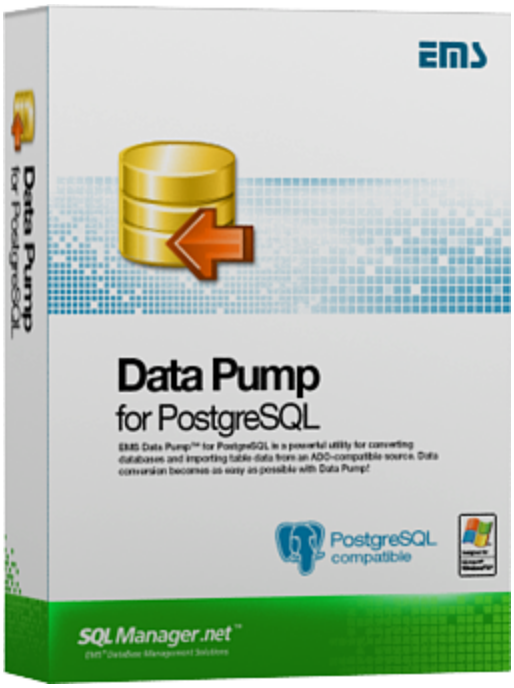
SQL Manager.net™

EMS® Software Development



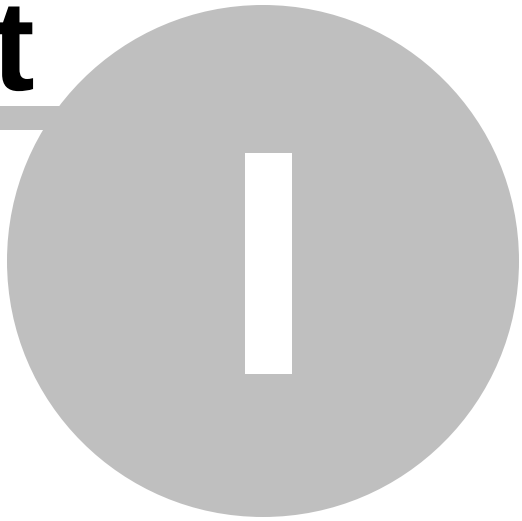
Data Pump for PostgreSQL User's Manual

Data Pump for PostgreSQL
User's Manual

Document generated on: 10.02.2023

Table of Contents

Part



1 Welcome to EMS Data Pump!

EMS Data Pump for PostgreSQL is an excellent utility for converting databases and importing table data from any ADO-compatible source (e.g. MS Access or any other database with ADO support) to PostgreSQL databases. The easy-to-use wizard application allows you to build the ADO connection string, select the source tables, fields, indexes and constraints for converting, view and edit the SQL script for generating target PostgreSQL database structure, select tables for importing data and customize a number of pump options. Database conversion becomes as easy as possible with **Data Pump for PostgreSQL**!

Visit our web-site: http://www.sqlmanager.net/ for details.

Key features

- User-friendly wizard interface
- Converting structure and data from any local and remote data source having an ADO provider
- Selecting source schemas, tables, fields and indexes for conversion
- Ability to create a new PostgreSQL database or connect to an existing one
- Ability to connect to target PostgreSQL server via SSH or HTTP tunnel
- Fast generation of tables, fields and indexes according to the source database structure
- Ability to specify schema for target database tables
- Easily customizable options for target database objects
- Ability to view/edit, execute and save SQL script for generating the target database structure
- Selecting tables and setting criteria for data import
- Unicode data support
- Full monitoring of the pumping process
- Ability to save all parameters specified within the current wizard session to a template
- Multilanguage GUI support
- Command-line utility to automate the data pump job with a template file used

Product information

Homepage:	http://www.sqlmanager.net/en/products/postgresql/datapump
Support Ticket System:	http://www.sqlmanager.net/support
Register on-line at:	http://www.sqlmanager.net/en/products/postgresql/datapump/buy

1.1 What's new

Version	Release date
Data Pump for PostgreSQL 3.1.3	February 10, 2023

What's new in EMS Data Pump?

- Implemented support for PostgreSQL 15.
- Some minor fixes and improvements.

See also:

Version history

1.2 System requirements

System requirements

- Microsoft Windows XP, Microsoft Windows Server 2003, Microsoft Windows Server 2008, Microsoft Windows Server 2008 R2, Microsoft Windows Server 2012, Microsoft Windows Server 2012 R2, Microsoft Windows Server 2016, Microsoft Windows Vista, Microsoft Windows 7, Microsoft Windows 8/8.1, Microsoft Windows 10
- 64MB RAM or more;
- 20MB of available HD space for program installation
- Possibility to connect to any local or remote PostgreSQL server as the target and an ADO-compatible source
- Supported PostgreSQL server versions: from 7.3 up to 15

1.3 Installation

If you are **installing Data Pump for PostgreSQL for the first time** on your PC:

- download the **Data Pump for PostgreSQL** distribution package from the download page available at our site;
- unzip the downloaded file to any local directory, e.g. *C:\unzipped*;
- run *PgDataPumpSetup.exe* from the local directory and follow the instructions of the installation wizard;
- after the installation process is completed, find the **Data Pump** shortcut in the corresponding group of Windows Start menu.

If you want to **upgrade an installed copy of Data Pump for PostgreSQL** to the latest version:

- download the **Data Pump for PostgreSQL** distribution package from the download page available at our site;
- unzip the downloaded file to any local directory, e.g. *C:\unzipped*;
- close **Data Pump** application if it is running;
- run *PgDataPumpSetup.exe* from the local directory and follow the instructions of the installation wizard.

See also:

System requirements

1.4 Registration

All purchases are provided by **Digital River** registration service. The **Digital River** order process is protected via a secure connection and makes on-line ordering by credit/debit card quick and safe.

Digital River 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, Check or Cash.

If you want to review your order information, or you have questions about ordering or payments please visit our Customer Care Center, provided by **Digital River.**

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 or passwords. Also you will receive a copy of registration keys or passwords 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.

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 or passwords and download links for archives of full versions. Also you will receive a copy of registration keys or passwords by e-mail. Please make sure to enter a valid e-mail address in your order. If you have not received the keys within 2 hours, please, contact us at sales@sqlmanager.net

Product distribution	MyCommerce/Digital River
EMS Data Pump for PostgreSQL (Business license) + 1-Year Maintenance*	Register Now!
EMS Data Pump for PostgreSQL (Business license) + 2-Year Maintenance*	
EMS Data Pump for PostgreSQL (Business license) + 3-Year Maintenance*	
EMS Data Pump for PostgreSQL (Non-commercial license) + 1-Year Maintenance*	
EMS Data Pump for PostgreSQL (Non-commercial license) + 2-Year Maintenance*	
EMS Data Pump for PostgreSQL (Non-commercial license) + 3-Year Maintenance*	
EMS Data Pump for PostgreSQL (Trial version)	Download Now!

***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 online, speed-through Maintenance Reinstatement/Renewal Interface. After reinitiating/renewal you will receive a confirmation e-mail with all the necessary information.

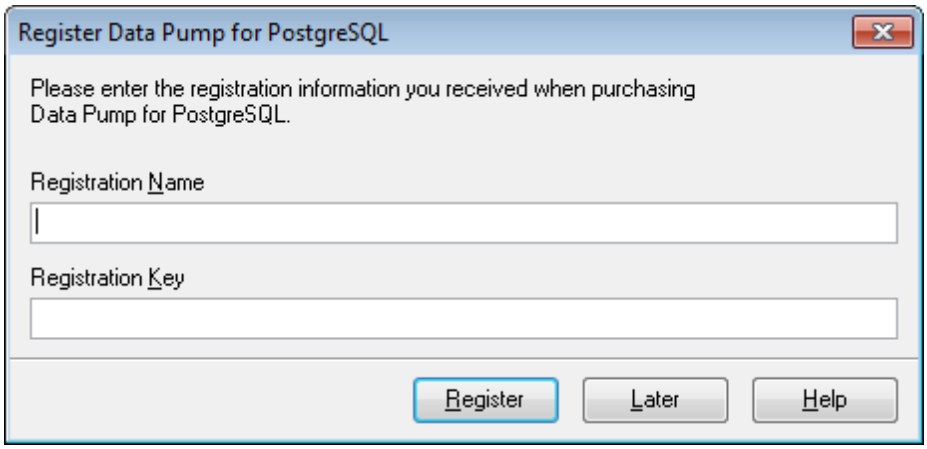
See also:

How to register EMS Data Pump

1.5 How to register EMS Data Pump

To **register** your newly purchased copy of **EMS Data Pump for PostgreSQL**, perform the following:

- receive the notification letter from **Digital River** 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 at the startup page.



See also:

Registration

1.6 Version history

Product name	Version	Release date
Data Pump for PostgreSQL	Version 3.1.2	March 4, 2020
Data Pump for PostgreSQL	Version 3.1.1	March 10, 2016
Data Pump for PostgreSQL	Version 3.1	June 20, 2011
Data Pump 2009 for PostgreSQL	Version 3.0.0.1	February 17, 2009
Data Pump 2006 for PostgreSQL	Version 2.2.0.1	February 21, 2007
Data Pump 2006 for PostgreSQL	Version 2.1.0.1	June 26, 2006

Version 3.1.2

- Implemented support for Azure, SQL Server 2017, MySQL Server 8 and PostgreSQL 12.
- The TRUNCATE statement is now used for deleting data from tables.
- MySQL. The 'MySQL has gone away' error has been fixed.
- PostgreSQL. Default values are now processed correctly.
- The total number of records for large tables is now displayed correctly.
- The error occured in case of using COPY statement for import. Fixed now.
- The error occured on loading the template with columns selection in the console version. Fixed now.
- Other fixes and improvements.

Version 3.1.1

- Renewed installation program containing lots of fixes.
- Milliseconds support for the Datatime and Timestamp MySQL types.
- The 'Out of memory' error occured on transferring a large amount of data from InterBase. Fixed now.
- The script for the indices having the same name was generated incorrectly for SQL Server and MySQL. Fixed now.
- Incorrect constraints refreshing in MS SQL Server with ODBC has been fixed.
- The 'Out of memory' error occured transferring the data from DBF/FoxPro using Advantage OLE DB Provider 11. Fixed now.
- The 'Access Violation' error occured at Step 5 of the wizard in some cases while importing data from MySQL. Fixed now.
- There was an error while reconnecting to the source MySQL server. Fixed now.
- Lots of other improvements and bug-fixes.

Version 3.1

- Added the possibility to transfer objects descriptions, if they are supported by provider.
- Added the option allowing to set sequence options after data importing.
- It is now possible to transfer views/queries data.
- Added the possibility to refresh the objects tree of source database.
- Now when importing data, foreign keys are disabled and recreated automatically.
- Support of the latest Server versions is implemented.
- Some other improvements and bugfixes.

Version 3.0

- Implemented Unicode data support

- Source DBMS schemas support has been implemented
- Added an option allowing one to enable/disable refreshing of source objects on connection
- Added an option specifying whether all source tables are selected by default at Step 3
- Quoting identifiers for the source database can be customized at Step 4 now
- Pumping BLOB fields using ODBC drivers v5.1 for MySQL was not possible. Fixed now
- Latest versions of target PostgreSQL DBMS are supported
- Improved schema support for target database
- It is now possible to clear tables before importing data
- It is now possible to generate DROP TABLE statements
- Using OIDs for pumped objects is now optional
- The wizard window is now resizable
- The possibility to restart the wizard is added at the last step
- It is now possible to specify the wizard step that will be opened after loading a template
- It is now possible to start data import after loading a template
- Implemented password encryption in templates and registry
- A number of minor bug-fixes and visual improvements

Scroll to top

Version 2.2

- It is now possible to enforce quoting for object names in the result script for data import automatically (when applied to "unrecognized" OLE DB/ODBC providers)
- Implemented Private Key support for SSH authentication
- Added the ability to specify the default schema for tables import
- Added French and German interface localizations for the utility
- A number of minor bug-fixes and visual improvements

Scroll to top

Version 2.1

- Two types of template formats have been implemented: the fixed one which points to a certain list of objects to be transferred, and the dynamic one which defines only excluded objects; the objects out of this list are to be transferred. Template files of old format are still supported
- Added the opportunity to connect through the SSH or HTTP tunnel
- The 'Recreate database' option is implemented in the GUI version of the utility: the options allows you to delete the existing database before creating a new one (this option has been only in the console version until now)
- Now the console version returns the error code (0 - successful termination, 1 - critical error, 2 – non-critical error)
- Added an option allowing one to specify whether the COPY statement is to be used instead of the INSERT INTO statement for data import to increase the transfer speed considerably

Full version history is available at http://www.sqlmanager.net/products/postgresql/datapump/news

Scroll to top

See also:

What's new

1.7 EMS Data Pump FAQ

Please read this page attentively if you have questions about **EMS Data Pump for PostgreSQL**.

Table of contents

Question/answer list

Q: What is EMS Data Pump?

A: **Data Pump for PostgreSQL** is an excellent utility for converting databases and importing table data from any ADO-compatible source (e.g. MS Access or any other database with ADO support) to PostgreSQL databases. The easy-to-use wizard application allows you to build the ADO connection string, select the source tables, fields, indices and constraints for converting, view and edit the SQL script for generating target PostgreSQL database, select tables to be imported and customize a number of pump options.

Q: What do I need to start working with EMS Data Pump?

A: First of all, you must have a possibility to connect to some local or remote PostgreSQL server to work with **Data Pump**. You can download PostgreSQL database server from http://www.postgresql.org/download/. Besides, you need your workstation to satisfy the system requirements of **Data Pump for PostgreSQL**.

Q: What is the easiest way to configure the template files for the PostgreSQL Data Pump console application?

A: You can configure the template files visually using the PostgreSQL Data Pump wizard. Set the required pump options at of the wizard, click the 'Tools' button and select the 'Save Template' popup menu item. All the options will be saved to the template file which can be used later in the console application.

Q: How can I register the product?

A: If you have already purchased **Data Pump for PostgreSQL**, you can register the product by entering the appropriate registration information. Please refer to Registration and How to register EMS Data Pump for details.

Q: Are there any limitations implied in the trial version as compared with the full one?

A: The trial version of the utility admits to the maximum of 10% of records to be imported for each table. In all other respects it does not differ from the full version as far as the functionality is concerned. That is, you can test all the features implemented in **Data Pump for PostgreSQL** within the 30-day trial period.

,

Q: I m trying to import data from MS Access. When I expand a table node, it shows indices(0), constraints(0) and I cannot select them. How can I fix this?

A: The solution to this problem is to use the Microsoft Jet OLE DB Provider for establishing connection to your source Access database.

Q: I create a database schema for the target server, go through the steps of the wizard, do not execute script at Step 5 and the available tables and selected tables panels are empty too. The Next button is disabled. What is wrong here?

A: This issue arises because of differences between the generated script and the schema already in existence. The problem is that some table names or field names may differ from the ones generated by Data Pump (the correspondence between the source and the target tables is verified by tables' and their fields' names). If your new (target) database structure differs from the source database structure, no objects will be available in the Available Tables and the Selected Tables lists at Step 7. The source metadata must match the target metadata for successful data import script execution.

Scroll to top

If you have any additional questions, contact us at our Support Center.

1.8 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.

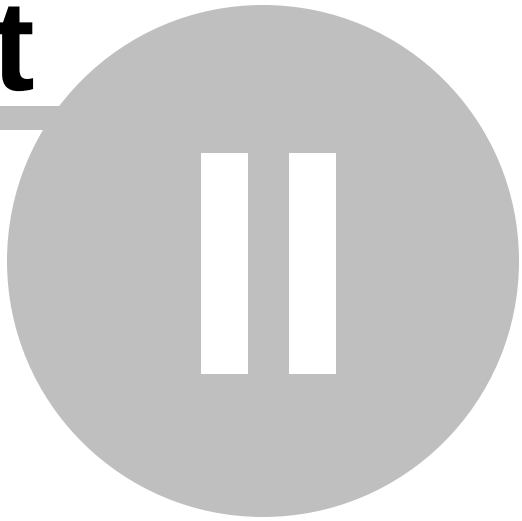
Advanced Query Builder for RAD Studio VCL
Advanced Query Builder for RAD Studio VCL is a powerful component for Delphi and C++ Builder intended for visual building SQL statements for the SELECT, INSERT, UPDATE and DELETE clauses.

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

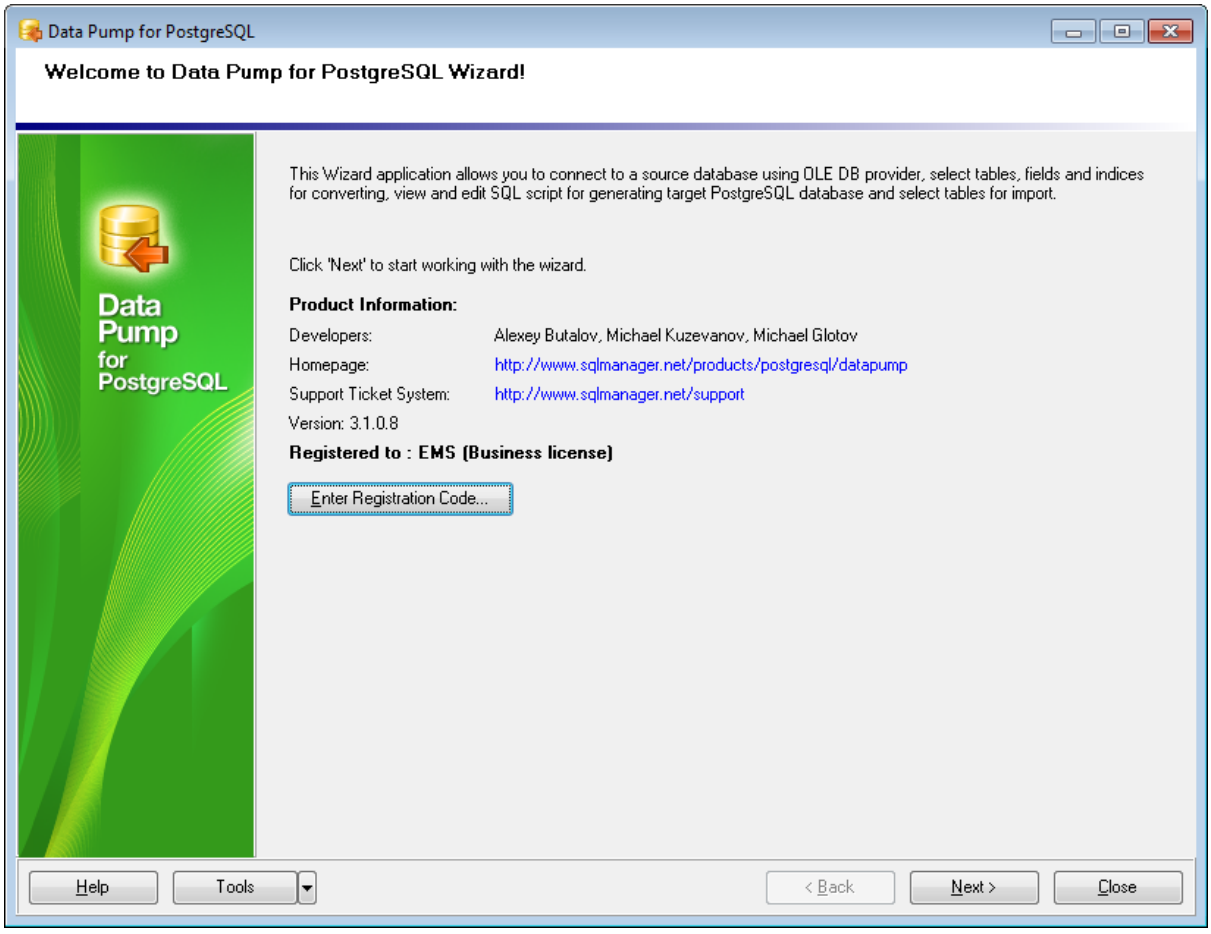
Part



2 Wizard Application

EMS Data Pump for PostgreSQL Wizard guides you through the entire pumping process and provides an easy-to-use graphical interface allowing you to set all data pumping parameters visually. Wizard application allows you to build the ADO connection string, select tables, fields, indexes and constraints for converting, view and edit the SQL script for generating the target PostgreSQL database and select tables for import.

Navigation through the steps of the wizard is performed with the help of the **Next>** and the **<Back** buttons.
Use the **Tools** button to call a menu allowing you to open the **Preferences** dialog or to **load/save a template**.



See also:
Console Application

2.1 Using Wizard Application

Go through the steps of the wizard and follow the wizard instructions to run the pumping process successfully.

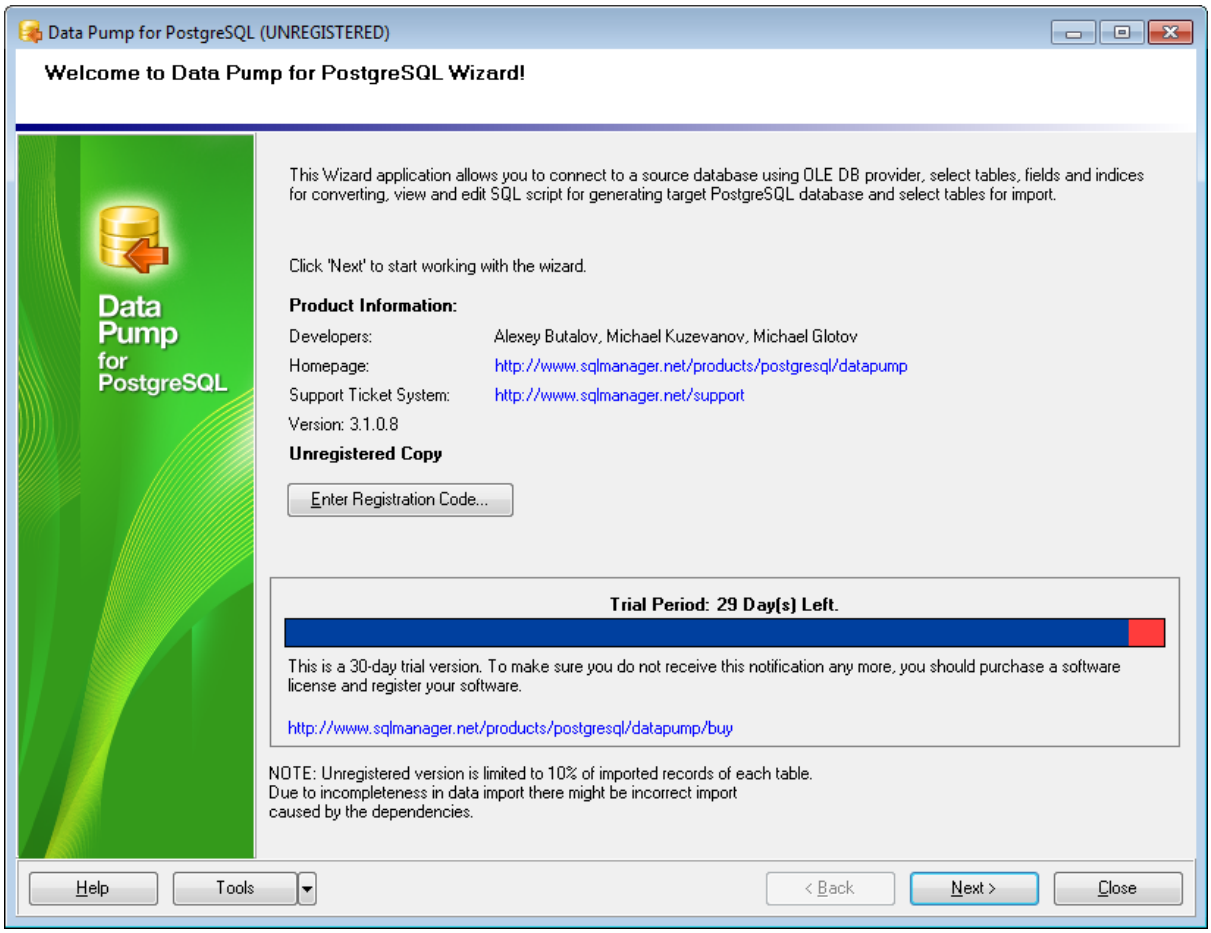
See also:

Using Console Application
Using Templates
Setting Program Preferences

2.1.1 Getting started

This is how **Data Pump for PostgreSQL** application wizard looks when you first start it.

This page allows you to view registration information. If you have not registered **Data Pump for PostgreSQL** yet, you can do it by pressing the **Enter Registration Code...** button and specifying your registration information.



Press the **Next** button to proceed to Step 1 of the wizard.

2.1.2 Step 1 - Setting connection properties

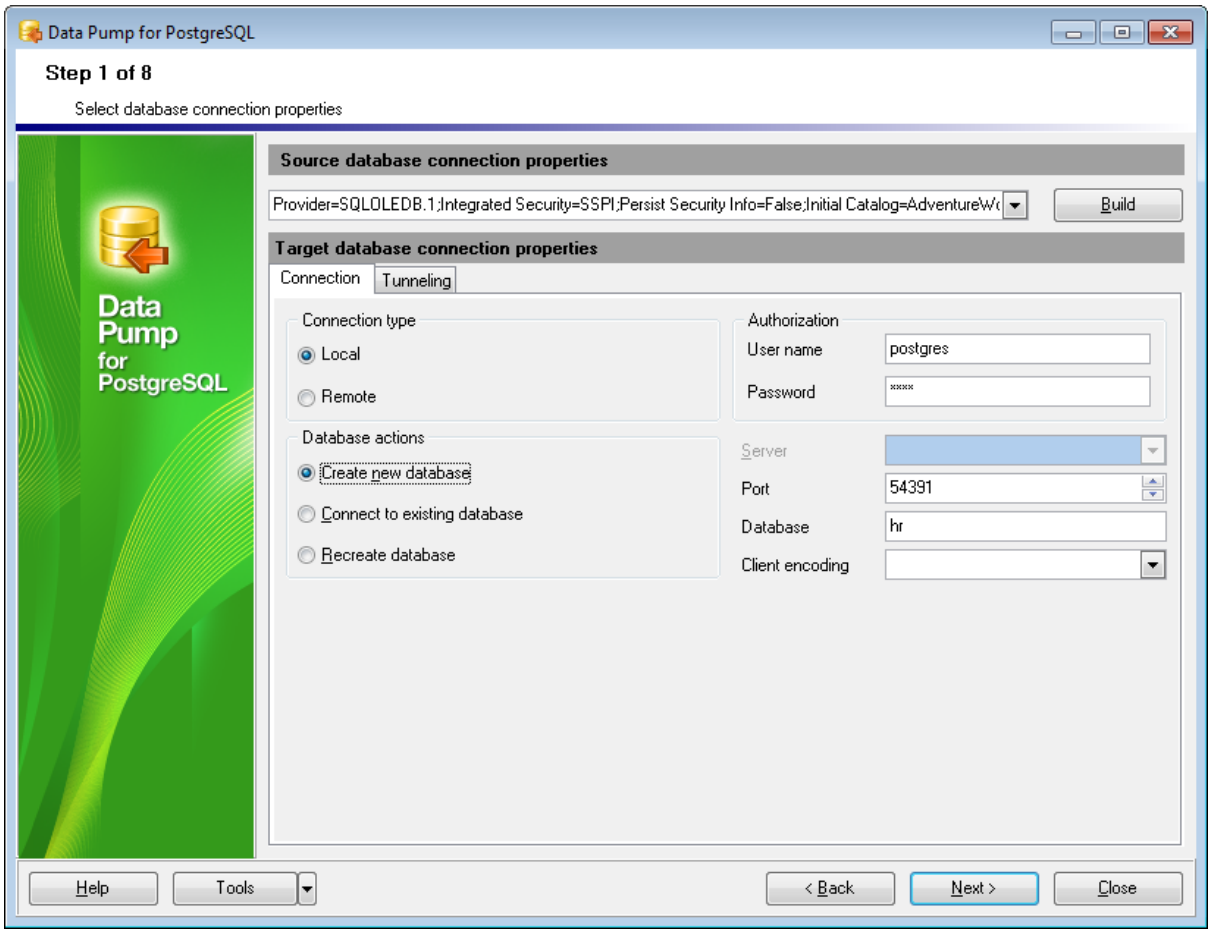
At **Step 1** of the wizard you should set the source and the target database connection properties.

Please note that you need to have sufficient privileges to be able to write to the destination database on PostgreSQL server.

Note: To establish connection to the source database, it is necessary to use additional software, namely data sources (OLE DB Providers and ODBC Drivers) which are distributed by DBMS developing companies and third-party developers. **EMS Data Pump** processes information received from the data source. The utility itself extracts neither data nor metadata from the source database. Therefore the functionality of **Data Pump** may be partially limited by the abilities of data source being used.

Source database connection properties

In order to set the *source database connection properties*, use Data Link Properties builder or select an initialization string from the list of previously built ones. To call the Data Link Properties dialog, press the **Build** button.



Target database connection properties
First select the **connection type**: *local* or *remote*.
Local connection is used to connect to PostgreSQL server running on the same machine where **Data Pump for PostgreSQL** is launched.
The **Remote** mode allows you to connect to PostgreSQL server running on another computer in the network.

Database actions
You can also set the preferable database action:

Create new database
Select this action to create a new database on the target PostgreSQL server to pump data into this database.

Connect to existing database
Select this action if the target database for pumping data already exists.

Recreate database
If this action is selected, the target database will be dropped and then recreated (**NB:** you will lose all data previously stored in the specified database in this case).

Authorization
Here you must specify valid authorization info to access the target PostgreSQL server: **User name** and **Password**.
The default superuser name is 'postgres' with the password specified during PostgreSQL server installation.

Server
For remote connection you should enter PostgreSQL host name in this field, or select one from the drop-down list.

Port
Use this field to specify a port to connect through.

Database
Type in the name of the target database for pumping data, or select one from the drop-down list.

If necessary, use the drop-down list to specify the preferable **Client encoding** to be used by the application.

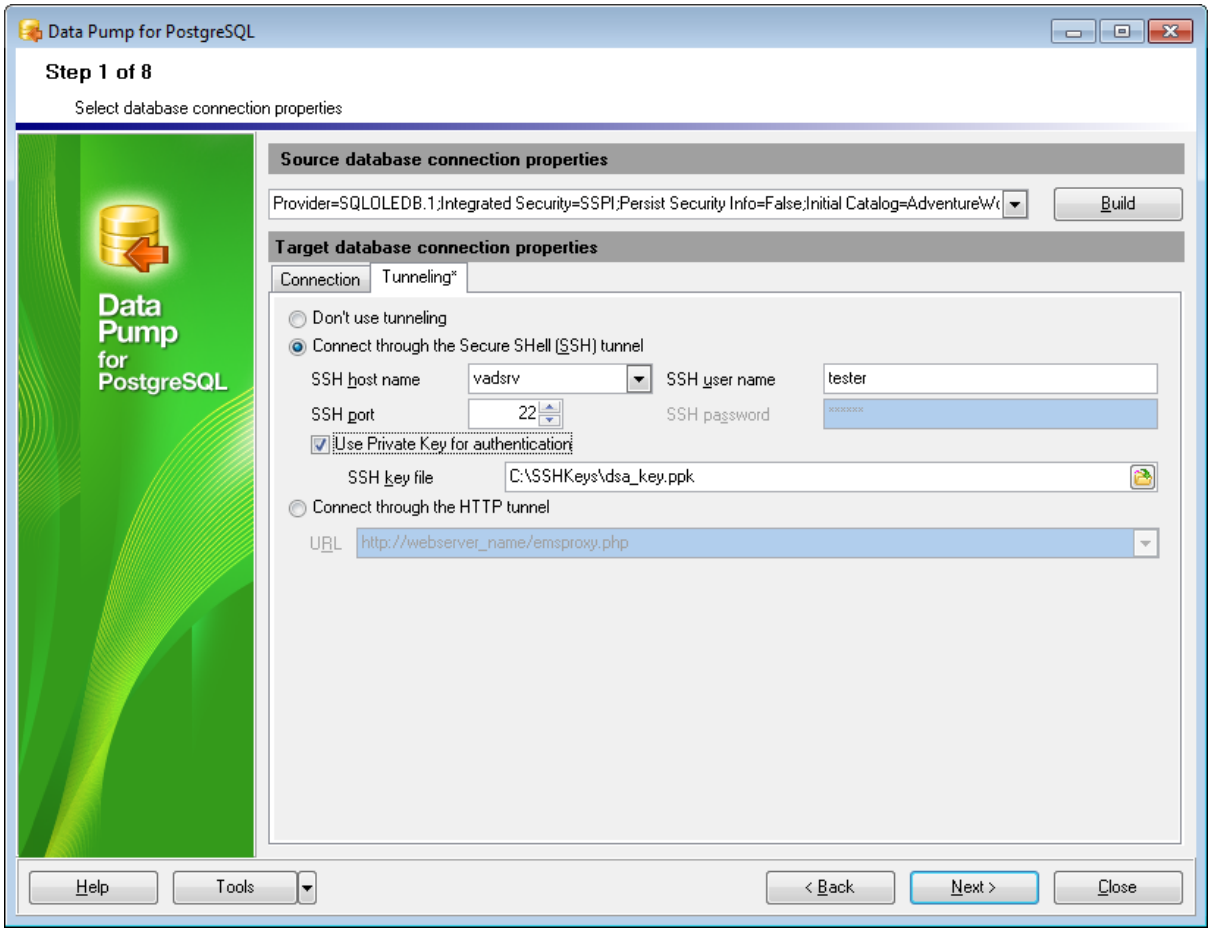
If you are using the EMS SQL Management Studio for PostgreSQL version of Data Pump for PostgreSQL then the **Select registered database** button is available. Click this button to pick a database already registered in the EMS SQL Management Studio in the Select Host or Database dialog.

Tunneling settings

To setup the connection via **SSH tunnel**, input the following values in the corresponding fields:

- **SSH host name** is the name of the host where SSH server is running
- **SSH port** indicates the port where SSH server is activated
- **SSH user name** stands for the user on the machine where SSH server is running (**Note:** it is a Linux/Windows user, not a user of PostgreSQL server)
- **SSH password** is the Linux/Windows user password

For details see SSH tunneling options.



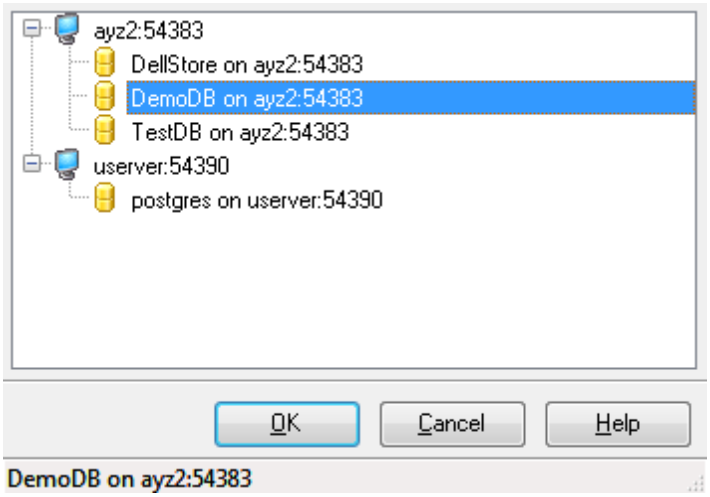
To use **HTTP tunneling**, just upload the tunneling script to the webserver where PostgreSQL server is located, or to any other webserver from which direct connections to your PostgreSQL server are allowed. This script exposes the PostgreSQL API as a set of web-services which is used by **Data Pump for PostgreSQL**.

For details see HTTP tunneling options.

When you are done, click the **Next** button to proceed to the selecting source schemas step of the wizard.

2.1.2.1 Selecting registered database

Use this dialog to select a database for pumping data. This dialog is available only in EMS SQL Management Studio version of Data Pump for PostgreSQL.



All databases registered in EMS SQL Management Studio for PostgreSQL are displayed in the list.

Select the necessary database and click the **OK** button.

Database registration information will be filled on the first step automatically.

2.1.2.2 Setting data link properties

The **Data Link Properties** dialog allows you to set the data link properties necessary to establish connection to the data source. The connection properties builder has its own help system. Click the **Help** button to get additional information concerning connection settings for ODBC.

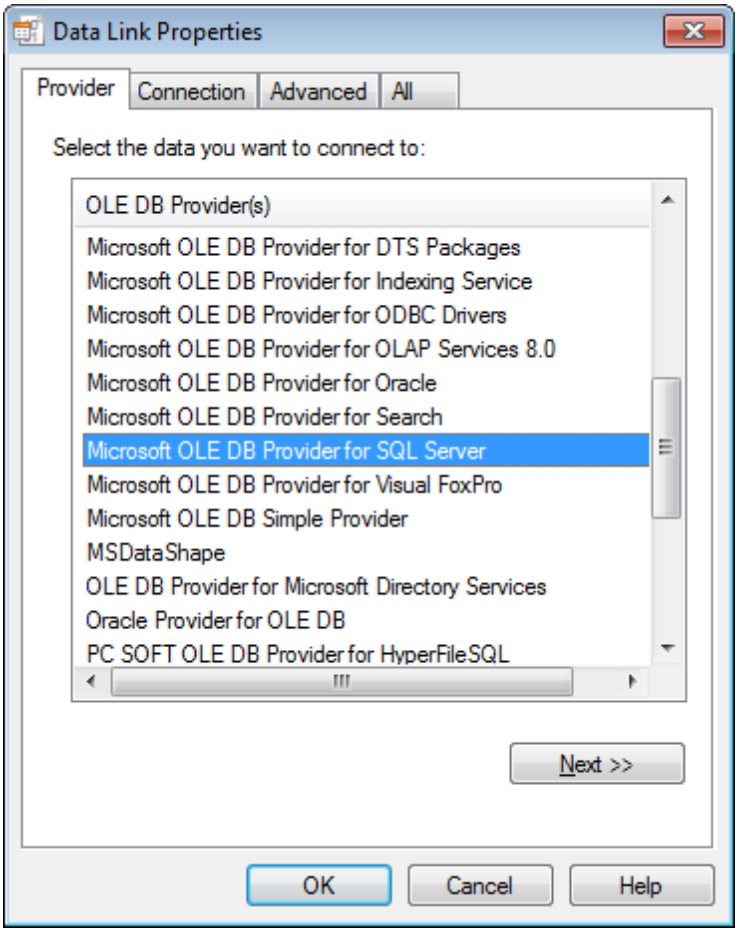
Use the tabs of the dialog to configure the connection string correctly.

Provider
Connection
Advanced
All

<< Back to setting connection properties

2.1.2.2.1 Provider

Within the **Provider** tab you can select the OLE DB provider to be used for connecting to the data source. For example, in order to connect to Microsoft® SQL Server™, select *Microsoft OLE DB Provider for SQL Server*.
In order to set connection via ODBC driver select *Microsoft OLE DB Provider for ODBC Drivers*.



Click the **Next** button to proceed to the Connection properties tab.

See also:
Connection
Advanced
All

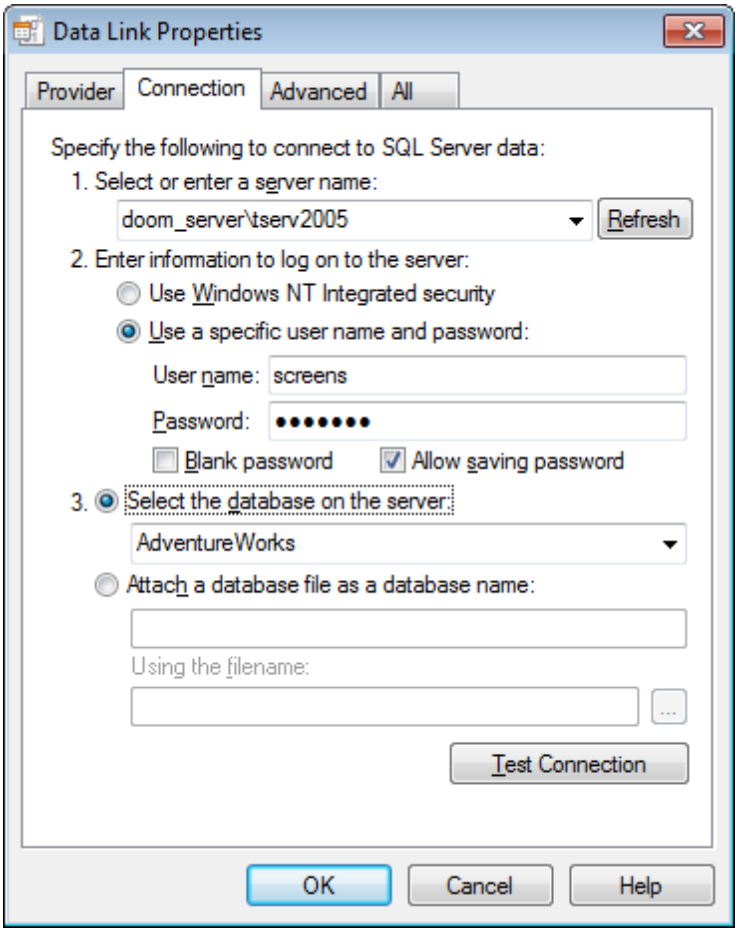
2.1.2.2.2 Connection

Within the **Connection** tab you can set the connection properties.

In order to set the connection properties correctly, follow the steps below.

- Type in the source server name, or select one from the drop-down list. You can get the actual server list by clicking the **Refresh** button.
- Depending on the preferable way to log on to the server, you can specify either *Use Windows NT Integrated security* or the database server account. If you have specified the specific server account to be used, enter *User name* and *Password* in the corresponding fields.
- Select a database on the server or specify the Microsoft database file (**.mdf*) which will be attached as a database.

Note: The set of available **Connection** options depends on the selected Ole DB Provider.



If you have selected *Microsoft OLE DB Provider for ODBC Drivers* the **Connection** tab is the following.

Click the **Test Connection** button to see if your connection settings are specified

correctly.

Use the **Advanced** and the **All** tabs of the dialog to access the corresponding pages available for the specified data source.

See also:

Provider

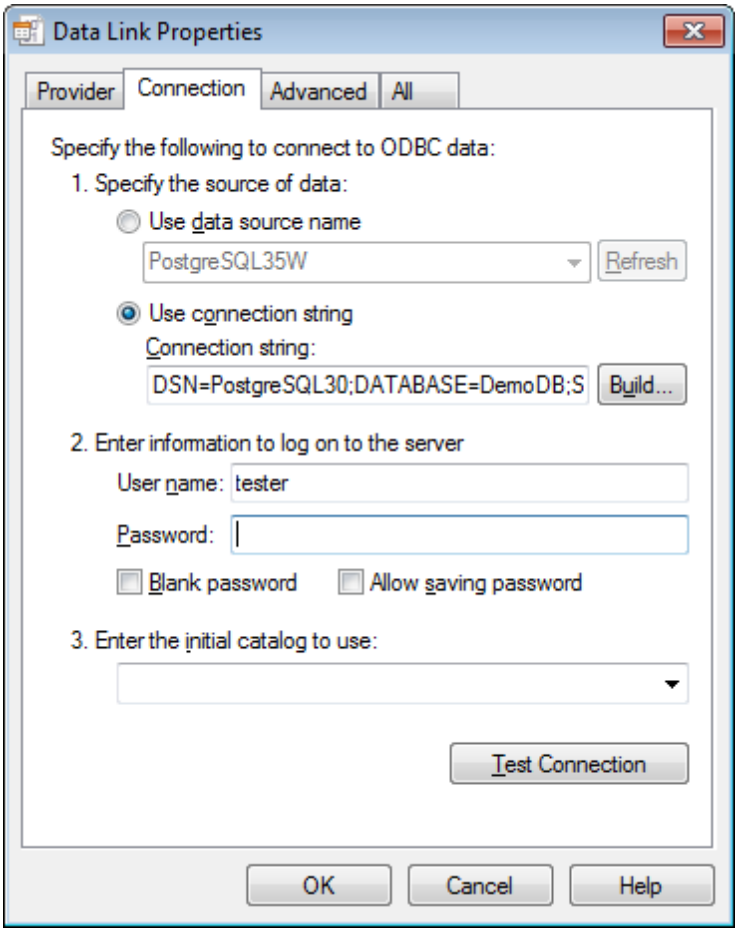
Advanced

All

2.1.2.2.2.1 Microsoft OLE DB Provider for ODBC

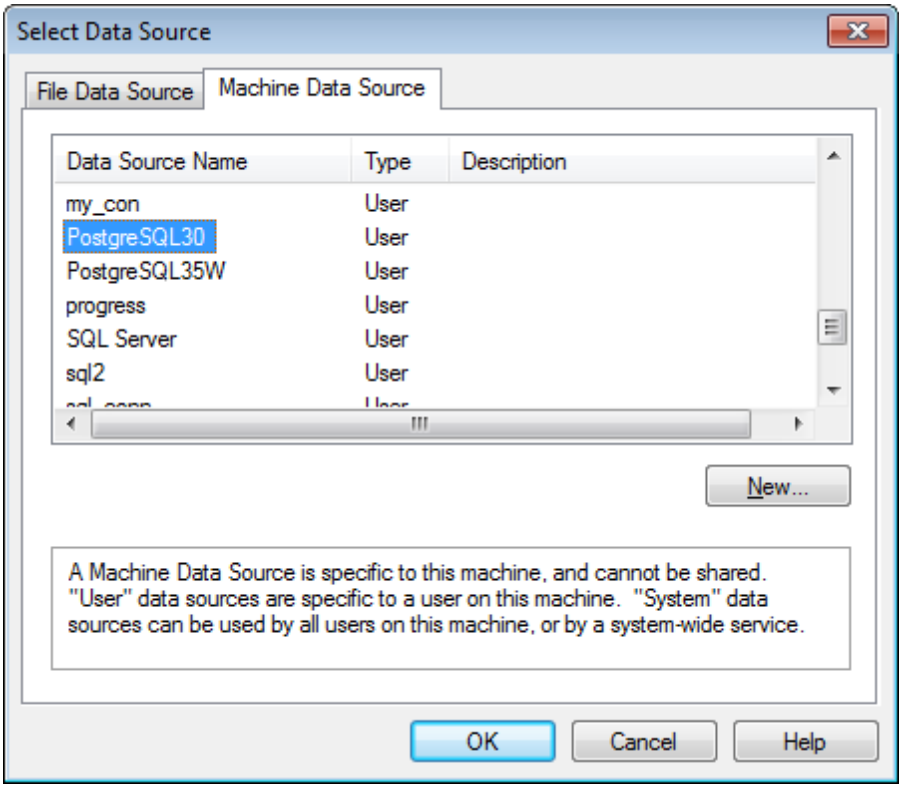
In order to create an ODBC using the new DSN correctly, follow the steps below:

- Select the **Use connection string** option and click the **Build...** button. Set data source properties in the opened Select Data Source dialog.
- Enter the log on information to connect to the server in the *User name* and *Password* fields. You may leave these fields empty as these connection parameters are provided in the created connection string.
- If necessary, select/deselect the available options: *Allow saving password*, *Blank password*.



Click the **Test Connection** button to see if your connection settings are specified correctly.

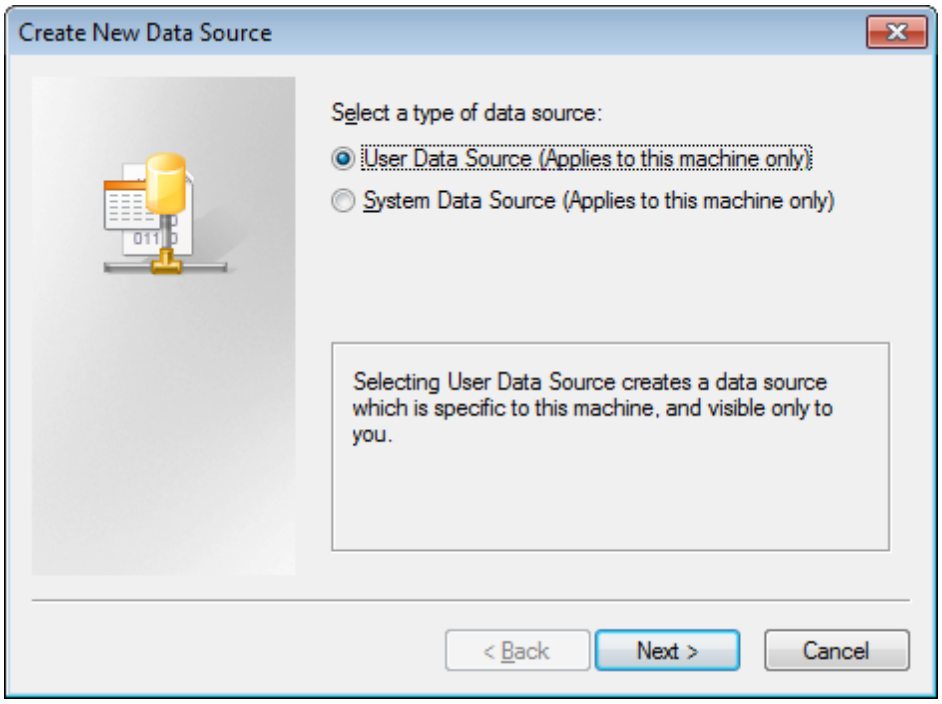
In the **Select Data Source** dialog proceed to the Machine Data Source tab in order to create a new DSN.



Click the **New...** button to launch the Create New Data Source wizard which is intended to build a new Data Source Name (DSN).

You can select existing DSN from the list as well.

On the first step of the wizard you need to select a data source type.

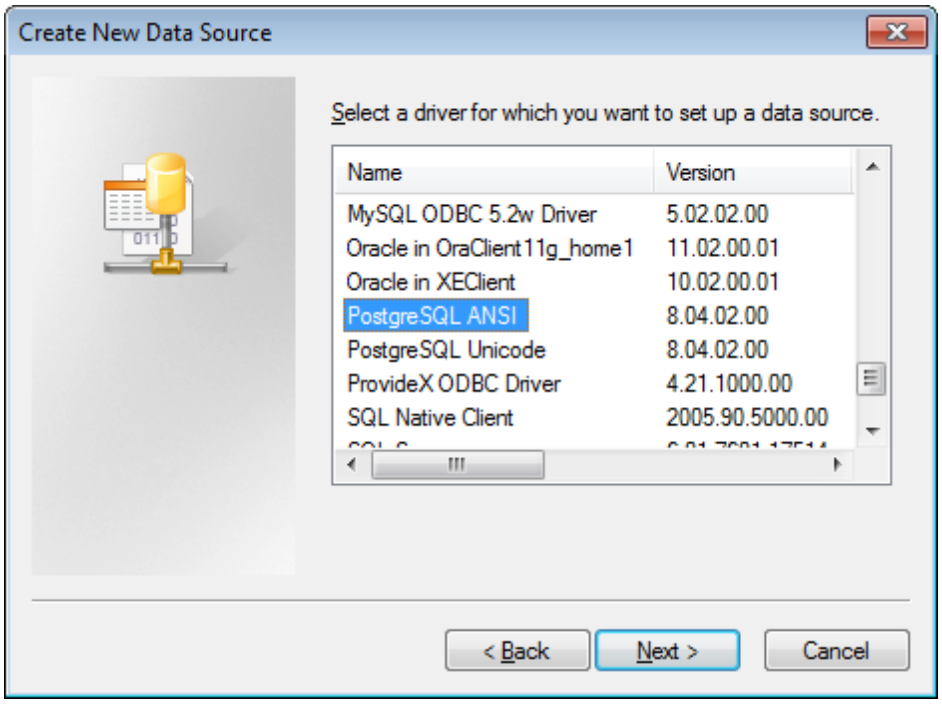


User Data Source
Selecting User Data Source creates a data source which is specific to this machine, and visible only to you.

System Data Source
Selecting a System Data Source creates a data source which is visible to any user, or service, logged into the machine.

Click the **Next** button to proceed to the Selecting driver step of the wizard.

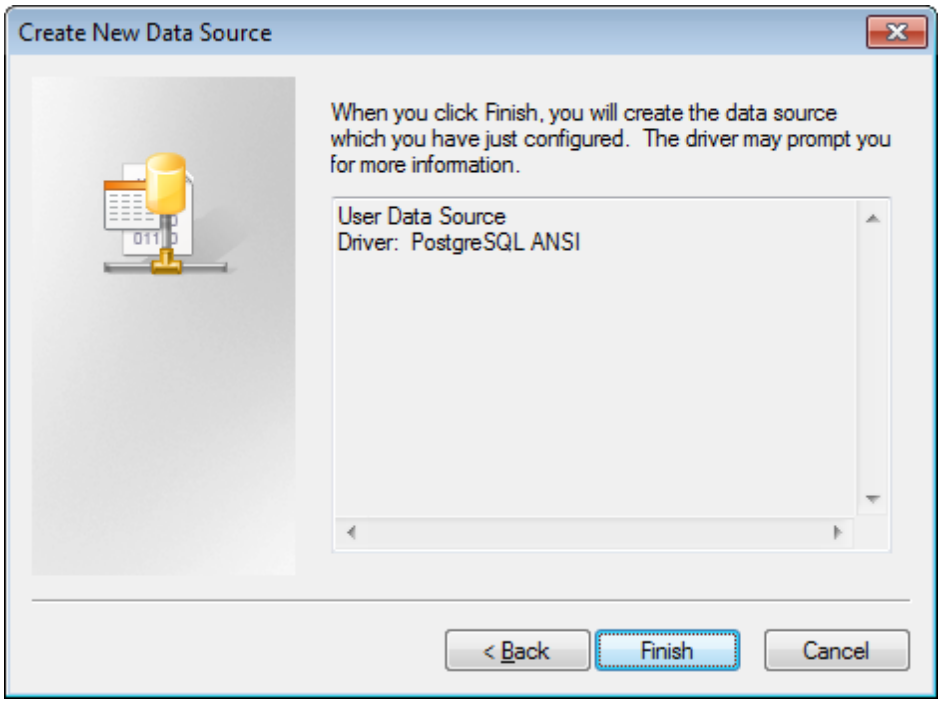
On this step of the wizard you are to select a driver for which you are adding a data source.
Choose the driver for which you are adding a data source. For example, PostgreSQL ANSI driver if you want to connect to the PostgreSQL server.



Any drivers that are installed on your machine are showed in the list.

Click the Next button to proceed to the Finish step of the wizard.

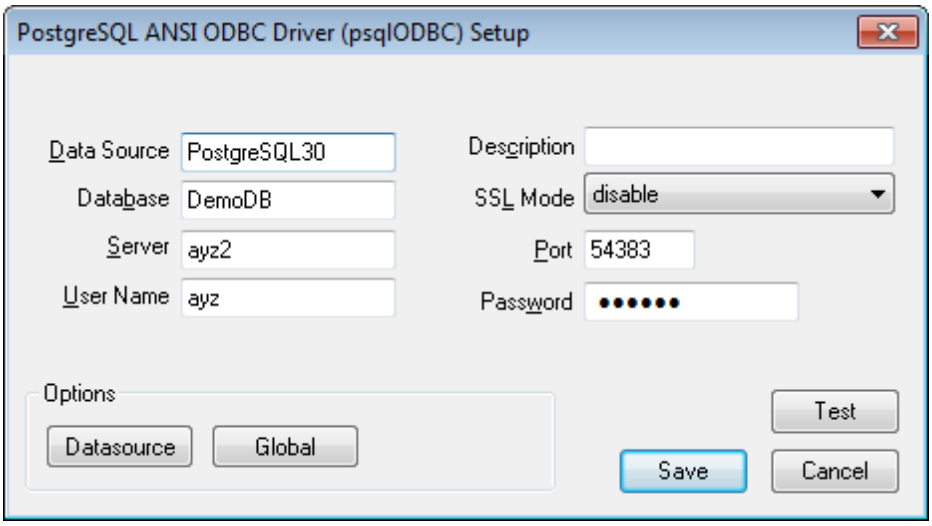
On this step of the wizard verify the choices made on the previous steps.



Click the **Finish** button to complete the wizard and launch the Setting specific ODBC

driver options dialog.

After the wizard completion a dialog appears which is used for setting ODBC driver parameters.

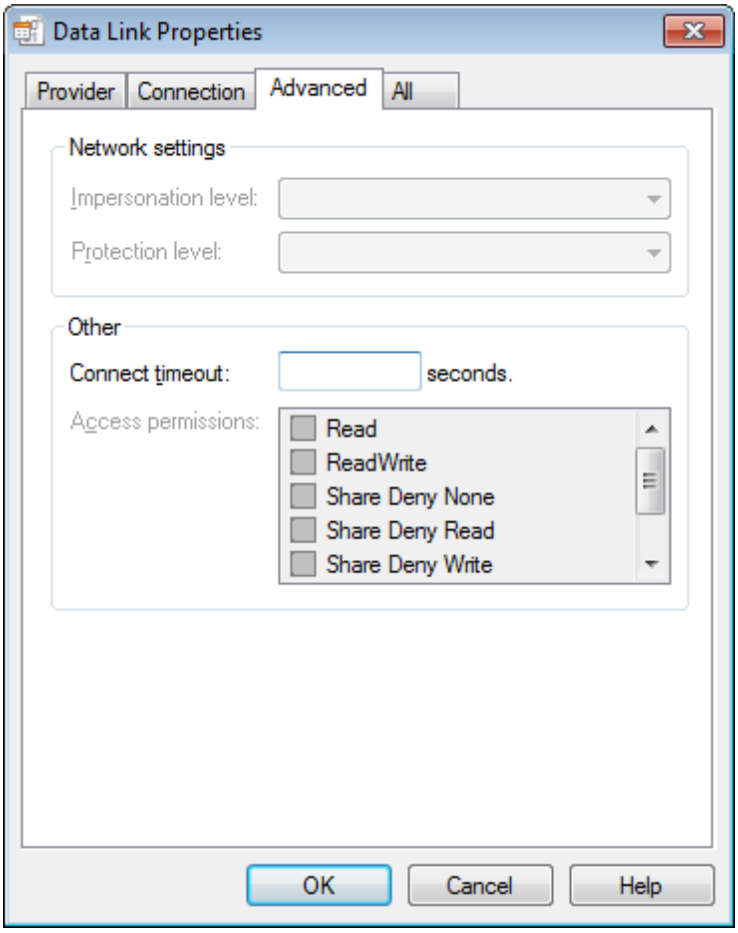


The set of options provided in the dialog box depends on the ODBC driver you have created.

2.1.2.2.3 Advanced

Within this tab you can specify the **advanced connection settings**: *impersonation level*, *protection level*, *connection timeout* and *access permissions*.

Note: The set of available **Advanced** options depends on the selected OLE DB Provider.



To view the summary for the current data source configuration, proceed to the **All** tab.

See also:
Provider
Connection
All

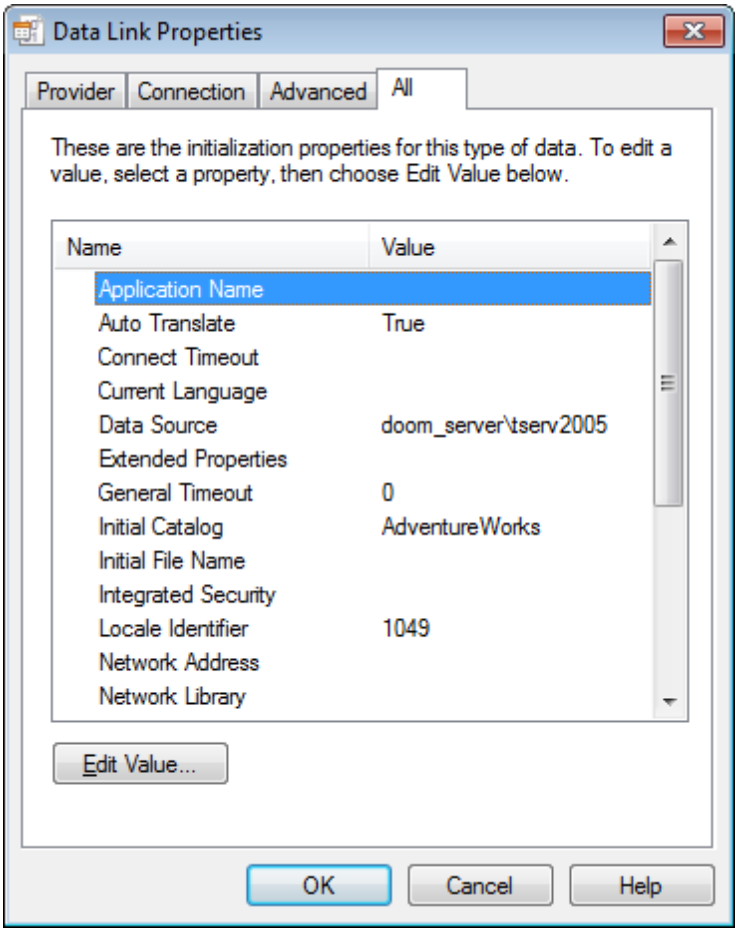
2.1.2.2.4 All

The **All** tab contains the initialization properties grid in which the properties specified within the **Data Link Properties** dialog are listed. If necessary, you can edit property values or reset them within the **Edit Property Value** dialog.

In order to set an option:

- select the option in the grid;
- click the **Edit Value...** button;
- specify the property value in the **Edit Property Value** dialog and click **OK** to apply the changes.

Note: The set of options available within the **All** tab depends on the selected OLE DB Provider.



When you are done, press **OK** to apply the changes and proceed to setting target database connection properties.

See also:

Provider

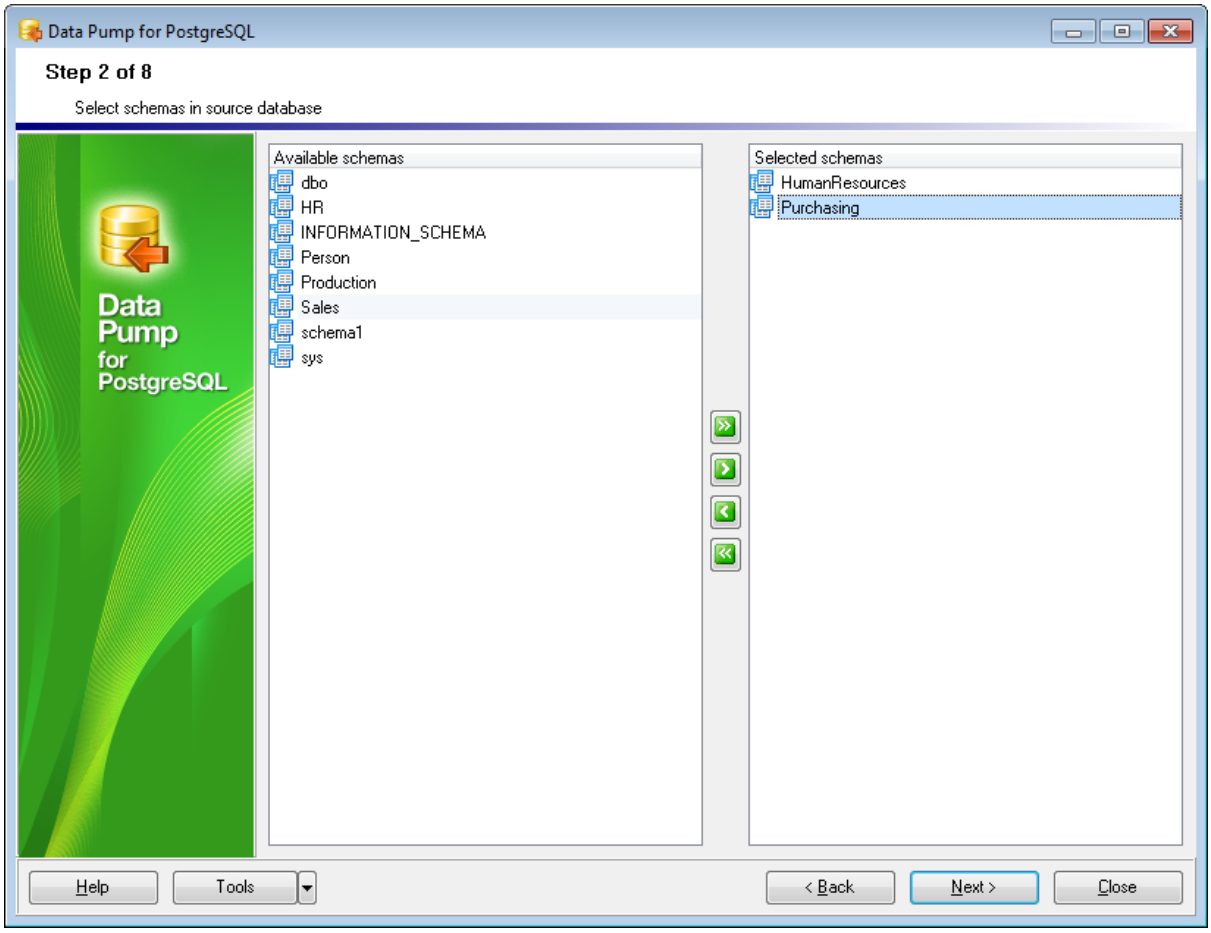
Connection

Advanced

2.1.3 Step 2 - Selecting source schemas

This step of the wizard allows you to specify schemas of the source database to pump objects from.
To select a schema, you need to move it from the **Available schemas** list to the **Selected schemas** list. Use the  buttons or drag-and-drop operations to move the schemas from one list to another.

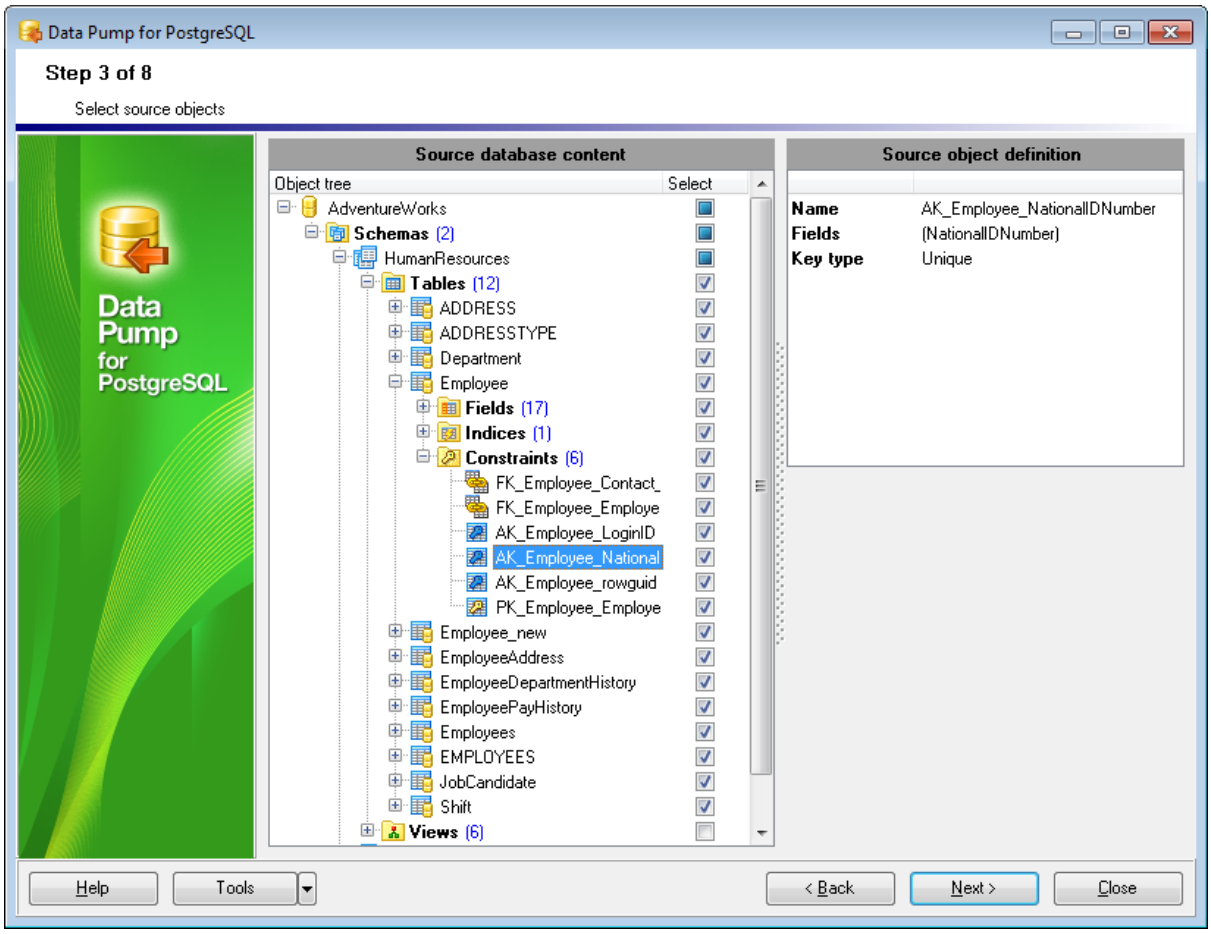
Hint: To select multiple schemas, hold down the *Shift* or *Ctrl* key while selecting the schema names.



Click the **Next** button to proceed to the selecting source objects step of the wizard.

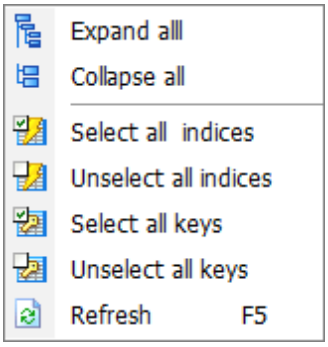
2.1.4 Step 3 - Selecting source objects

At this step you can view the tree of source objects in the **Source database content** area and select them for pumping using the corresponding check boxes.
The **Source object definition** area displays definition for a currently selected source database object.



Note: In the target database all selected views will be converted to tables.

The **context menu** of the **Source database content** area can be used to select/unselect table subobjects and to expand/collapse the tree nodes. To call the context menu, right-click any node in the *Source database content* area.



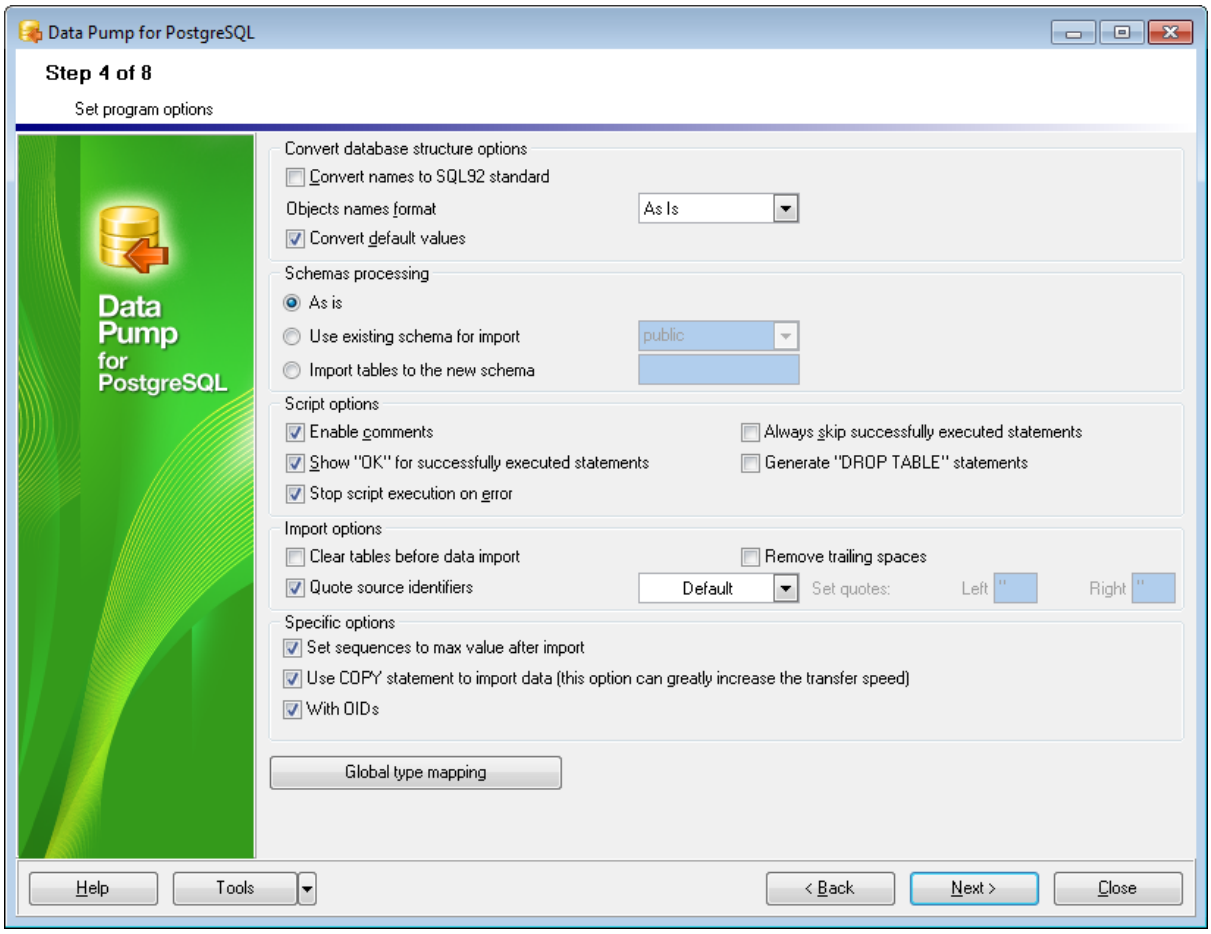
The **context menu** of the **Source database content** area allows you to:

 expand all nodes of the tree
 collapse all nodes of the tree
 select all indexes for pumping
 unselect all indexes
 select all keys for pumping
 unselect all keys
 refresh the tree

Click the **Next** button to proceed to the setting pump options step of the wizard.

2.1.5 Step 4 - Setting options

At this step of the wizard you can define pump options: *database structure* conversion options, options pertaining to *schemas processing*, *SQL script* execution, *data import* preferences, *server-specific options*, and perform source-to-target *type mapping*.



Convert database structure options

This group of options allows you to customize database structure conversion process.

Convert names to SQL92 standard

This option brings all the database object names to conformity with SQL92 naming rules.

Object names format

The drop-down list allows you to specify whether object names are to be converted to upper/lower case or left without changes (*As Is*).

Convert default values

Check this option to convert the default values of the source database to default values for the destination database.

Schemas processing

This group of options allows you to specify how PostgreSQL schemas will be processed by the application.

As is
If this option is selected, the tables will be placed into the default schema in the target database.
Use existing schema for import
If this option is selected, the tables will be placed into an existing schema. Use the drop-down list to select the schema in the PostgreSQL database where the source tables are to be pumped into.
Import tables to the new schema
If this option is selected, a new schema will be created and the tables will be placed into this schema. Use the edit-box to enter a name for the new schema.

Script options
This group of options allows you to customize the SQL script generation and its execution process.

Enable comments
If this option is selected, comments will be included into the body of the script.

Show "OK" for successfully executed statements
If this option is selected, records for successfully executed statements will be listed in the **Script execution information** area at Step 6.

Always skip successfully executed statements
If this option is selected, all successfully executed statements will be skipped upon subsequent script execution.

Generate "DROP TABLE" statements
Select the option to add the *DROP* statements for tables in the script.

Stop script execution on error
If this option is checked, the script execution will be stopped if an error occurs.

Import options
This group of options allows you to customize data import process.

Clear tables before import data
Select this option to empty the tables in the PostgreSQL database before data import at Step 8.

Quote source identifiers
This option enables/disables quoting source object identifiers.
If enabled, use the drop-down list to select the preferable value:
- *Default* to use the default quotes for the data source being used;
- one of available quoting patterns: *"..."*, *'...'*, *`...`*, *[...]*, *(...)*, *{...}*, *<...>*;
- *Custom* to specify any other characters to be used as left/right quotes.

Remove trailing spaces
If this option is checked, unused space at the end of source data strings will be cut off.

Specific options
This group allows you to define server-specific options.

Set sequences to max value after import
Use this option to enforce the maximum value for the sequences after the import operation is complete.

Use COPY statement to import data
Enables PostgreSQL COPY statement instead of INSERT statement. This option can be used to speed up data import process.

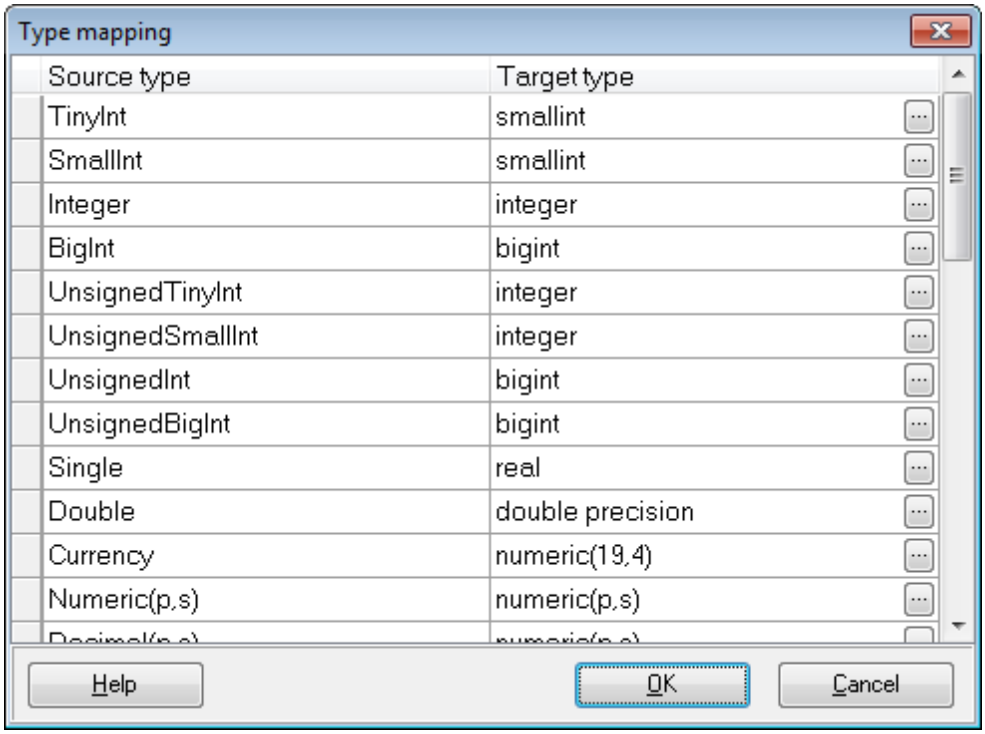
With OIDs
Enable this option to use OIDs (object identifiers) for pumped objects.

Press the **Global type mapping** button to open the **Type mapping** dialog allowing you to view and edit source-target data type correspondences.

When you are done, click the **Next** button to proceed to editing target objects.

2.1.5.1 Global type mapping

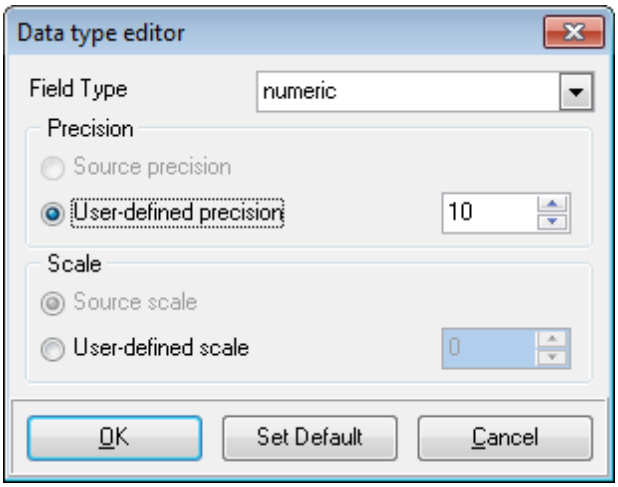
In this dialog you can view and edit the type conversion grid where the correspondence between the data types of the OLE DB Provider being used and the target database server (i.e. PostgreSQL) data types is displayed. You can use this grid to set conversion preferences for each of the types.



The **Data type editor** dialog can be opened by clicking the ellipsis button to the right of the record to be edited.

In the **Data type editor** dialog you can specify whether the source field type returned by the OLE DB Provider is to be converted to a data type of the target database server (i.e. PostgreSQL).

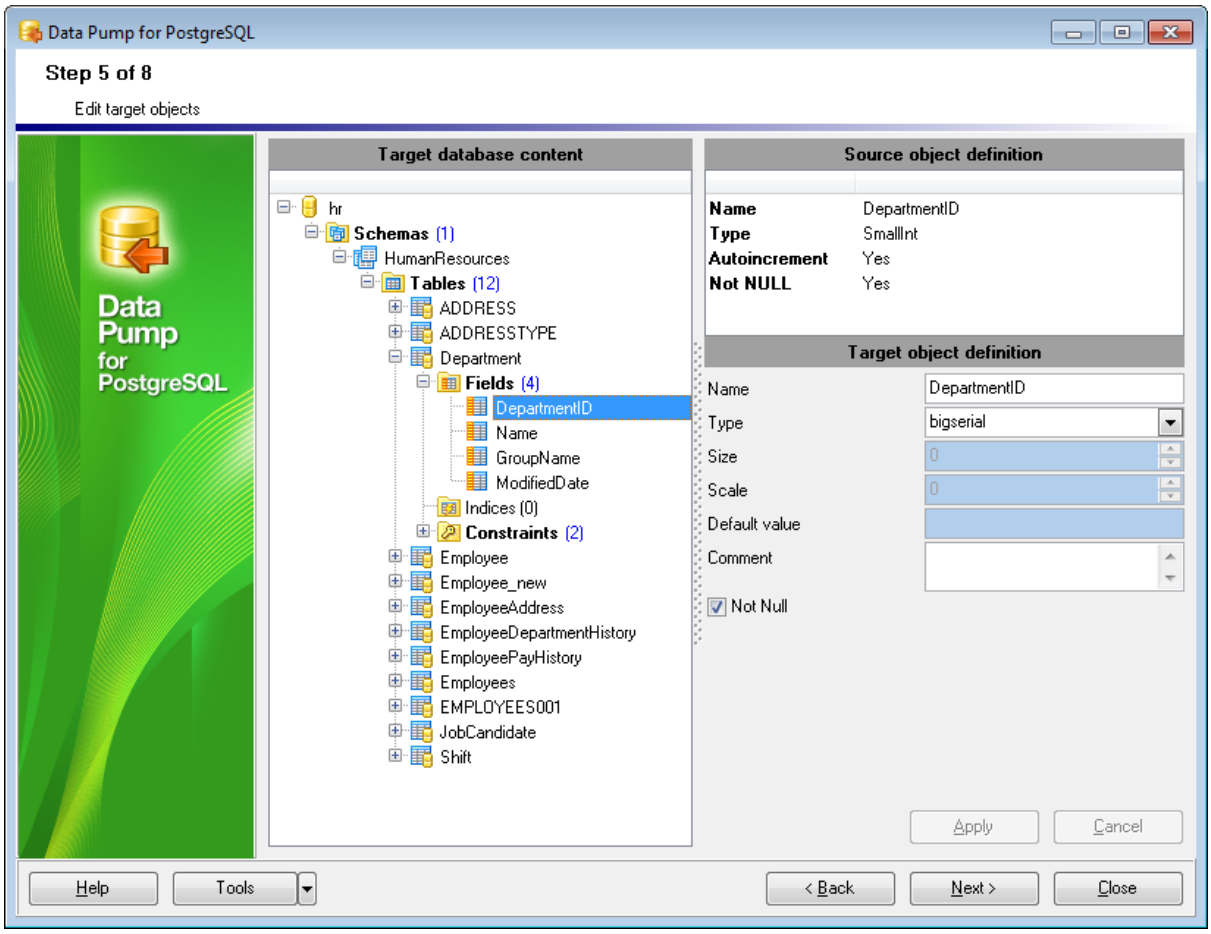
For certain data types you can set **scale**, **size** and/or **precision**.



[<< Back to setting options]

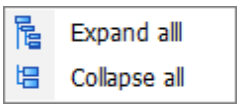
2.1.6 Step 5 - Editing target objects

At this step of the wizard you can browse the estimated content of the *target database* in a tree of objects, view definitions of the corresponding *source objects* and edit the *target object definitions*.



The tree of the target PostgreSQL database objects is displayed in the **Target database content** area.

The **context menu** of the **Target database content** area can be used to expand/ collapse the tree nodes. To call the context menu, right-click any node in the *Target database content* area.



The **context menu** of the **Source database content** area allows you to:
- expand all nodes of the tree
- collapse all nodes of the tree

Depending on the current selection in the *Target database content* area, the **Source object definition** area displays the properties of the corresponding source database

object.

The **Target object definition** area allows you to view and edit the target PostgreSQL database object properties:

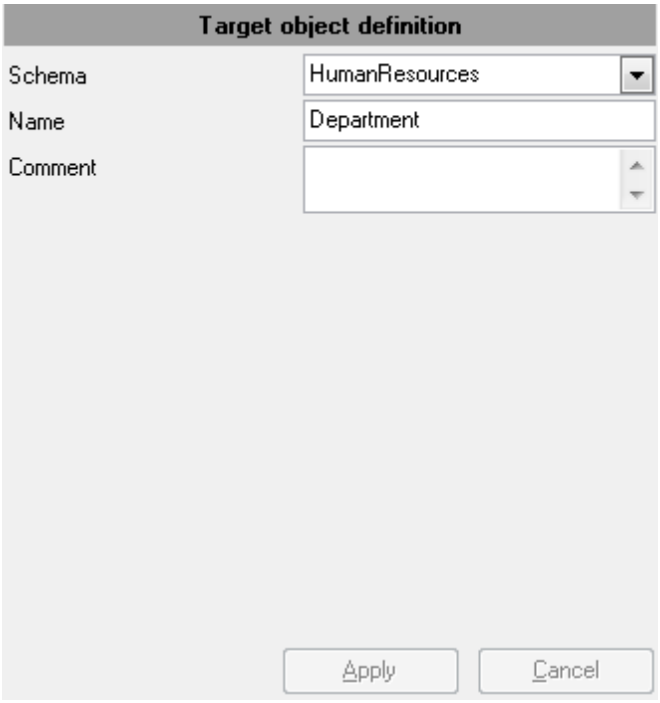
- for tables
- for table fields
- for table indexes
- for primary key / unique key / foreign key constraints

To apply and discard the changes made within the **Target object definition** area, use the **Apply** and the **Cancel** buttons respectively.

Click the **Next** button to proceed to Viewing/editing generated script.

2.1.6.1 Target table properties

The **Target object definition** area of Step 5 allows you to view and edit the target **table** properties:



Schema - use the drop-down list of schemas available in the PostgreSQL database to select the one in which the table is to be allocated.

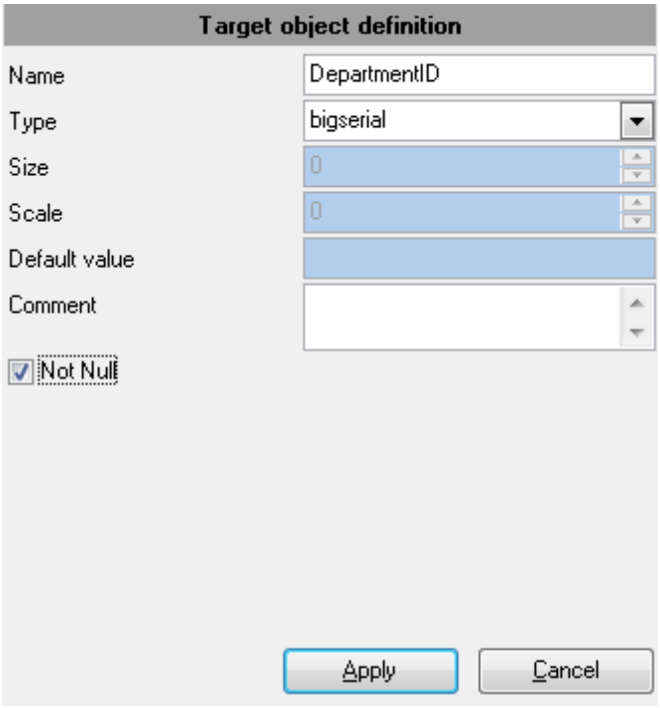
Name - specifies the table name.

Comment
If necessary, add a comment for the object.

[<< Back to editing target objects]

2.1.6.2 Target field properties

The **Target object definition** area of Step 5 allows you to view and edit the target **field** properties:



Name - specifies the field name

Type - use the drop-down list of data types supported by PostgreSQL to select the one you wish to be applied to the currently selected field

Size - set the size of the field (for string types)

Scale - specify the scale for the field values (if available for the selected data type)

Default value - enter the value that inserted records will get by default (if necessary)

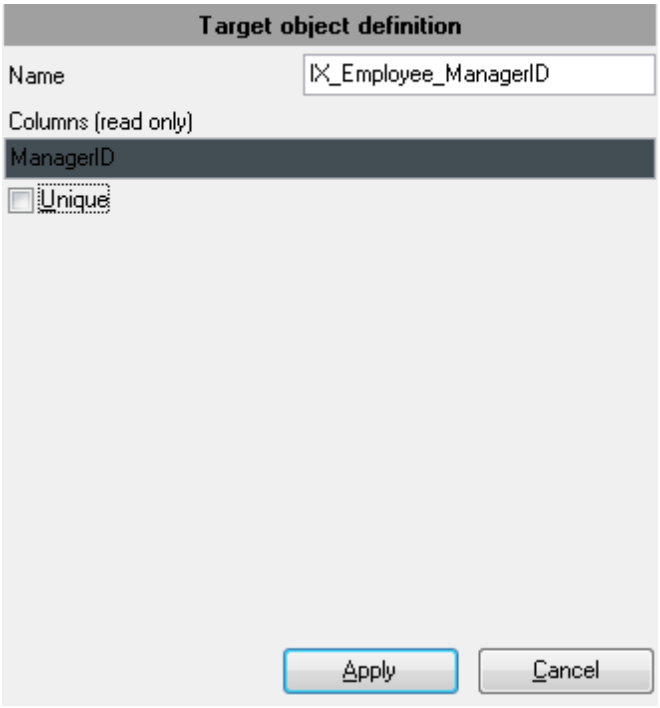
Comment
If necessary, add a comment for the object.

☑ **Not NULL** - determines nullability for the field (whether the field is **NULL** / **NOT NULL**)

<< Back to editing target objects

2.1.6.3 Target index properties

The **Target object definition** area of Step 5 allows you to view and edit the target **index** properties:



Name - specifies the index name

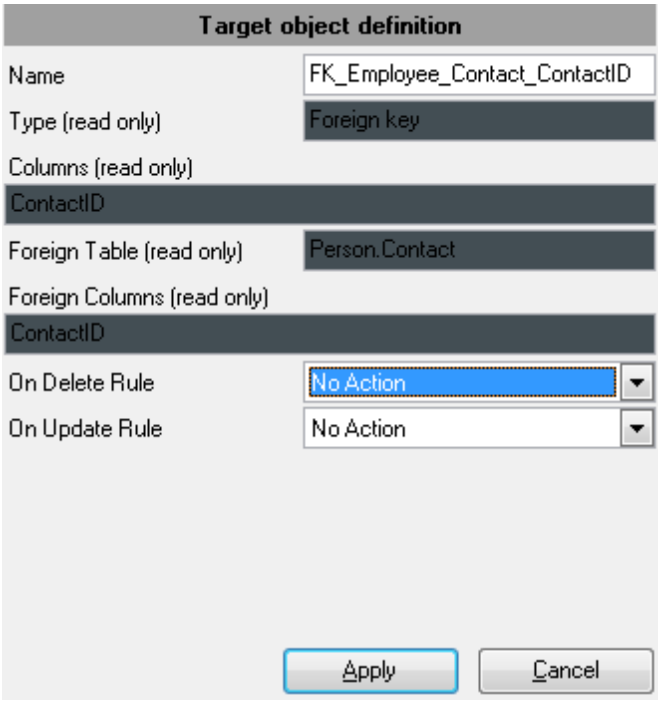
Columns (read only) - view the column(s) used by the index

Unique - determines uniqueness of the index, causes the system to check for duplicate values in the table each time data are added

<< Back to editing target objects

2.1.6.4 Target key properties

The **Target object definition** area of Step 5 allows you to view and edit the target **Primary key** / **Unique key** / **Foreign key** properties:



Name - specifies the key name

Type (read only) - view the constraint type: *Primary key*, *Unique key*, *Foreign key*

Columns (read only) - view the column(s) used by the constraint

Foreign Table (read only) - view the name of the foreign table

Foreign Columns (read only) - view the foreign columns (for *Foreign key*)

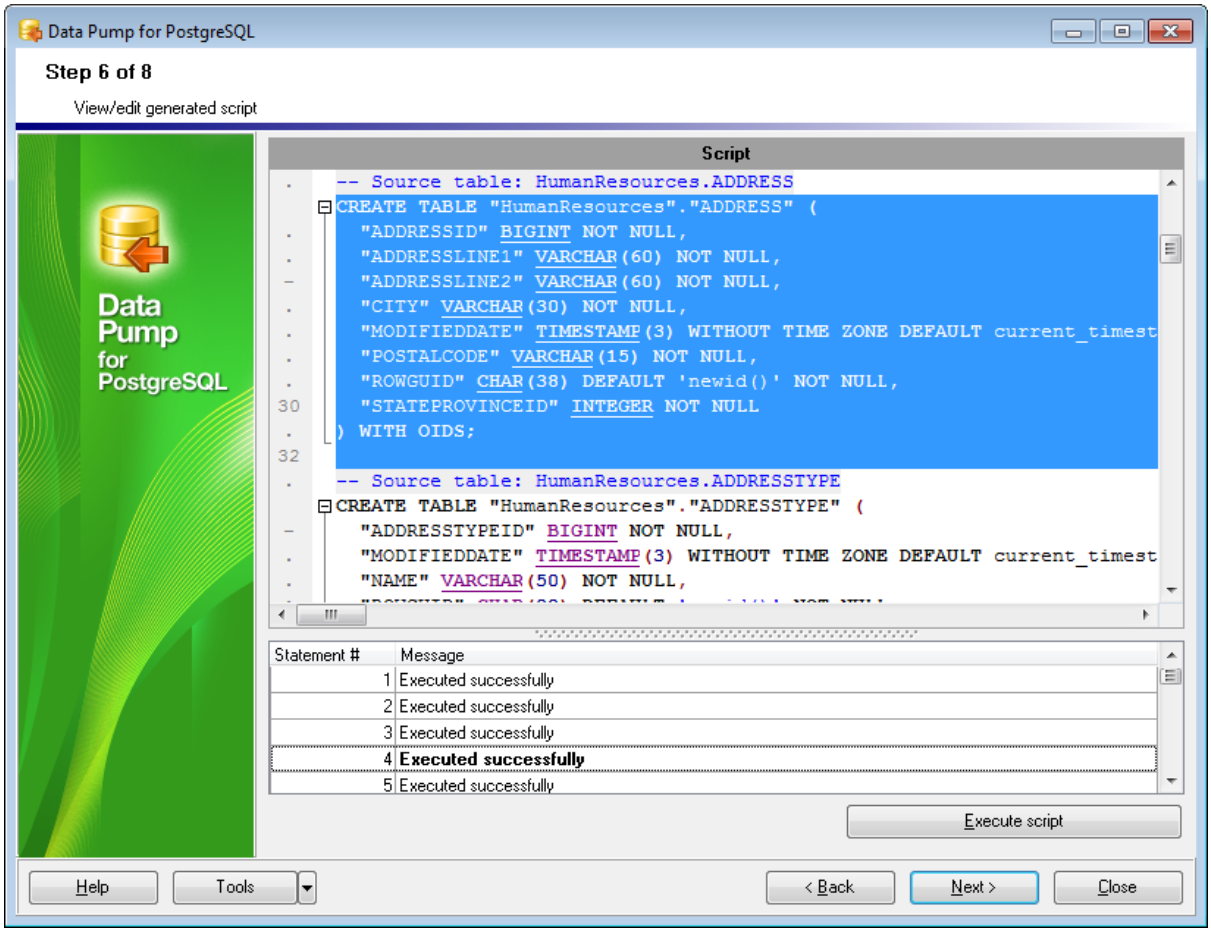
ON DELETE rule - specifies the ON DELETE rule: *NO ACTION*, *RESTRICT*, *CASCADE*, *SET NULL*, *SET DEFAULT* (for *Foreign key*)

ON UPDATE rule - specifies the ON UPDATE rule: *NO ACTION*, *RESTRICT*, *CASCADE*, *SET NULL*, *SET DEFAULT* (for *Foreign key*)

<< Back to editing target objects

2.1.7 Step 6 - Viewing/editing generated script

At this step you can view, edit and execute the generated SQL script.



The **Script** editor area allows you to view, edit and save the script for future use, if necessary.

Right-click within the **Script** editor area to call the context menu allowing you to perform a number of operations with the script text.

Use the **Execute Script** button to perform the entire script execution.

The **Script execution information** area contains the list of records for executed SQL statements and status messages.
When a record is selected in the list, the corresponding statement is highlighted in the **Script** area for your convenience. You can also use the context menu of the grid to perform a number of operations with the statements.

When you are done, click the **Next** button to proceed to selecting tables for data import.

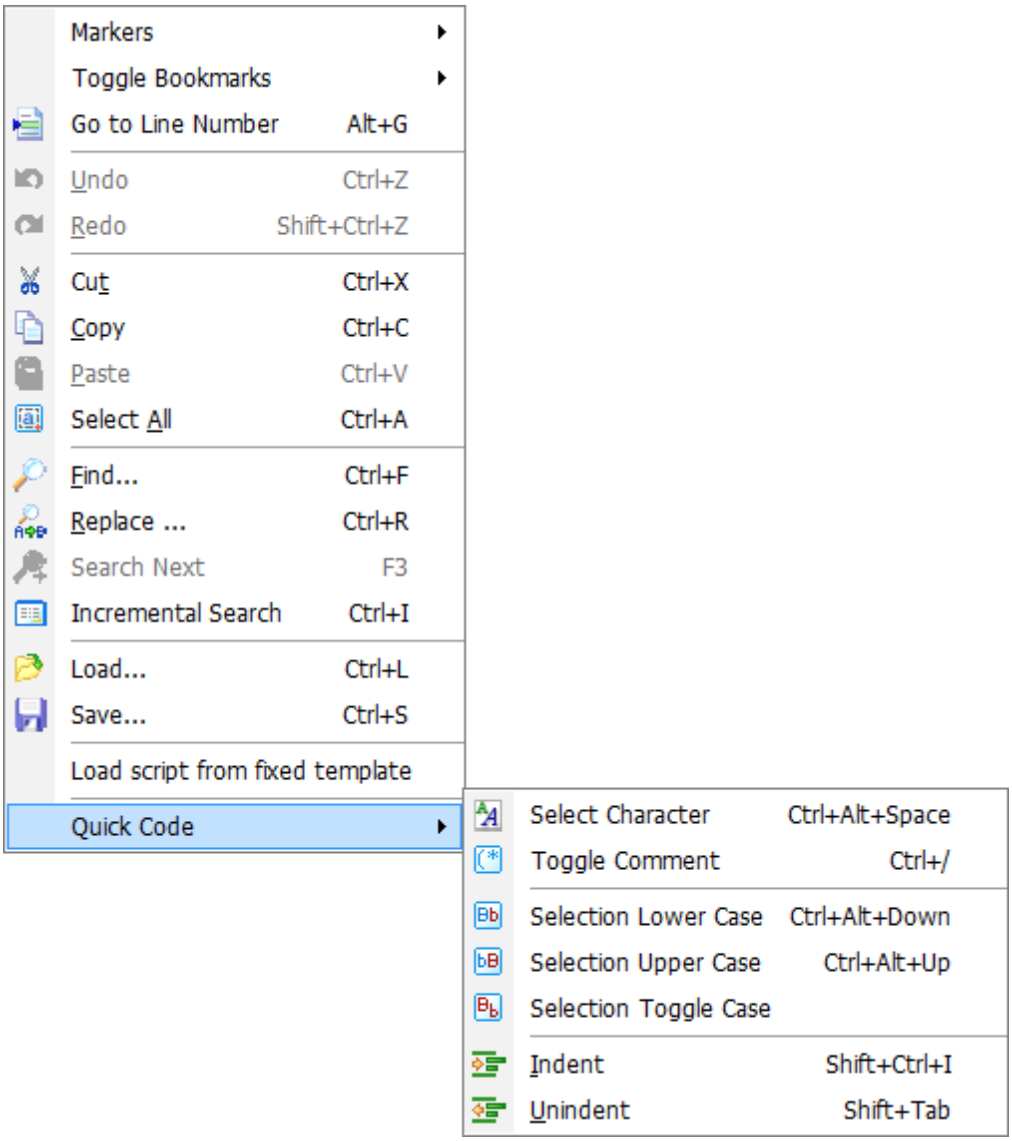
2.1.7.1 Using context menus

The **context menus** available at Step 6 provide additional functionality for working with the script. Find the description of context menu items below.

The **context menu** of the **Script** editor area contains most of the standard text-processing functions (*Cut*, *Copy*, *Paste*, *Select All*) and functions for working with the script as a whole, e.g. you can set a *marker*, *move the cursor to a particular line*, *change case* of selected text, etc. Most of these operations can be also performed with the corresponding hot keys used.
Implementation of the Find Text / Replace Text dialogs and **Incremental search** contributes to more efficient work with the SQL code.

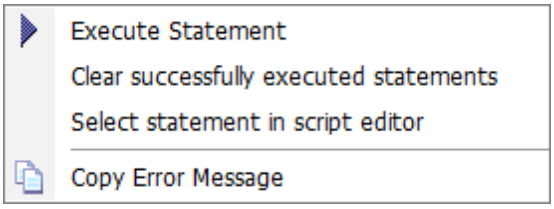
The context menu allows you to:

- manage markers: *Drop Marker*, *Collect Marker*, *Swap Marker*;
- toggle bookmarks allowing you to navigate through the query text and jump to a line with a particular number;
- perform editing operations: *Undo/Redo*, *Cut*, *Copy*, *Paste*, *Select all*;
- perform search and replace operations;
- save/load a script to/from an external **.sql* file;
- load SQL script from a *fixed template* (for details see Save template options);
- use the *Quick Code* group allowing you to toggle comments for code fragments, change case of the selected text, indent/unindent code lines.



You can also use the **context menu** of the **Script execution information** grid to:

- *execute* the currently selected SQL statement;
- *clear successfully executed statements* from the Script execution information list;
- *select a statement in script editor*;
- *copy the error message* (if any) to the clipboard.

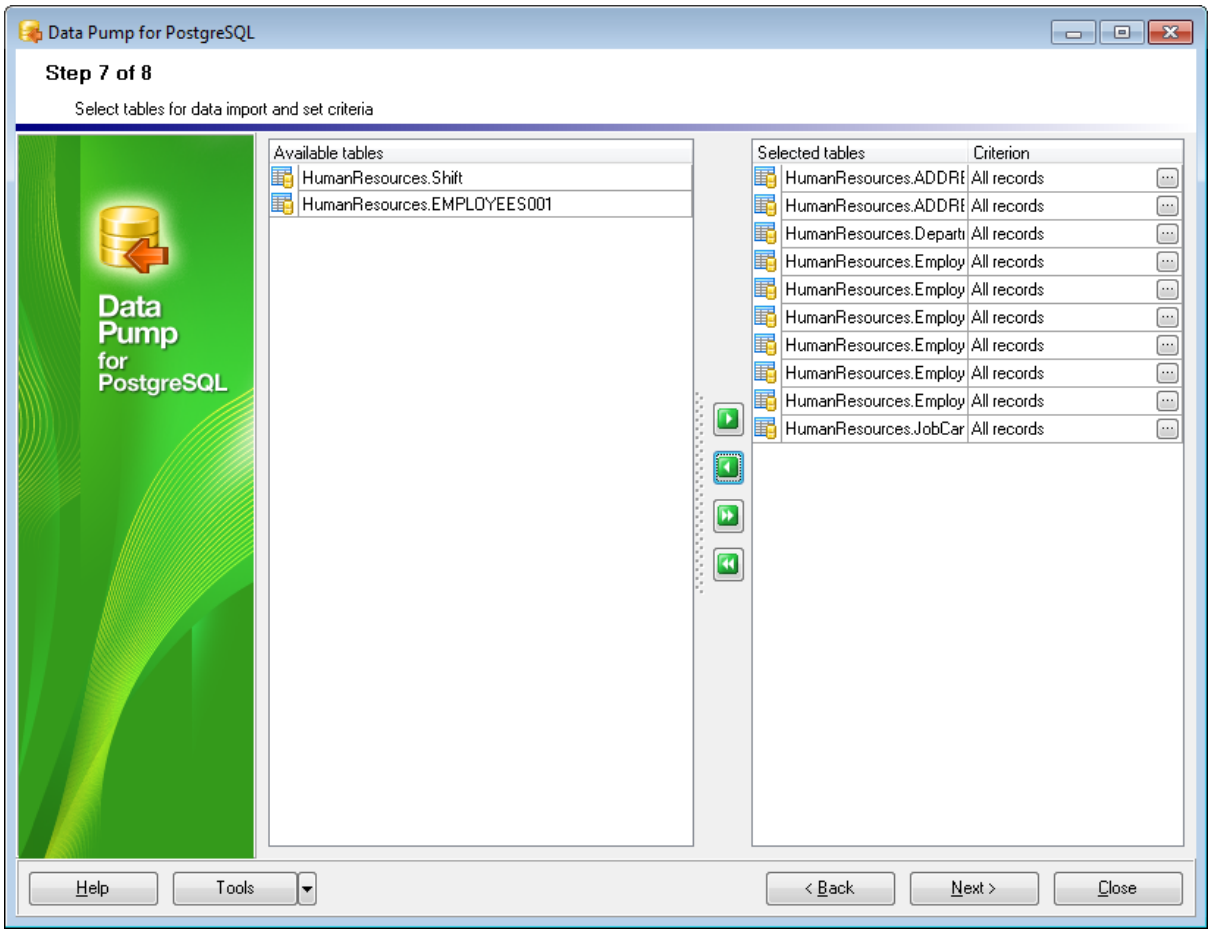


[<< Back to viewing/editing generated script]

2.1.8 Step 7 - Selecting tables for data import

At this step of the wizard you can specify tables to import data from.
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.

Hint: To select multiple tables, hold down the *Shift* or *Ctrl* key while selecting the table names.



You can also set SQL criteria for fetching data from the corresponding source table with a *WHERE* clause used within the Criterion Editor dialog.

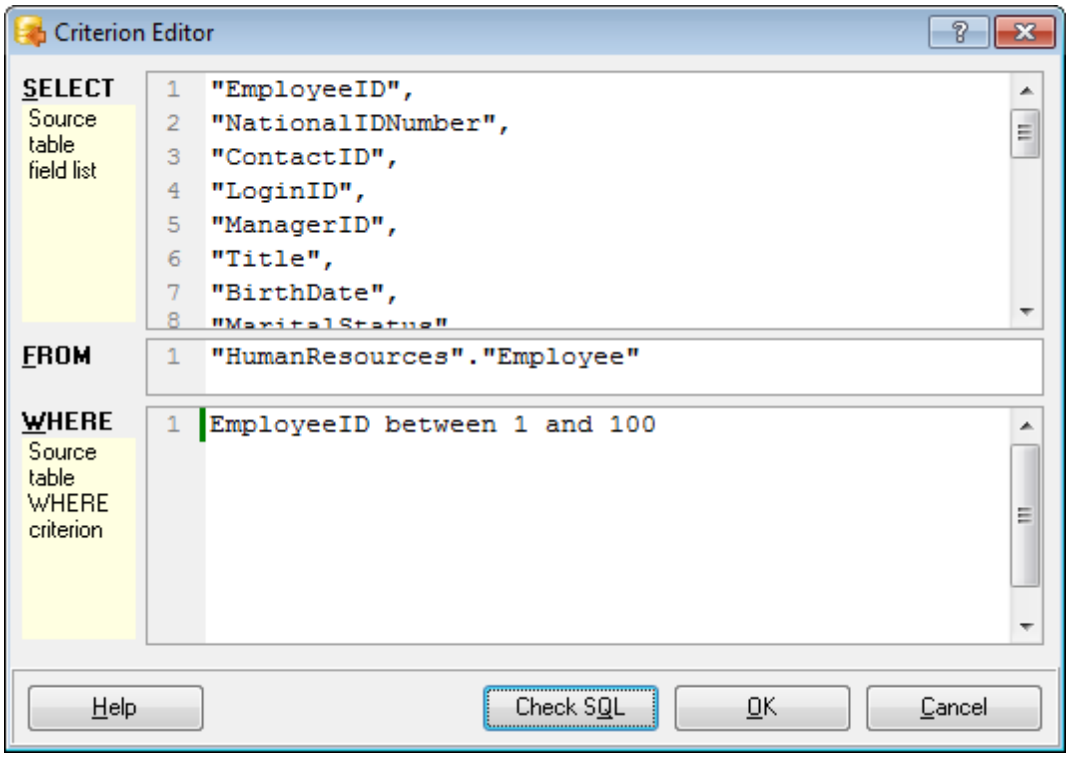
To open **Criterion Editor**, use the ellipsis button next to the *Criterion* value.

Note that the **Available tables** list only contains tables for which structure has been already created (e.g. at the previous step), i.e. you can only choose to import data to those tables which already exist in the target PostgreSQL database.

When you are done, click the **Next** button to proceed to importing data.

2.1.8.1 Criterion Editor

Criterion Editor allows you to edit the *SELECT* query for the source table. The record set returned upon this query execution will be imported to the target PostgreSQL table at Step 8.



The **Criterion Editor** dialog contains three editing areas, each corresponding to a specific clause of the *SELECT* statement:

- *SELECT <source table field list>*
- *FROM <table name>*
- *WHERE <source table WHERE criterion>*

A number of features for efficient SQL viewing and editing are implemented. For more information see Viewing/editing SQL script and Using script area context menu.

Use the **WHERE <source table WHERE criterion>** area of **Criterion Editor** to set a SQL criterion (e.g. *"SALARY" BETWEEN 10000 AND 100000*) for fetched source data that will be imported to the target PostgreSQL table.

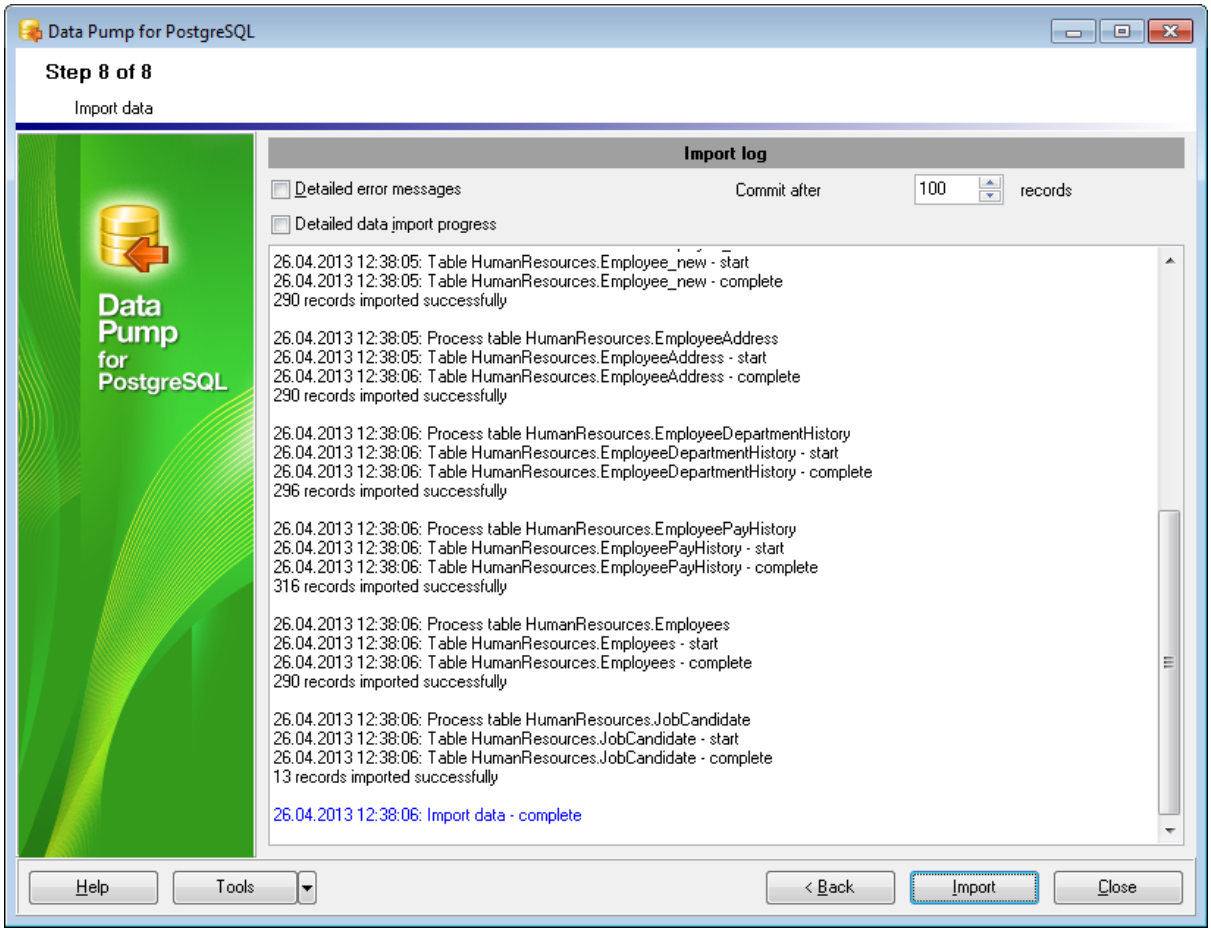
Hint: You can also **check SQL** for your statement to make sure that the syntax is correct, and to see how many records are fetched by the query.

Click **OK** to apply the criterion for the selected table or **Cancel** to exit the editor and return to Step 6.

<< Back to selecting tables for data import

2.1.9 Step 8 - Importing data

This step of the wizard allows you to start importing data and view the log of the data import process.



Detailed error messages
Use this option to toggle detailed/simple error messages in the import log.

Commit after ... records
The spin-edit control allows you to define the number of records after which the *COMMIT* statement will be inserted.

Detailed data import process
Use this option to toggle detailed/simple data import progress.

Click the **<Back** button if you need to change any settings before importing data.
Click the **Import** button to complete the importing process.

The **context menu** of the **Import log** area allows you to **save** the log to an external file.

Note: If necessary, you can use the **Tools** button to call a menu allowing you to **restart the wizard**.

Do not forget to save data pump template if you need to repeat the process with the same (or similar) settings later.

2.2 Using Configuration Files (Templates)

Data Pump for PostgreSQL allows you to store data conversion settings in external **.dpc* files (**templates**) which can be used to repeat data pump process as many times as you need.

You can load previously saved templates to the wizard application if you need to make any changes before pumping data, or you can run it with the console application for faster processing.

- Saving templates
- Loading templates

See also:

Using Wizard Application

Setting Program Preferences

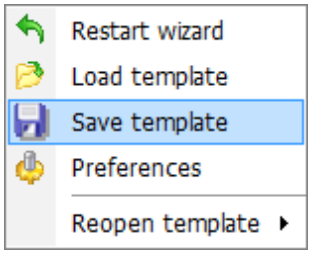
Configuration file format

2.2.1 Saving templates

Having configured the pumping process according to your needs using Steps 1-8 of **Data Pump** wizard, you can **save** all the settings into a **.dpc* **template** file at the final Importing data step.

In order to save a template:

- press the **Tools** button;
- select the **Save template** item from the popup menu;
- configure your template using the Save template options dialog.

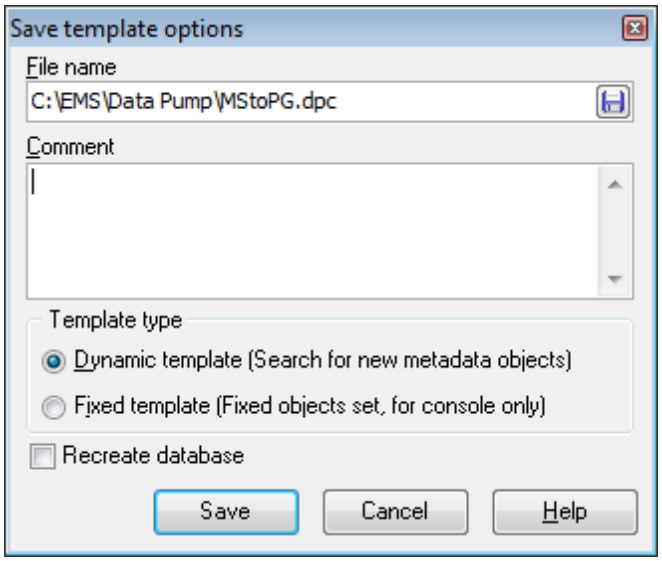


See also:

Loading templates

2.2.1.1 Save template options

The **Save template options** dialog allows you to configure your template according to your needs.



File name
Enter the path to the template file and its name, or press the button to browse for location using the **Save As** dialog.

Comment
If necessary, add a comment for the newly created template.

Template type
The radio options group allows you to specify the type of the template to be created.

Dynamic template
Stores settings and objects list inside; if there are any new objects in the source database, they will be automatically added to the source objects list.

Fixed template
In this type of template the set of objects is fixed. This selection might be useful if you frequently use the console version of the utility.

Recreate database
If this option is selected, the console version of **Data Pump** will drop the existing database, then create a new one and pump the source database into the newly created database.

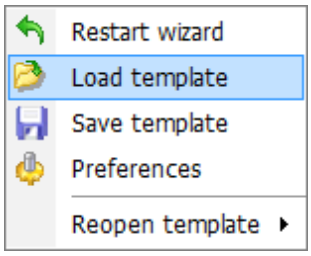
<< Back to saving templates

2.2.2 Loading templates

If you have a saved **Data Pump** template, you can **load** it any time with all the settings specified at Steps 1-8 (from database connection properties to importing data) applied.

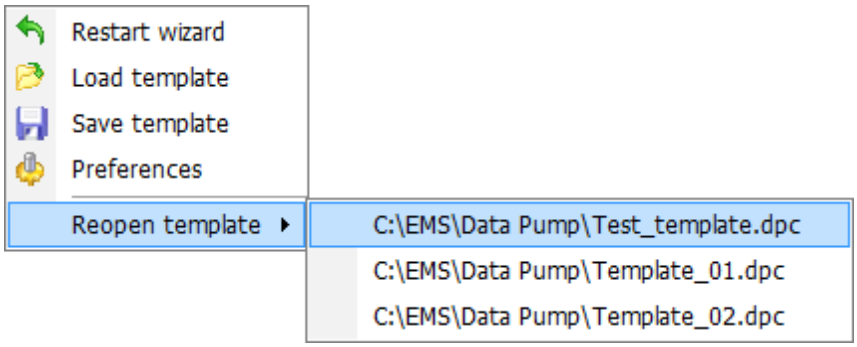
In order to load a template:
- press the **Tools** button;
- select the **Load template** item from the popup menu;
- use the **Open file** dialog to select the **Data Pump** template file (**.dpc*).

After loading a template you can automatically open any wizard step or immediately start the importing process (i.e. all the previous steps will be passed with the settings stored in the template), as specified in the General section of the Preferences dialog.



The **Tools** menu of **Data Pump for PostgreSQL** also allows you to load one of previously loaded templates quickly by using the **Reopen Template** function.

In order to reopen a template:
- press the **Tools** button;
- select the **Reopen template** item from the popup menu;
- select one of previously loaded template files from the submenu.



See also:
Saving templates

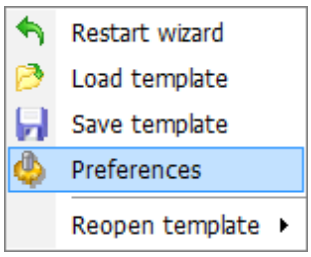
2.3 Setting Program Preferences

Data Pump for PostgreSQL provides full customization of the program interface by setting various options within the **Preferences** dialog. This chapter is intended to inform you how to use these options.

General options
These options define general behavior of **Data Pump for PostgreSQL**.

Language
This page allows you to select an interface language to be applied for your copy of **Data Pump for PostgreSQL**.

Interface
This branch contains several pages with a number of options allowing you to customize the application interface style according to your liking.



See also:
Using Wizard Application
Using Templates

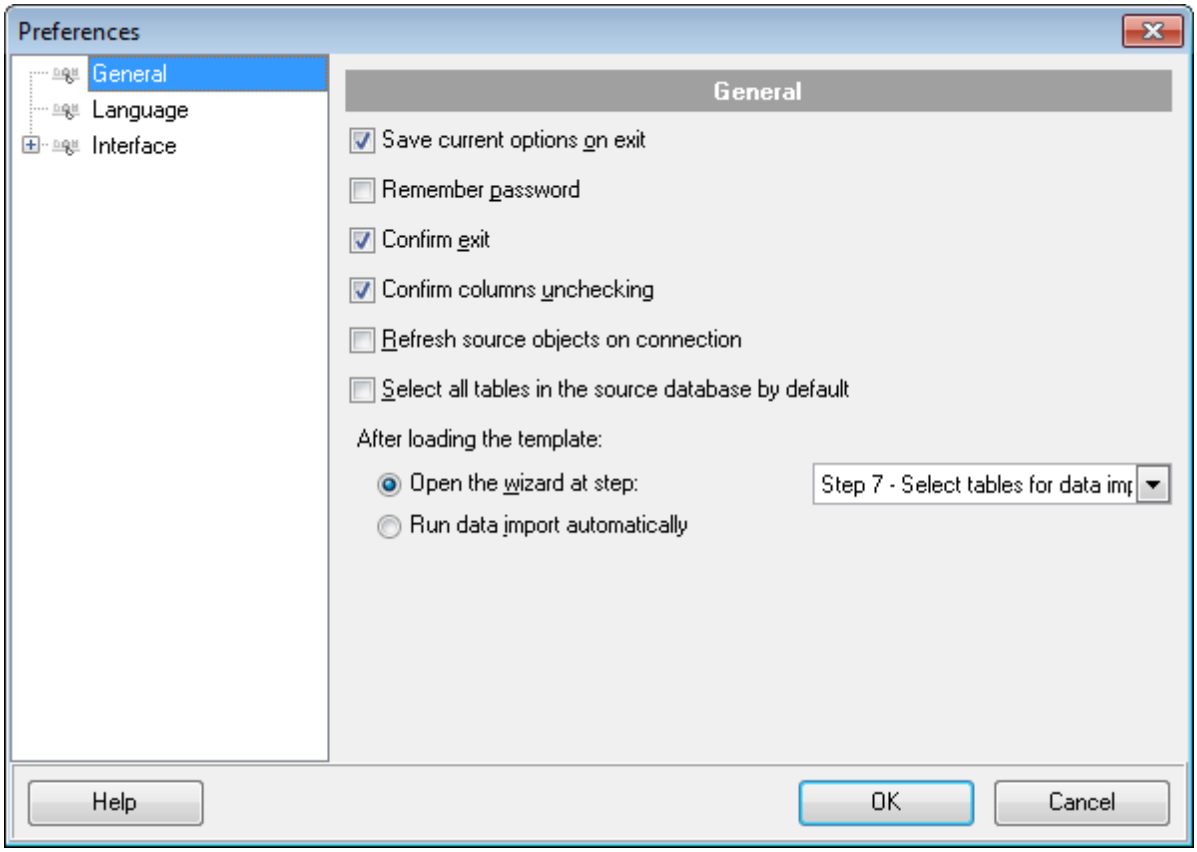
2.3.1 General

This page allows you to define general options of the application.

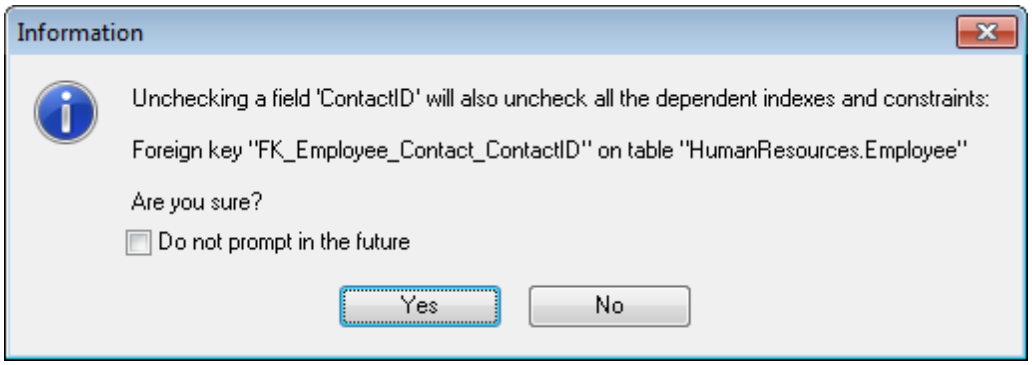
☑ **Save current options on exit**
If this option is checked, your current options will be saved upon exiting **Data Pump for PostgreSQL**.

☑ **Remember password**
If this option is checked, the password used for accessing PostgreSQL database will be saved, i.e. you will not have to enter it each time you run the utility.

☑ **Confirm exit**
If this option is checked, you will be prompted to confirm exiting the application every time you close **Data Pump for PostgreSQL**.



☑ **Confirm columns unchecking**
If you uncheck an object which has dependent objects at Step 3, you will be prompted to confirm this action (see the picture below).



☑ **Refresh source objects on connection**
This option specifies whether source objects are refreshed upon connection to the data source or not.

☑ **Select all tables in the source database by default**
This option specifies whether all source tables are selected at Step 3 by default.

After loading the template
This group allows you to define the action taken after loading a template:

◉ *Open the wizard at step*
If this option is selected, after loading a template the wizard will be opened at the specified step. Use the drop-down list to select the step to be opened.

◉ *Run data import automatically*
If this option is selected, after loading a template data import will be started immediately according to the template settings.

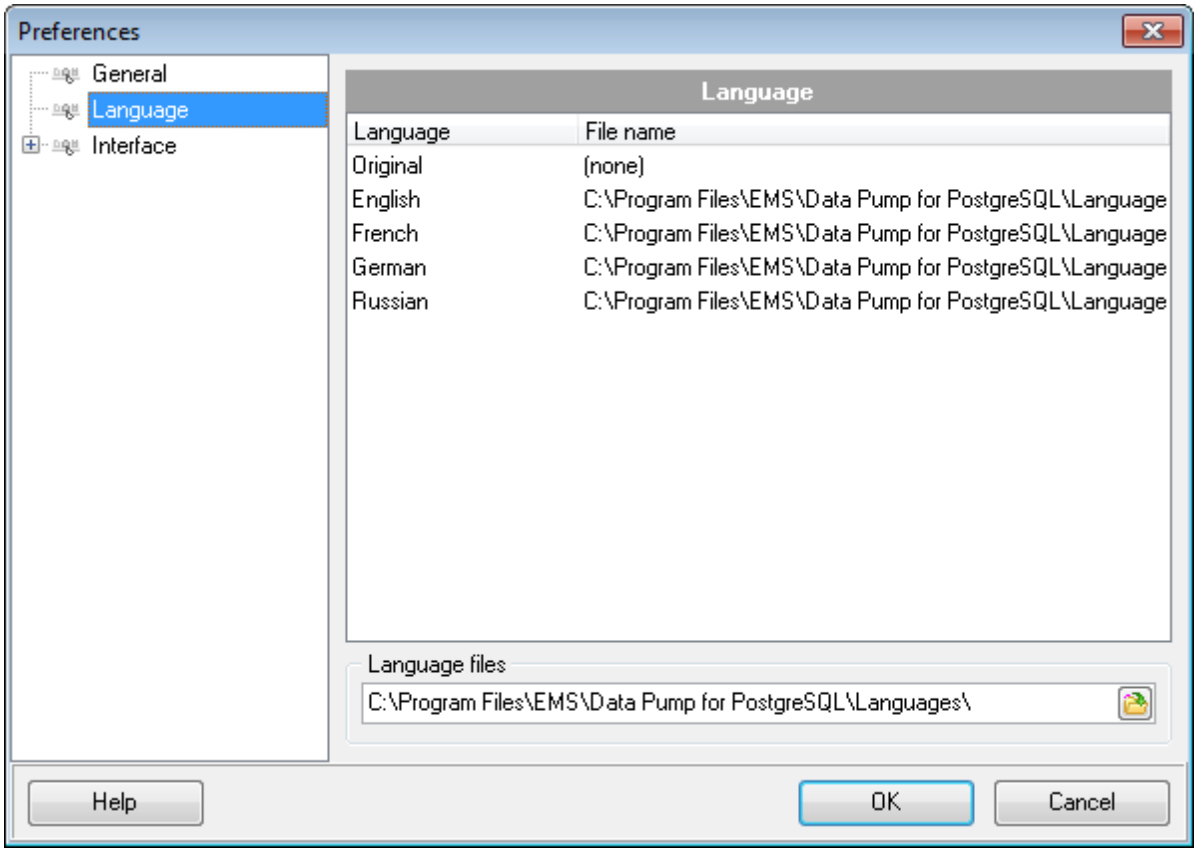
See also:

Language

Interface

2.3.2 Language

The **Language** page is provided for managing **Data Pump** localization files.



In the **Language** area the list of available languages and the names of the corresponding localization (**.lng*) files is displayed.

Language files

Use the button to specify the directory where the **.lng* files are stored by default.

See also:

General

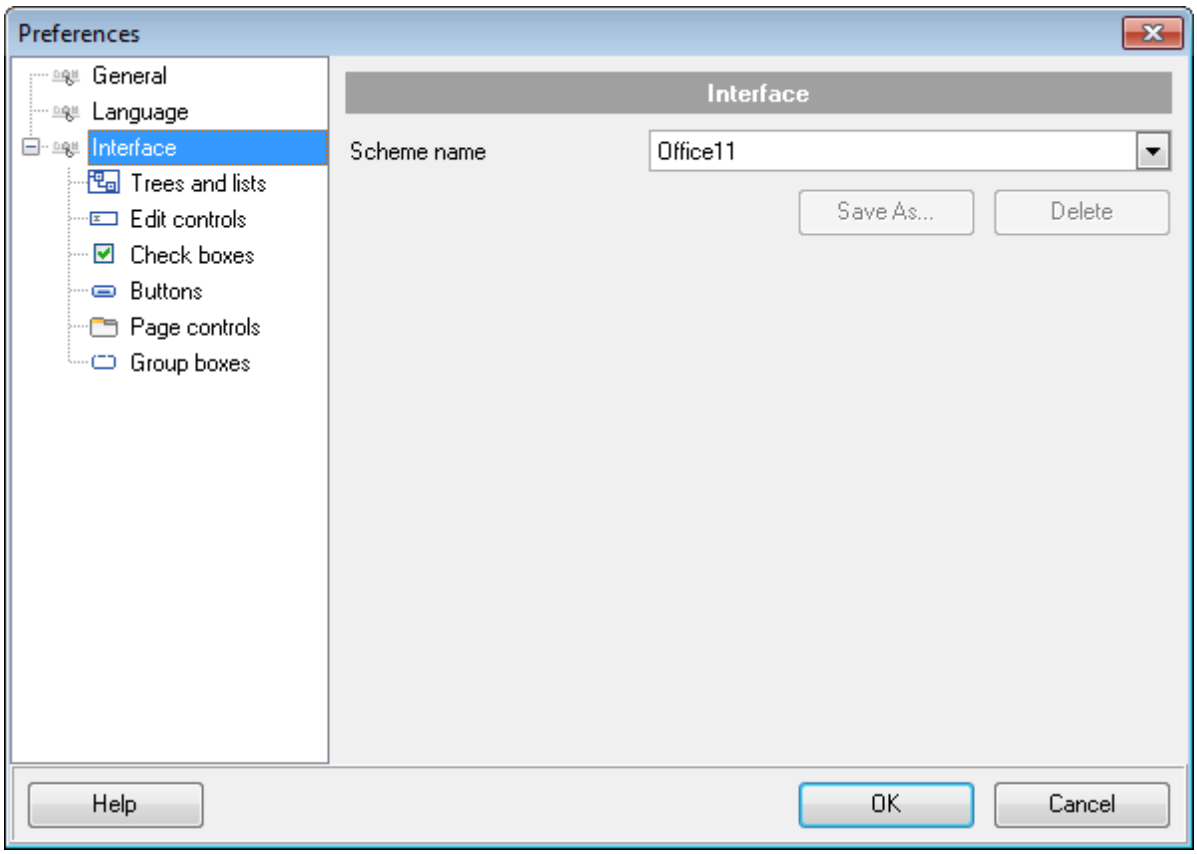
Interface

2.3.3 Interface

The **Interface** section of the **Preferences** dialog allows you to customize the application interface style according to your liking.

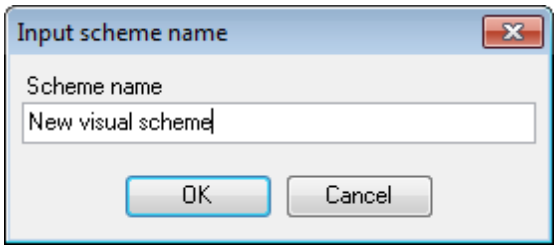
- Trees and lists
- Edit controls
- Check boxes
- Buttons
- Page controls
- Group boxes

Use the **Scheme name** drop-down list to select an interface scheme according to your liking: *Classic*, *Office XP* style, *Windows XP* native style, etc.



It is also possible to create one's own interface scheme, if necessary:

- set your preferences within the available branches of the **Interface** node (*Trees and Lists*, *Edit Controls*, *Check Boxes*, *Buttons*, *Group Boxes*);
- return to the **Interface** page and click the **Save As** button;
- specify the scheme name in the **Input scheme name** dialog.



Note: For your convenience the previews illustrating the changes are displayed in the **Sample** area of each branch of the **Interface** node.

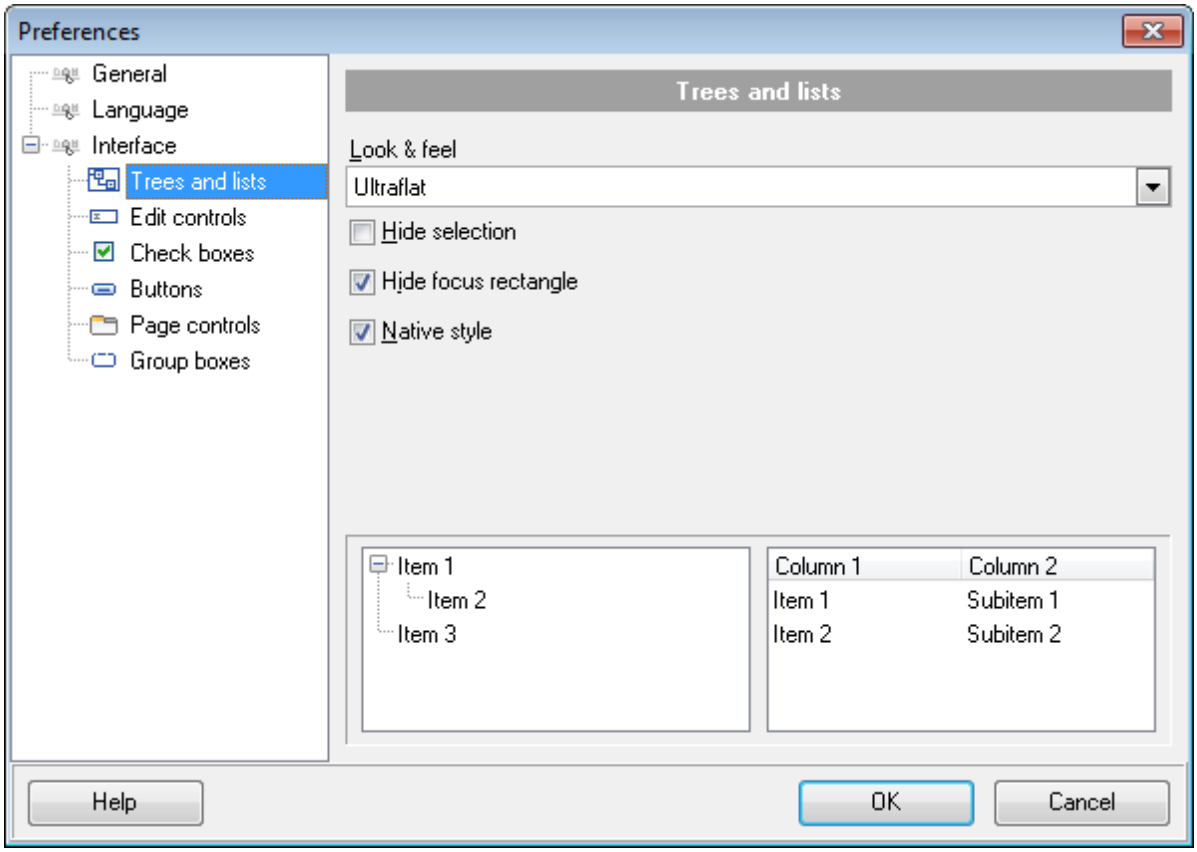
See also:

General

Language

2.3.3.1 Trees and lists

Use the **Trees and lists** section to view and edit the corresponding options.



Look & feel
This setting determines the manner in which tree and list elements are painted. Use the drop-down list to select the painting style that will be applied to the trees and lists:
Standard
Flat
UltraFlat

Hide selection
This option specifies how selected tree nodes and list items are displayed when focus leaves the tree or list control.
If this option is enabled, selected nodes look like other nodes. Otherwise, selected nodes/ items are highlighted within the tree/list.

Hide focus rectangle
This option determines whether a focus rectangle is displayed around the focused tree node or list item within the tree or list control.
If this option is disabled, the focused node/item is not highlighted but the focus rectangle is displayed around it.

Native style
This option determines whether the native Windows style will be applied to the trees and lists.

The option has the highest priority for trees and lists. If this option is selected, the tree nodes and list items are painted according to the native Windows style, regardless of other painting settings.

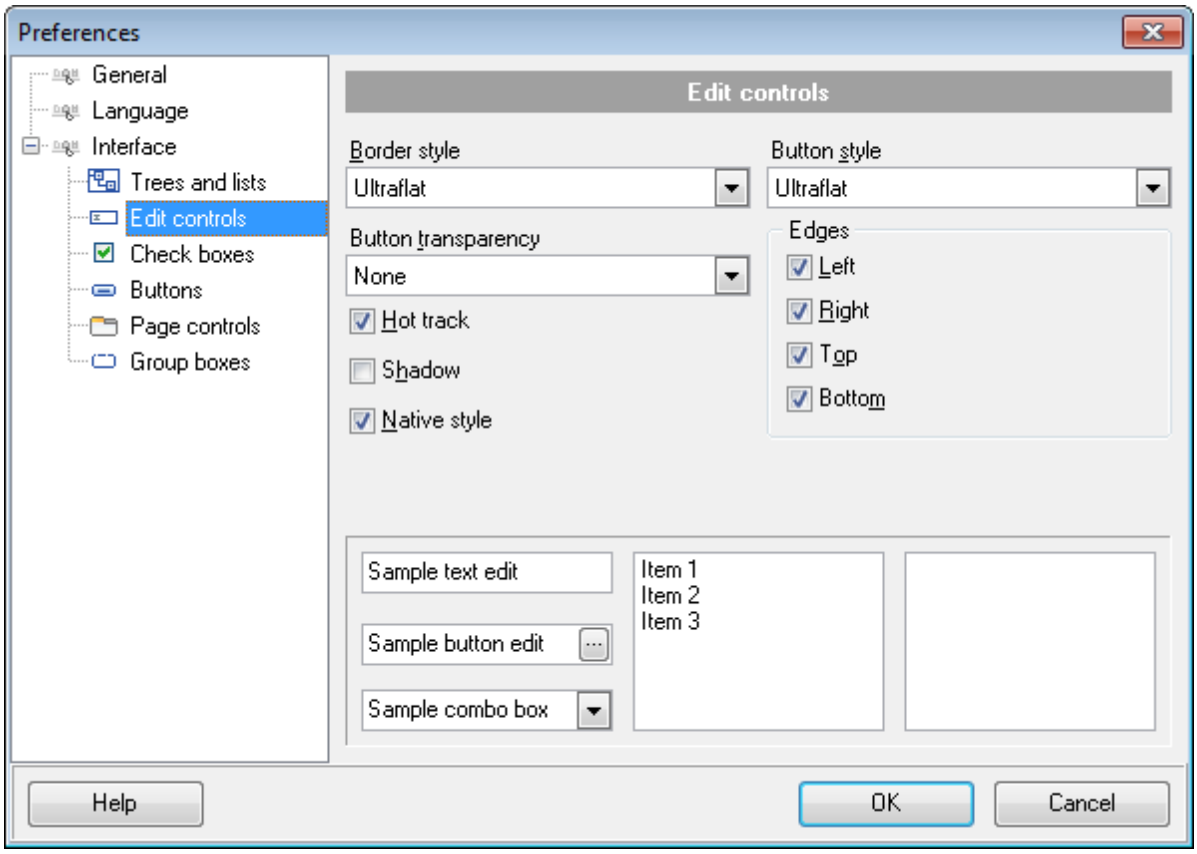
Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:

Edit controls
Check boxes
Buttons
Group boxes

2.3.3.2 Edit controls

Use the **Edit controls** section to customize the appearance of various **Data Pump for PostgreSQL** edit controls: *Border style*, *Button style*, *Button transparency*, etc.



Border style / **Button style**
Use these drop-down lists to specify the style around an editor (the edit control **borders**) and select the painting style that will be applied to the edit control **buttons** (ellipsis button, arrow-down combo-box button, etc.) respectively:
None
Single
Thick
Flat
3D
UltraFlat

Button transparency
Represents the button transparency mode within an editor. Use the drop-down list to specify the transparency that will be applied to the edit control **buttons** (ellipsis button, arrow-down combo-box button, etc.):
None (a button is always displayed in a non-transparent fashion)
Inactive (a button is drawn when the editor has focus or when the mouse cursor is positioned over the button; otherwise, the button is transparent)
Always (a button is always transparent)
Hide inactive (a button is drawn only when the editor has focus; otherwise, the button is invisible)

Hide unselected (a button is drawn when the editor has focus or when the mouse cursor is positioned over the editor region; otherwise, the button is invisible).

Edges
This group defines which edges are displayed within an editor. Check/uncheck the boxes to hide/show individual edges of the edit controls:
- *Left* (if unchecked, the left border edge is invisible)
- *Right* (if unchecked, the right border edge is invisible)
- *Top* (if unchecked, the top border edge is invisible)
- *Bottom* (if unchecked, the bottom border edge is invisible)

Hot track
This option specifies whether editor items are highlighted when the mouse cursor is positioned over an edit control. Select this option to highlight an edit control in response to mouse movements.

Shadow
If this option is selected, a shadow is displayed for the edit controls.

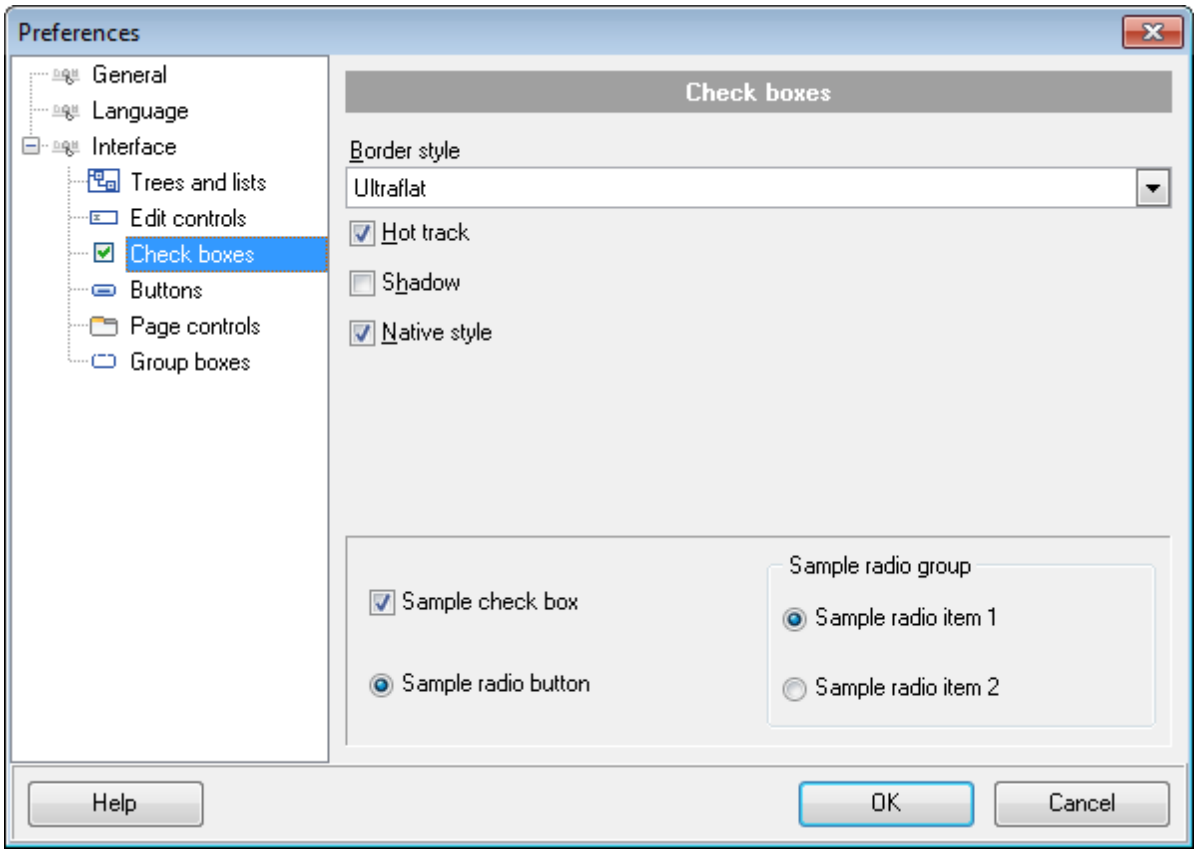
Native style
This option determines whether the native Windows style will be applied to the edit controls.
The option has the highest priority for edit controls. If this option is selected, the edit controls are painted according to the native Windows style, regardless of other painting settings.

Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:
Trees and lists
Check boxes
Buttons
Group boxes

2.3.3.3 Check boxes

Use the **Check boxes** section to customize the *border style* and the appearance of *check boxes* and *radio buttons*.



Border style
This setting determines the manner in which check box and radio group borders are painted. Use the drop-down list to select the painting style that will be applied to the check boxes and radio groups:
None
Single
Thick
Flat
3D
UltraFlat

Hot track
This option specifies whether check boxes are highlighted when the mouse cursor is positioned over the check box controls. Select this option to highlight check boxes in response to mouse movements.

Shadow
If this option is selected, a shadow is displayed for the check boxes and radio groups.

Native style
This option determines whether the native Windows style will be applied to the check

boxes and radio buttons.
The option has the highest priority for check boxes and radio buttons. If this option is selected, the check boxes and radio buttons are painted according to the native Windows style, regardless of other painting settings.

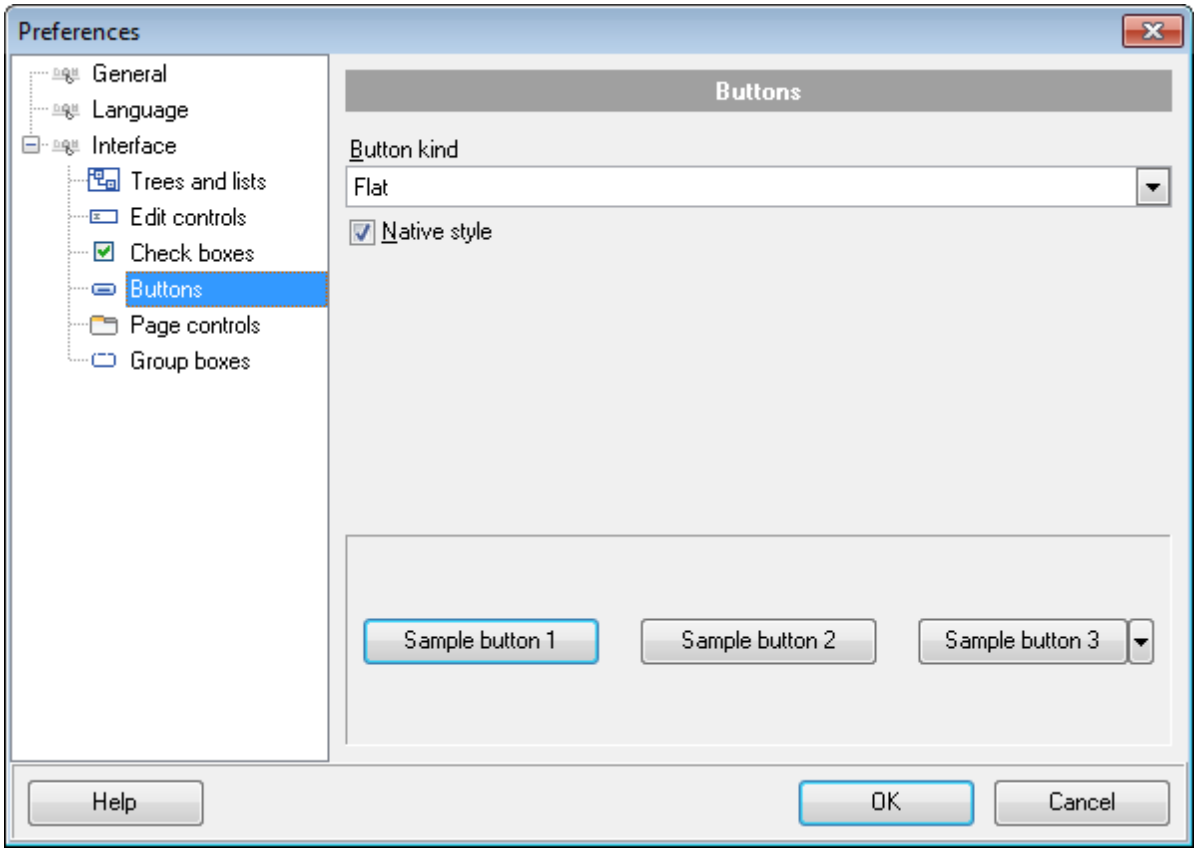
Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:

Trees and lists
Edit controls
Buttons
Group boxes

2.3.3.4 Buttons

Use the **Buttons** section to customize **Data Pump** *buttons*.



Button kind
This setting determines the manner in which a button is painted. Use the drop-down list to select the painting style that will be applied to buttons:
Standard
Flat
UltraFlat

Native style
This option determines whether the native Windows style will be applied to the buttons. The option has a higher priority than the **Button kind** setting. If this option is selected, the buttons are painted according to the native Windows style, otherwise the **Button kind** selection is applied.

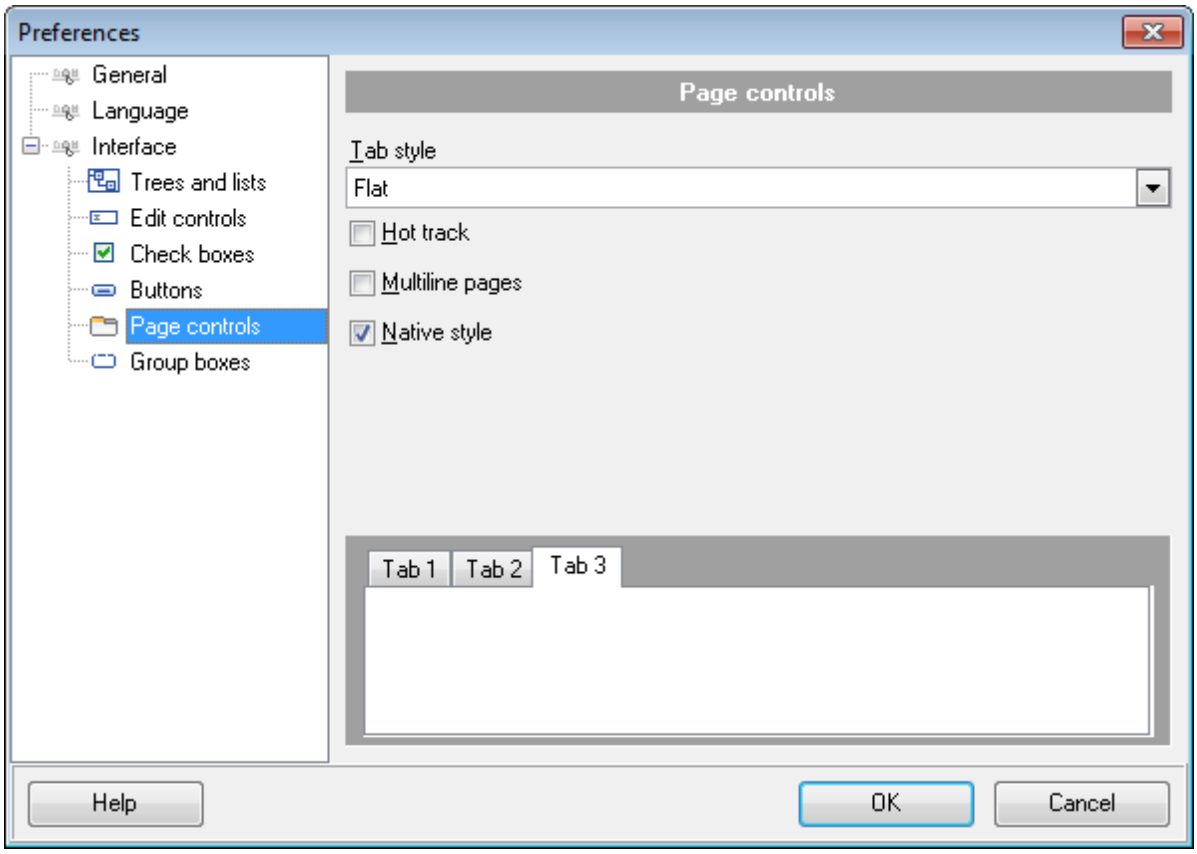
Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:
Trees and lists
Edit controls
Check boxes

Group boxes

2.3.3.5 Page controls

Use the **Page controls** section of the **Preferences** dialog to customize the style of all **Data Pump** *page controls*.



Tabs are visual elements of **tab controls**. Their purpose is to identify pages and switch between them. Once a tab is clicked, the corresponding page is selected.
Pages are container controls that represent the contents of tab controls. Tab controls contain a single page, whose context is to be updated each time the selected tab changes. **Page controls** contain the number of pages equal to the number of tabs.

Tab style
Use the drop-down list to select the painting style that will be applied to the tab controls:
Tabs (tabs are painted as notebook tabs)
Buttons (the selected tab is painted as a pressed button, other tabs are painted as released buttons)
Flat (tabs are painted as notebook tabs, but appear lowered slightly)

☑ **Hot track**
This option specifies whether tab captions are highlighted when the mouse pointer hovers over tabs. Select this option to enable tab highlighting.

☑ **Multiline pages**
This option specifies whether tabs are arranged across several rows.
If this option is enabled, tabs are automatically arranged into the minimum number of rows required to fit all of them. If this option is disabled, tabs are displayed within a single row.

Native style
This option determines whether the native Windows style will be applied to the tab controls.
The option has the highest priority for the tab controls. If this option is selected, the tabs are painted according to the native Windows style, regardless of other painting settings.

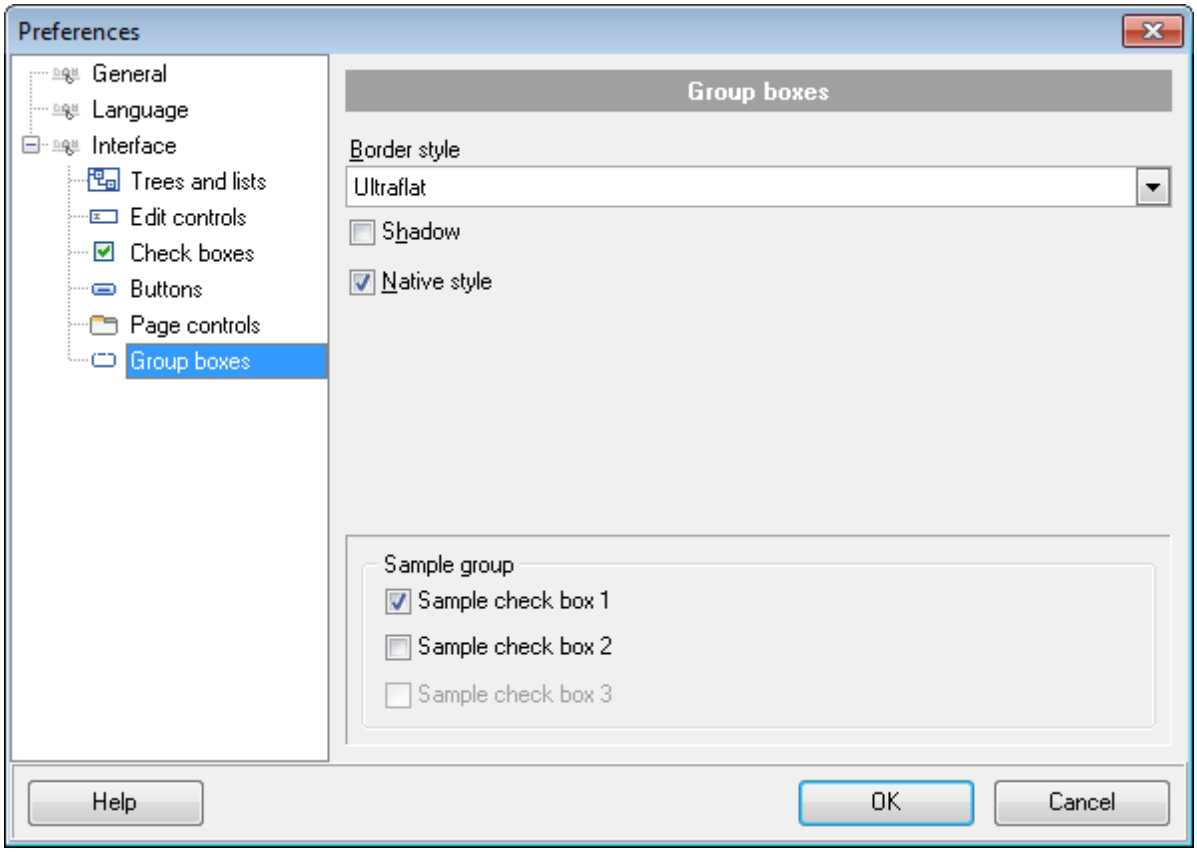
Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:

Trees and lists
Edit controls
Check boxes
Buttons
Group boxes

2.3.3.6 Group boxes

Use the **Group boxes** section to customize all **Data Pump** *group boxes* to your liking.



Border style
This setting determines the manner in which group box borders are painted. Use the drop-down list to select the painting style that will be applied to the group boxes:
None
Single
Thick
Flat
3D
UltraFlat

☑ **Shadow**
If this option is selected, a shadow is displayed for the group boxes.

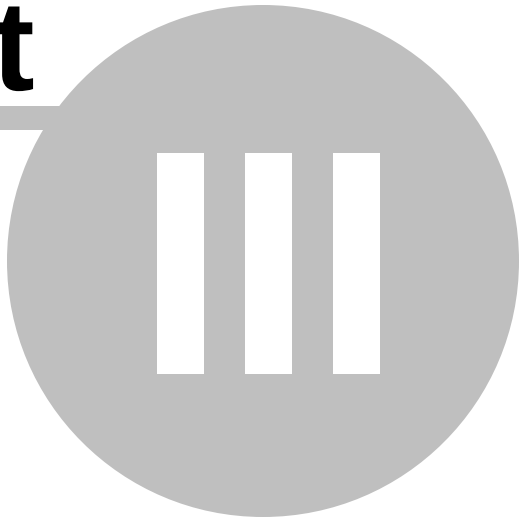
☑ **Native style**
This option determines whether the native Windows style will be applied to the group boxes.
The option has the highest priority for the group boxes. If this option is selected, the group boxes are painted according to the native Windows style, regardless of other painting settings.

Note: The **Native style** option is currently supported for the Windows® XP operating system only.

See also:

Trees and lists
Edit controls
Check boxes
Buttons

Part



3 Console Application

Additionally to **the GUI version** which is implemented in the form of a wizard application, the installation package of **Data Pump for PostgreSQL** includes **the console version** which is intended for being run from Windows command line with a template file name used as the execution parameter.

- Using Console Application

See also:

Wizard Application

3.1 Using Console Application

All the options for the pumping process are set in **template** (**.dpc*) files. A template can be also used in the **Console version** of **Data Pump for PostgreSQL**.

To create a template file, follow the instructions below:
- start **Data Pump for PostgreSQL** Wizard Application;
- set all the required options in all steps of the wizard;
- test the pumping process at the last step;
- save all generation options in the template file.

The easiest way to start **Data Pump** console application is to double-click the generated **.dpc* template file. The other way is to enter the command line and type the appropriate command.

Usage:

<path to Data Pump for PostgreSQL console application>\PgDataPumpC.exe TemplateFile [-L]

TemplateFile
Stands for the **.dpc* template file to be used as the console version execution parameter

[-L]
Selects current localization set in Wizard Application (GUI)

Example:

"C:\Program Files\EMS\Data Pump for PostgreSQL\PgDataPumpC.exe" DataPump1.dpc -L

Note that when you use the console version of **Data Pump**, the result script is always executed and the importing operation is always performed as well.

Note: The following exit codes can be returned by Data Pump for PostgreSQL to the operating system after performing the latest task:
0 - successful completion;
1 - error(s) occurred during task performing;
2 - fatal error occurred. The task was not performed.

See also:

Using wizard application

Part

4 Appendix

4.1 SSH tunneling options

SSH (Secure Shell Host) protocol is used to heighten computer security when working with Unix systems on the Internet. SSH uses several encryption algorithms of different reliability. The spread of SSH is also related to the fact that a number of *nix operating systems (e.g. FreeBSD) include SSH server in their standard distributions. To learn more about SSH, please visit http://openssh.org.

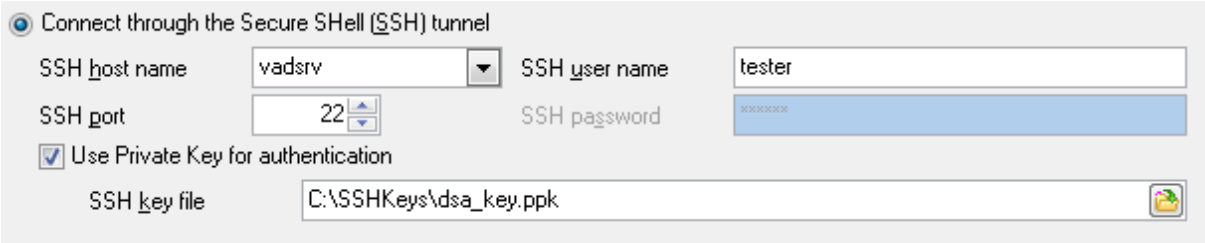
The SSH tunneling feature of **Data Pump** is a means of ensuring secure connection to PostgreSQL servers when working over insecure connection channels. You can also use SSH tunnel to get access to the remote PostgreSQL servers when port 3306 is closed for external connections for some reasons.

The connection via SSH tunnel works in the following way.
First, a connection is established and the process of authentication between SSH client built in **Data Pump** and remote PostgreSQL server is performed. Then all incoming and outgoing information between the application and PostgreSQL is transmitted through SSH server with the help of a communication port (regularly port 22), and SSH server transfers this information directly to PostgreSQL server.

To setup the connection via **SSH tunnel**, input the following values in the corresponding fields:

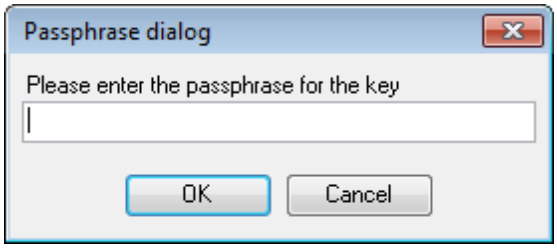
- **SSH host name** is the name of the host where SSH server is running
- **SSH port** indicates the port where SSH server is activated
- **SSH user name** stands for the user on the machine where SSH server is running (**Note:** it is a Linux/Windows user, not a user of PostgreSQL server)
- **SSH password** is the Linux/Windows user password

Please note that PostgreSQL **host name** should be set relatively to the SSH server in this case. For example, if both PostgreSQL and SSH servers are located on the same computer, you should specify *localhost* as **host name** instead of the server external host name or IP address.



Use Private Key for authentication
If the SSH encryption is enabled on the SSH server, a user can generate a pair of cryptographic keys (the **Private key** and the **Public key**). The **Public key** is placed on the SSH server, and the **Private key** is the part you keep secret inside a secure box that can only be opened with the correct passphrase (or an empty string as the passphrase). When you wish to access the remote system, you open the secure box with your passphrase (if any), and use the private key to authenticate yourself with the Public key on the remote Linux computer.

SSH Key file
Specify the location (the secure box) of the **Private key** file on your local machine.
Supported Private Key file formats are:
OpenSSH
Putty
SSH.com
Note that you need to trust your local machine not to scrape your passphrase or a copy of your Private key file while it is out of its secure box.



4.2 HTTP tunneling options

HTTP tunneling is a method that allows one to connect to a database and transmit data between the program and a PostgreSQL server through the HTTP/HTTPS protocols using port 80 which is used by a regular Internet browser. This method is used to connect to the remote PostgreSQL server of a hosting company when direct connection is not available (e.g. for security reasons).

The HTTP tunnel works in the following way.
All outgoing queries and commands sent by the client's software are encoded and transmitted through the HTTP/HTTPS protocol using port 80 to the special script that decodes the received data, sends it to PostgreSQL server for processing and then sends the result back. This method requires the HTTP server (Apache) and PHP with PostgreSQL to be installed on the remote server. Normally this software is provided by a hosting company that offers Linux hosting solutions.

To use **HTTP tunneling**, just upload the tunneling script to the webserver where PostgreSQL server is located (e.g. to the location where other PHP scripts are stored), or to any other webserver from which direct connections to your PostgreSQL server are allowed. This script exposes the PostgreSQL API as a set of web-services used by **Data Pump for PostgreSQL**.

If your webserver complies with the requirements and the script is installed correctly, you will see the message "EmsProxy v 1.31" (version can be different) in your browser when opening the http://<your_webserver>/emsproxy.php page.

In case of using this connection method the response will be slower as compared to the direct connection or the SSH Tunneling method, since the data are XML encoded and HTTP is stateless by nature. However, all the features of **Data Pump for PostgreSQL** are available.

Note that the *emsproxy.php* script file is included into the distribution package and can be found in **Data Pump** installation directory.

4.3 Find Text dialog

The **Find Text** dialog is provided for quick and flexible searching for specified text within the working area of **Data Pump** editors.

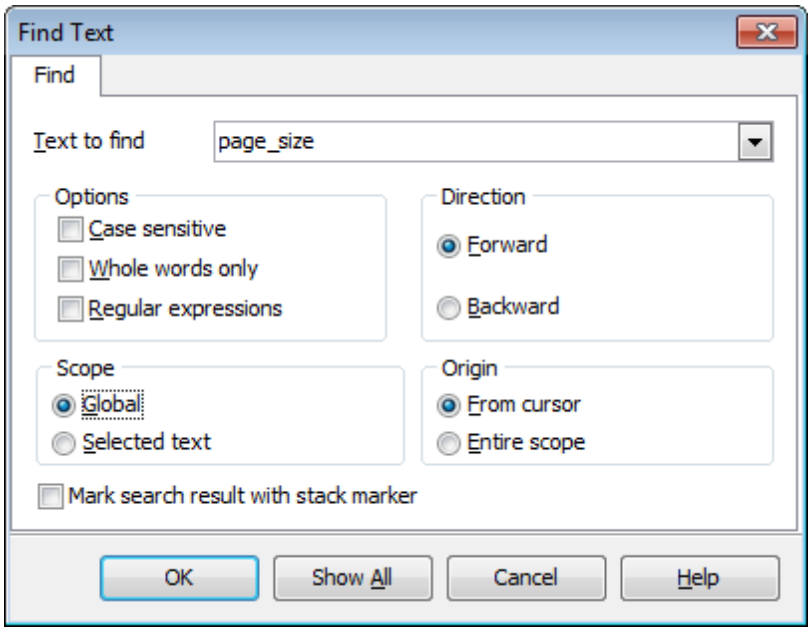
Text to find
Enter a search string in this box. The Arrow-Down button which can be found next to the input box allows you to select any of the previously entered search strings.

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 search string.



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.

Mark search result with stack marker
The option toggles marking search results. If this option is selected, stack markers are set at all search positions - this makes it possible to jump from one marker (search result) to another within the text.

Click the **Show All** button to highlight every occurrence of the search string.

4.4 Replace Text dialog

The **Replace Text** dialog is provided for searching and replacing text within the working area of **Data Pump** 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.

Text to replace
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.

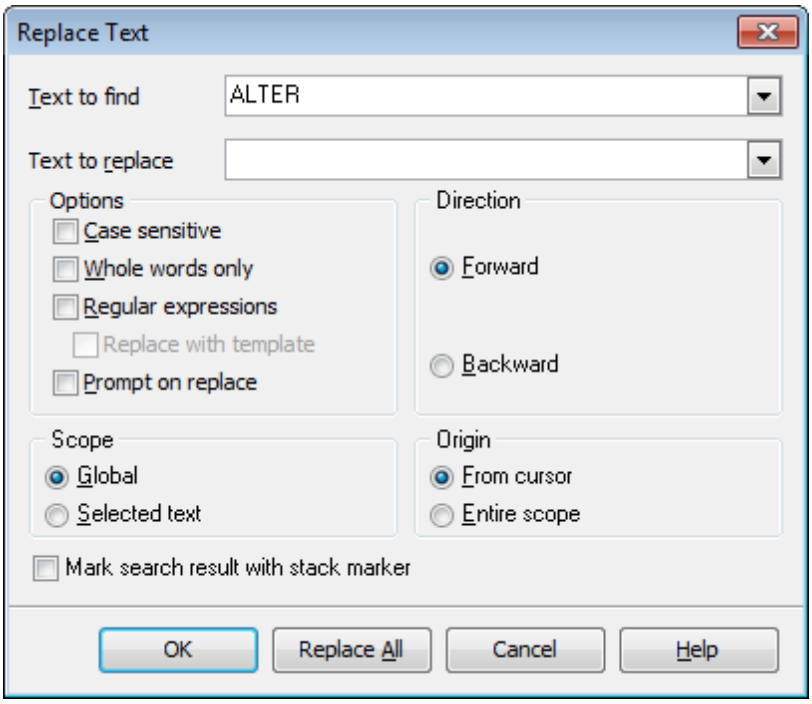
☑ **Whole words only**
Use this option to search for words only (with this option off, the search string might be found within longer words.)

☑ **Regular expressions**
Recognizes regular expressions in the search string.

☑ **Replace with template**
This option requires the **Regular expressions** option selection.
Enable this option to use regular expressions in the **Text to replace** field. Expression used in this field will be applied to each string that matches the **Text to find** expression.

Note: The syntax of regular expressions that can be used in the Text to find and the Text to replace fields 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.

☑ **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.



Direction

Forward
Searches and replaces from the current position to the end of the working area.

Backward
Searches and replaces from the current position to the beginning of the working area.

Scope

Global
Searches and replaces within the entire working area, in the direction specified by the *Direction* setting.

Selected text
Searches and replaces 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 and replace process 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 and replace process 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.

Mark search result with stack marker
The option toggles marking search results. If this option is selected, stack markers are set

at all search positions - this makes it possible to jump from one marker (search result) to another within the text.

Click the **Replace All** button to replace every occurrence of the search string. If you have checked the **Prompt on replace** option, the confirmation dialog box appears upon each occurrence of the search string.

4.5 Configuration file format

The **configuration** (**template**) **file** used by **Data Pump for PostgreSQL** is divided into several sections, each corresponding to a particular group of settings specified at different steps of the GUI application wizard.

[#General#]
This section stores general information about the utility:

Parameter	*Description*
Product	internal product name
Version	major version

[Source]
This section stores the connection properties used to access the data source. The connection string is built according to parameters specified in the Data Link Properties dialog.

Parameter	*Description*
ConnectionString	connection string for the data source

[#Comment#]
This section stores the template file comment as specified optionally in the Save template options dialog:

Parameter	*Description*
Line<N>	comment text

where *N* stands for the comment line identifier

Example:

```
Line0=Data Pump for PostgreSQL
Line1=Template file #1
Line2=Pumping tables from MS Access to PostgreSQL
```

[Options]
This section stores data conversion and script execution options. The parameters correspond to the values specified at Step 4 and Step 8 of the Wizard application.

Parameter	*Description*
CommitAfter	number of records after which the *COMMIT* statement is inserted
DisplayErrorMessages	*0* = error messages are hidden in the import log *1* = error messages are displayed in the import log
DetailErrorMessages	*0* = simple error messages in the import log *1* = detailed error messages in the import log
ShowTableProgress	*0* = simple data import progress in the import log *1* = detailed data import progress in the import log
ConvertNames	*0* = object names are not converted *1* = object names are converted to conform to SQL92 naming rules
NamesFormat	*nfAsIs* = object names are left without changes *nfLowerCase* = object names are converted to lower case

	nfUpperCase = object names are converted to upper case
ConvertDefaultValues	*0* = default values of the source database are not converted *1* = default values of the source database are converted to default values for the destination database
EnableScriptComments	*0* = script comments are disabled *1* = script comments are included into the body of the script
DropIfExistDbStatement	the parameter is not used by **Data Pump for PostgreSQL**
StopScriptOnError	*0* = script execution is not stopped on errors *1* = script execution is stopped if an error occurs
ShowExecutedStatements	*0* = the **Script execution information** area does not display any information upon script execution at Step 6 *1* = records for successfully executed statements are listed in the **Script execution information** area at Step 6
AlwaysSkipSuccExecuted	*0* = upon subsequent script execution at Step 6 you are prompted to choose whether successfully executed statements should be skipped or not *1* = successfully executed statements are skipped upon subsequent script execution at Step 6
ClearTablesBeforeImport	*0* = tables in the PostgreSQL database are not cleared before data import at Step 8 *1* = tables in the PostgreSQL database are cleared before data import at Step 8
TrimStringSpaces	*0* = source data strings *1* = unused space at the end of source data strings is cut off
QuoteNamesInImportSelQuery	*0* = quoting source object identifiers is disabled *1* = quoting source object identifiers is enabled
QuoteTemplate	*qtCustom* = custom quoting characters (**QuoteCharLeft**, **QuoteCharRight**) *qtDefault* = the default quotes for the data source in use *qtDoubleQuote* = "..." *qtSingleQuote* = '...' *qtApostrophe* = `...` *qtSquareBracket* = [...] *qtRoundBracket* = (...) *qtBrace* = {...} *qtAngleBracket* = <...>
QuoteCharLeft	specifies the left quoting character (if **QuoteTemplate** = *qtCustom*)
QuoteCharRight	specifies the right quoting character (if **QuoteTemplate** = *qtCustom*)
SchemasMapping	*smAsIs* = the tables are placed into the default schema in the target database *smExisting* = the tables are placed into an existing schema (**SchemasMappingExist**) *smNew* = a new schema is created (**SchemasMappingNew**) and the tables are placed into this schema
SchemasMappingExist	name of the existing schema in the PostgreSQL database to pump the source tables into (if **SchemasMapping** = *smExisting*)
SchemasMappingNew	name of the new schema in the PostgreSQL database to pump the source tables into (if **SchemasMapping** = *smNew*)
GenerateDropTableStmt	*0* = no *DROP* statements are generated *1* = the *DROP* statements are added for tables in the script
UseCopyStatement	*0* = INSERT statement is used to import data

1 = COPY statement is used to import data

[Target]
This section stores connection parameters for the *target* PostgreSQL database. The parameters correspond to the values entered at Step 1 of the Wizard application and are obligatory.

Parameter	*Description*
Host	host where the target database resides (if **Remote** = *1*)
Login	PostgreSQL login
Password	password to identify the login (encrypted)
RemoteIndex	*0* = local connection *1* = remote connection
DataBase	target PostgreSQL database name
DatabaseActions	*cdtCreate* = a new database is created for pumping data *cdtSelectExisting* = an existing database is selected for pumping data *cdtRecreate* = an existing database is dropped and then recreated
Charset	client character set specified for the connection
Port	port on which PostgreSQL is listening
UseSSL	*0* = SSL protocol is not used *1* = SSL protocol is used for interchanging data with remote PostgreSQL server
UseCompProtocol	*0* = compressed data protocol is not used *1* = compressed data protocol is used when connecting to the database server
QuotesIdentifiers	*0* = object identifiers are not quoted *1* = object identifiers are quoted with a backquote (''')
InteractiveMode	*0* = *wait_timeout* variable value from my.cnf file is used for connection timeout *1* = *interactive_timeout* variable value from my.cnf file is used for connection timeout
TunnelType	indicates the tunneling type being used: SSH, HTTP, or none (**TunnelType** = *ttNotUse*)
SSHHostName	name of the host where SSH server is running
SSHPort	port on which SSH server is activated
SSHUserName	user on the machine where SSH server is running
SSHPassword	password to identify SSH server user (encrypted)
SSHKeyFile	path to the Private Key used for the SSH connection (if **SSHUseKeyFile** = *True*)
SSHUseKeyFile	*True* = SSH Private Key is used *False* = SSH Private Key is not used
HTTPUrl	URL to the *emsproxy.php* script file uploaded to your web-server (for **HTTP** tunneling)

[{TBL}<table_name>]
Sections of this type contain table properties (those that were changed at Step 5 of the Wizard application) and data import options specified at Step 7.
Note: Sections of this type are only available in *dynamic* templates (for details see Save template options).

Parameter	*Description*
Name	table name
NameSpace	PostgreSQL schema to save the table into
Excluded	*1* = the table is excluded from data import at Step 7
WhereClause	text of the WHERE condition for pumping data

[{FLD}<field_name>]
Sections of this type contain field properties (those that were changed at Step 5 of the Wizard application).
Note: Sections of this type are only available in *dynamic* templates (for details see Save template options).

Parameter	*Description*
Name	field name
Type	data type applied to the field
NotNULL	*0* = nullable
	1 = NOT NULL
AutoIncrement	*0* = non-autoincrement field
	1 = autoincrement field

[{IND}<index_name>]
Sections of this type contain index properties (those that were changed at Step 5 of the Wizard application).
Note: Sections of this type are only available in *dynamic* templates (for details see Save template options).

Parameter	*Description*
Name	index name
Unique	*0* = non-unique index
	1 = unique index

[{FK}<key_name>]
Sections of this type contain key properties (those that were changed at Step 5 of the Wizard application).
Note: Sections of this type are only available in *dynamic* templates (for details see Save template options).

Parameter	*Description*
Name	key name
DeleteAction	'on delete' rule (for foreign keys):
	0 = NO ACTION
	1 = RESTRICT
	2 = CASCADE
	3 = SET NULL
	4 = SET DEFAULT
UpdateAction	'on update' rule (for foreign keys):
	0 = NO ACTION
	1 = RESTRICT
	2 = CASCADE
	3 = SET NULL
	4 = SET DEFAULT

[TypeMapping]
This section stores source-target type mapping options. The parameters correspond to the values specified in the Type mapping dialog.
Note: This section is only available in *dynamic* templates (for details see Save template options).

[ExcludedFromStructure]
This section stores the list of source objects excluded from structure conversion at Step 3 of the Wizard application.
Note: This section is only available in *dynamic* templates (for details see Save template options).

Parameter	*Description*
{NSP}<schema_name>	excluded schema (if available)
{TBL}<table_name>	excluded table
{IND}<index_name>	excluded index
{FK}<key_name>	excluded key

[SelectedTables]
This section stores the list of source and target tables, as specified at Step 3 and Step 5 of the Wizard application.
Note: This section is only available in *fixed* templates (for details see Save template options).

Parameter	*Description*
TableCount	number of tables selected for pumping
SrcTableNS<N>	source schema name (if available)
SrcTableNM<N>	source table name
TrgTableNS0<N>	target schema name
TrgTableNM<N>	target table name

where *N* stands for the table identifier

Example:
```
[SelectedTables]
TableCount=2
SrcTableNS0=dbo
SrcTableNM0=EMPLOYEE
TrgTableNM0=EMPLOYEE
SrcTableNS1=dbo
SrcTableNM1=DEPARTMENT
TrgTableNM1=DEPARTMENT
```

[Table<N>]
Sections of this type contain the list of source and target fields of the table (*N* stands for the table identifier in **[SelectedTables]**), as specified at Step 3 and Step 5 of the Wizard application.
Note: This section is only available in *fixed* templates (for details see Save template options).

Parameter	*Description*
SrcField<N>	source field name
TrgField<N>	target field name
FieldCount	number of table fields selected for pumping

where *N* stands for the field identifier

Example:

```
[Table1]
SrcField0=DepartmentID
TrgField0=DEPARTMENTID
SrcField1=Name
TrgField1=NAME
SrcField2=GroupName
TrgField2=GROUPNAME
SrcField3=ModifiedDate
TrgField3=MODIFIEDDATE
FieldCount=4
```

[FixedInfo]

This section stores the result script as presented at Step 6 of the Wizard application.
Note: This section is only available in *fixed* templates (for details see Save template options).

Parameter	*Description*
Line<N>	script text

where *N* stands for the script line identifier

Credits

Software Developers:

Michael Glotov

Dmitry Schastlivtsev

Alexey Butalov

Michael Kuzevanov

Technical Writers:

Dmitry Doni

Semyon Slobodenyuk

Olga Ryabova

Cover Designer:

Tatyana Makurova

Translators:

Anna Shulkina

Serge Fominikh

Team Coordinators:

Dmitry Schastlivtsev

Alexander Chelyadin

Roman Tkachenko