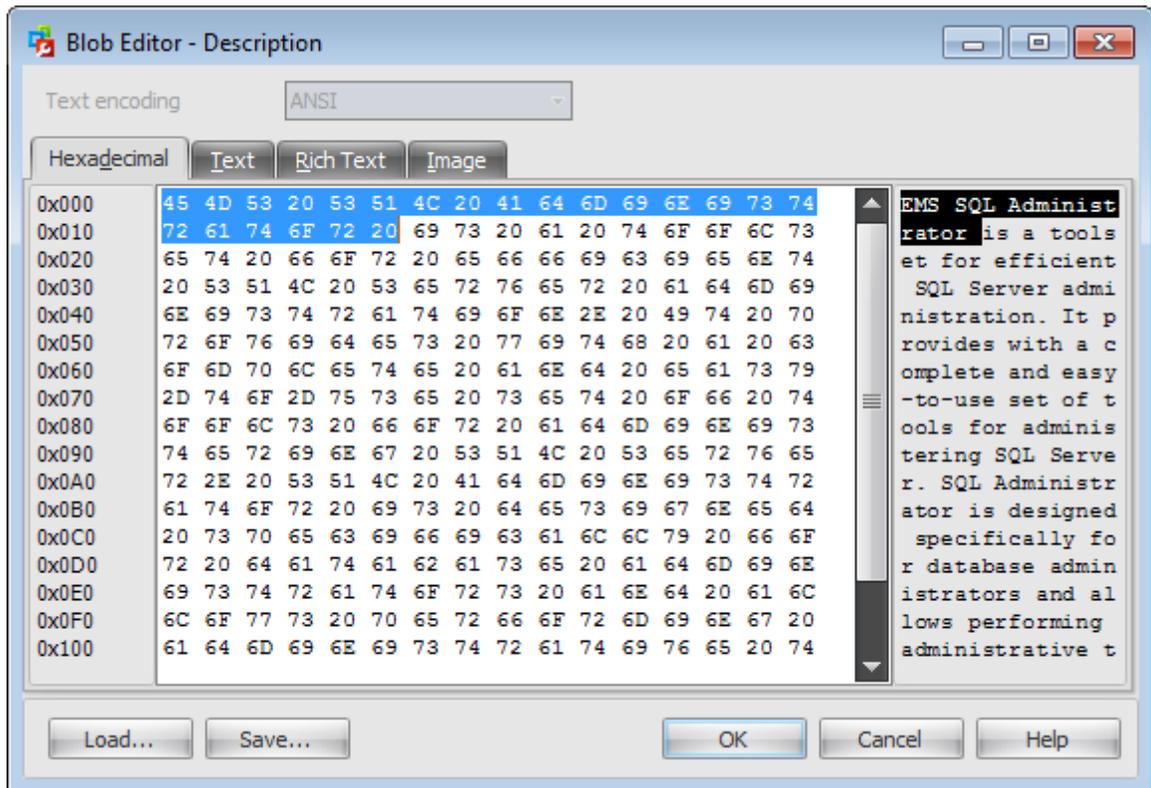
- [Hexadecimal](#)
- [Text](#)
- [Rich Text](#)
- [Image](#)

**Note:** For string data types only [Text](#) and [Rich Text](#) tabs are available.

### 11.5.1 Editing as Hexadecimal

The **Hexadecimal** tab allows you to view/edit the BLOB data as hexadecimal.

The working area of the Hexadecimal tab consists of two blocks: the editable hexadecimal presentation of data and its text analogue.

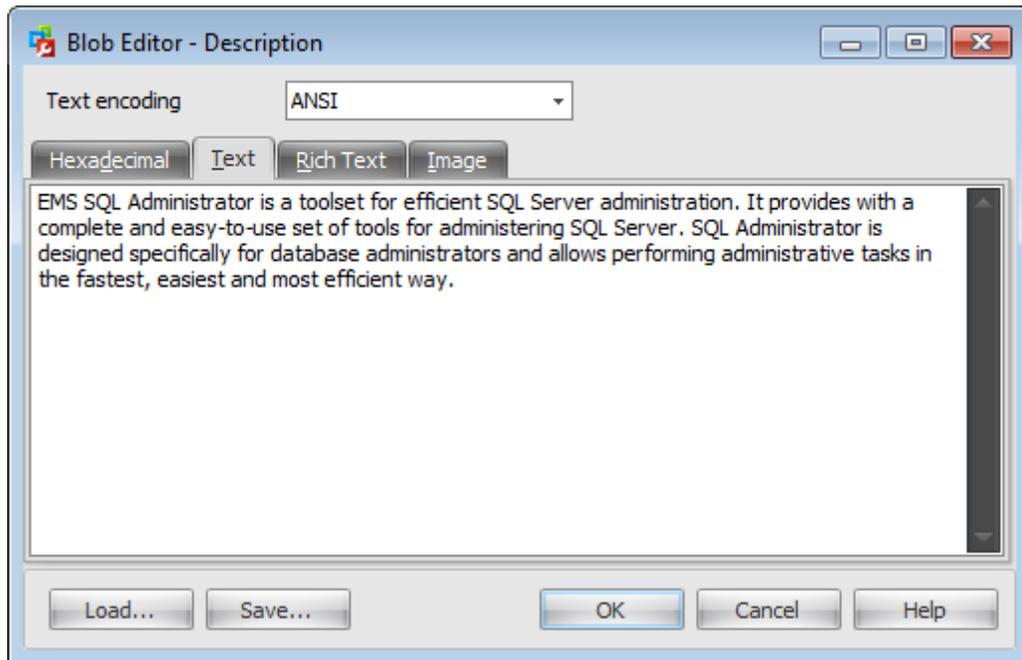


Using the the **Load...** and **Save...** buttons of the BLOB Editor you can save the hexadecimal data to a file, or load data from a file.

## 11.5.2 Editing as Text

The **Text** tab allows you to view/edit the BLOB data as plain text.

You can use the *Cut*, *Copy*, *Paste*, *Select All*, *Undo*, *Delete* context menu items for editing the text efficiently.



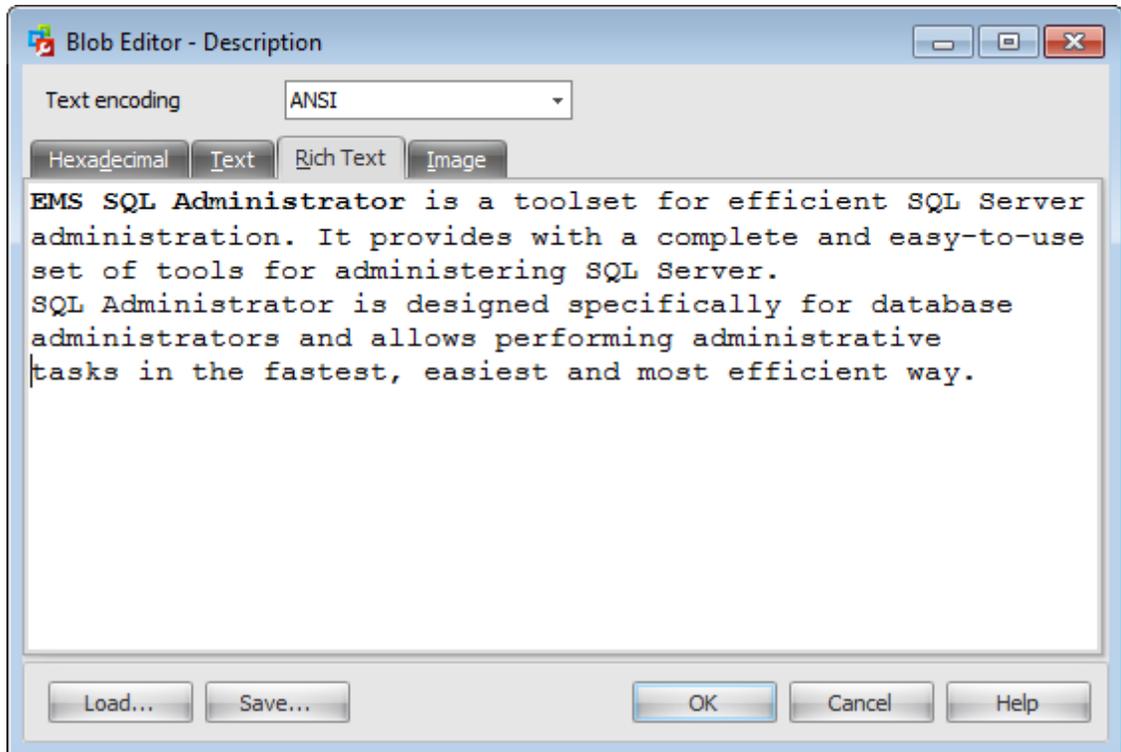
### **Text encoding**

Use the drop-down list to select the encoding for the text in the field. Possible values are: *ANSI*, *Unicode*, *Unicode Big Endian* and *UTF-8*.

Use the **Save...** and the **Load...** buttons to save the text to a \*.txt file, or load text from a file.

### 11.5.3 Editing as Rich Text

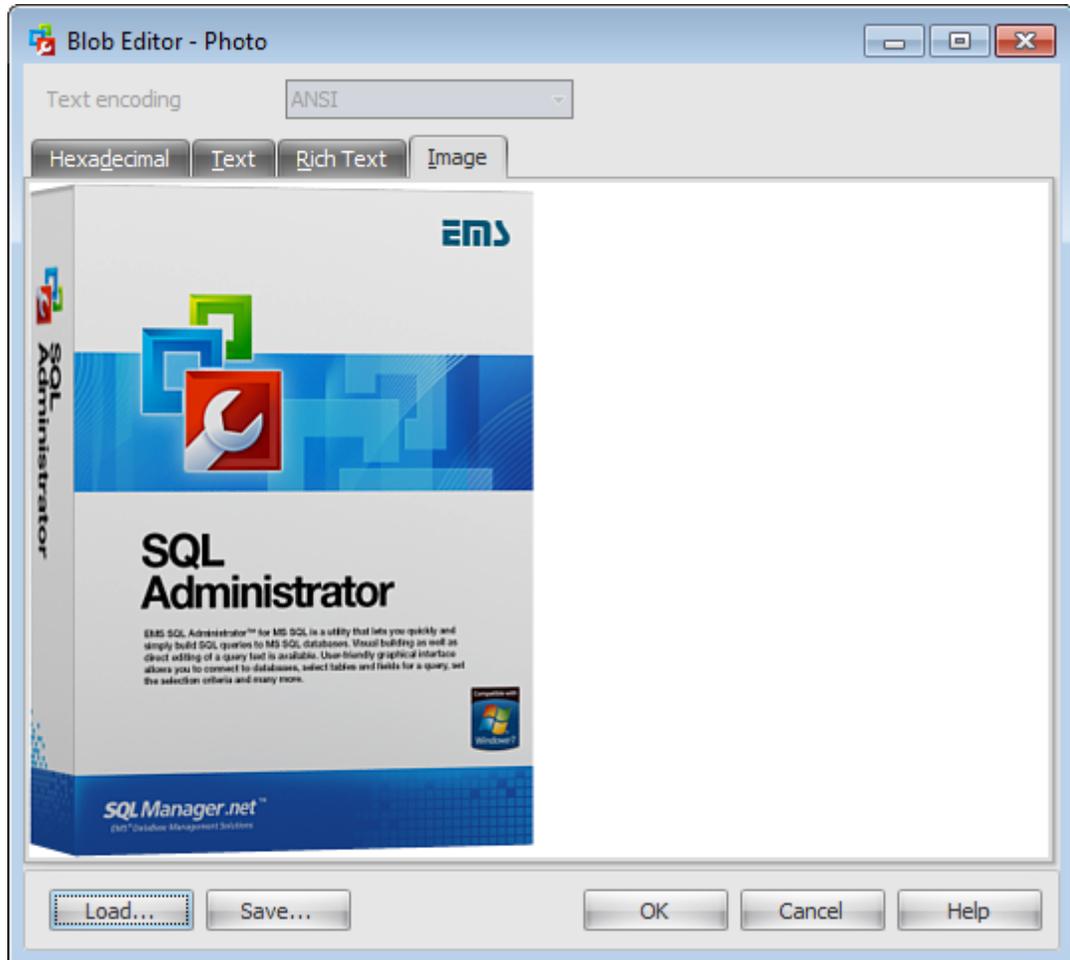
The **Rich Text** tab allows you to view/edit the BLOB data in Rich Text format (RTF). The text is presented in the formatted view.



Use the **Save...** and the **Load...** buttons to save the Rich Text to a \*.rtf file, or load text from a file.

### 11.5.4 Editing as Image

The **Image** tab allows you to view the BLOB data as image.

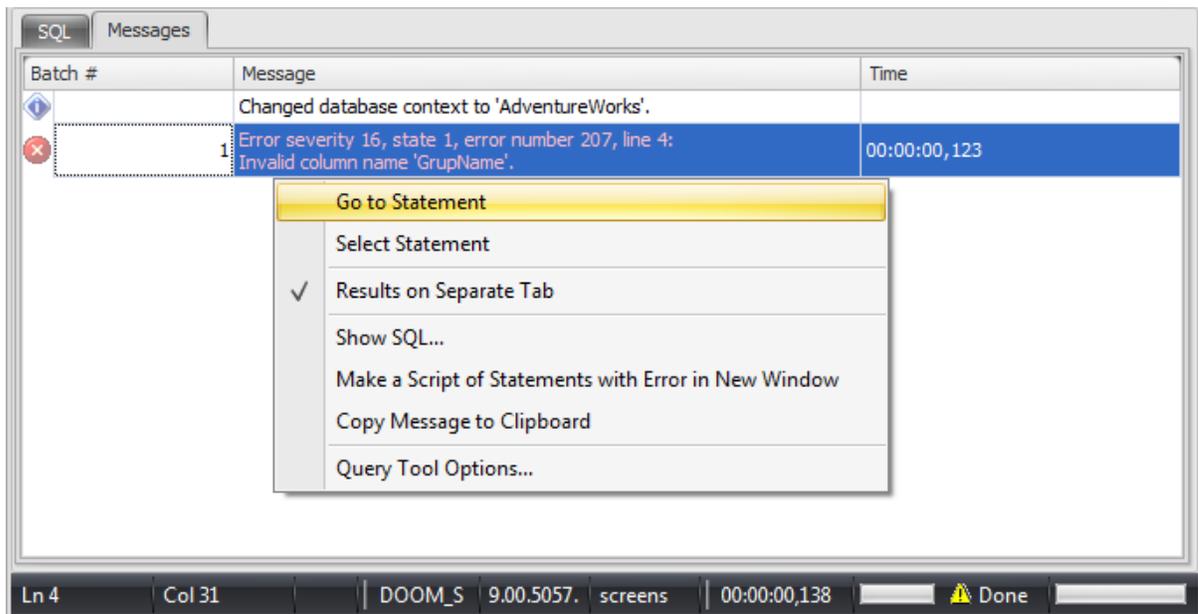


Use the **Save...** and the **Load...** buttons to save the image to a \*.png, \*.wmf, \*.ico, \*.png, \*.gif, \*.ico, \*.exif, \*.tiff, \*.emf or \*.jpg file, or load an image from a file.

## 11.6 Viewing messages

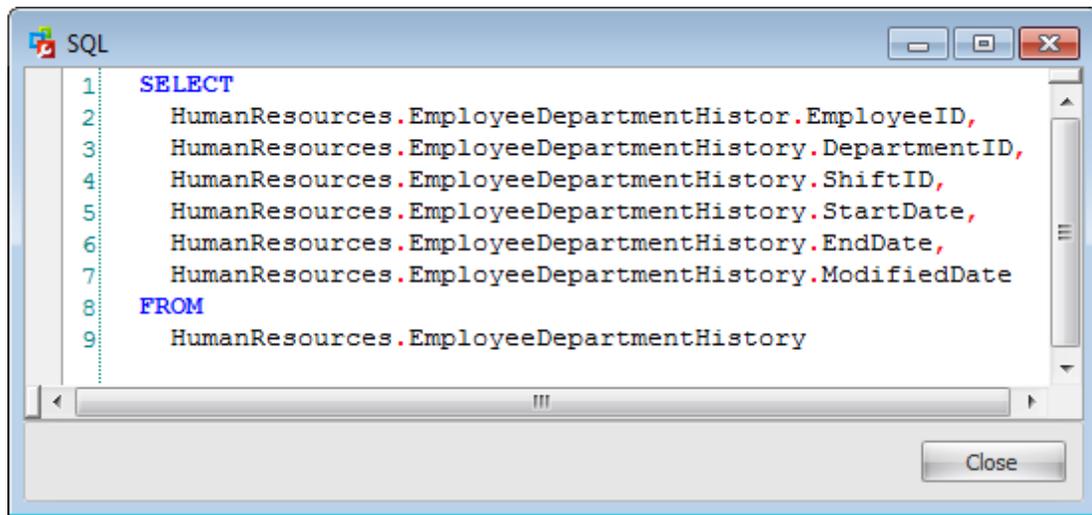
The **Messages** tab allows you to view messages appearing on a query execution. Using this tab you can view log entries containing the following details:

- date and time of the query execution;
- text of the query;
- number of rows fetched and fetch time, or the text of the error (if any).



The **context menu** of the messages area allows you to:

- *Go to Statement* - switch to the SQL tab and set the cursor at the beginning of the SQL statement with this error;
- *Select Statement* - switch to the SQL tab and select the statement with this error;
- *Results on Separate Tab* - set whether the query results should be displayed on the separate tab or under the editor area;
- *Show SQL...* - open SQL statement in the separate window;



- **Make a Script of Statements with Error in New Window** - copy the text of the script with this error to a new query and switch to it;
- Copy Message to Clipboard;
- launch the [Query Tool Options](#) dialog.

---

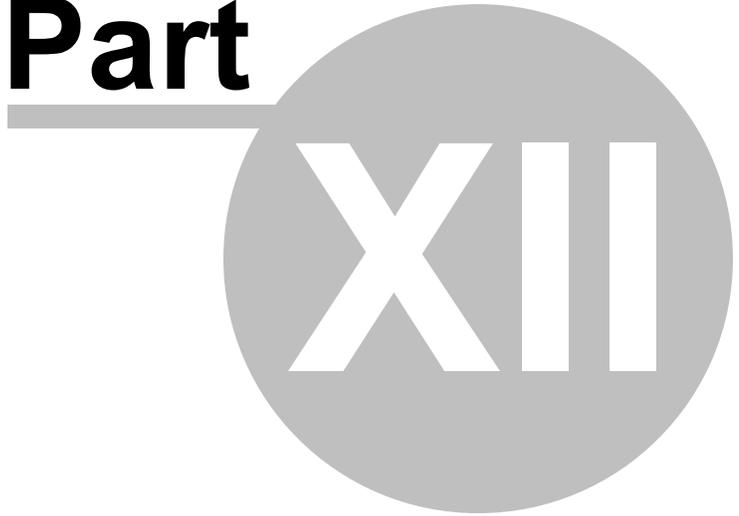
**See also:**

[Using toolbar](#)

[Working with the SQL Editor area](#)

[Executing queries](#)

**Part**

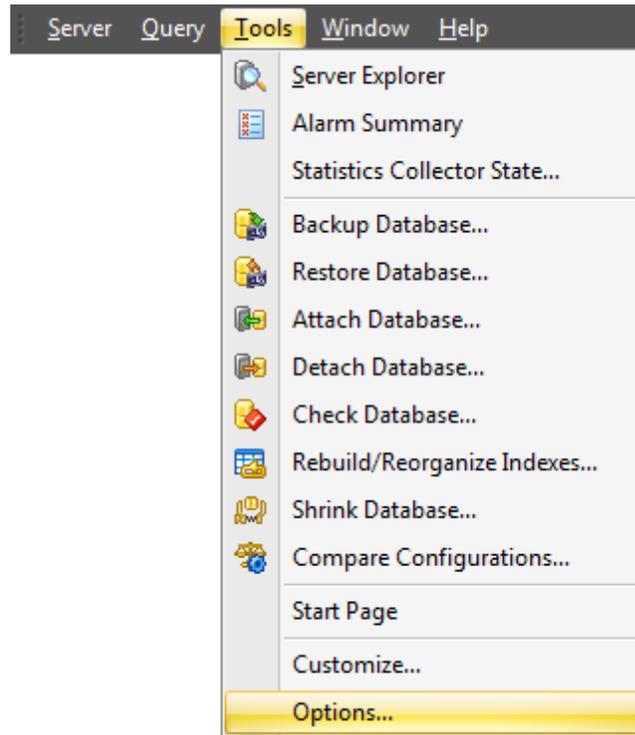


**XII**

## 12 Options

**Options** allow you to customize general options of the SQL Administrator.

To open the **Options** window, select the **Tools | Options...** [main menu](#) item.



The following list contains the six groups of **options** available in SQL Administrator.

- [Environment](#)
- [Server Explorer](#)
- [Statistics Collector](#)
- [Query tool](#)
- [SQL Editor](#)
- [Appearance](#)

---

**See also:**

[Getting Started](#)

[Servers](#)

[Database Maintenance](#)

[Storage](#)

[Security](#)

[SQL Server Agent](#)

[Performance](#)

[SQL Server Log](#)

[Query Editor](#)

## 12.1 Environment

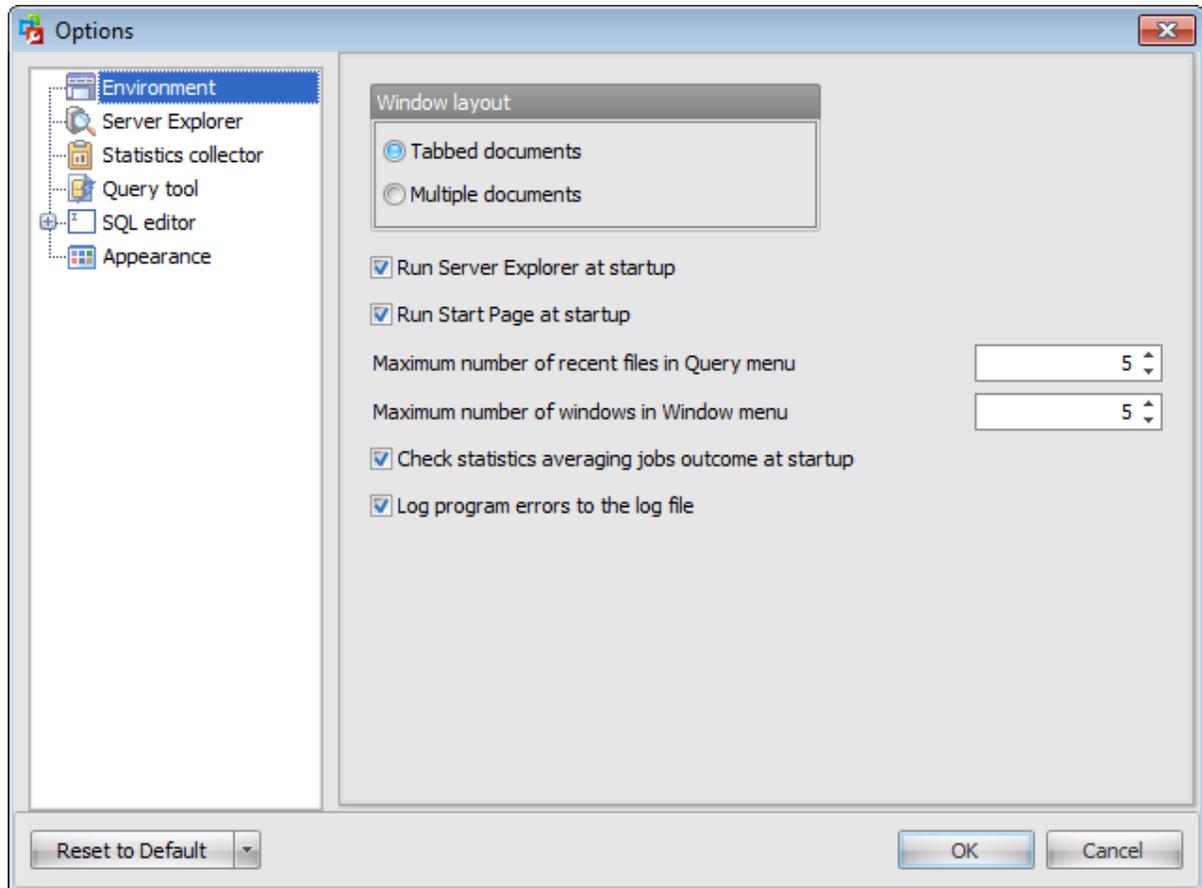
### Window layout

#### **Tabbed documents**

If you select this layout then work areas will be displayed in separate tabs of the main window.

#### **Multiple documents**

If you select this layout then all work areas are displayed in separate windows.



#### **Run Server Explorer at startup**

If you check this option then the [Server Explorer](#) will be displayed when the program starts.

#### **Run Start Page at startup**

If you check this option then the [Start Page](#) will be displayed on the program start.

#### **Maximum number of recent files in Query menu**

Set the maximum number of recent query files to be displayed in the [Query menu](#).

#### **Maximum number of windows in Window menu**

Set the maximum number of opened windows to be displayed in the Window menu

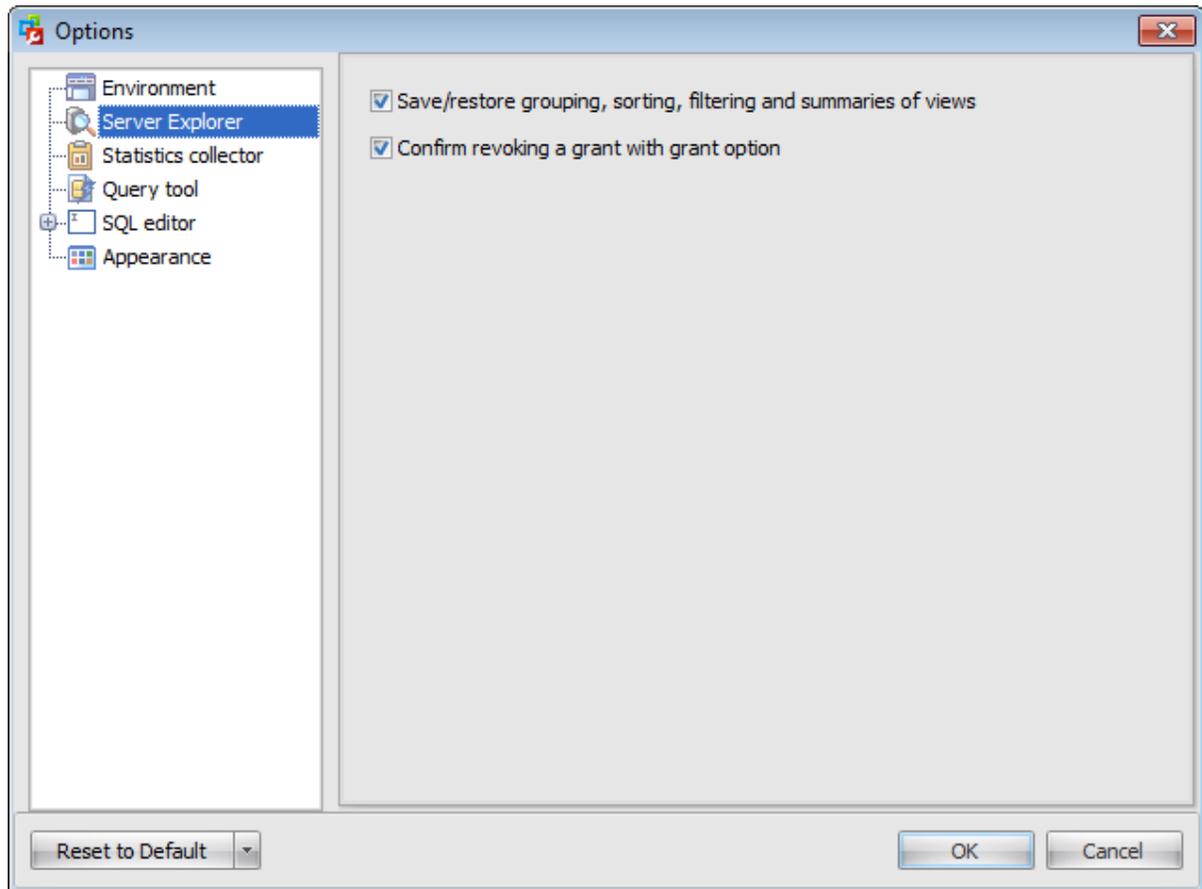
**Check statistics averaging jobs outcome at startup**

If you check this option the pop-up dialog will inform you about failed jobs at startup.

**Log program errors to the log file**

Check this option to enable logging occurred errors to the *SQLAdmin.log* file. This file is stored in the program folder.

## 12.2 Server Explorer



**Save/restore grouping, sorting, filtering and summaries of views**

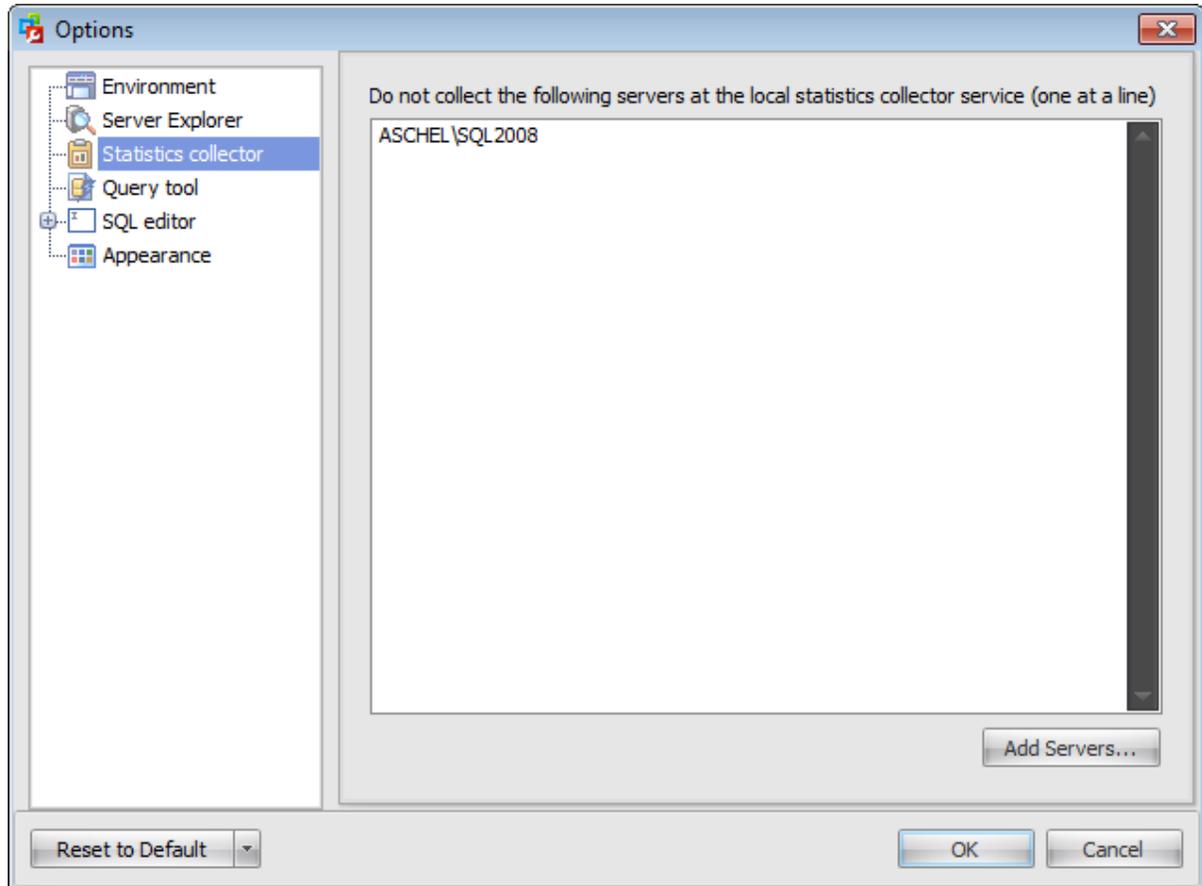
If this option is checked then all grouping, sorting, filtering conditions applied to grids will remain the same on the next connection.

**Confirm revoke a grant with grant option**

If you check this option then the SQL Administrator will request a confirmation on revoking saved granted permissions with grant options.

## 12.3 Statistics Collector

This section allows you to define servers which you do not want the statistics to be collected for.



You can type in servers names manually or click the **Add Servers...** button and choose among the servers in the list appeared.

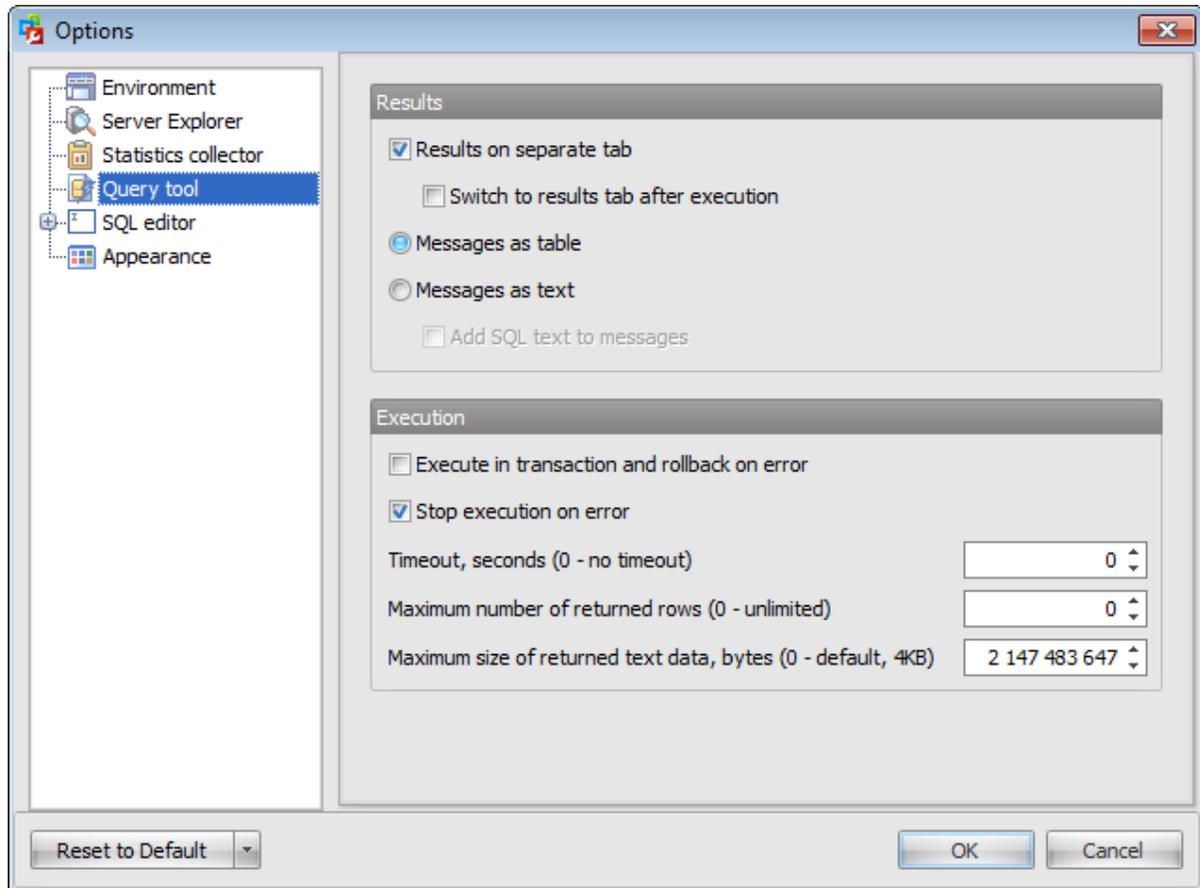
## 12.4 Query tool

### Results on separate tab

If this option is checked then query results will be displayed on a separate tab of the [Query Editor](#).

### Switch to results tab after execution

Check this option to view query results on the results tab immediately after the query execution



### Messages as table

If this option is checked then Query Editor messages on the **Messages** tab are displayed as a table.

### Messages as text

If this option is checked then Query Editor messages on the **Messages** tab are displayed as a text. Such view allows copying error messages.

### Add SQL text to messages

Check this option to add SQL statements to messages

### Execute in transaction and rollback on error

Check this option to rollback a transaction automatically in case of an occurred error.

**Stop execution on error**

If this option is checked then the query execution will be stopped in case of error.

**Timeout, seconds**

Set the number of seconds for SQL Server to wait before canceling the query. A value of 0 indicates an infinite wait, or no time-out.

**Maximum number of returned rows**

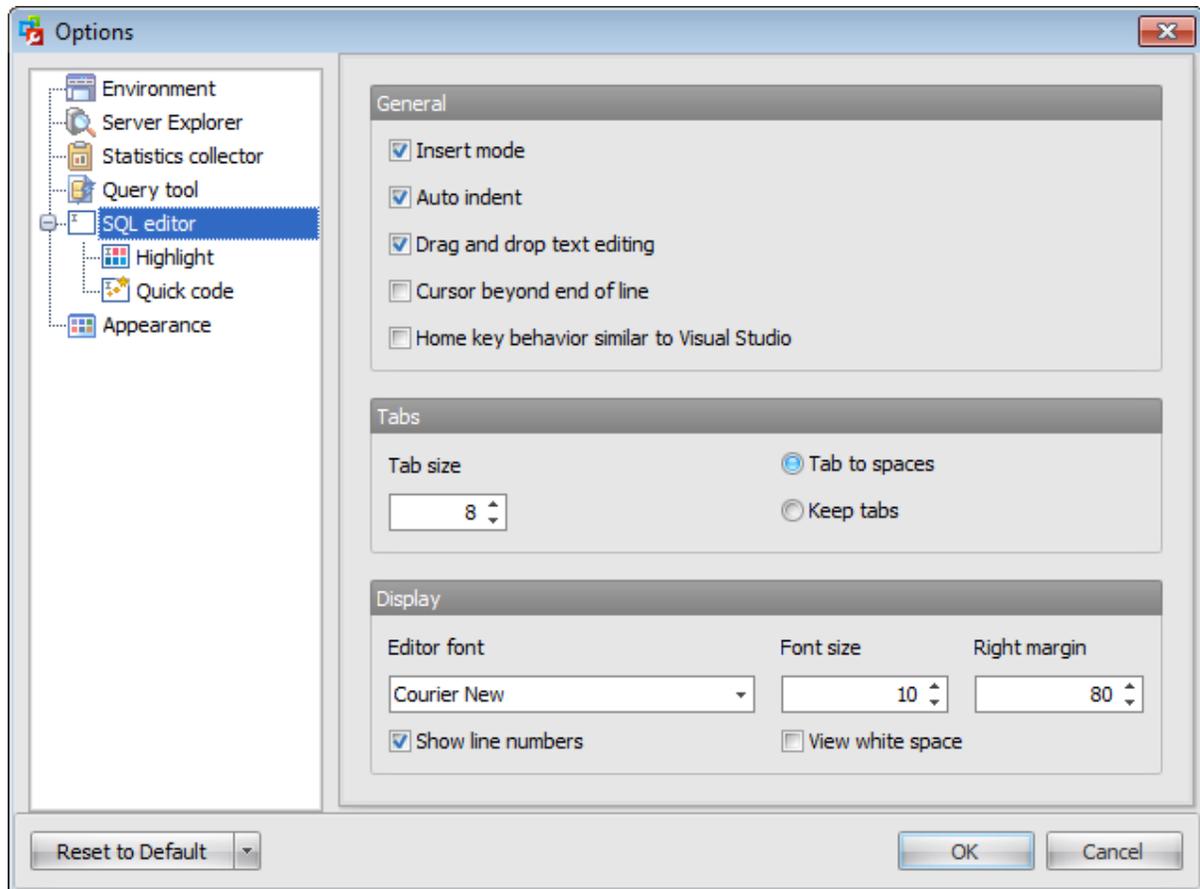
Specify the maximum number of rows to return after the query execution.

**Maximum size of returned text data, bytes**

Define the maximum size (in bytes) of text data to return after the query execution.

## 12.5 SQL Editor

Use this tab to set general options of [Query Editor](#).



### General

#### **Insert mode**

If this option is checked, the insert symbols mode is on by default.

#### **Auto indent**

If this option is checked, each new indentation is the same as the previous one when editing SQL text.

Indents and outdents are used in the process of text editing to make the source code easier to read.

#### **Drag and drop text ending**

This option allows to drag and drop selected text.

#### **Cursor beyond end of line**

If this option is checked, the horizontal position of a cursor is kept. If you move the cursor (using the *Up* and *Down* arrow keys) onto a line having length less than the current cursor horizontal position, it will be positioned after the last symbol of the line.

#### **Home key behaviour similar to Visual Studio**

If this option is checked then the cursor is placed at the beginning of the string on the *Home* key press.

### Tabs

#### Tab size

Define the tab length used when editing a text.

Specify the way the TAB key is processed:  **Tab to spaces** (insert space characters) or  **Keep tabs** (insert a tab character).

### Display

#### Editor font

Use the drop-down list to set the font of a query text.

#### Font size

Use the drop-down list to set the font size of a query text.

#### Right margin

Define the position of the right text margin in the editor window.

#### Show line number

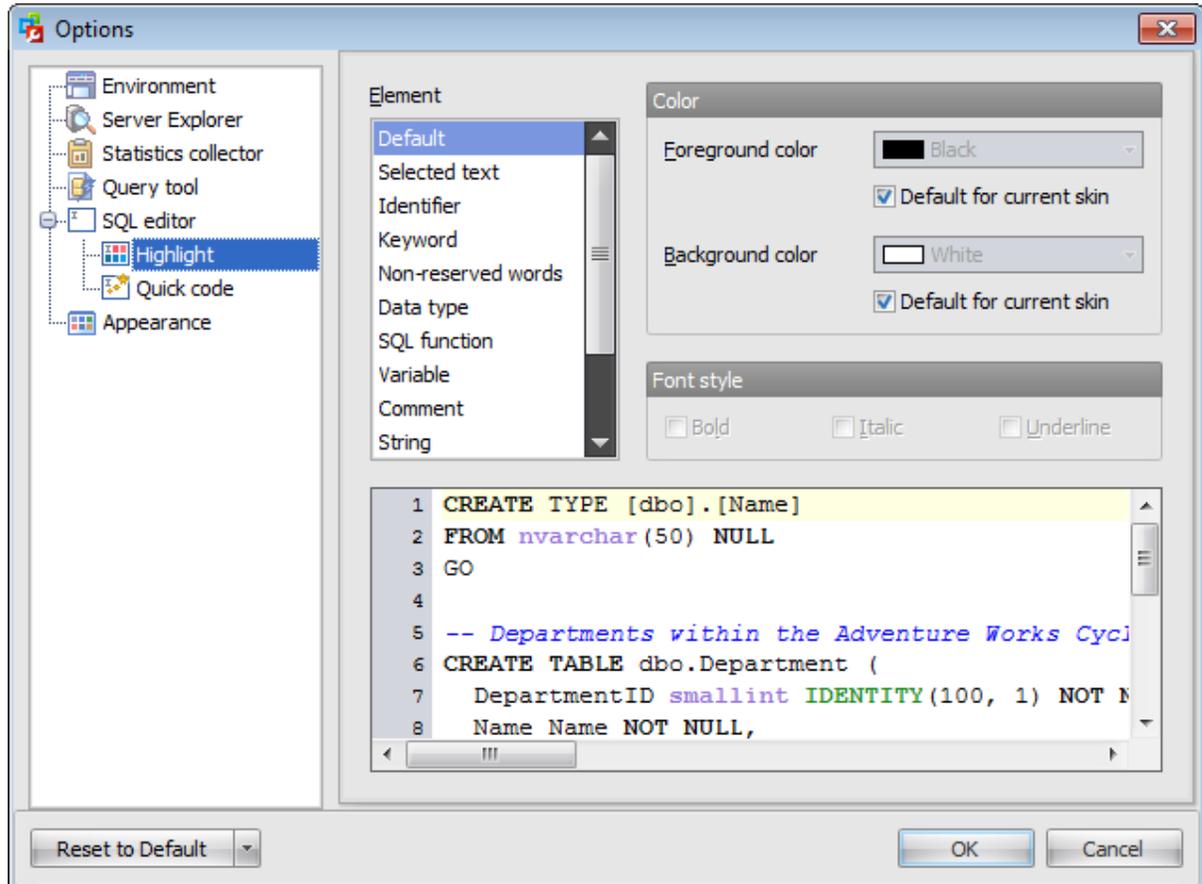
If this option is checked, line numbers are displayed in the SQL text editor window.

#### View white space

Check this option to show white spaces in the text of a query.

### 12.5.1 Highlight

The **Highlight** section of the **SQL Editor Options** dialog allows you to specify the **font styles, foreground and background colors** of the text used by the editor to mark out various text elements: *selected text, identifier, keyword, non-reserved words, data type, SQL function, variable, comment, string, number, current line, error.*



The **Element** list contains all elements available in SQL editor of the program. For your convenience the preview area (under the **Element** list) illustrates the changes being made to each of the elements.

Select the text foreground/background colors from the **Foreground color / Background color** drop-down lists. If you check the  **Default for current skin** box, the default color will be applied.

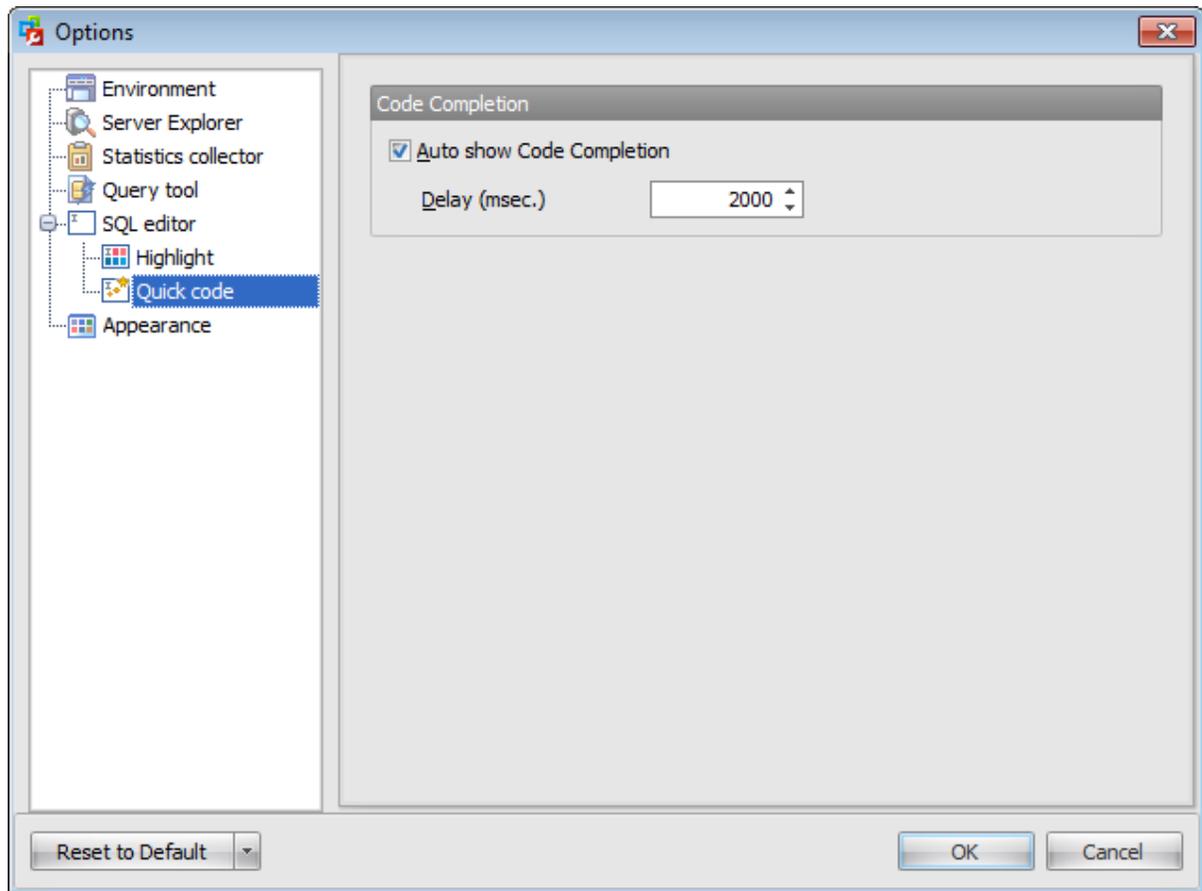
There is also a possibility to set the font style in the **Font style** group:

*Bold*

*Italic*

Underline

## 12.5.2 Quick code



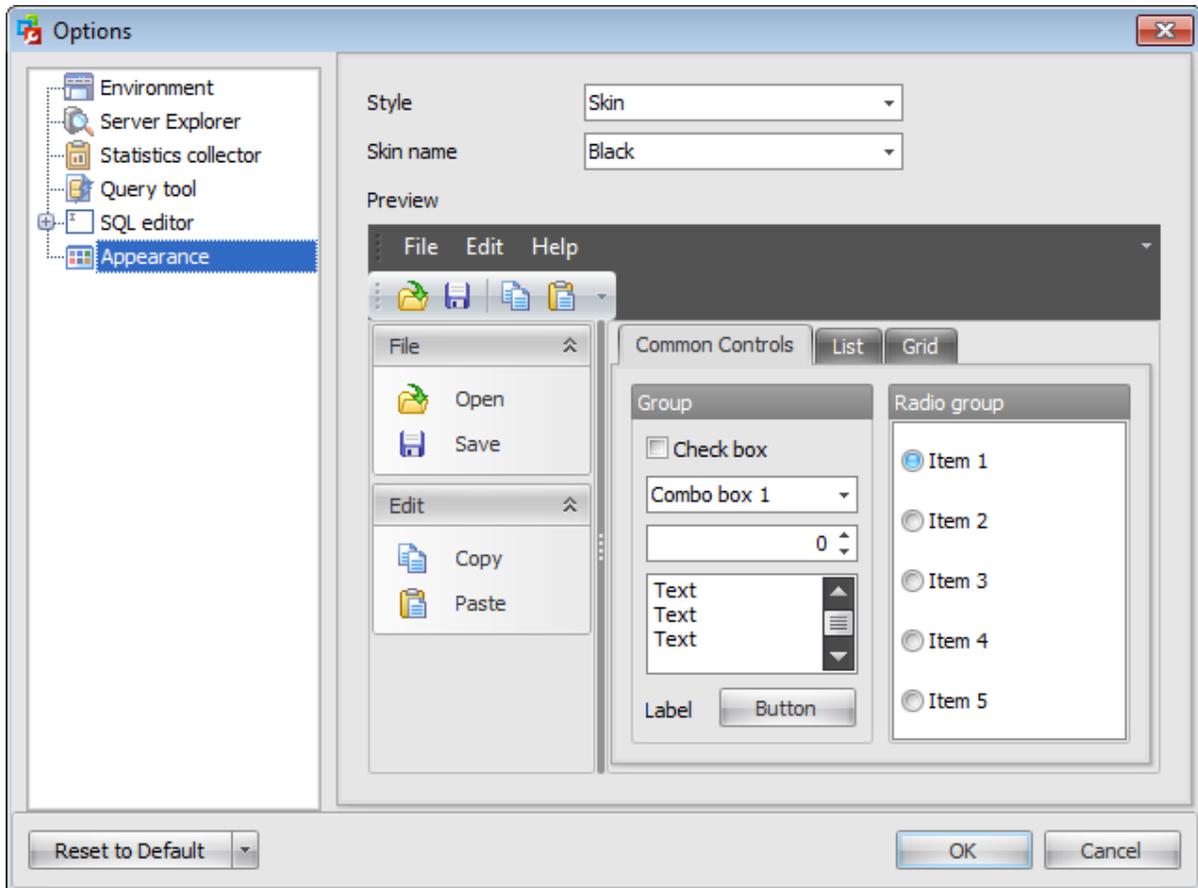
### **Auto show Code Completion**

If this option is checked, then on typing the first word characters in the SQL text editor you will be offered some variants for the word completion in a popup list (an analogue of the **Code Insight** feature in **Delphi IDE**). The popup list will appear after a period of time defined by the **Delay (msec.)** option.

## 12.6 Appearance

The **Appearance** section of the **Options** dialog allows you to choose the application interface style according to your liking.

Use the **Style** drop-down list to select an interface scheme according to your liking: *Windows XP, Flat, Ultra flat, Style 3D, Office 2003* or *Skin*. If you chose the *Skin* style, then define **Skin name** using corresponding drop-down list.



**Preview** area shows you the view of the chosen scheme.



**Part**

**XIII**

## 13 Appendix

### 13.1 Shortcuts

#### Objects list

<i>Insert</i>	Create a new object
<i>Enter</i>	Edit selected object(s)
<i>Delete</i>	Drop selected object(s)

#### Query Editor

<i>Ctrl+N</i>	Create a new query
<i>Ctrl+O</i>	Load the saved query from file
<i>Ctrl+S</i>	Save a query
<i>Ctrl+Shift+S</i>	Save all queries
<i>Ctrl+X; Shift+Del</i>	Cut
<i>Ctrl+C; Ctrl+Insert</i>	Copy
<i>Ctrl+V;</i>	Paste
<i>Shift+Insert</i>	
<i>Ctrl+A</i>	Select all
<i>Ctrl+Z; Alt+Back</i>	Undo
<i>Ctrl+Y</i>	Redo
<i>Ctrl+K+K; Ctrl+B+T</i>	Toggle a bookmark
<i>Ctrl+Shift+Num</i>	Toggle a numbered bookmark
<i>Ctrl+K+N; Ctrl+B+N</i>	Go to next bookmark
<i>Ctrl+K+P; Ctrl+B+P</i>	Go to previous bookmark
<i>Ctrl+Num</i>	Go to numbered marker
<i>Ctrl+B+C</i>	Clear all markers
<i>Ctrl+K+L</i>	Clear unnumbered markers
<i>Ctrl+F</i>	Search for text using the <a href="#">Find Text</a> dialog
<i>Ctrl+H</i>	Replace text
<i>F3</i>	Find next
<i>Ctrl+K+C</i>	Comment
<i>Ctrl+K+U</i>	Uncomment
<i>Tab</i>	Indent selection
<i>Shift+Tab</i>	Unindent selection
<i>Ctrl+Shift+U</i>	Make uppercase
<i>Ctrl+U</i>	Make lowercase
<i>Ctrl+Alt+U</i>	Toggle case
<i>Ctrl+R+W</i>	View special characters
<i>Ctrl+Space</i>	Show code completion
<i>Ctrl+G</i>	Go to line...
<i>Shift+Alt+Up</i>	Select column upwards
<i>Shift+Alt+Down</i>	Select column downwards
<i>Shift+Alt+Left</i>	Select column symbol-by-symbol to the left
<i>Shift+Alt+Right</i>	Select column symbol-by-symbol to the right
<i>Ctrl+Shift+Alt+Left</i>	Select column word-by-word to the left
<i>Ctrl+Shift+Alt+Right</i>	Select column word-by-word to the right
<i>t</i>	
<i>Ctrl+Left</i>	Go to the previous word
<i>Ctrl+Right</i>	Go to the next word
<i>Ctrl+Shift+Left</i>	Select to the previous word
<i>Ctrl+Shift+Right</i>	Select to the next word
<i>Ctrl+PageDown</i>	Go to the last line on the page
<i>Ctrl+PageUp</i>	Go to the first line on the page

*Ctrl+Shift+PageDown* Select text to the last line on the page  
*Ctrl+Shift+PageUp* Select text to the first line on the page  
*Ctrl+Shift+Alt+PageDown* Select column to the beginning of the page  
*Down*  
*Ctrl+Shift+Alt+PageUp* Select column to the end of the page  
*Up*  
*Shift+Home* Select text to the beginning of the line  
*Shift+End* Select text to the end of the line  
*Ctrl+Shift+Alt+Home* Select column from the current cursor position to the beginning of the  
*e* first line  
*Ctrl+Shift+Alt+End* Select column from the current cursor position to the beginning of the  
last line  
*Insert* Toggle Insert/Replace mode  
*Ctrl+Back* Delete symbols from the current cursor position to the beginning of the  
word

**Working with windows**

*Ctrl+Tab* Switch to the next tab  
*F6* Switch to the next window  
*Ctrl+F6* Switch to the previous window

## 13.2 Find & Replace text dialog

The **Find&Replace Text** dialog is provided for quick and flexible searching for specified text and replacing text within the working area of SQL Administrator editors.

### Text to find

Enter a search string in this box. The Arrow-Down button which can be found next to the input box allows you to select any of the previously entered search strings.

### Replace with

This box allows you to enter a string to replace the search string. The Arrow-Down button which can be found next to the input box allows you to select any of the previously entered strings. To replace the search string with an empty string, leave this input box blank.

### Options

**Case sensitive**

This option can be used to differentiate uppercase characters from lowercase ones during the search process.

**Whole words only**

Use this option to search for words only (with this option off, the search string might be found within longer words).

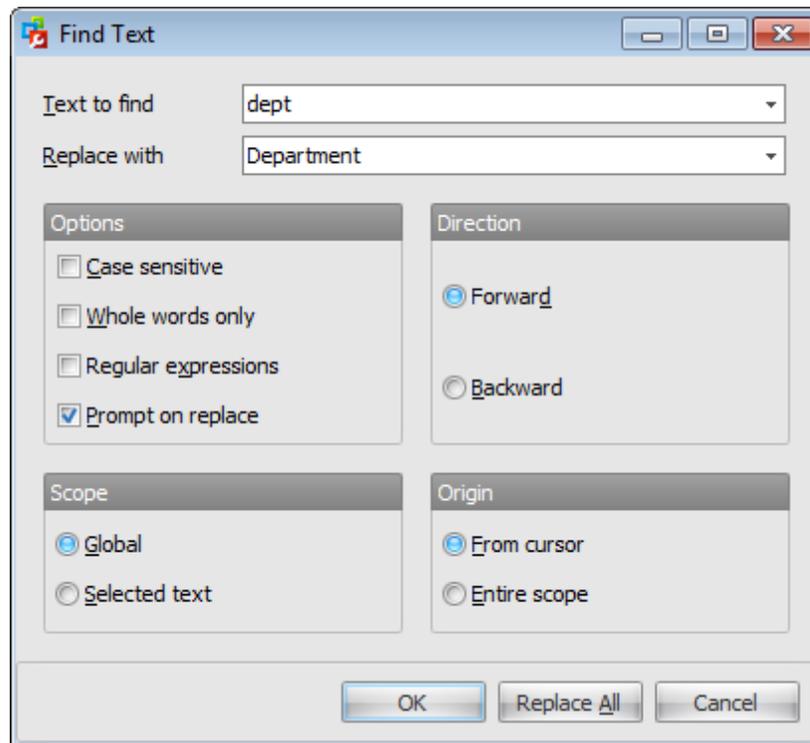
**Regular expressions**

Recognizes regular expressions in the **Text to find** field.

**Prompt on replace**

Check this option if you wish to be prompted before replacing upon each occurrence of the search string. When this option is off, the search string is replaced automatically.

**Note:** The syntax of regular expressions that can be used in the **Text to find** field is similar to that used in Perl regular expressions. Comprehensive information about it can be found at <http://perldoc.perl.org/perlre.html#Regular-Expressions>.



### Direction

#### **Forward**

Searches from the current position to the end of the working area.

#### **Backward**

Searches from the current position to the beginning of the working area.

### Scope

#### **Global**

Searches within the entire working area, in the direction specified by the *Direction* setting.

#### **Selected text**

Searches only within the currently selected text, in the direction specified by the *Direction* setting. You can use the mouse or block commands to select a block of text.

### Origin

#### **From cursor**

The search starts at the cursor's current position, and then proceeds either forward to the end of the scope, or backward to the beginning of the scope depending on the *Direction* setting.

#### **Entire scope**

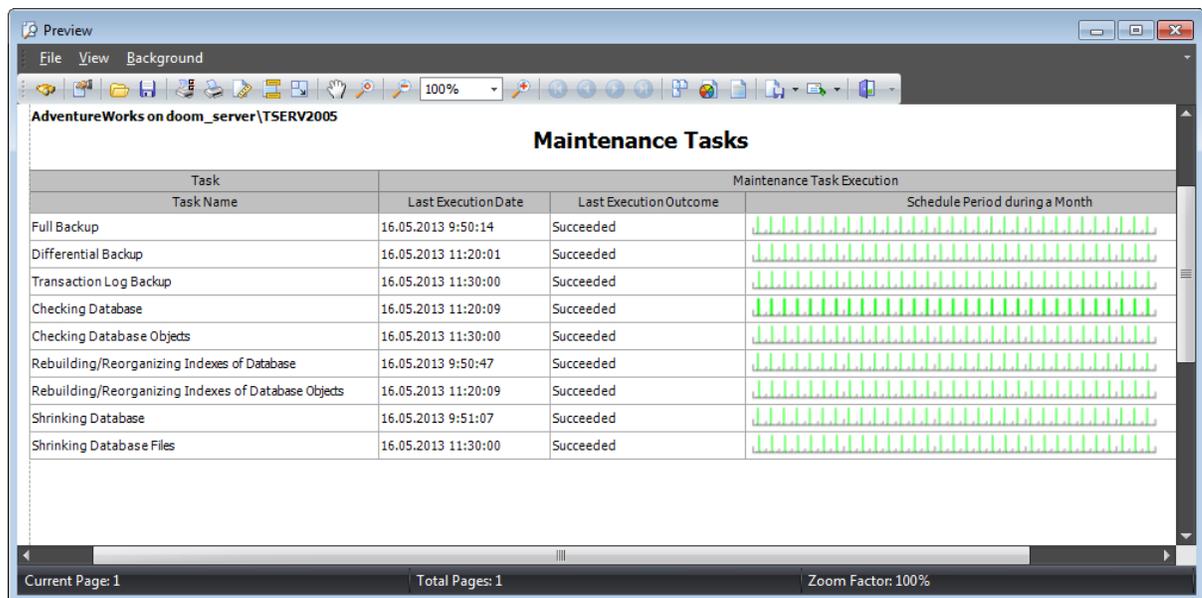
The search covers either the entire block of selected text or the entire script (no matter where the cursor is in the Editor area) depending upon the *Scope* options.

### 13.3 Print preview

You can make a preview of the report that you want to print. Using the preview window you can export report data to a file or change the report appearance, i.e. change a document background and add watermarks.

The report can be created on data contained in different grids and chart views located in tabs of the [Server Explorer](#) including [Alarm summary](#). Also you can create a report on the [result of an SQL query](#).

To view the report select the  **Print** context menu item or click the corresponding toolbar button.



Task	Maintenance Task Execution		
Task Name	Last Execution Date	Last Execution Outcome	Schedule Period during a Month
Full Backup	16.05.2013 9:50:14	Succeeded	
Differential Backup	16.05.2013 11:20:01	Succeeded	
Transaction Log Backup	16.05.2013 11:30:00	Succeeded	
Checking Database	16.05.2013 11:20:09	Succeeded	
Checking Database Objects	16.05.2013 11:30:00	Succeeded	
Rebuilding/Reorganizing Indexes of Database	16.05.2013 9:50:47	Succeeded	
Rebuilding/Reorganizing Indexes of Database Objects	16.05.2013 11:20:09	Succeeded	
Shrinking Database	16.05.2013 9:51:07	Succeeded	
Shrinking Database Files	16.05.2013 11:30:00	Succeeded	

The report includes the following information:

- report creation date and time;
- server name;
- database name (for reports pertaining to databases);
- report name;
- grid with columns relevant to the report.

The main menu of the **Preview** window allows you to:

- open page setup dialog;
- start printing the report;
- export the content to any of the available formats: *PDF file*, *HTML file*, *MHT file*, *RTF file*, *Excel file*, *CSV file*, *Text file*, *Image file* and then send it via E-mail;
- close the **Preview** window;
- change page layout: Facing or Continuous;
- enable/disable toolbar;
- enable/disable status bar;
- customize toolbar;

- set report background and watermark.

The toolbar of the **Preview** window allows you to:

- search for text within the document;
- customize print options;
- open a previously saved report (\*.prnx file);
- save the current report to an external \*.prnx file;
- start printing the report;
- open page setup dialog;
- customize report header and footer;
- set scaling options;
- enable/disable hand tool;
- adjust zoom options;
- navigate within the printing report pages;
- set the number of pages displayed in the window;
- set report background and watermark;
- export the content to any of the available formats: *PDF file, HTML file, MHT file, RTF file, Excel file, CSV file, Text file, Image file* and then send it via E-mail;
- close the **Preview** window.

Availability:

**Full** version                      **Yes**

**Free** version                      **No**

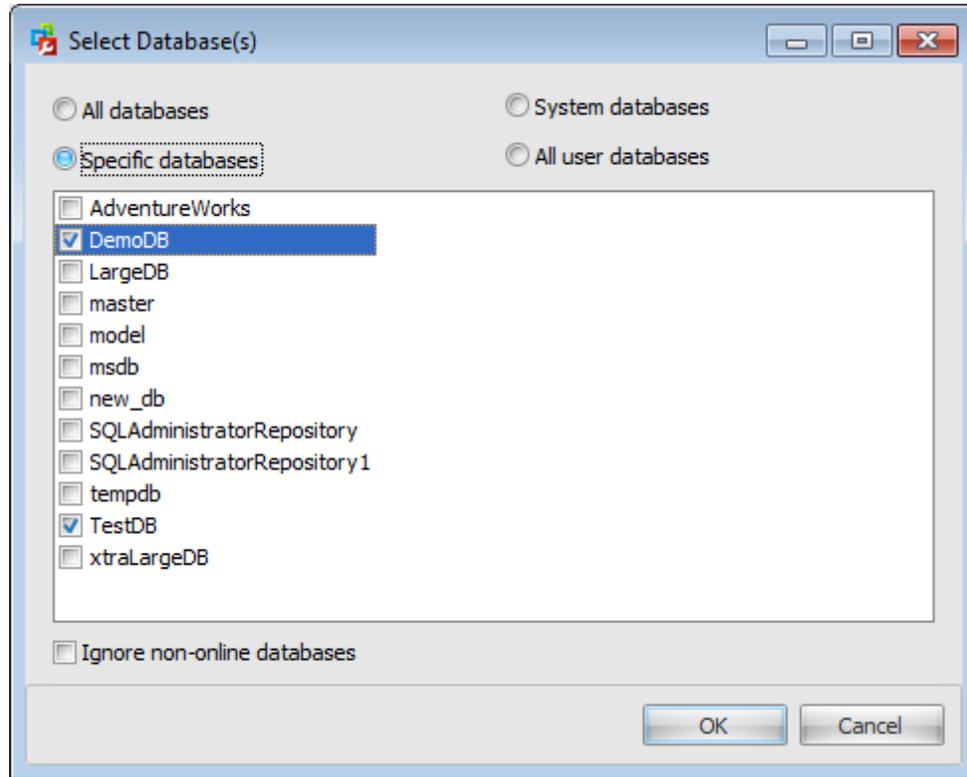
**Note:** To compare all features of the **Full** and the **Free** versions of SQL Administrator, refer to the [Feature Matrix](#) page.

## 13.4 Database states

-  - online;
-  - offline;
-  - standby (read-only for restore log);
-  - restoring;
-  - emergency;
-  - emergency and read-only;
-  - single-user;
-  - emergency and single-user;
-  - standby and single-user;
-  - standby, single-user and read-only;
-  - standby and read-only;
-  - read-only;
-  - recovering, recovery pending, suspect.

## 13.5 Select databases dialog

The **Select Databases** dialog appears each time you choose the *Select...* item of the **Database(s)** drop-down list when creating or editing some [maintenance tasks](#).



This list displays all databases registered on the server. Here you can define databases against which a maintenance task should be run.

**All databases**

Generate a maintenance plan that runs maintenance tasks against all Microsoft SQL Server databases except *tempdb*.

**System databases**

Generate a maintenance plan that runs maintenance tasks against each of the SQL Server system databases except *tempdb*. No maintenance tasks are run against user-created databases.

**All user databases**

Generate a maintenance plan that runs maintenance tasks against all user-created databases. No maintenance tasks are run against the SQL Server system databases.

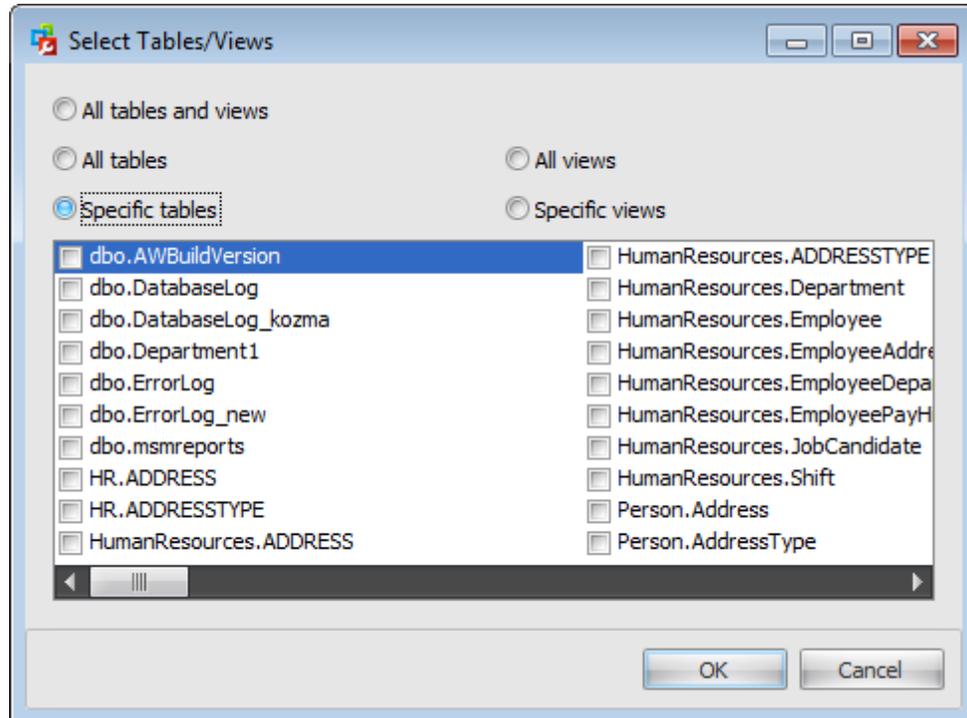
If you select the  **Specific databases** option then choose among registered databases to define databases for maintenance task by checking the corresponding boxes.

**Ignore non-online databases**

Check this option to backup databases only in case they are online.

## 13.6 Select objects dialog

The **Select Tables/Views** dialog appears each time you choose the *Select...* item of the **Object(s)** drop-down list when creating or editing some [maintenance tasks](#).



To run the task against all tables and views/tables/views of the selected database choose the corresponding options.

If you select the  **Specific tables** or the  **Specific views** option then choose among existing tables/views to specify the tables or indexes affected by this task by checking the corresponding boxes.

## 13.7 Using templates

For your convenience the ability to use templates is provided by SQL Administrator. A template is a named collection of wizard options stored in a file.

Instead of performing a long chain of routine steps all the time you can save all the options of the wizard for future use as a template file. Select the **Templates | Save Template** drop-down menu item, specify the template file name and set an optional comment for the template file.

When starting the wizard next time, you can load the template by selecting the **Templates | Load Template** drop-down menu item.

Note that saving/loading of templates is possible at any step of the wizard.

# Credits

**Software Developers:**

*Alexander Zhiltsov*

**Technical Writers:**

*Olga Radchenko*

*Tatyana Ovsyannikova*

**Cover Designer:**

*Tatyana Makurova*

**Translators:**

*Anna Shulkina*

*Serge Fominikh*

**Team Coordinators:**

*Alexander Zhiltsov*

*Alexander Chelyadin*

*Roman Tkachenko*