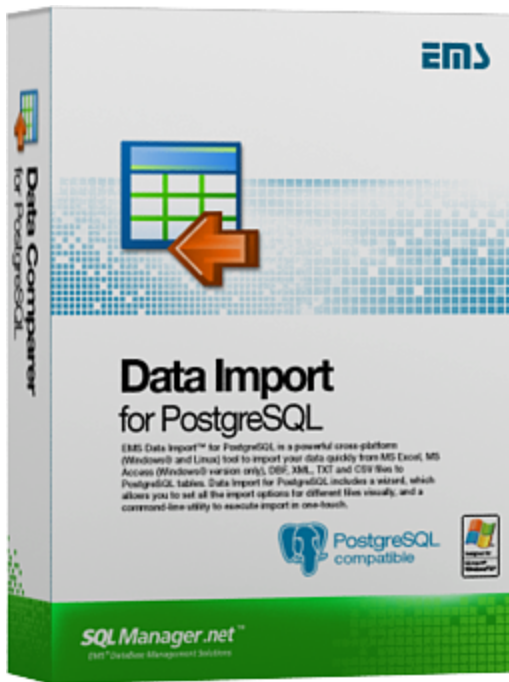


SQL Manager.net™

EMS® Software Development



Data Import for PostgreSQL User's Manual

© 1999-2023 EMS Software Development

Data Import for PostgreSQL User's Manual

© 1999-2023 EMS Software Development

All rights reserved.

This manual documents EMS Data Import for PostgreSQL

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

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

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

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

Document generated on: 12.07.2023

Table of Contents

Part I Welcome to EMS Data Import utility!	6
What's new	7
System requirements	8
Installation	9
How to buy Data Import	10
How to register Data Import	12
EMS Data Import FAQ	13
Other EMS Products	15
Part II Wizard application	22
Working with wizard application	23
Getting started	23
Step 1 - Setting connection properties	24
Selecting registered database.....	26
Step 2 - Selecting files to import	27
Step 3 - Setting XML file type	31
Step 4 - Mapping fields	31
MS Excel 97-2003.....	32
MS Excel.....	33
MS Word.....	36
MS Access.....	37
DBF.....	38
XML.....	40
XML Document.....	41
TXT.....	43
CSV.....	44
HTML.....	46
ODS.....	47
ODT.....	49
Step 5 - Setting base data formats	50
Format specifiers.....	52
Step 6 - Settings data formats for each field	55
Step 7 - Specifying import mode	59
Step 8 - Selecting key columns	61
Step 9 - Setting common options	62
Step 10 - Defining scripts	63
Step 11 - Start of data import process	64
Using data import configuration files	66
Setting program preferences	68
General	69
Skipped steps	71
Languages	73

Part III Console application	76
Working with console application	77
Configuration file format	78
Part IV Appendix	82
Supported file formats	82
SSH tunneling options	83
HTTP tunneling options	84
Add/Edit field	85
Add/Edit Range	86

Part



1 Welcome to EMS Data Import utility!

EMS Data Import for PostgreSQL is a powerful program to import your data quickly from MS Excel, MS Word, MS Access, DBF, TXT, CSV, Open Document, HTML files to PostgreSQL tables. It provides adjustable import parameters, including source data formats for all the fields and destination data formats for the selected fields, commit options, number of records to skip, etc. Data Import for PostgreSQL includes a wizard which allows you to set all the import options for different files visually, and a command-line utility to execute import in one-touch.

Visit our web-site: <https://www.sqlmanager.net/> for details.

Key features

- Import from most popular formats: MS Excel, MS Access, XML, DBF, TXT, CSV, MS Word, Open Document Format and HTML
- Importing data to one or several tables or views from different databases
- Automatically creates table structure
- Special batch insert commands allow to import data at the maximum possible speed
- A number of import modes - Insert All, Insert New, Insert or Update, and others
- Secure Shell (SSH) and HTTP tunneling support
- UNICODE support
- Adjustable import parameters for each source file
- Saving all import parameters set on current wizard session
- Command-line utility to import data using the configuration file
- Powerful visual options module
- User-friendly localizable wizard interface

Product information

Homepage: <https://www.sqlmanager.net/products/postgresql/dataimport>

Support Ticket <https://www.sqlmanager.net/support>

System:

Register online at: <https://www.sqlmanager.net/products/postgresql/dataimport/buy>

1.1 What's new

Version**Data Import for PostgreSQL 4.1.0****Release date**

July 12, 2023

What's new in EMS Data Import?

- Support for Windows 11 ARM implemented.
- Restart of the Wizard is now available at the last step.
- Check for column name length added on creating a new table.
- Default values for separator and quotes are available for CSV now.
- The password is now saved correctly to the template file.
- XML files without header are now processed with no issues.
- Implemented localization in the command-line version.
- Exit codes are corrected for working in the console version.
- Other improvements and fixes.

1.2 System requirements

System requirements for Data Import for PostgreSQL

- Microsoft® Windows XP, Microsoft® Windows 2003 Server, Windows® 2008 Server, Microsoft® Windows Vista, Microsoft® Windows 7, Microsoft® Windows 8, Microsoft® Windows 2012 R2 Server, Microsoft® Windows 2012 Server, Microsoft® Windows 8.1, Microsoft® Windows 10, Microsoft® Windows 2016 Server, Microsoft® Windows 2019 Server, Microsoft® Windows 11, Microsoft® Windows 11 ARM
- 512 MB RAM or more
- 50MB of available HD space for program installation
- Possibility to connect to any local or remote PostgreSQL server
- Supported PostgreSQL server versions: from 7.3 up to 15

1.3 Installation

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

- download the Data Import 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 *PgImportSetup.exe* from the local directory and follow the instructions of the installation wizard;
- after the installation process is completed, find the Data Import shortcut in the corresponding group of Windows Start menu.

[Installation FAQs](#)

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

- download the **Data Import 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 Import application if it is running;
- run *PgImportSetup.exe* from the local directory and follow the instructions of the installation wizard.

See also:

[System requirements](#)

| 8 |

1.4 How to buy Data Import

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.

To obtain **MORE INFORMATION** on this product, visit us at <https://www.sqlmanager.net/products/postgresql/dataimport>

Product distribution	MyCommerce/Digital River
EMS Data Import for PostgreSQL (Business license) + 1-Year Maintenance*	Buy Now!
EMS Data Import for PostgreSQL (Business license) + 2-Year Maintenance*	
EMS Data Import for PostgreSQL (Business license) + 3-Year Maintenance*	
EMS Data Import for PostgreSQL (Non-commercial license) + 1-Year Maintenance*	
EMS Data Import for PostgreSQL (Non-commercial license) + 2-Year Maintenance*	
EMS Data Import for PostgreSQL (Non-commercial license) + 3-Year Maintenance*	
EMS Data Import 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
- Access to personalized sqlmanager.net account
- online, speed-through maintenance renewal

- Advanced and exclusive notification of software promotions
- "Maintenance Owner ONLY" product promotions

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.

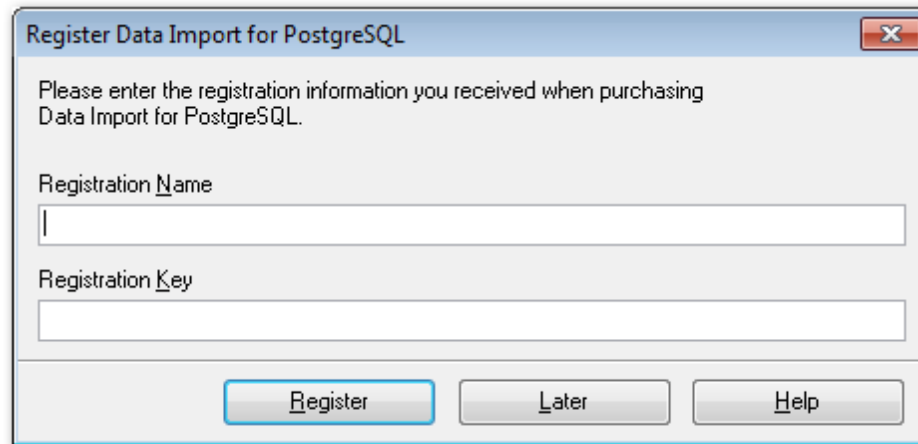
See also:

[How to register Data Import](#)^[12]

1.5 How to register Data Import

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

- receive the notification letter from **Share-it!** 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](#)^[23].



Register Data Import for PostgreSQL

Please enter the registration information you received when purchasing Data Import for PostgreSQL.

Registration Name

Registration Key

Register Later Help

See also:

[How to buy Data Import](#)^[10]

1.6 EMS Data Import FAQ

Please read this page attentively if you have questions about Data Import for PostgreSQL.

Table of contents

- [What is EMS Data Import for PostgreSQL?](#)^[13]
- [What do I need to start working with EMS Data Import for PostgreSQL?](#)^[13]
- [What is the difference between the Import feature of SQL Manager for PostgreSQL and the Data Import for PostgreSQL utility?](#)^[13]
- [Are there any limitations implied in the trial version as compared with the full one?](#)^[13]
- [What is the easiest way to configure the template files for the Data Import console application?](#)^[14]
- [I work behind a firewall \(router\). Can EMS Data Import connect to a remote PostgreSQL server?](#)^[14]
- [Why cannot I connect to PostgreSQL Server?](#)^[13]

Question/answer list

Q: What is EMS Data Import for PostgreSQL?

A: EMS Data Import for PostgreSQL is a powerful tool to import your data quickly from MS Access, MS Excel, DBF, TXT, CSV, XML, MS Excel 2007, MS Word 2007, HTML, ODF files to PostgreSQL tables. It provides adjustable import parameters, including source data formats for all the fields and destination data formats for the selected fields, commit options, number of records to skip, etc. Data Import includes a wizard which allows you to set all the import options for different files visually, and a command-line utility to execute import in one-touch.

Q: What do I need to start working with EMS Data Import for PostgreSQL?

A: First of all, you must have a possibility to connect to some local or remote PostgreSQL server to work with Data Import. You can download PostgreSQL database server from <https://www.postgresql.org/download/>. Besides, you need your workstation to satisfy the [system requirements](#)^[8] for Data Import for PostgreSQL.

Q: What is the difference between the Import feature of SQL Manager for PostgreSQL and the Data Import for PostgreSQL utility?

A: First of all, Data Import for PostgreSQL works faster as it is a considerably lighter application. Besides, it provides additional features for query building, namely:

- import data to several tables simultaneously;
- import data to tables selected from different databases;
- the command-line version of the utility to import data using the configuration (template) file with all import settings;
- automatically creates table structure.

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

A: Actually the trial version of the utility only allows you to import 20% of records into each table. With this limitation, you can test all the features implemented in Data Import for PostgreSQL within the 30-day trial period.

Note: There is a 100 records (instead of 20%) limitation when importing from CSV,

TXT or MS Access files.

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

A: You can configure the template files for each table visually using the Data Import Wizard application. Set the required options and select the Tools | Save Template menu item. All the options will be saved to the template file which can be used by the console application.

Q: I work behind a firewall (router). Can EMS Data Import connect to a remote PostgreSQL server?

A: If due to the security policy accepted in your corporate network or by your host provider you cannot connect to your PostgreSQL server directly through TCP/IP (for example a firewall prevents this), you can use SSH or HTTP tunneling feature of SQL Manager to fulfill your task.

Q: Why cannot I connect to PostgreSQL Server?

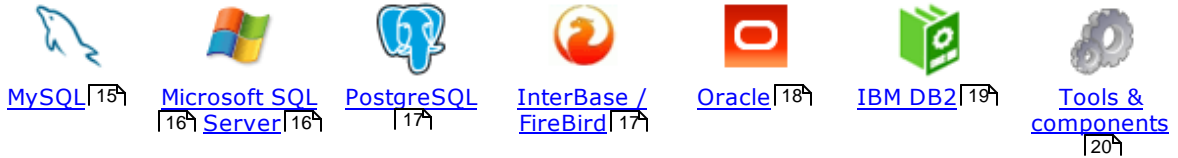
A: 1. Make sure that there is a record for your client PC in pg_hba.conf file on PostgreSQL server. This file can be found in database cluster's data directory.
2. If you connect to the remote server, then the TCP/IP connections should be allowed on the remote server. In this case you need to edit file postgresql.conf on the server: just add the name or TCP/IP of the client PC in the listen_addresses list. Or you can as well put * character; in this case the connection from any host will be allowed. You can get the same result if you run the server with -i option, which is the same as listen_addresses = '*'.
3. Make sure that you enter the correct login/password.
4. You use password authentication, but it is not supported in Data Import.

[Scroll to top](#) 











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

1.7 Other EMS Products

Quick navigation



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](#)^[15]

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](#) ¹⁵

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](#)^[15]

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](#)¹⁵

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](#)^[15]

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.

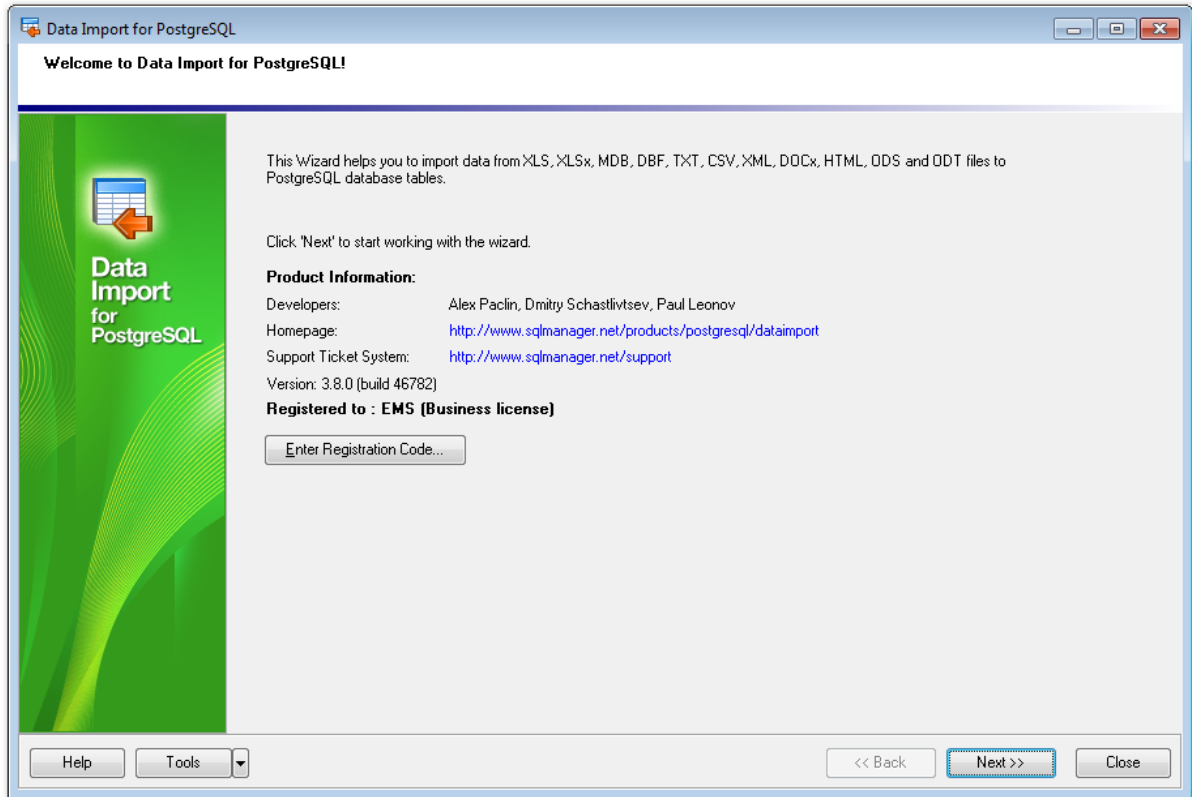
[Scroll to top](#)^[15]

Part



2 Wizard application

Data Import for PostgreSQL wizard application provides easy-to-use wizard interface to set all data import parameters visually.



[Working with wizard application](#)^[23]

[Using configuration files](#)^[66]

[Setting program preferences](#)^[68]

See also:

[Console Application](#)^[76]

2.1 Working with wizard application

Follow the steps of the wizard to import data to PostgreSQL tables for your needs.

[Getting started](#)^[23]

[Step 1 - Setting connection properties](#)^[24]

[Step 2 - Selecting files to import](#)^[27]

[Step 3 - Setting XML file type](#)^[31]

[Step 4 - Mapping fields](#)^[31]

[Step 5 - Setting base data formats](#)^[50]

[Step 6 - Setting data formats for each field](#)^[55]

[Step 7 - Specifying import mode](#)^[59]

[Step 8 - Selecting key columns to exclude from import process](#)^[61]

[Step 9 - Setting common options](#)^[62]

[Step 10 - Defining scripts to execute before and after import](#)^[63]

[Step 11 - Start of data import process](#)^[64]

See also:

[Using data import configuration files](#)

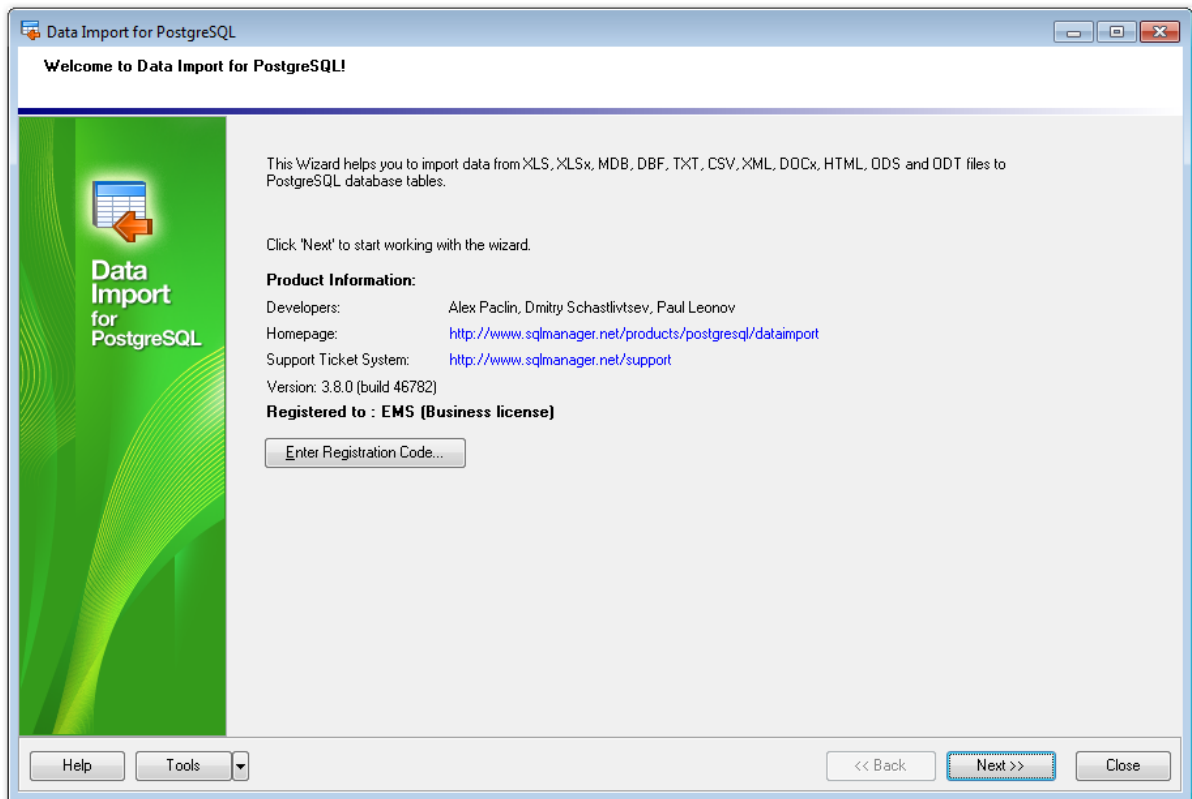
^[66]

[Setting program preferences](#)^[68]

2.1.1 Getting started

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

This page allows you to view registration information. If you have not registered Data Import for PostgreSQL yet, you can do it by pressing the **Register** button and [entering your registration information](#)^[12].



Press the **Next** button to proceed to [Setting connection properties](#)^[24].

See also:

[How to buy Data Import](#)^[10]

2.1.2 Step 1 - Setting connection properties

At this step you should specify necessary settings to establish **connection** to the target PostgreSQL database.

The screenshot shows the 'Data Import for PostgreSQL' dialog box, Step 1 of 11, titled 'Set PostgreSQL server connection options'. The dialog is divided into a green sidebar on the left and a main configuration area on the right. The sidebar contains the product logo and the text 'Data Import for PostgreSQL'. The main area is titled 'Connection Properties' and contains the following fields and options:

- Host:** A dropdown menu with 'tester' selected.
- Port:** A dropdown menu with '54111' selected.
- DB to connect:** A text field with 'postgres' entered.
- SSL mode:** A dropdown menu with 'Require' selected.
- Client certificate:** A text field with 'D:\SSL\postgresql.crt' and a browse button.
- Client key:** A text field with 'D:\SSL\postgresql.key' and a browse button.
- Root certificate:** A text field with 'D:\SSL\root.crt' and a browse button.
- Revocation list:** A text field with a browse button.
- Authentication options:**
 - Don't use tunneling
 - Connect through the Secure SHell (SSH) tunnel
 - SSH host:** A dropdown menu with 'testssh' selected.
 - SSH port:** A text field with '22' entered.
 - Use Private Key for authentication
 - SSH key file:** A text field with 'C:\SSHKeys\dsa_key.ppk' and a browse button.
 - Connect through the HTTP tunnel
 - URL:** A dropdown menu with 'http://webserver_name/emsproxy.php' selected.
- Login:** A text field with 'postgres' entered.
- Password:** A text field with masked characters (dots).
- SSH login:** A text field with 'tester' entered.
- SSH password:** A text field.

At the bottom of the dialog, there are buttons for 'Help', 'Tools', '<< Back', 'Next >>', and 'Close'.

Connection settings

For connection you also enter PostgreSQL host name in the **Host** field. The **Host** drop-down list contains the list of already registered hosts. For both types you should enter PostgreSQL port to connect through in the **Connection port** field.

Afterwards you should specify *authorization* settings: **Login** and **Password**. The default superuser name is 'postgres' with the password specified during PostgreSQL server installation.

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

If you are using the EMS SQL Management Studio for PostgreSQL version of Data Import 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](#)^[26] dialog.

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

Tunneling settings

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

- **SSH host** is the name of the host (IP address) where SSH server is running
- **SSH port** indicates the port where SSH server is activated (default is "22")
- **SSH login** 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

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

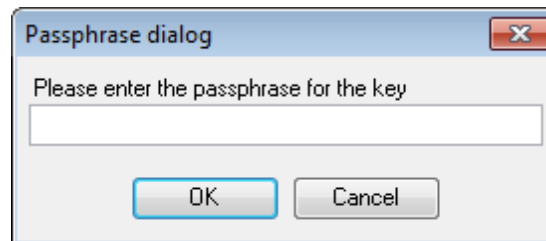
Specify the location (the secure box) of the **Private key** file on your local machine. Supported Private Key file formats are:

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.



To use [HTTP tunneling](#)^[84], 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 Import for PostgreSQL.

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

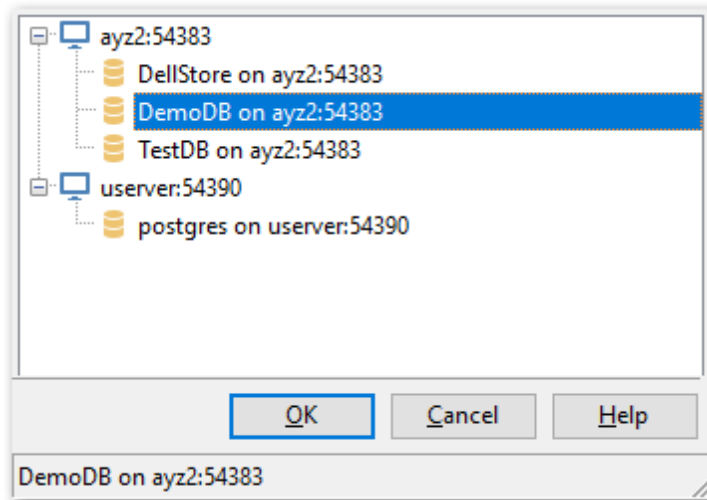
If you get any difficulties connecting to the database, it can be useful to look through [FAQs](#)^[13].

NOTE: The **Tools** button gets access to saving/loading import templates or adjusting program [Preferences](#)^[68].

When you are done, press the **Next** button to proceed to the [next step](#)^[27].

2.1.2.1 Selecting registered database

Use this dialog to select a database for importing data. This dialog is available only in EMS SQL Management Studio version of Data Import 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](#)^[24] automatically.

2.1.3 Step 2 - Selecting files to import

At this step you should **select source file(s) to be imported**.

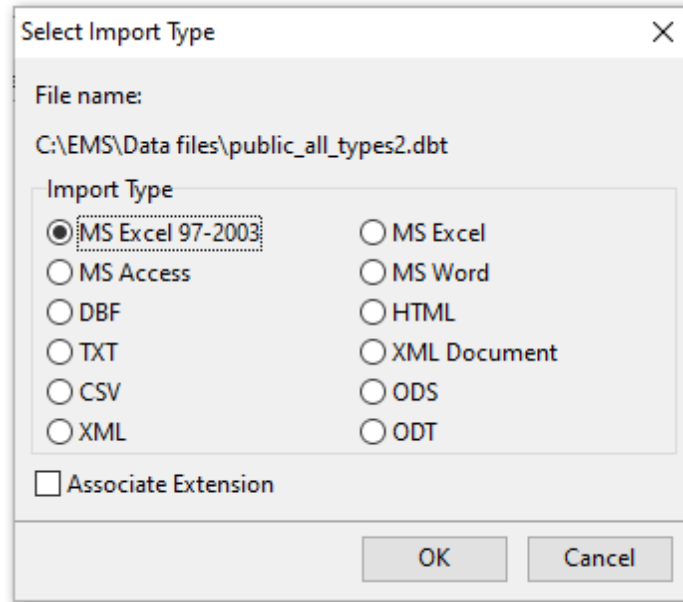
Click the  **Add File** button to select the source file name using the **Open file...** dialog. Repeat this operation to add more source files (if necessary).

You can choose among the following types of the source data file:


- *MS Excel*
- *MS Excel 2007*
- *MS Word 2007*
- *MS Access*
- *DBF*
- *TXT*
- *CSV*
- *HTML*
- *XML Data Packet*
- *XML MS Access*
- *XML Doc*
- *ODS*
- *ODT*



The open file dialog allows you to set a filter on the source file format.

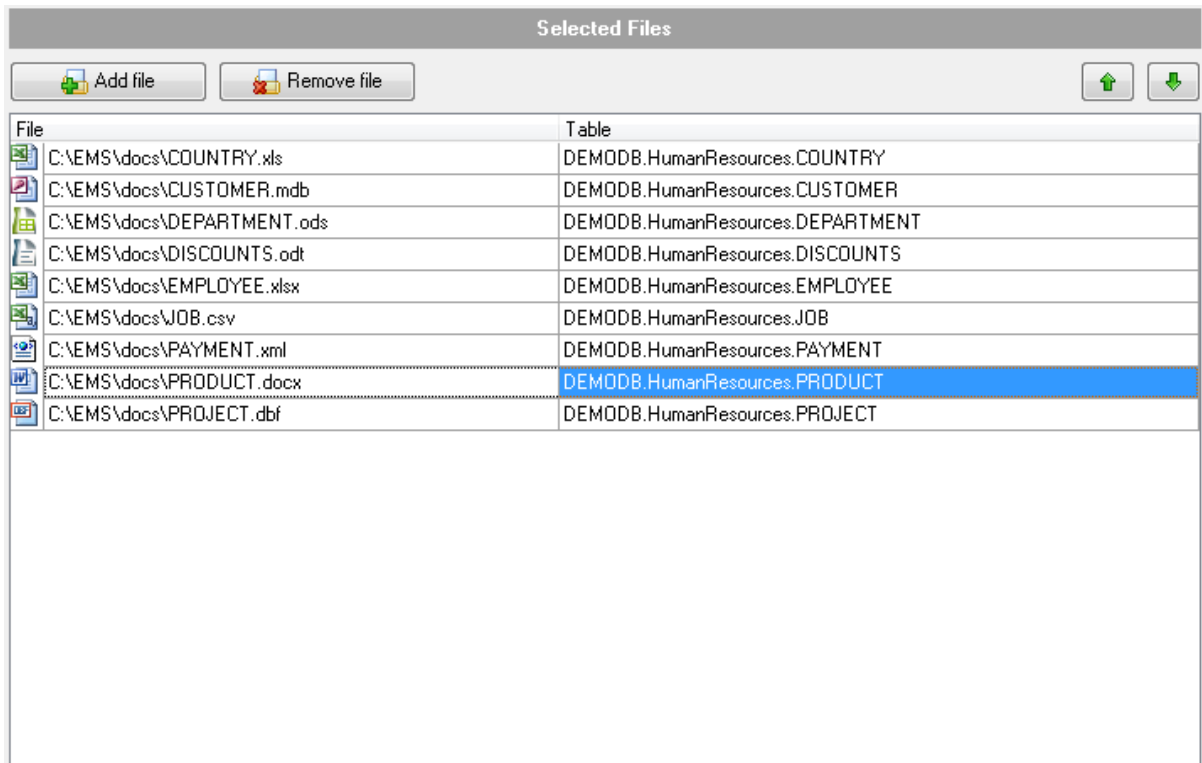
If you have selected the file of a format that is not supported the **Select Import Type** dialog appears.



Here you can select which import type for supported file formats should be applied to import the selected file.

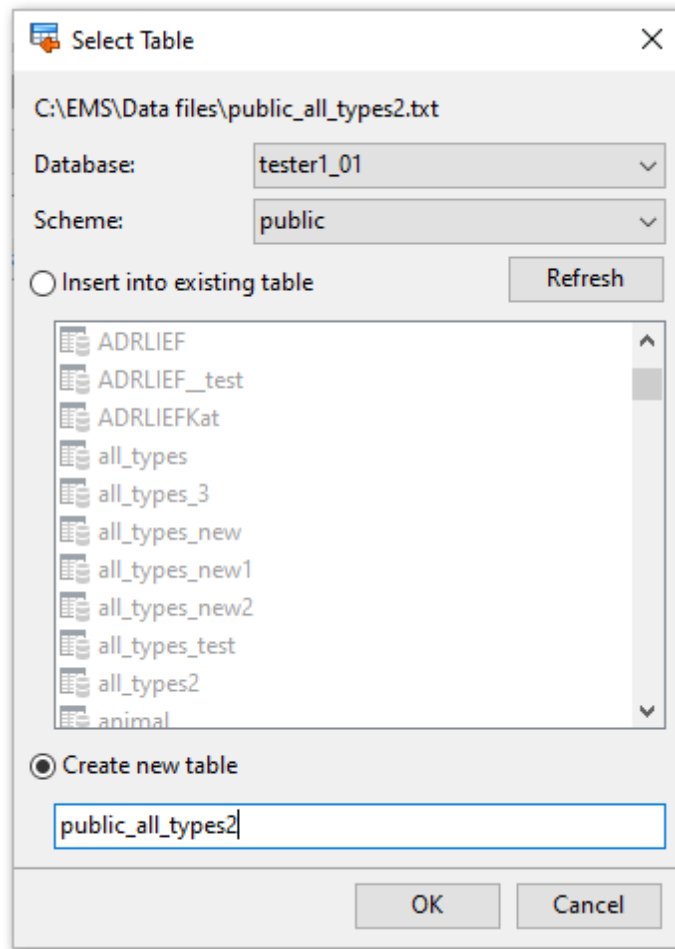
To delete a file from the list, select it and click the  **Remove File** button.

You can also change the order of the source files in the list using the **Up**  and the **Down**  buttons.



Note: For the spreadsheet files: if you need to import data from different sheets into different tables you are to include that multi-sheet spreadsheet file into the list several times (equal to the number of sheets to be imported to different tables).

When a file is selected the **Select Table** dialog appears:



File name

Displays the name of the imported file and full path.

Database field allows you to specify the target database.

Insert into existing table


Select this option to import data to the existing table. The active area contains the list of tables in the selected database. You can refresh the list by clicking the **Refresh** button.

Create new table

Use this option to create import destination table.

Schema

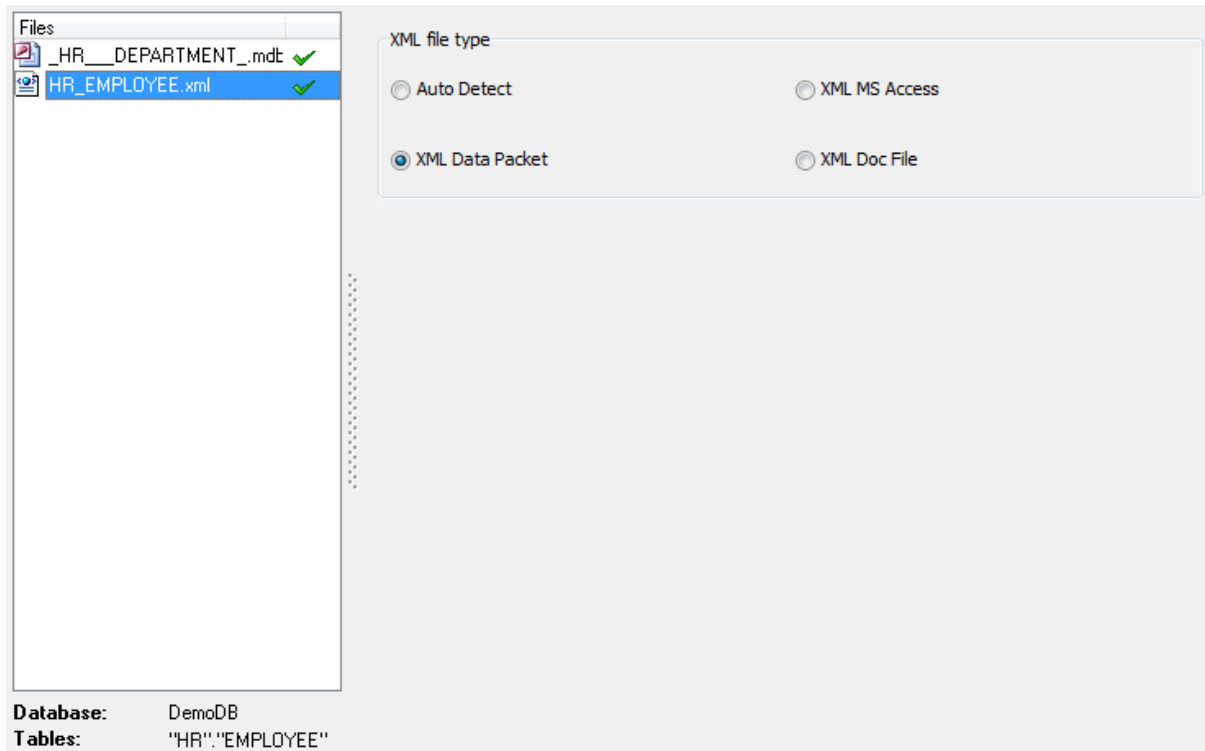
Use the drop-down list to select the schema in which the table should be created.

To change the target PostgreSQL table that has been already assigned to a source data file, select the table in the list and press the ellipsis  button to call the **Select Table** dialog again.

When you are done, press the **Next** button to proceed to the [next step](#)^[31].

2.1.4 Step 3 - Setting XML file type

This step appears if you have selected *.xml file for importing data from.



XML file type

Select the type of *.xml file from which you are importing data:

- Auto Detect* - file type will be detected automatically whether it is XML MS Access or XML data packet file;
- XML MS Access* - select this option if your *.xml file has the same structure as files exported from MS Access;
- XML Data Packet* - select this option if your *.xml file has data representation format used by Embarcadero;
- XML Doc File* - select this option to map fields of a [Generic XML document](#)^[41] manually.

When you are done, press the **Next** button to proceed to the [next step](#)^[31].

2.1.5 Step 4 - Mapping fields

This step of the wizard allows you **to set correspondence** between columns of the source file and fields of the target PostgreSQL table, according to the source data format.

- [MS Excel 97-2003](#)^[32]
- [MS Excel mapping](#)^[33]
- [MS Word mapping](#)^[36]
- [MS Access mapping](#)^[37]
- [DBF mapping](#)^[38]
- [XML mapping](#)^[40]
- [XML Document mapping](#)^[41]

- [TXT mapping](#)^[43]
- [CSV mapping](#)^[44]
- [HTML Document](#)^[46]
- [ODS mapping](#)^[47]
- [ODT mapping](#)^[49]

To get more information about the file formats, see the [Supported file formats](#)^[82] page.

When you are done, press the **Next** button to proceed to the [next step](#)^[50].

2.1.5.1 MS Excel 97-2003


Table


Select the sheet for importing data from.


Select the needed source file from the list. Then specify ranges in the grid for the target and source fields:

- select a field of the target PostgreSQL table in the **Fields** list;
- proceed to the sheet grid: click a column caption to select the whole column or click the row number to select the whole row;
- the selected column/row of the source file gets green highlight, and a new range indicating the source and target fields correspondence appears in the **Range** list;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

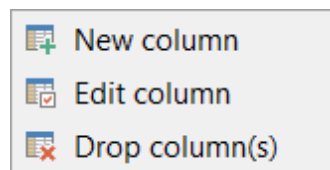
you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have



different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity. You can also use

the  **Auto fill Rows** to set the correspondence between destination table fields and source file rows.

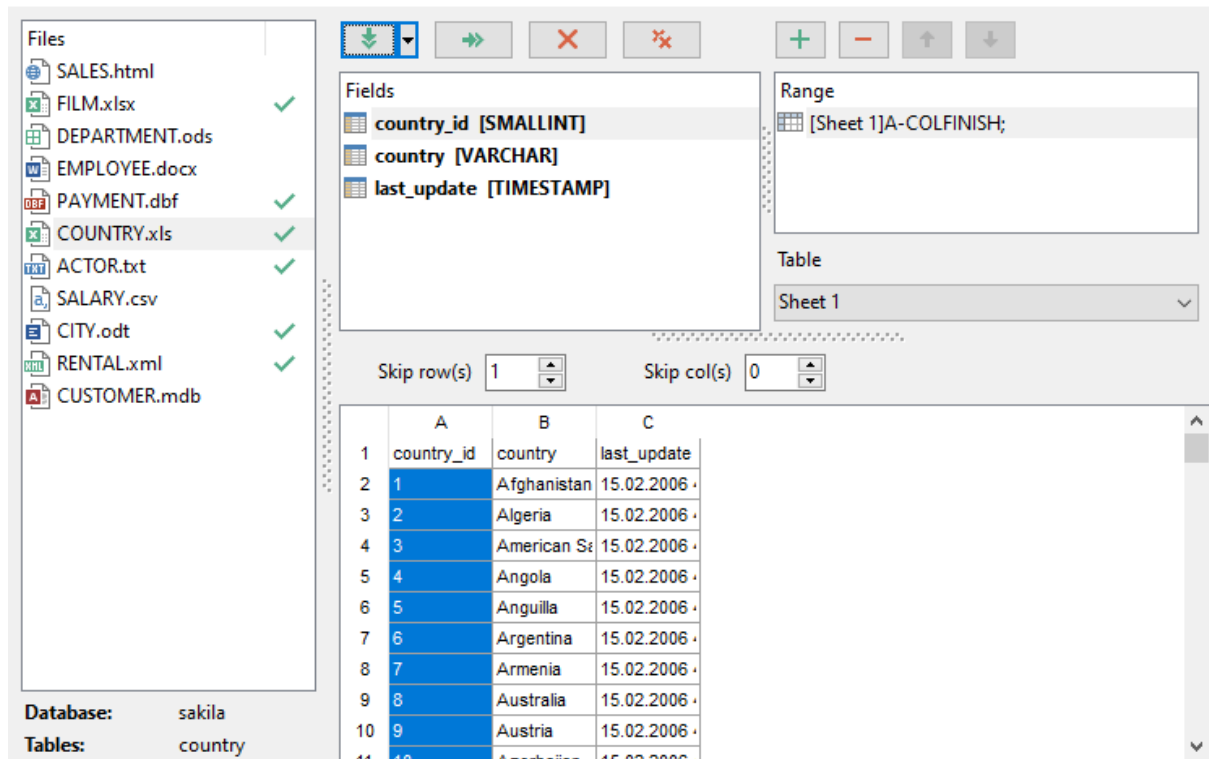
If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.




Note: If table was properly created or already exists, it will be marked with a tick . If an error occurs during table creation, this table will be marked with a cross .


If necessary, you can choose to skip a defined number of the source file columns and/or rows using the **Skip Col(s)** and **Skip Row(s)** spin-edits.



If your spreadsheet file contains several sheets, you are able to set different mapping for each sheet.

To clear ranges for a field, select the field in the **Fields** list and press the  **Clear** button.

To clear all ranges specified for the target table fields, press the  **Clear All** button.

To [set a range of data](#)^[86] to be imported from the file, use the  **Add range** button.

To remove a range, use the  **Delete range** button.

Using the  **Move Up** and the  **Move Down** buttons you can change the order of ranges applied to data.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.2 MS Excel

Table

Select the sheet for importing data from.


Select the needed source file from the list. Then specify ranges in the grid for the target and source fields:


- select a field of the target PostgreSQL table in the **Fields** list;
- proceed to the sheet grid: click a column to assign the column to the selected target

table field;

- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

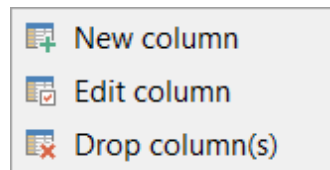
you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have

different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.



Note: The number of the column for which correspondence was set is displayed at the **Col.** control. You can use it for setting the fields correspondence as well.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.




If necessary, you can choose to **skip** a defined number of the source file rows or columns using the **Skip lines** or **Skip cols** spin-edit.


Note: If table was properly created or already exists, it will be marked with a tick . If an error occurs during table creation, this table will be marked with a cross .

Database: sakila
Tables: film


	A	B	C	D	E	F	G	H
1	film_id	title	description	release_yea	language_id	original_lang	rental_durat	rental_r
2	1	ACADEMY I	A Epic Dram	2006	1	null	6	0,99
3	2	ACE GOLDF	A Astoundin	2006	1	null	3	4,99
4	3	ADAPTATIO	A Astoundin	2006	1	null	7	2,99
5	4	AFFAIR PRE	A Fanciful D	2006	1	null	5	2,99
6	5	AFRICAN EC	A Fast-Pace	2006	1	null	6	2,99
7	6	AGENT TRU	A Intrepid Ps	2006	1	null	3	2,99
8	7	AIRPLANE S	A Touching	2006	1	null	6	4,99


To select the column click its header. To select the row click its number. To select several cells at once click the right cell and the last cell holding the "Shift" button.

To remove a correspondence, select the field in the **Fields** list and press the  **Clear Field Ranges** button.

To remove all correspondences, press the  **Clear All Ranges** button.

If your spreadsheet file contains several sheets, you are able to set different mapping for each sheet.

To clear ranges for a field, select the field in the **Fields** list and press the **Clear**  button.

To clear all ranges specified for the target table fields, press the **Clear All**  button.

To [set a range of data](#)^[86] to be imported from the file, use the **Add range**  button.

To remove a range, use the **Delete range**  button.

Using the **Move Up**  and the **Move Down**  buttons you can change the order of ranges applied to data.


Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.


2.1.5.3 MS Word

Specify ranges in the grid for the target and source fields:

- select a field of the target PostgreSQL table in the **New fields** list;
- proceed to the sheet grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have

different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.

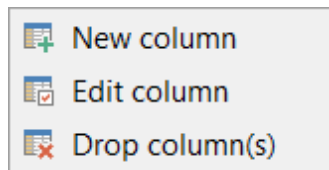
Table

In case the file contains several tables select the require one from the dropdown menu.



The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.

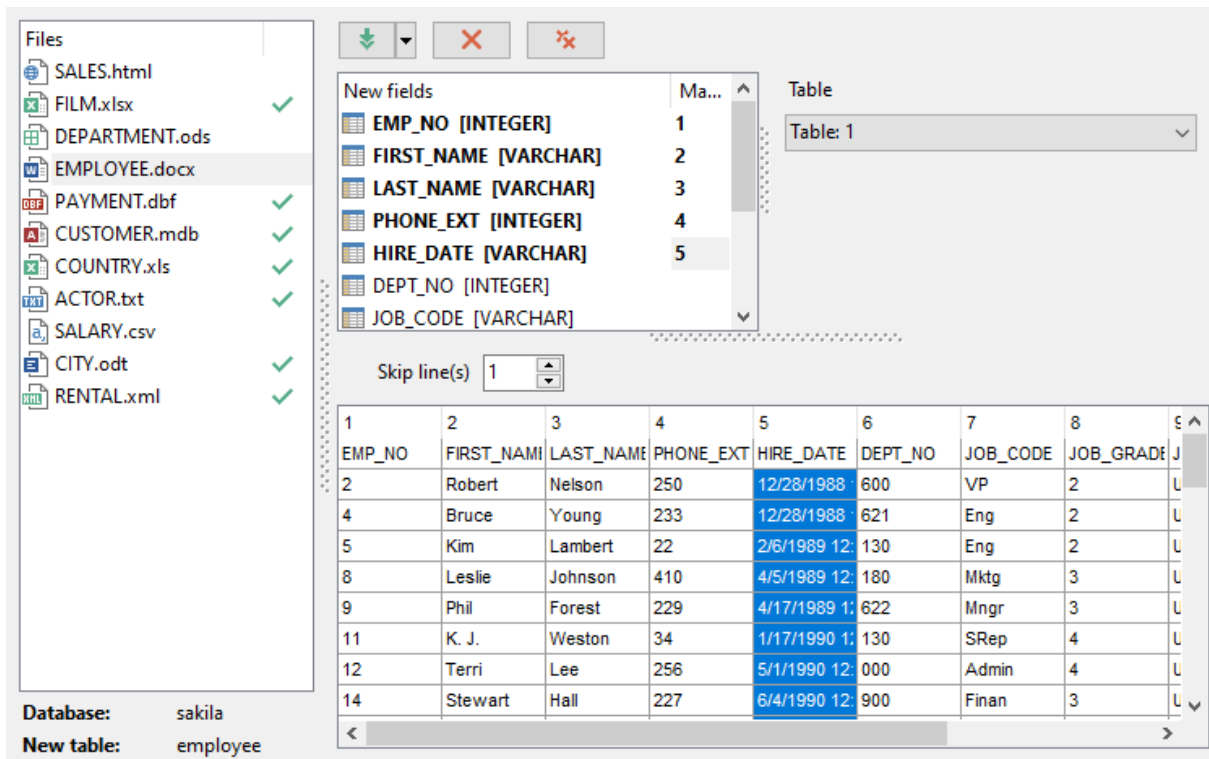
If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.


You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.




If necessary, you can choose to **skip** a defined number of the source file rows using the **Skip lines** spin-edit.

Note: If table was properly created or already exists, it will be marked with a tick . If an error occurs during table creation, this table will be marked with a cross .



To remove a correspondence, select the field in the **New fields** list and press the  **Clear** button.

To remove all correspondences, press the  **Clear All** button.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.4 MS Access

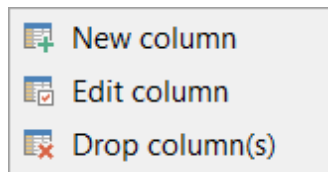
Switch between existing **Access table** and custom **Access SQL query** to retrieve data from.

Set correspondence between the source MS Access fields and the target PostgreSQL table fields:

- select a field of the target PostgreSQL table in the **Table fields** list;
- proceed to the sheet grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.



Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.

If you choose a query as the data source, you also can load a SQL query from a *.sql file or save the current query text to a file using the **Load from File...** and the **Save to File...** buttons correspondingly.

Use the **Auto Fill Columns** button to set correspondence between the source and target fields automatically on the basis of their order.

CUST_NO	CUSTOMER	CONTACT_F	CONTACT_L	PHONE_NO	ADDRESS_I	ADDRESS_I	CITY
1001	Signature Dr	Dale J.	Little	(619) 530-2	15500 Pacifi		San Diego
1002	Dallas Techn	Glen	Brown	(214) 960-2	P. O. Box 47		Dallas
1003	Buttle, Griffi	James	Buttle	(617) 488-1	2300 Newbu	Suite 101	Boston
1004	Central Ban	Elizabeth	Brocket	61 211 99 8	66 Lloyd Str		Manchester
1005	DT Systems	Tai	Wu	(852) 850 4	400 Connau		Central Hong
1006	DataServe Ir	Tomas	Bright	(613) 229 3	2000 Carling	Suite 150	Ottawa
1007	Mrs. Beauve		Mrs. Beauve		P.O. Box 22		Pebble Beac
1008	Anini Vacati	Leilani	Briggs	(808) 835-7	3320 Lawwai		Lihue
1009	Max	Max		22 01 23	1 Emerald C		Turtle Island
1010	MPM Corpor	Miwako	Miyamoto	3 880 77 19	2-64-7 Sasa		Tokyo
1011	Dynamic Inte	Victor	Granges	01 221 16 5	Florhofgass		Zurich

To remove a correspondence, select the field in the Fields list and press the **Clear** button.

To remove all correspondences, press the **Clear All** button.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.5 DBF

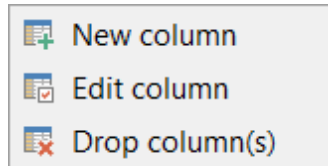
Set correspondence between the source DBF columns and the target PostgreSQL table fields:

- select a field of the target PostgreSQL table in the **Table Fields** list;

- proceed to the grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.



Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.

Use the **Auto Fill Columns** button to set correspondence between the source and target fields automatically on the basis of their order.

Charset

Use this field to specify the source file character set.

Skip deleted rows

Use the option to exclude records marked as deleted in source DBF file.

The screenshot shows the data import interface. On the left, a 'Files' list contains various source files, with 'PAYMENT.dbf' selected and marked with a green checkmark. Below the list, the database is identified as 'sakila' and the target table as 'payment'. In the center, a 'Fields' table maps source fields to target fields. On the right, the 'Charset' is set to 'None' and the 'Skip deleted rows' option is checked. At the bottom, a data preview table shows 12 rows of data.

Fields	Mapped
payment_id [SMALLINT]	PAYMENT_I
customer_id [SMALLINT]	CUSTOMER
staff_id [TINYINT]	STAFF_ID
rental_id [INTEGER]	RENTAL_ID
amount [DECIMAL]	AMOUNT
payment_date [DATETIME]	PAYMENT_I
last_update [TIMESTAMP]	LAST_UPD/

PAYMENT_ID	CUSTOMER_	STAFF_ID	RENTAL_ID	AMOUNT	PAYMENT_C	LAST_UPDAT
1	1	1	76	2.9900	25.05.2005	15.02.2006
2	1	1	573	0.9900	28.05.2005	15.02.2006
3	1	1	1185	5.9900	15.06.2005	15.02.2006
4	1	2	1422	0.9900	15.06.2005	15.02.2006
5	1	2	1476	9.9900	15.06.2005	15.02.2006
6	1	1	1725	4.9900	16.06.2005	15.02.2006
7	1	1	2308	4.9900	18.06.2005	15.02.2006
8	1	2	2363	0.9900	18.06.2005	15.02.2006
9	1	1	3284	3.9900	21.06.2005	15.02.2006
10	1	2	4526	5.9900	08.07.2005	15.02.2006
11	1	1	4611	5.9900	08.07.2005	15.02.2006
12	1	1	5244	4.9900	09.07.2005	15.02.2006

To remove a correspondence, select the field in the Fields list and press the **Clear** button.

To remove all correspondences, press the **Clear All** button.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

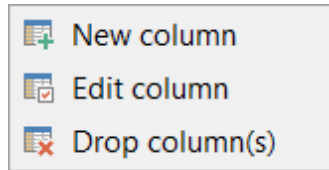
2.1.5.6 XML

Set correspondence between the source XML columns and the target PostgreSQL table fields:

- select a field of the target PostgreSQL table in the **Table Fields** list;
- proceed to the grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.

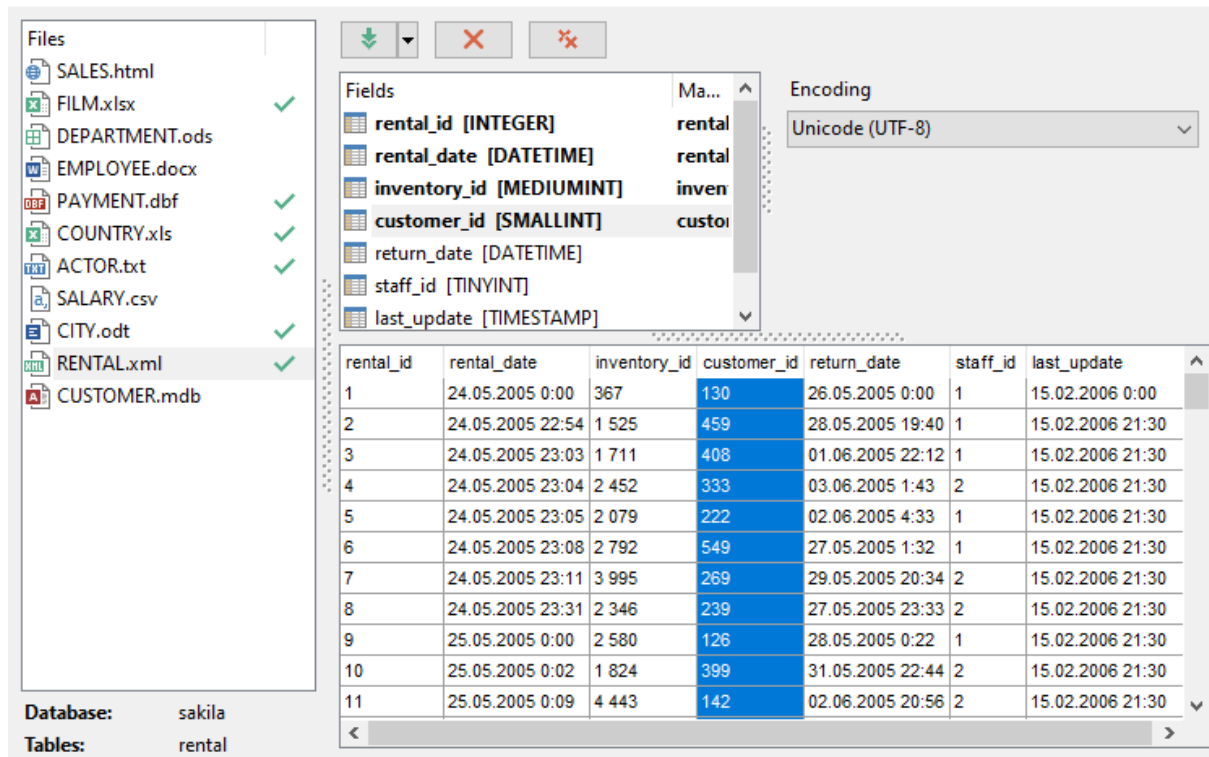


Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.

Use the **Auto Fill Columns** button to set correspondence between the source and target fields automatically on the basis of their order.

Encoding

Use this field to select the file encoding.



To remove a correspondence, select the field in the Fields list and press the **Clear** button.

To remove all correspondences, press the **Clear All** button.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.


2.1.5.7 XML Document

In order to set mapping of a Generic XML document, you should first select the desired XML Document in the **Files** list. The tree-like structure of source document is displayed in the area located to the right from the **New fields** list. Select a node in the tree to get its relative path or type the path manually in the editor below (the path must be specified in the XPath format). Upon pressing the **Fill grid** button the grid gets filled with *Sub Nodes Text* or *Attributes* values of the selected node.

Note: if the source XML document contains huge amount of data, building the tree may take a long time.

Set correspondence between the source XML file columns and the target PostgreSQL table fields:

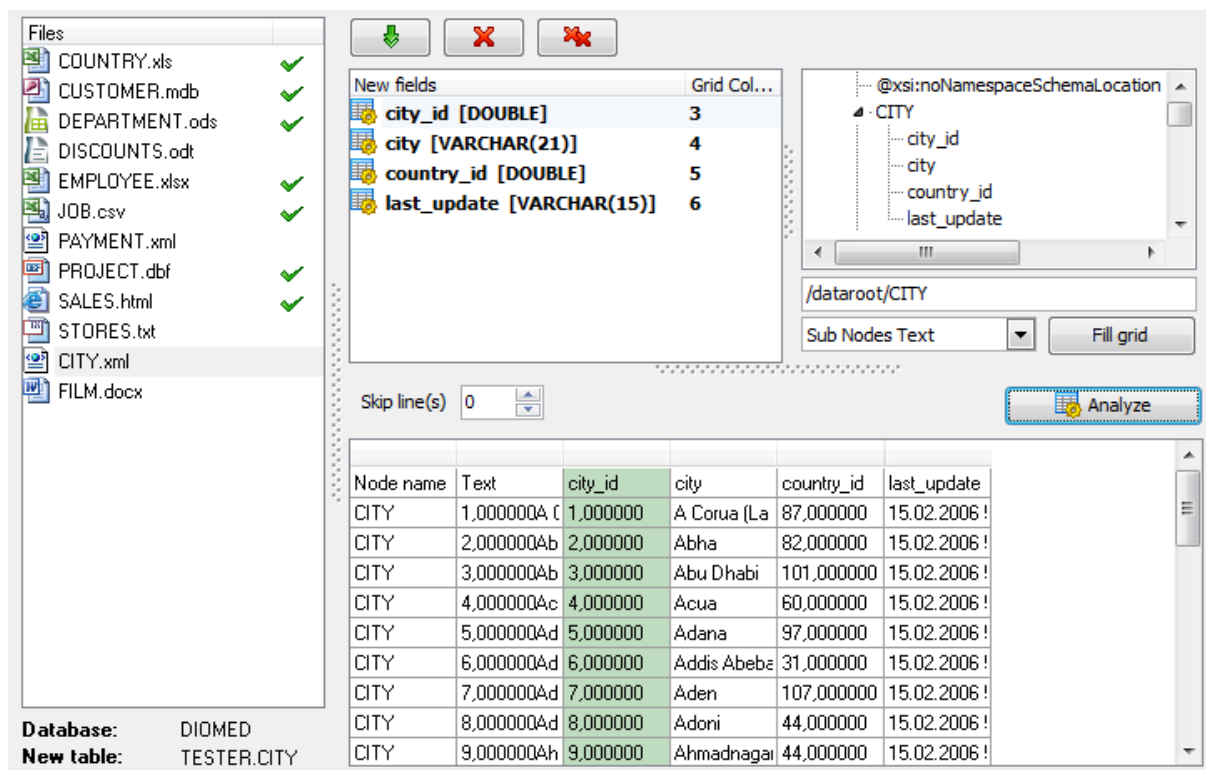
- select a field of the target PostgreSQL table in the **Fields** list;
- proceed to the source grid viewer area: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

You can use the  **Auto Fill Columns** button to set correspondence between the source and target fields automatically according to their order (mapping is started from the first attribute value in this case).

When you import data to a newly created table, the **Analyze** button appears. Use this button for automatic field creation.

The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.

If necessary, you can choose to **skip** a defined number of the source file lines using the **Skip Lines** spin-edit.




The screenshot shows the data import interface. On the left, a 'Files' list contains various files with green checkmarks. The main area shows a 'New fields' table with the following data:

New fields	Grid Col...
city_id [DOUBLE]	3
city [VARCHAR(21)]	4
country_id [DOUBLE]	5
last_update [VARCHAR(15)]	6

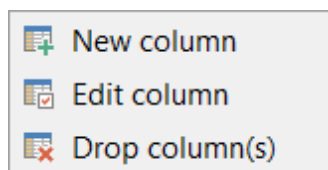
Below this is a 'Skip line(s)' spin-edit set to 0 and an 'Analyze' button. At the bottom, a data grid shows the following information:


Node name	Text	city_id	city	country_id	last_update
CITY	1,000000A	1,000000	A Corua (La	87,000000	15.02.2006!
CITY	2,000000Ab	2,000000	Abha	82,000000	15.02.2006!
CITY	3,000000Ab	3,000000	Abu Dhabi	101,000000	15.02.2006!
CITY	4,000000Ac	4,000000	Acua	60,000000	15.02.2006!
CITY	5,000000Ad	5,000000	Adana	97,000000	15.02.2006!
CITY	6,000000Ad	6,000000	Addis Abeba	31,000000	15.02.2006!
CITY	7,000000Ad	7,000000	Aden	107,000000	15.02.2006!
CITY	8,000000Ad	8,000000	Adoni	44,000000	15.02.2006!
CITY	9,000000Ah	9,000000	Ahmadnagai	44,000000	15.02.2006!

At the bottom left, the database is 'DIOMED' and the new table is 'TESTER.CITY'.

To remove all correspondences, press the  **Clear All** button.

If you choose to import data to the newly created table, at this step you need to create necessary fields. Use the corresponded context menu item to [manage fields](#)^[85].



Note: If table was properly created or already exists, it will be marked with a tick . If

an error occurs during table creation, this table will be marked with a cross **✘**.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.8 TXT

Set correspondence between the source text file columns and the target PostgreSQL table fields:

- select a field of the target PostgreSQL table in the **Fields** list;
- double-click in the text viewer area to add vertical separators delimiting the source column bounds;
- click the area between the separators to assign the column to the selected target table field - the selected source column gets black highlight;
- repeat the operation for all the fields you need to be included in the import process.


The **Fields** list also allows you to view the following values:

Pos represents the starting point of your selection;

Length displays the width of the selected area.

You can change these parameters manually or by moving the slider in the grid.

If necessary, you can choose to **skip** a defined number of the source file lines using the **Skip lines** spin-edit.

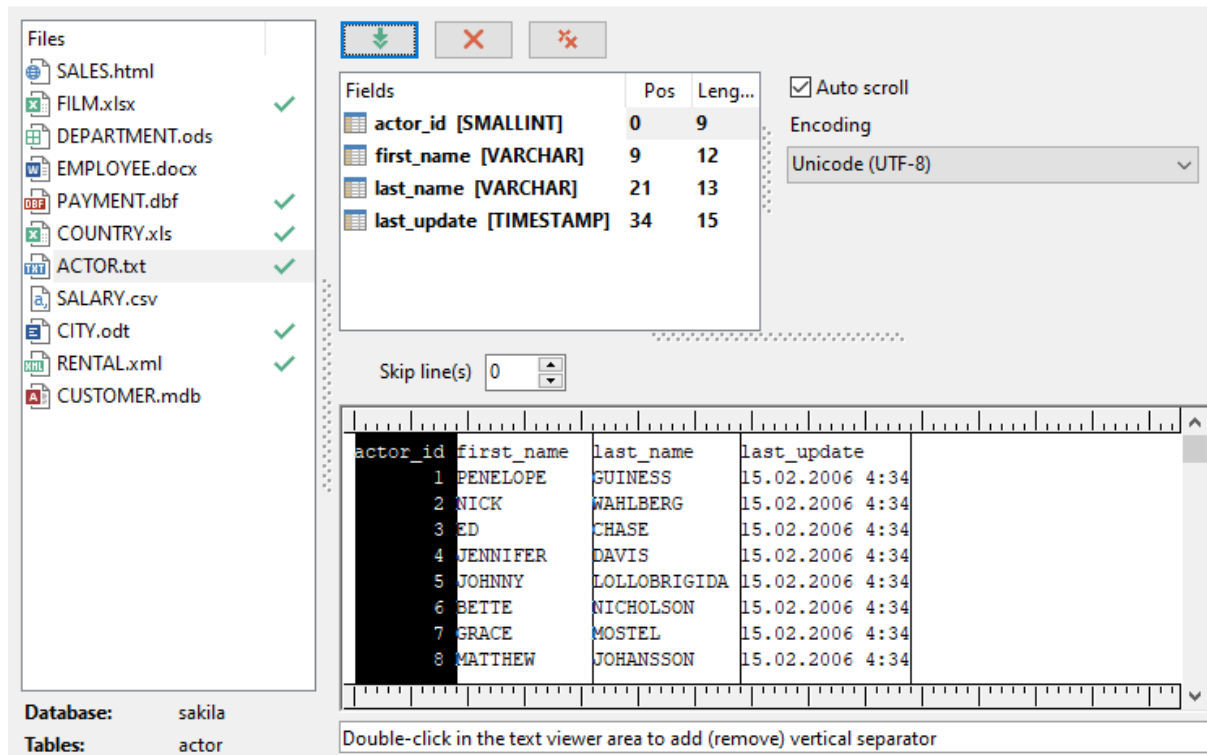
If the source text file and the destination PostgreSQL table have the same order of columns, you can use the  **Auto Fill Columns** button to set correspondence between them automatically.


The source file character set can be defined at the **Encoding** field.

Auto scroll

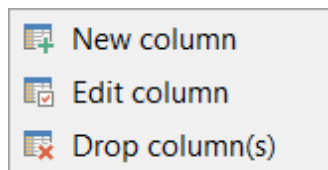
This option scrolls the document automatically when you switch to the next field for mapping.

When you import data to a newly created table, the **Analyze** button appears. Use this button for automatic field creation.



To remove all correspondences, press the  **Clear All** button.

If you choose to import data to the newly created table, at this step you need to create necessary fields. Use the corresponded context menu item to [manage fields](#)^[85].



Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.



Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.9 CSV

Set correspondence between the target table fields and the source CSV file columns:

- select the **Delimiter** and **Quote** characters for CSV files using the corresponding drop-down lists of the **CSV Parameters** group;
- select a field of the target PostgreSQL table in the **Fields** list;
- proceed to the source grid viewer area: click a caption to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

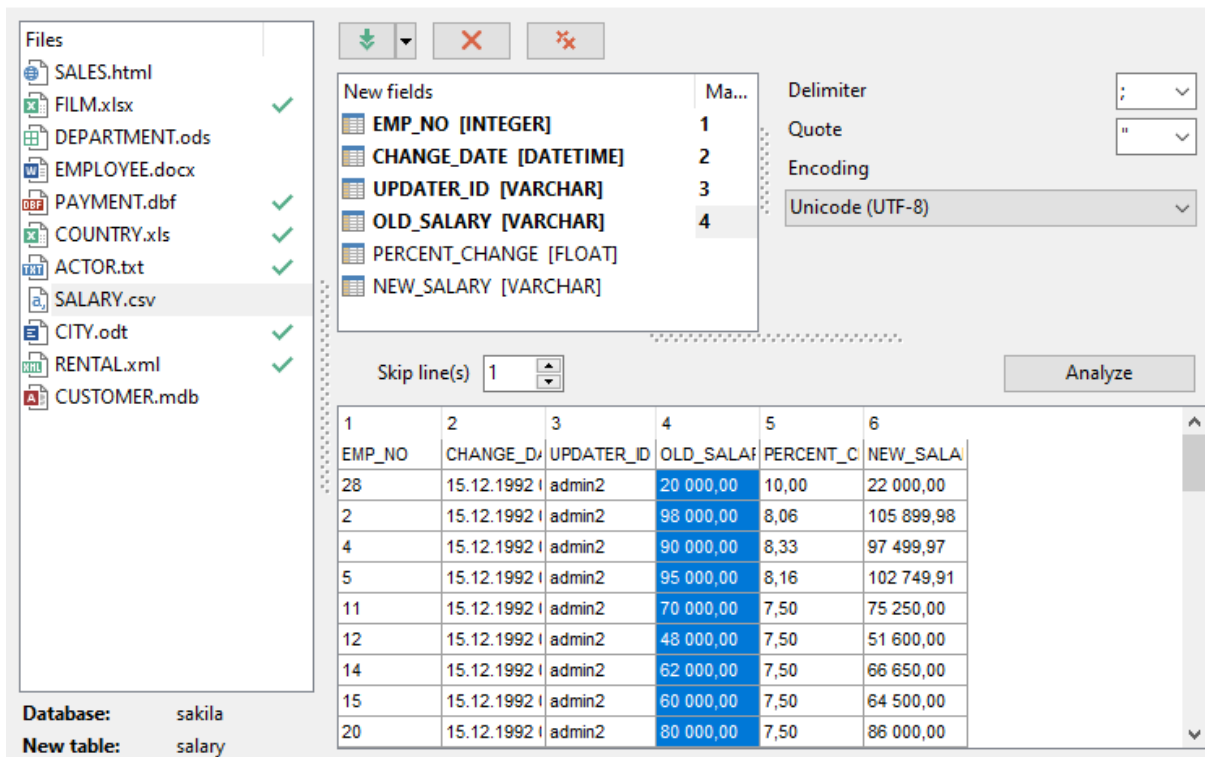
The source file character set can be defined at the **Encoding** field.

If the source file and the destination PostgreSQL table have the same order of columns, you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.

When you import data to a newly created table, the **Analyze** button appears. Use this button for automatic field creation.

If necessary, you can choose to **skip** a defined number of the source file rows using the **Skip lines** spin-edit.


The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.



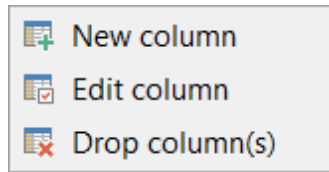
The screenshot displays the Data Import for PostgreSQL interface. On the left, a file list includes SALES.html, FILM.xlsx, DEPARTMENT.ods, EMPLOYEE.docx, PAYMENT.dbf, COUNTRY.xls, ACTOR.txt, SALARY.csv, CITY.odt, RENTAL.xml, and CUSTOMER.mdb. The 'SALARY.csv' file is selected. The 'New fields' list shows: EMP_NO [INTEGER] (Ma... 1), CHANGE_DATE [DATETIME] (2), UPDATER_ID [VARCHAR] (3), OLD_SALARY [VARCHAR] (4), PERCENT_CHANGE [FLOAT], and NEW_SALARY [VARCHAR]. The 'Skip line(s)' is set to 1. The 'Analyze' button is visible. Below, a data grid shows the following data:

1	2	3	4	5	6
EMP_NO	CHANGE_D	UPDATER_ID	OLD_SALARY	PERCENT_C	NEW_SALARY
28	15.12.1992	admin2	20 000,00	10,00	22 000,00
2	15.12.1992	admin2	98 000,00	8,06	105 899,98
4	15.12.1992	admin2	90 000,00	8,33	97 499,97
5	15.12.1992	admin2	95 000,00	8,16	102 749,91
11	15.12.1992	admin2	70 000,00	7,50	75 250,00
12	15.12.1992	admin2	48 000,00	7,50	51 600,00
14	15.12.1992	admin2	62 000,00	7,50	66 650,00
15	15.12.1992	admin2	60 000,00	7,50	64 500,00
20	15.12.1992	admin2	80 000,00	7,50	86 000,00

Database: sakila
New table: salary

To remove all correspondences, press the  **Clear All** button.

If you choose to import data to the newly created table, at this step you need to create necessary fields. Use the corresponded context menu item to [manage fields](#)⁸⁵.



Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.


Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.


2.1.5.10 HTML

Set correspondence between the target table fields and the source HTML file columns:

- select a field of the target PostgreSQL table in the **New fields** list;
- proceed to the source grid viewer area: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have

different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.

The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.

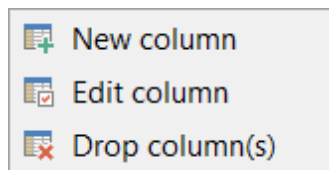


Table number


Select the number of the table for retrieving data from the dropdown menu. Can be used if the HTML file consists more than one table.

If necessary, you can choose to **skip** a defined number of the source file rows using the **Skip lines** spin-edit.

Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.

Database: sakila
New table: sales

1	2	3	4	5	6	7	8	9
PO_NUMBER	CUST_NO	SALES_REP	ORDER_STA	ORDER_DA1	SHIP_DATE	DATE_NEED	PAID	C
V91E0210	1,004	11	shipped	3/4/1991 12:	3/5/1991 12:	null	y	1
V92E0340	1,004	11	shipped	10/15/1992	10/16/1992	10/17/1992	y	7
V92J1003	1,010	61	shipped	7/26/1992 1:	8/4/1992 12:	9/15/1992 1:	y	1
V93J2004	1,010	118	shipped	10/30/1993	12/2/1993 1:	11/15/1993	y	3
V93J3100	1,010	118	shipped	8/20/1993 1:	8/20/1993 1:	null	y	1
V92F3004	1,012	11	shipped	10/15/1992	1/16/1993 1:	1/16/1993 1:	y	3
V93F3088	1,012	134	shipped	8/27/1993 1:	9/8/1993 12:	null	n	1
V93F2030	1,012	134	open	12/12/1993	null	null	y	1

To remove all correspondences, press the  **Clear All** button.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.


2.1.5.11 ODS


The **OpenDocument** format is used by Mobile Office, as well as other well-known desktop applications, such as OpenOffice, StarOffice and KOffice. **ODS** stands for **OpenDocument Spreadsheet** (*.ods).

Specify ranges in the grid for the target and source fields:

- select a field of the target PostgreSQL table in the **New fields** list;
- proceed to the sheet grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

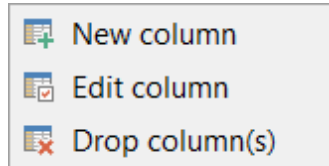
you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have

different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.

The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.

If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.

You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.



Table


Select the sheet for importing data from.


If necessary, you can choose to skip a defined number of the source file rows using the **Skip line(s)** spin-edit.

Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.

The screenshot shows the data import interface. On the left, a 'Files' list contains various source files, each with a green checkmark indicating successful import. The 'Database' is 'sakila' and the 'New table' is 'department'. The 'New fields' list includes: DEPT_NO [INTEGER], DEPARTMENT [VARCHAR], HEAD_DEPT [INTEGER], MNGR_NO [INTEGER], BUDGET [INTEGER], LOCATION [VARCHAR], and PHONE_NO [VARCHAR]. The 'Skip line(s)' spinner is set to 1. The 'Table' dropdown is set to 'Sheet1'. Below, a data preview table shows the following data:

A	B	C	D	E	F	G
DEPT_NO	DEPARTMENT	HEAD_DEPT	MNGR_NO	BUDGET	LOCATION	PHONE_NO
000	Corporate H	null	105	1000000	Monterey	(408) 555-1:
100	Sales and M	000	85	2000000	San Francis	(415) 555-1:
600	Engineering	000	2	1100000	Monterey	(408) 555-1:
900	Finance	000	46	400000	Monterey	(408) 555-1:
180	Marketing	100	null	1500000	San Francis	(415) 555-1:
620	Software Pr	600	null	1200000	Monterey	(408) 555-1:
621	Software De	620	null	400000	Monterey	(408) 555-1:
622	Quality Asst	620	9	300000	Monterey	(408) 555-1:
623	Customer St	620	15	650000	Monterey	(408) 555-1:

To remove a correspondence, select the field in the **New fields** list and press the  **Clear** button.

To remove all correspondences, press the  **Clear All** button.

If your OpenDocument Spreadsheet file contains several sheets, you are able to set

different mapping for each sheet.

Click the **Next** button to proceed to the [Setting base data formats](#)^[50] step of the wizard.

2.1.5.12 ODT

The **OpenDocument** format is used by Mobile Office, as well as other well known desktop applications such as OpenOffice, StarOffice and KOffice. **ODT** stands for **OpenDocument Text** (.odt).


Table


Select the sheet for importing data from.

Specify ranges in the grid for the target and source fields:

- select a field of the target PostgreSQL table in the **New fields** list;
- proceed to the **Table** grid: click a column to assign the column to the selected target table field;
- the selected column of the source file gets gray highlight;
- repeat the operation for all the fields you need to be included in the import process.

If the source file and the destination PostgreSQL table have the same order of columns,

you can use the  **Auto Fill Columns** mode (default) to set correspondence between them automatically. If source file and destination PostgreSQL table have

different order of columns but identical names you can use the  **Auto Fill by Captions** mode to set the correspondence based on name's identity.

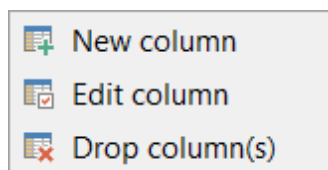
Table

In case the file contains several tables select the require one from the dropdown menu.

The number of the column which correspondence is set for are displayed at the **Grid Column** column. You can use it for setting the fields correspondence as well.

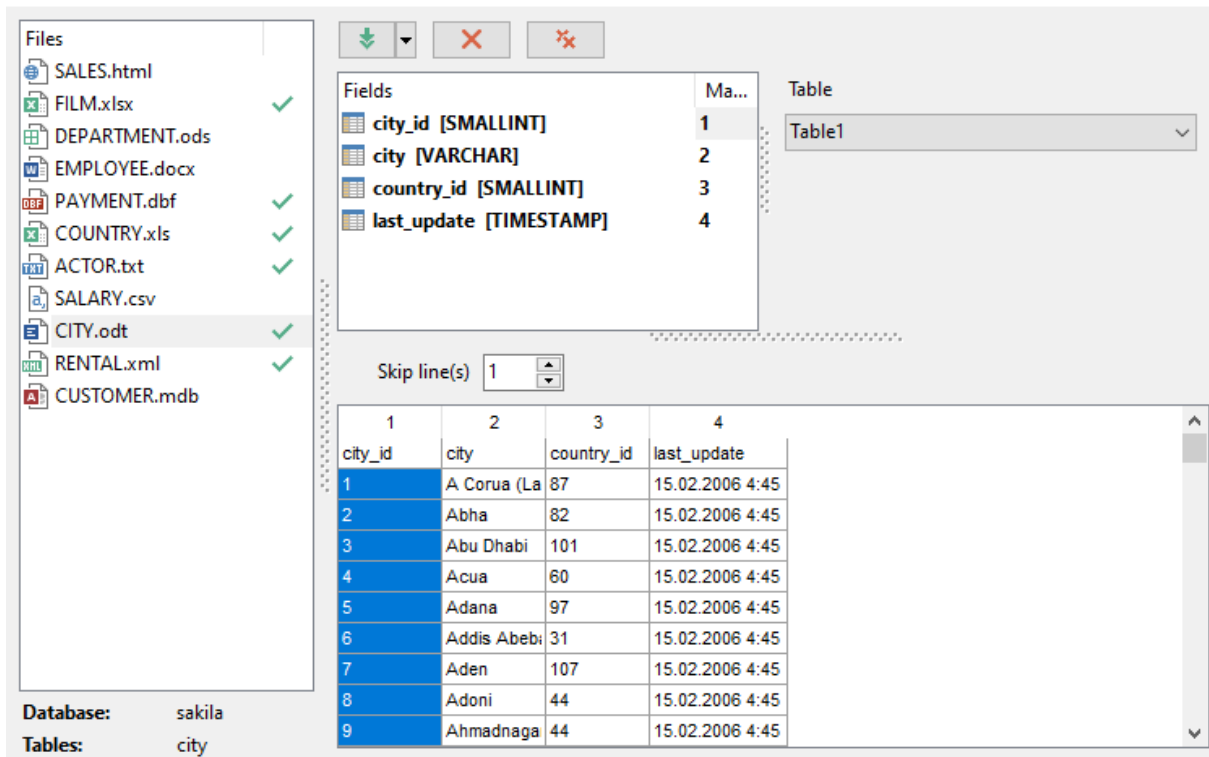
If new table creation was selected at the [previous step](#)^[27], then fields needed for the import procedure will be created automatically.


You can [manage destination table fields](#)^[85] if needed. Use the corresponded context menu item to **New/Edit/Drop** field.



If necessary, you can choose to skip a defined number of the source file rows using the **Skip line(s)** spin-edit.

Note: If table was properly created or already exists, it will be marked with a tick ✓. If an error occurs during table creation, this table will be marked with a cross ✗.



To remove a correspondence, select the field in the **Fields** list and press the  **Clear** button.

To remove all correspondences, press the  **Clear All** button.

If your OpenDocument Text file contains several tables, you are able to set different mapping for each table.

Click the **Next** button to proceed to the [Setting base data formats](#) ^[50] step of the wizard.

2.1.6 Step 5 - Setting base data formats

This step of the wizard provides a number of options for setting **base formats** for each source data file. The specified format should match the source data representation.

The screenshot shows the 'Data Import for PostgreSQL' interface. On the left, a 'Files' list contains several files, with 'COUNTRY.xls' selected. The main area is divided into three sections: 'Separators', 'Constants', and 'Date/Time formats'. The 'Separators' section has input fields for Decimal (comma), Thousand (#160), Date (period), and Time (colon). The 'Constants' section has input fields for Boolean true (True), Boolean false (False), and Null values (Null). The 'Date/Time formats' section has dropdown menus for Date (dd.MM.yyyy) and Time (h:mm). At the bottom, the 'Database' is set to 'DIOMED' and the 'Table' is 'TESTER.COUNTRY'. There is a 'Skip this step' checkbox.

Separators

These options set the separator for the source file to ensure correct import. Make sure that they strongly correspond to the source file data.

Decimal

Set a character to delimit the decimal parts of the imported numbers.

Thousand

Set a character to separate the digit groups in the imported numbers.

Date

Set a character to separate the year, month and day parts of date values.

Time

Set a character to separate the hour, minute and second parts of time values.

Constants

Set the values as they are represented in the source file for correct recognition.

Boolean True

Set one or more variants of TRUE value representation in the imported table, e.g. 'Yes' or '+'. Use a new line for each additional option.

Boolean False

Set one or more variants of FALSE value representation in the imported table, e.g. 'No' or '-'. Use a new line for each additional option.

Null Values

Set one or more variants of NULL value representation in the imported table, e.g. 'Null'.

Use a new line for each additional option.

Date/Time formats

Use these fields to set date and time formats so that they strongly correspond to the source file format for correct processing of data.

For more information refer to the [Format specifiers](#)^[52] page.

NOTE: Date and Time formats only reflect data format view, not separators. Separators must be set in the **Separators** group above.

Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#)^[69] section of the [Preferences](#)^[68] dialog.

When you are done, press the **Next** button to proceed to the [next step](#)^[55].

2.1.6.1 Format specifiers

The following format specifiers are supported in the format string:

Float/Integer format

0

Digit place holder. If the value being formatted has a digit in the position where the '0' appears in the format string, then that digit is copied to the output string. Otherwise, a '0' is stored in that position in the output string.

#

Digit placeholder. If the value being formatted has a digit in the position where the '#' appears in the format string, then that digit is copied to the output string. Otherwise, nothing is stored in that position in the output string.

.

Decimal point. The first '.' character in the format string determines the location of the decimal separator in the formatted value; any additional '.' characters are ignored.

,

Thousand separator. If the format string contains one or more ',' characters, the output will have thousand separators inserted between each group of three digits to the left of the decimal point. The placement and number of ',' characters in the format string does not affect the output, except to indicate that thousand separators are wanted.

E+

Scientific notation. If any of the strings 'E+', 'E-', 'e+', or 'e-' are contained in the format string, the number is formatted using scientific notation. A group of up to four '0' characters can immediately follow the 'E+', 'E-', 'e+', or 'e-' to determine the minimum number of digits in the exponent. The 'E+' and 'e+' formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents. The 'E-' and 'e-' formats output a sign character only for negative exponents.

Date/Time format

c

Displays the date using the format using the Short Date Format, followed by the time using the Long Time Format. The time is not displayed if the date-time value indicates midnight precisely.

d

Displays the day as a number without a leading zero (1-31).

dd

Displays the day as a number with a leading zero (01-31).

ddd

Displays the day as an abbreviation (Sun-Sat) using the strings of the Short Day Names.

dddd

Displays the day as a full name (Sunday-Saturday) using the strings of the Long Day Names.

dddddd

Displays the date using the Short Date Format.

ddddddd

Displays the date using the Long Date Format.

e

Displays the year in the current period/era as a number without a leading zero (Japanese, Korean and Taiwanese locales only).

ee

Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).

g

Displays the period/era as an abbreviation (Japanese and Taiwanese locales only).

gg

Displays the period/era as a full name. (Japanese and Taiwanese locales only).

m

Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.

mm

Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.

mmm

Displays the month as an abbreviation (Jan-Dec) using the strings given of the Short Month Names.

mmmm

Displays the month as a full name (January-December) using the strings of the Long Month Names.

yy

Displays the year as a two-digit number (00-99).

yyyy

Displays the year as a four-digit number (0000-9999).

h

Displays the hour without a leading zero (0-23).

hh

Displays the hour with a leading zero (00-23).

n

Displays the minute without a leading zero (0-59).

nn

Displays the minute with a leading zero (00-59).

s

Displays the second without a leading zero (0-59).

ss

Displays the second with a leading zero (00-59).

z

Displays the millisecond without a leading zero (0-999).

zzz

Displays the millisecond with a leading zero (000-999).

t

Displays the time using the Short Time Format.

tt

Displays the time using the Long Time Format.

am/pm

Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.

a/p

Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.

ampm

Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.

/

Displays the date separator character using the Date Separator.

:

Displays the time separator character using the Time Separator.

'xx'/"xx"

Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

2.1.7 Step 6 - Settings data formats for each field

This step of the wizard allows you to set **formats** for each imported field separately, in case additional formatting is required.

Select a field in the list and use the **Field Customization** group to adjust format options that will be applied to this field only: *generator value*, *generator step*, *constant value*, *NULL value*, *default value*, *function*, *script*, *left/right quotation*, *quotation action*, *character case*, *character set*.

Generator value

Use this field to set the initial value of the autoincrement field.

Generator step

Set the step of the autoincrement field. If it is 0 or if the field is of *Identity* one, then the value of the generator will be ignored.

Constant value

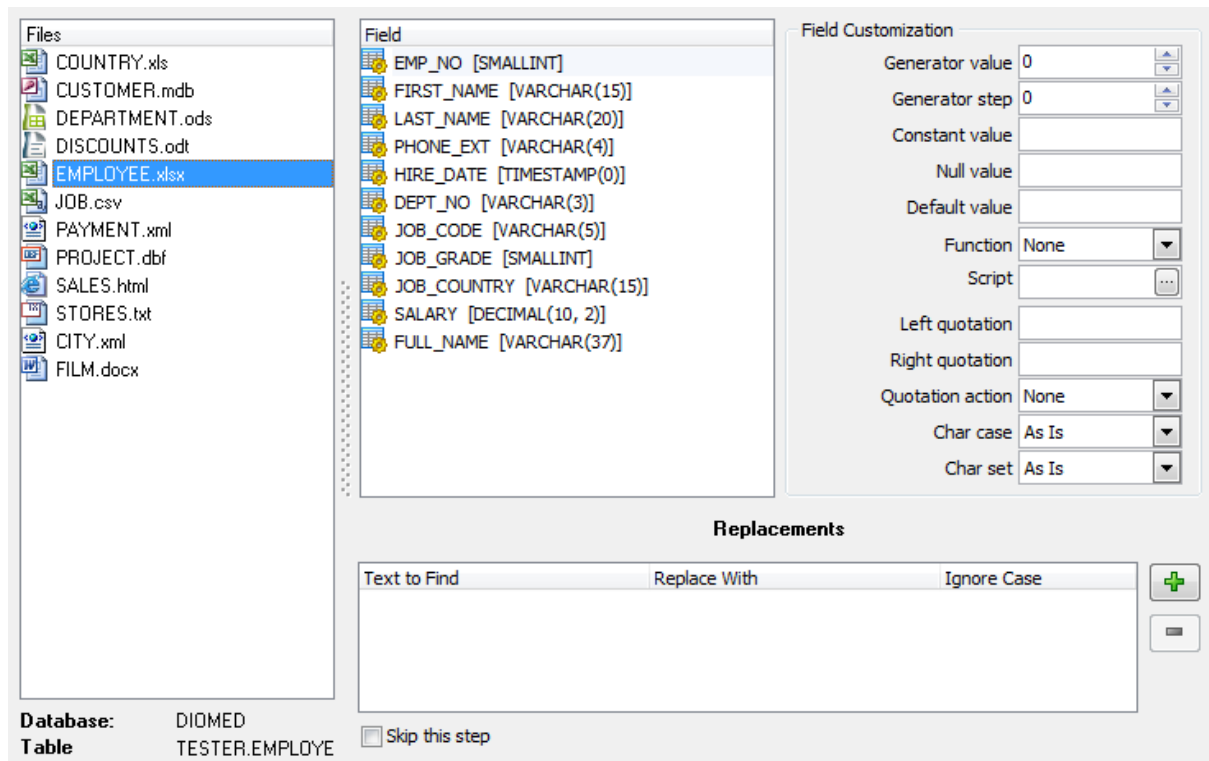
Use this field to set a constant value for the field.

Null value

Set the value which will be considered as NULL to set the default value.

Default value


Set the default value of the NULL field.

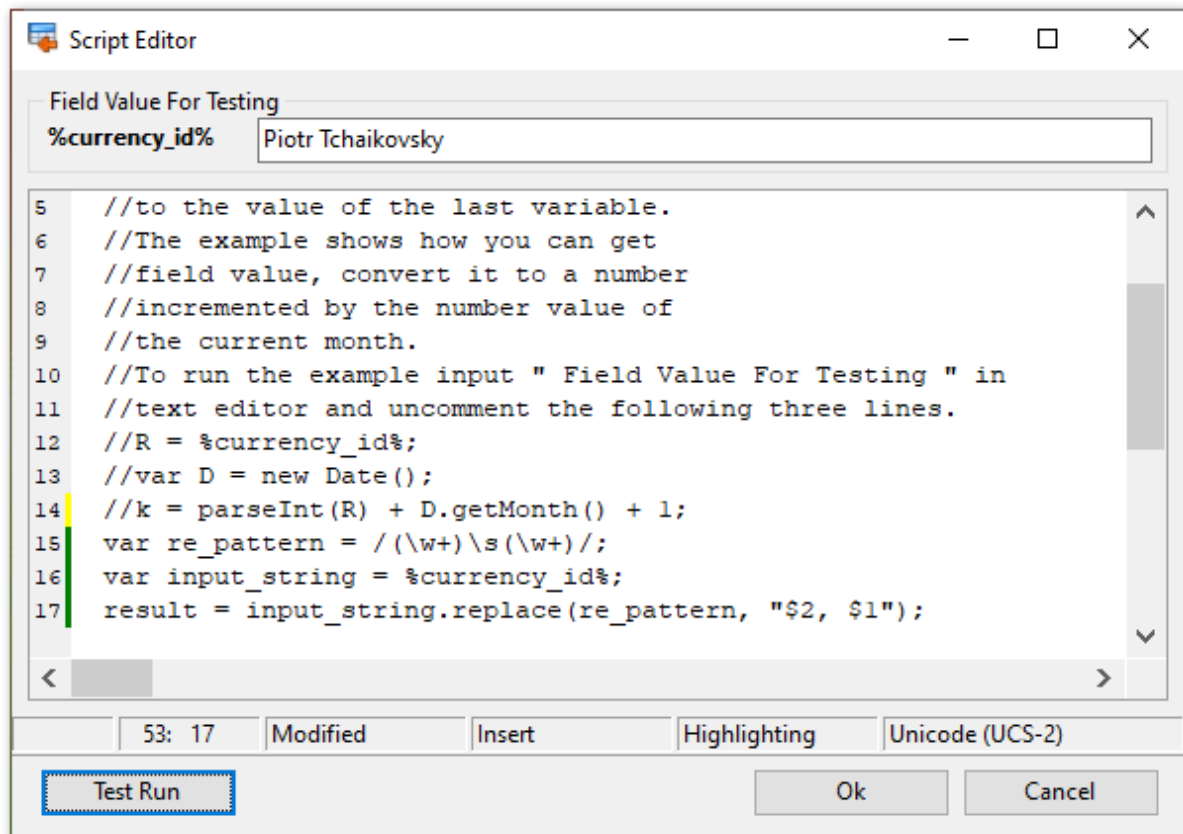


Function

Use the drop-down list to select one of the function return values that can be set in the field: *Date, Time, Date&Time, Long File Name, Short File Name*.

Script

Press the  button to add/edit the script in the **Script Editor**.



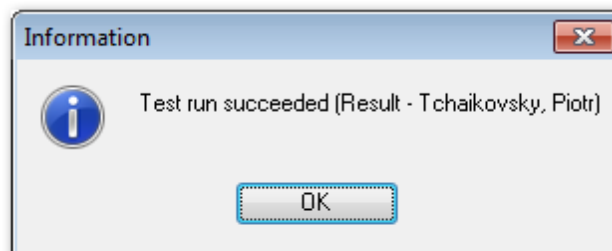
Use the **Script Editor** window to enter the script for the field processing. Here you can use all features of the MS JScript.

The picture above shows the example of the Jscript use for the field *Name* processing. The entered code converts the 'XXXX YYYY' string to the 'YYYY, XXXX' one, for example 'Piotr Tchaikovsky' -> 'Tchaikovsky, Piotr'.

If you want to test your code and see the result of its execution then enter the test field value in the **Field Value For Testing** box and click the **Test Run** button.

If the execution fails you see the message box with the description of an occurred mistake.

If the execution succeeds you see the message box with the result as it is shown on the picture below.



Quotation

Left quotation

Set a character or a number of characters to denote quoting in the imported string.

Right quotation

Set a character or a number of characters to denote unquoting in the imported string.

Quotation action


You can select the *Add* item to add quotation marks to the imported string, the *Remove* item to remove all the quotation marks from the imported string, or the *None* item to save the original quotation marks.

String conversion**Char case**

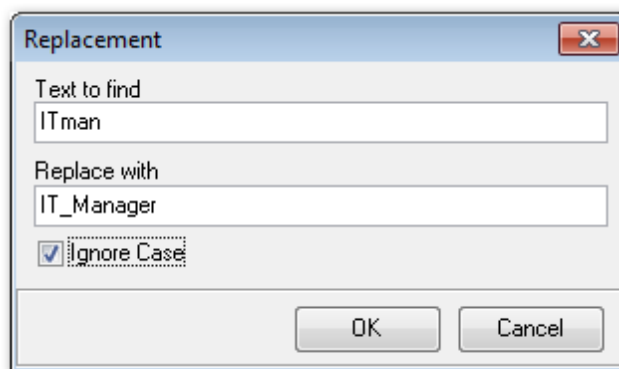
Set the case of the imported string. *As Is* keeps the original string unchanged, *Upper* sets the whole string to the upper case, *Lower* sets the whole string to the lower case, *UpperFirst* sets the first character of the string to the upper case, *UpperFirstWord* sets the first character of each word to the upper case.


Char set

Set the char set of the imported string to *ANSI* or *OEM*. *As Is* saves the original character set of the string.

The **Replacements** area allows you to set the text you need to be replaced during data import into the selected field. Press the **Add Replacement...**  button to specify a new replacement options using the **Add Replacement** dialog.

Define the text to replace and the value to replace with in the appeared dialog window. Check the **Ignore Case** option to make replacement case-insensitive.



To remove a replacement, select it in the list and click the **Delete Replacement...**  button.

Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#) section of the [Preferences](#) dialog.

When you are done, press the **Next** button to proceed to the [next step](#).

2.1.8 Step 7 - Specifying import mode

This step of the wizard allows you to define **the records processing mode** as *Insert All*, *Insert New*, *Update*, *Update or Insert*, *Delete*, *Delete or Insert* mode.

Import Mode

- **Insert all:** all records from the source file are inserted into the tables irrespective of whether any records exist in the destination table or not
- **Insert new:** already existing records are skipped, and new records are inserted into the destination table
- **Update:** all existing records are updated from the source file
- **Update or insert:** already existing records are updated and new records are inserted into the destination table
- **Delete:** already existing records are deleted
- **Delete or insert:** existing records are deleted and new records are inserted into the destination table

Here is an **example** of some import modes offered by Data Import utility:

DB Table		Source file	
ID	DATA	A	B
1	a	1	c
2	b	2	d
4	f	3	e

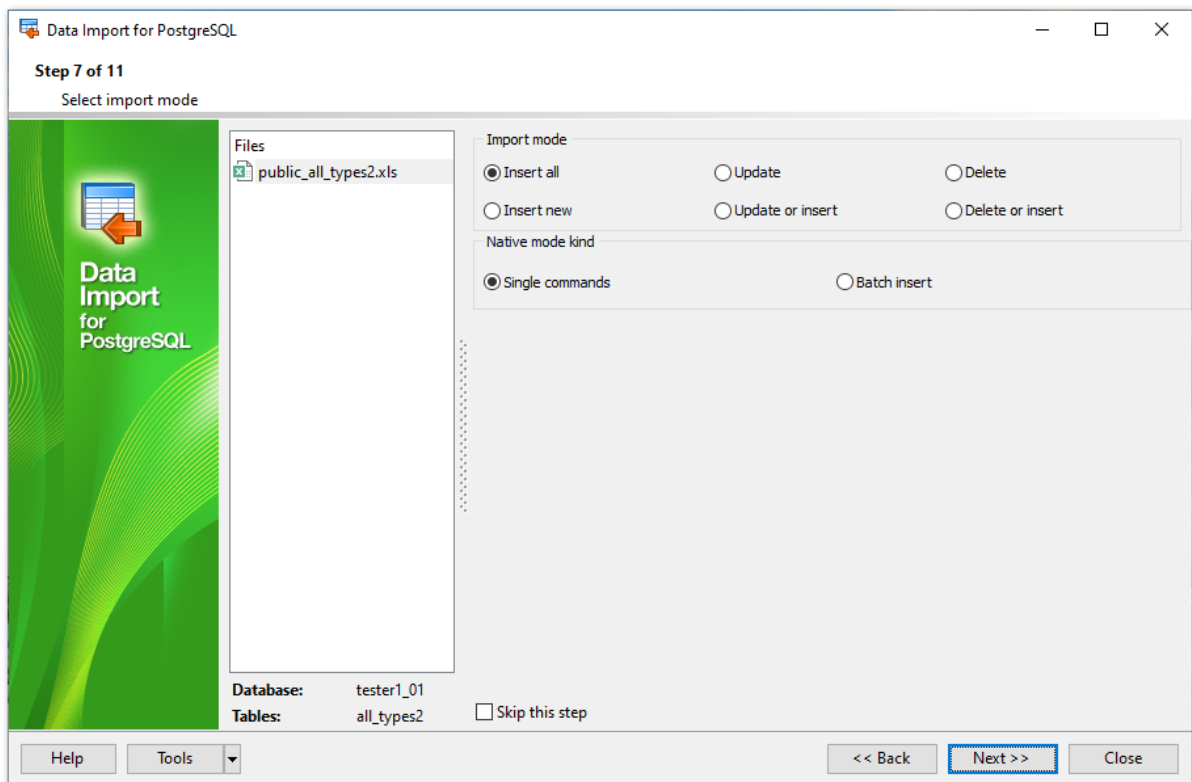
All import modes (except for the **Insert All** mode) are based on primary key values information. In order to perform import operations with these modes used you need to have matches between the source file primary key column(s) and the destination table primary key column(s).

Insert mode	Insert all	Insert new	Update																																
Result	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>c</td> </tr> <tr> <td>1</td> <td>a</td> </tr> <tr> <td>2</td> <td>b</td> </tr> <tr> <td>2</td> <td>d</td> </tr> <tr> <td>3</td> <td>e</td> </tr> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	1	c	1	a	2	b	2	d	3	e	4	f	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>a</td> </tr> <tr> <td>2</td> <td>b</td> </tr> <tr> <td>3</td> <td>e</td> </tr> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	1	a	2	b	3	e	4	f	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>c</td> </tr> <tr> <td>2</td> <td>d</td> </tr> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	1	c	2	d	4	f
ID	DATA																																		
1	c																																		
1	a																																		
2	b																																		
2	d																																		
3	e																																		
4	f																																		
ID	DATA																																		
1	a																																		
2	b																																		
3	e																																		
4	f																																		
ID	DATA																																		
1	c																																		
2	d																																		
4	f																																		

Insert mode	Update or insert	Delete	Delete or insert																				
Result	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>c</td> </tr> <tr> <td>2</td> <td>d</td> </tr> <tr> <td>3</td> <td>e</td> </tr> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	1	c	2	d	3	e	4	f	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	4	f	<table border="1"> <thead> <tr> <th>ID</th> <th>DATA</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>e</td> </tr> <tr> <td>4</td> <td>f</td> </tr> </tbody> </table>	ID	DATA	3	e	4	f
ID	DATA																						
1	c																						
2	d																						
3	e																						
4	f																						
ID	DATA																						
4	f																						
ID	DATA																						
3	e																						
4	f																						

It is applied to all other import modes, except for the **Insert all** mode. For all these modes (except for the **Insert all** mode) it is obligatory to select the primary key fields. This field (or fields) is used as key field to identify specific data in the target database.

The key columns for these operations are defined at the [Selecting key columns](#)^[61] step of the wizard.



If the *Update* value is selected for Import Mode, then you need to specify the Primary Key field(s) that will serve for data identification of your source file with the data of the target table. The source file must contain the column(s) that will correspond to the Primary Key field of the target table. If the target table contains a record in which the value of the primary key field coincides with the value of the corresponding column of the source table, the data stored in this table record will be updated. You can specify the field(s) which will be used for identification at [Step 7](#)^[61] of the wizard. This field(s) will only serve for identification and will not be imported.

Native mode kind

The *Native* mode uses the *Single Commands* method that serves to generate and execute single SQL commands on the server.

If necessary, you can use the *Batch Insert* mode to import a data set as a batch.

Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#)^[69] section of the [Preferences](#)^[68] dialog.

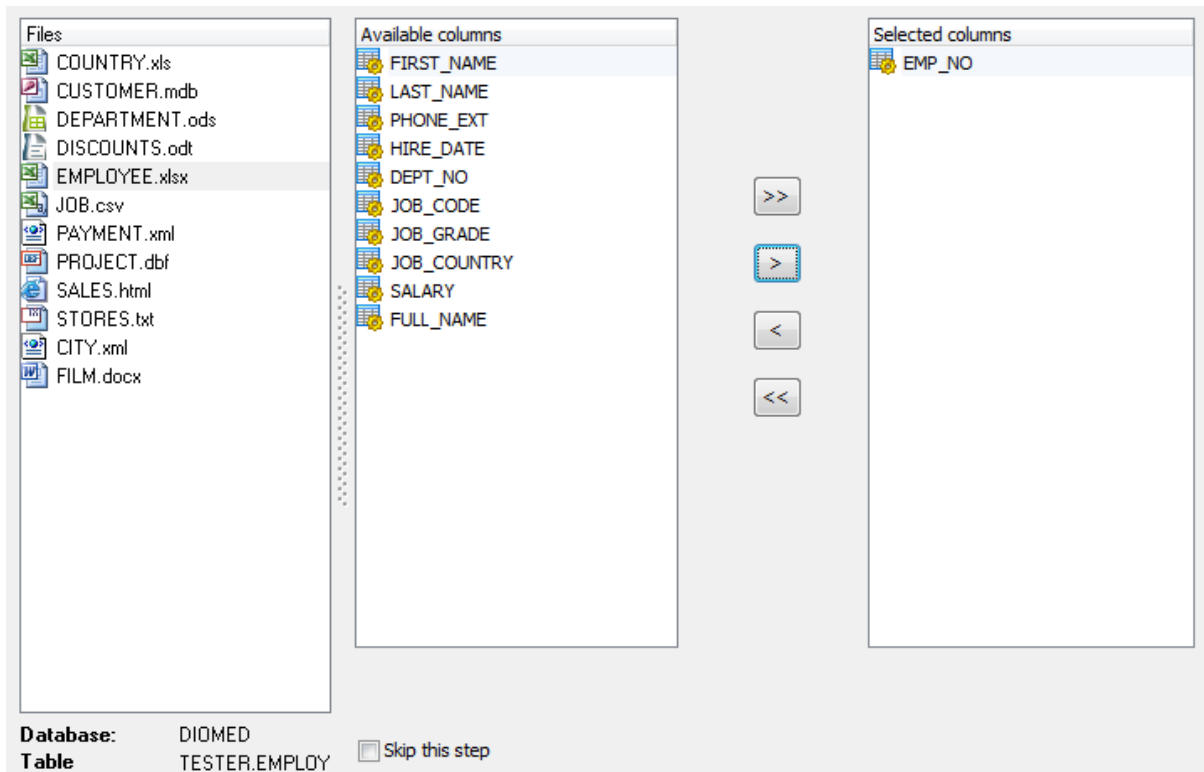
When you are done, press the **Next** button to proceed to the [next step](#)^[61].

2.1.9 Step 8 - Selecting key columns

This step of the wizard allows you to select the fields of the table to be used as the **key fields** for the import process.

This step is not available if the *Insert all* option has been selected on the [previous step](#)^[59].

To select a field, you need to move it from the **Available Columns** list to the **Selected Columns** list. Use the , , , buttons or drag-and-drop operations to move the fields from one list to another.



Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#)^[69] section of the [Preferences](#)^[68] dialog.

dialog.

When you are done, press the **Next** button to proceed to the [next step](#)⁶².

2.1.10 Step 9 - Setting common options

Using this step of the wizard you can set **final import options**.

Commit

Commit when done

Check this option to commit the transaction after all records are imported.

Commit after each ... records

These control allows you to define the number of records in each block to be supplemented with the COMMIT statement.

Rollback on error

Check this option to rollback the transaction if an error occurred.

Record Count

Import all records

Specifies that all records of the source file will be imported.

Import only ... record(s)

Specifies the number of records to be imported.

Import empty rows


If this option is checked then empty records are imported.

Auto trim values

Check this function to remove all leading and trailing white-space characters from all imported records.

Save Result SQL Script to File

This option allows you to save the result SQL script of the import operation to an external *.sql file on your disk drive.

Specify the script file name and select its location using the  button which calls the **Save As...** dialog.

Files

- COUNTRY.xls
- CUSTOMER.mdb
- DEPARTMENT.ods
- DISCOUNTS.odt
- EMPLOYEE.xlsx
- JOB.csv
- PAYMENT.xml
- PROJECT.dbf
- SALES.html
- STORES.txt
- CITY.xml
- FILM.docx

Commit

Commit when done

Commit after each records

Rollback on error

Record Count

Import all records

Import only records

Import empty rows

Auto trim values

Save result SQL script to file

C:\EMS\docs\COUNTRY.sql

Database: DIOMED

Table: TESTER.COUNTR

Skip this step

Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#)^[69] section of the [Preferences](#)^[68] dialog.

When you are done, press the **Next** button to proceed to the [next step](#)^[63].

2.1.11 Step 10 - Defining scripts

This step of the wizard allows you to define scripts to be **executed before and after import process** for each database.

Select a database to define the script for. Type the text of the script to be executed before the import operation in the **Before Import Script** area, the script to be executed after the import operation in the **After Import Script** area.

Select another database and add scripts for it, if necessary. By default, the After Import Script is not executed if import fails. If you need to execute it anyway, use the **Execute after import script on import fail** option at [Step 11](#)^[64].

You can also save and load *Before Import* and *After Import* scripts using the corresponding **Save...** and **Load...** buttons.

The screenshot shows two text input fields for SQL scripts. The top field, titled "Before import script", contains the following text:

```
1 /*Specify the script that will executed
2 BEFORE data import*/
```

Below this field are three buttons: "Load...", "Save...", and "Clear".

The bottom field, titled "After import script", contains the following text:

```
1 /*Specify the script that will executed
2 AFTER data import*/
```

Below this field are three buttons: "Load...", "Save...", and "Clear".

At the bottom of the dialog is a checkbox labeled "Skip this step".

Skip this step

Check this option to skip the current step in the future. To edit the list of skipped steps, use the **Skipped Steps** group available in the [General](#)^[69] section of the [Preferences](#)^[68] dialog.

When you are done, press the **Next** button to proceed to the [last step](#)^[64].

2.1.12 Step 11 - Start of data import process

This step of the wizard is intended to inform you that all import options have been set, and you can start the import process. If everything is correct, press the **Import** button to start the process. If you want to change something, you can return to any of the wizard steps using the **Back** button.

Continue if [before import](#)^[63] script is executed with errors

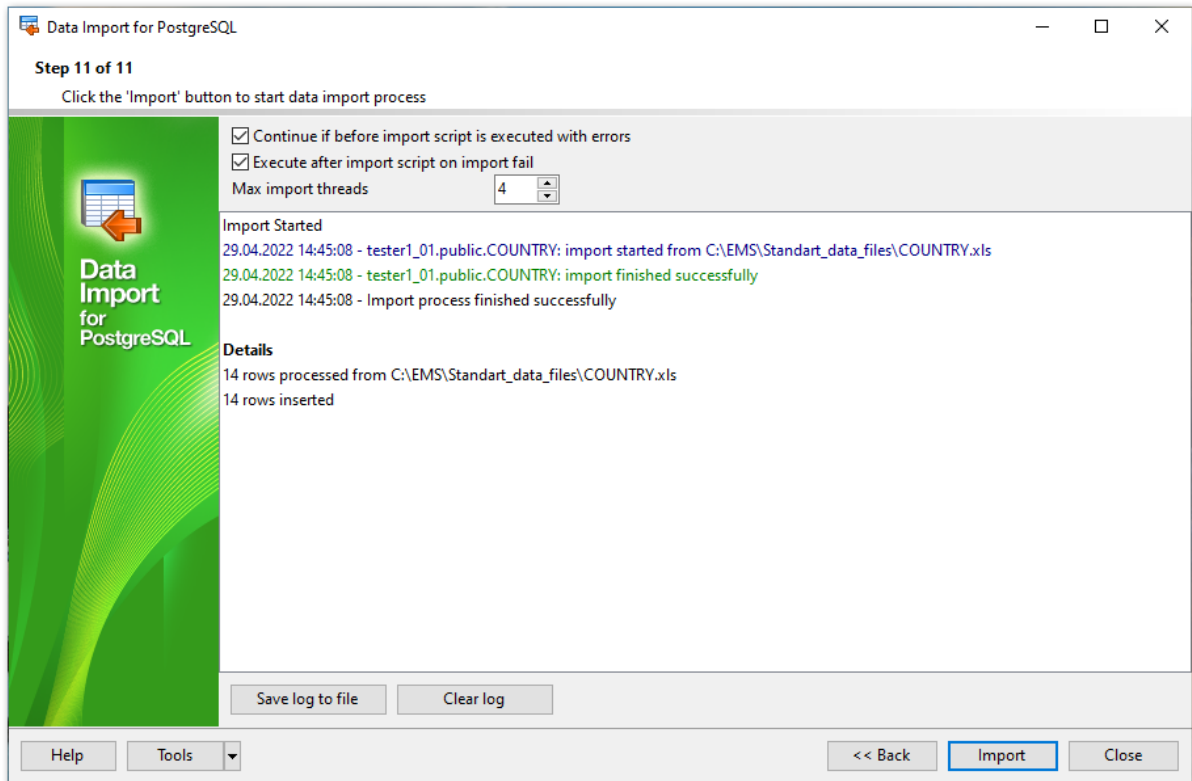
Check this option to ignore errors in *Before Script* execution during import.

Execute after import script on import fail

Check this option to execute After import script, specified at [Step 10](#)^[63], not depending on import result. Otherwise the script is not executed if import fails.

Max import threads

Set the value from 0 to 64 to adjust import performance.



Save log to file

This button calls the **Save file** dialog which allows you to save the on-screen log to a file.

Clear log

Pressing this button clears the on-screen log area, removing all messages.

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

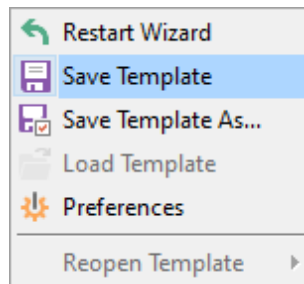
Click the **Import** button to start the import process.

2.2 Using data import configuration files


Data Import for PostgreSQL allows you to store its configuration settings in external *.itm template files if you need to perform the data import process repeatedly.

You can load previously saved configuration settings to the application [wizard](#)^[22] if you need to make some changes before data comparison, or you can run it with the [console application](#)^[76] for quicker data import.

Data Import templates are saved/loaded within the **Save template options/Load template** dialog. To open this dialog, press the **Tools** button and select the **Save template/Load template** popup menu item.

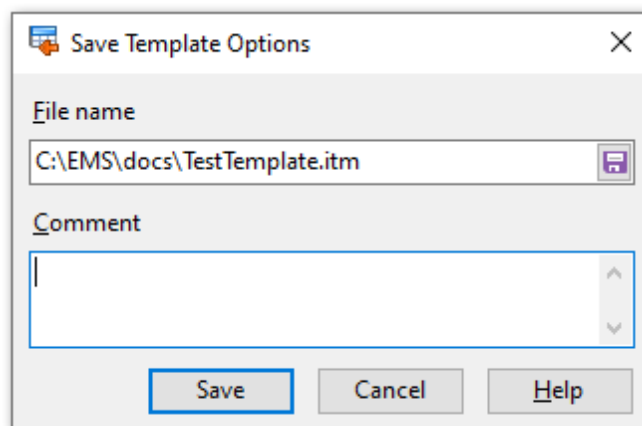


File name

When saving template, specify the template file name and select its location using the  button which calls the **Save Template As...** dialog.

Comment

The comment field of the template description.



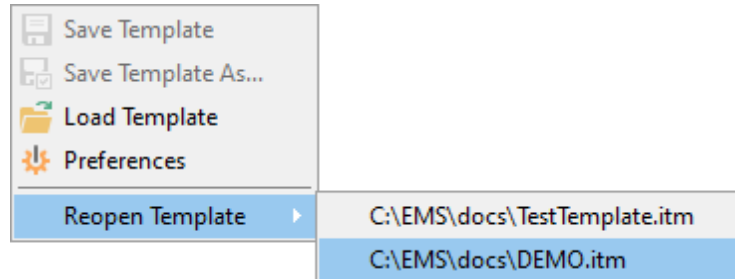
If you need to repeat data importing process with the same or similar settings later, it is reasonable to save all the settings you entered on the [Start of data import process](#)^[64] step of the Wizard.

To quickly return to Step 1, use the **Restart Wizard** button, which is only available at the last step.

Please note that loading a template is only available at the [Getting started](#)^[23] and the

[Setting connection properties](#)^[24] steps of the Wizard.

If necessary, you can **Save template** at any step of the wizard using the corresponding popup menu item of the **Tools** menu.



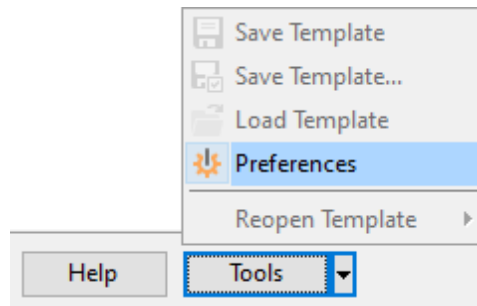
See also:

[Working with wizard application](#)^[23]

[Setting program preferences](#)^[68]

2.3 Setting program preferences

Data Import 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 all these options.



[Setting general options](#)^[69]

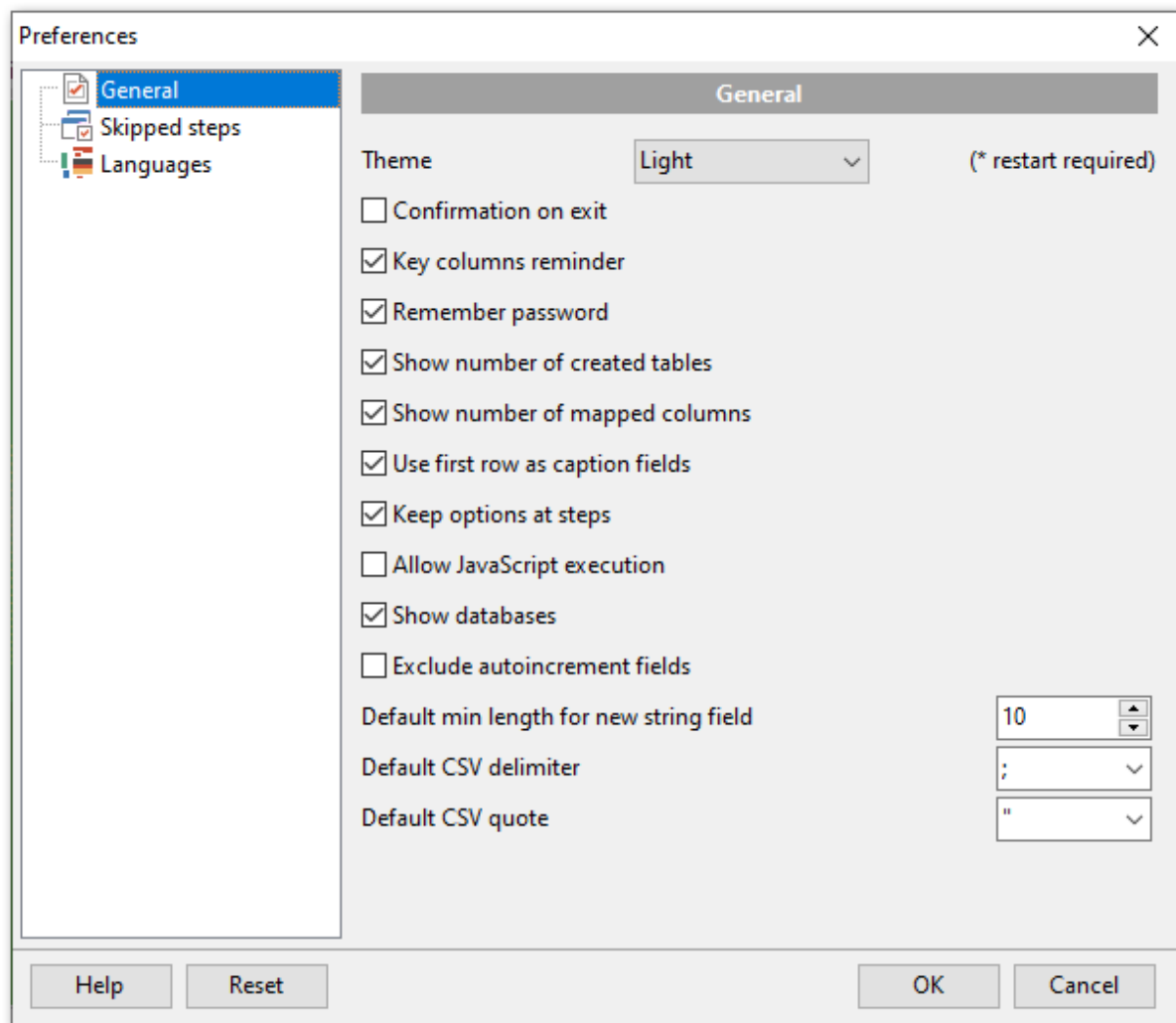
These options define general behavior of Data Import for PostgreSQL.

[Skipped steps](#)^[71]

Selecting steps to skip by default.

[Selecting program language](#)^[73]

This page allows you to select a language to be applied for your copy of Data Import for PostgreSQL.

**Reset**

Click this button to set all the program settings to default.

See also:

[Working with wizard application](#)^[23]

[Using data import configuration files](#)^[66]

2.3.1 General**General****Theme**

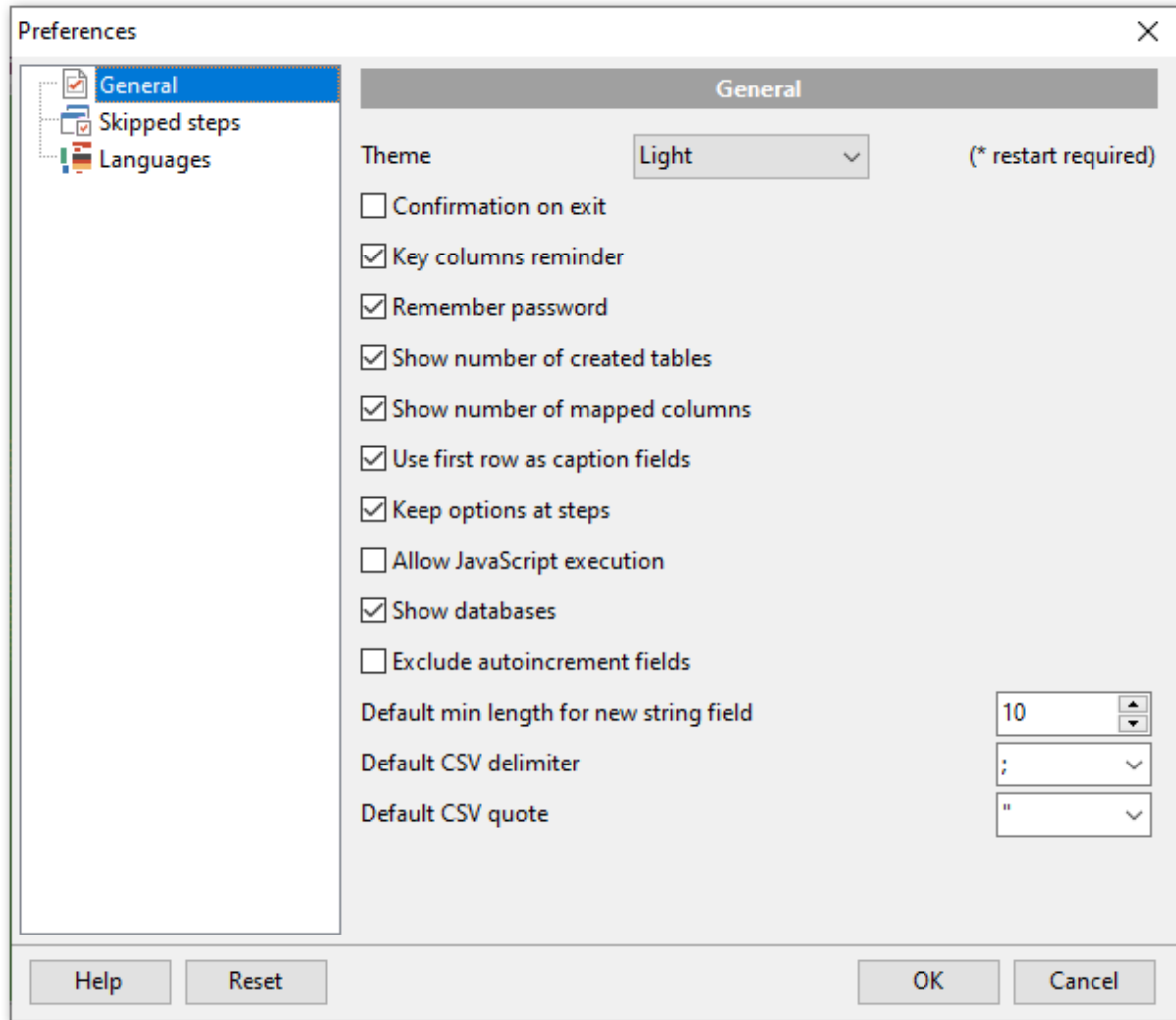
Select the main color theme for the application: Light or Dark.

Confirmation on exit

Enables/disables confirmation upon exiting the program.

Key columns reminder

Use this option to enable/disable prompts to define key columns.



Remember password

Setting this option allows you to save passwords used for access to the databases automatically upon closing the application. Please note that checking this option saves the latest password used for connection to the database (including the SSH server password).

Show number of created tables

Use this option to allow/disallow appearance of the correspondent information window.

Show number of mapped columns

Displays the number of mapped fields on proceeding to [Step 4](#)^[31].

Use first row as caption fields

Enable this option to use first rows as captions when mapping fields.

Keep options at steps

If this option is checked all your import settings will remain the same the next time the program is started.

 Allow JavaScript execution

If this option is checked then the script entered on the [Step 6](#)^[55] is always executed.

 Show databases

This option enables/disables the 'Select Database' drop-down list in 'Select Table' dialog at [Step 2](#)^[27] of the wizard. If you check this option, you can select the database from the list, otherwise you must type the database name manually.

 Exclude autoincrement fields

If this option is checked then autoincrement fields will not be mapped on [auto mapping](#)^[31].

Default min length for new string field

Specify the default length of the new field in case you are importing data into a new table.

Default CSV delimiter

Specify the default delimiter for CSV files format.

Default CSV quote

Specify the default quottas for CSV files format.

See also:

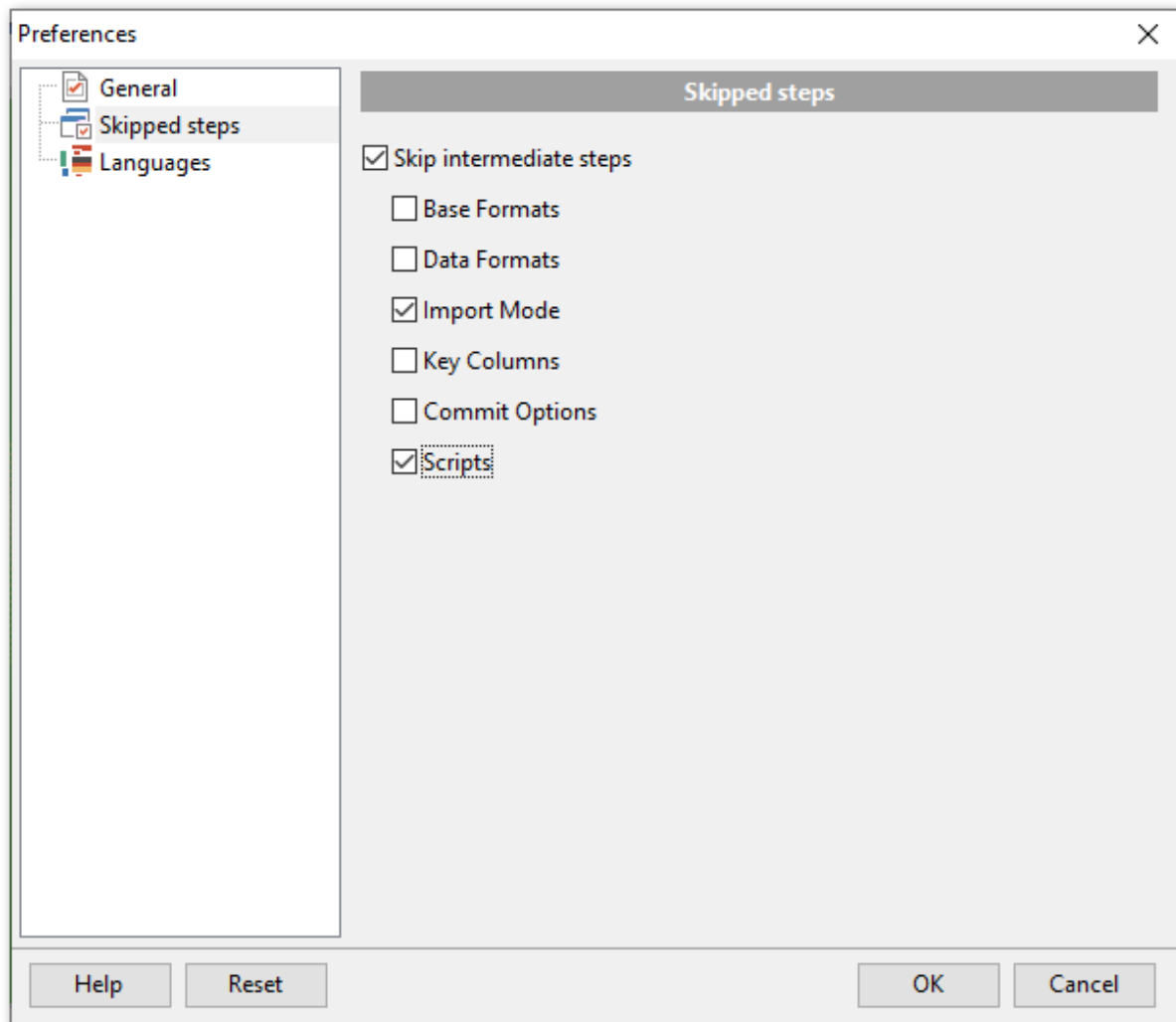
[Selecting program language](#)

^[73]

[Skipped steps](#)^[71]

2.3.2 Skipped steps

Use this option group to define the Wizard application steps to be skipped.



Skip intermediate steps

This option determines the behavior of the *Next* and *Back* buttons. When this option is disabled, clicking the *Next* button leads to sequential passing through the list of imported objects. When this option is enabled, the steps checked in the **Skipped steps** group will be skipped.

Base Formats

[Setting base data formats](#)^[50]

Data Formats

[Settings data formats for each field](#)^[55]

Import Mode

[Specifying import mode](#)^[59]

Key Columns

[Selecting key columns](#)^[61]

Commit Options

[Setting common options](#)^[62]

Scripts

[Defining scripts](#)^[63]

See also:

[Setting general options](#)^[69]

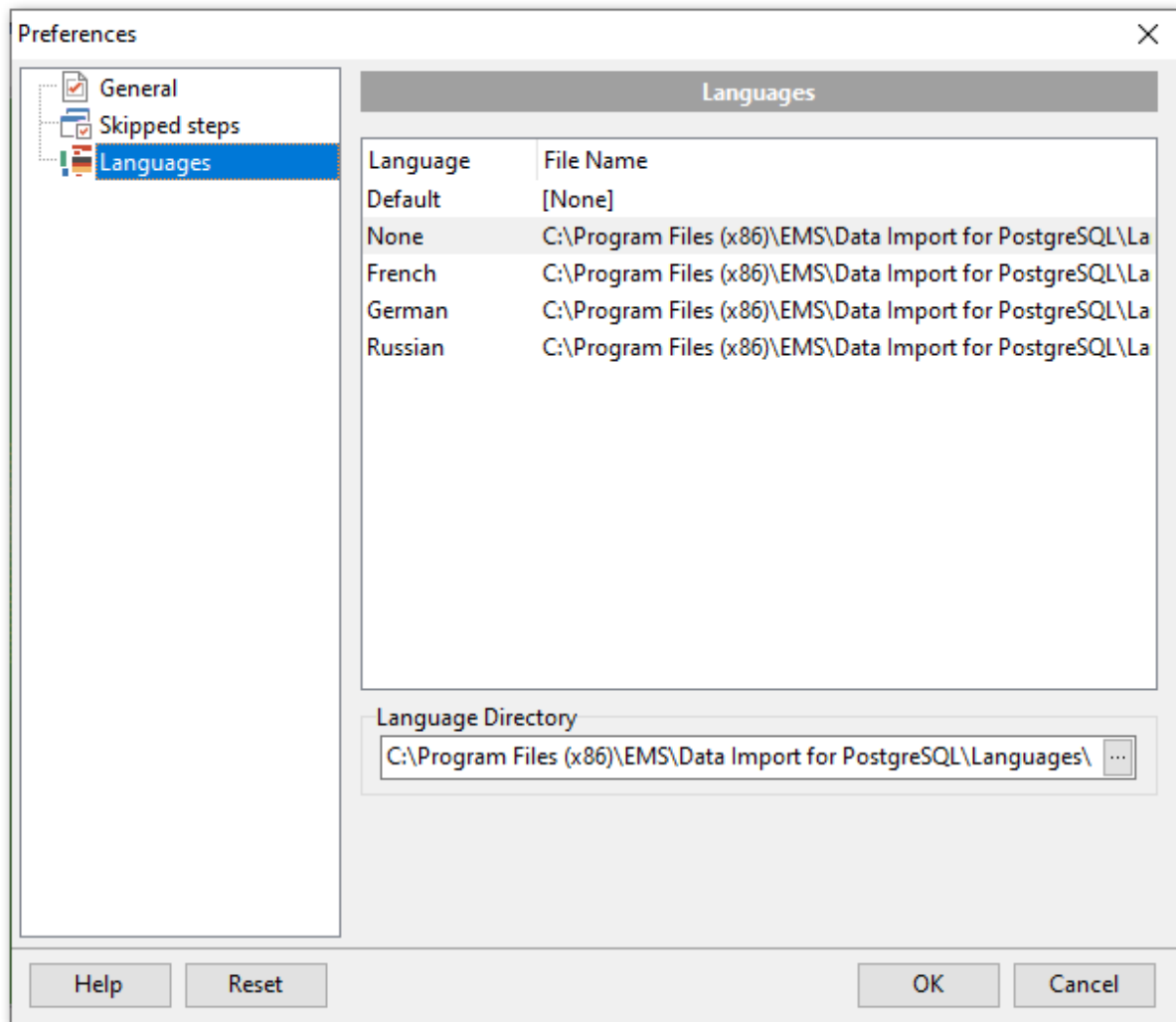
[Languages](#)^[73]

2.3.3 Languages


The **Languages** page is provided for managing Data Import localization files.

You can create your own **.lng* files similar to those available in the `%program_directory%\Languages` folder, add them to the list of available languages and set the new language as the program interface language.

The table lists all the languages available for localization and the corresponding **.lng* files.



Language Directory

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

See also:

[Setting general options](#) ^[69]

[Skipped steps](#) ^[71]

Part



3 Console application

Additionally to **the GUI version** which is implemented in the form of a wizard application, the installation package of Data Import 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.

```
C:\Program Files\EMS\Data Import for PostgreSQL>PgImportC.exe_
```

Data Import for PostgreSQL command line utility is intended for quick and powerful data import to PostgreSQL tables.

[Working with console application](#)^[77]

[Configuration file format](#)^[78]

See also:

[Wizard Application](#)^[22]

3.1 Working with console application

All the import options are set in **template** (*.itm) files. A template can be also used in the **Console version** of Data Import for PostgreSQL.

To create a template file, follow the instructions below:

- start Data Import [Application wizard](#)^[22];
- set all the required options in all steps of the wizard;
- test the import process at the last step;
- [save all import options in the template](#)^[66].

Console application also generates the *.log file that contains information about the import operation completion.

The easiest way to start Data Import for PostgreSQL console application is to double-click the generated *.itm template. The other way is to enter the command line and type the appropriate command.

Usage:

```
<path to Data Import for PostgreSQL console application>\PgImportC.exe TemplateFile [-  
datafile=<source file name>] [-B]
```

datafile=<filename1>;<filename2>;<filename3>

Indicates the source file(s) for data import. The values you specify replace the names of the files in the template. The number of files should strongly correspond to the number of files in the template. Multiple files are separated with a semicolon.

TemplateFile

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

[-B]

Use this parameter in the command line to run the console version of Data Import for PostgreSQL in the background mode.

Example:

```
"C:\Program Files\EMS\Data Import for PostgreSQL\PgImportC.exe" "C:  
\EMS\DataImport\Template1.itm" -B -datafile=test150283.xls
```

Note: The following exit codes can be returned by Data Import 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:

[Working with wizard application](#)^[23]

[Configuration file format](#)^[78]

3.2 Configuration file format

The configuration file is divided into several sections.

[Connection]

This section contains parameters for connecting to the server: *Host*, *Port*, *Charset*, *tunneling settings*, *Login*, *Password* (encrypted). These parameters are obligatory.

[PREFERENCES]

This section contains the general settings of the utility.

ImportToViews

1 - enabled, 0 - disabled

The following sections are unique for each imported file and contain the appropriate postfix (*FILE_00*, *FILE_01*, *FILE_02*, etc.)

In these sections the import options for each particular file are stored.

[FILE_00]

In the main file section the following parameters are used:

Database (destination database name), *Table* (destination table), *FileName* (the source file name)

ImportType

Indicates the format of the source file: 0 stands for MS Excel, 1 for MS Access, 2 - DBF, 3 - TXT, 4 - CSV, 5 - XML, 6 - MS Excel 2007, 7 - MS Word 2007, 8 - HTML, 9 - XML Document, 10 - ODS, 11 - ODT

The next section is specific to the file format and is named correspondingly, e.g. *[FILE_00_XLS]*. In this section the following specific file type options are set:

[FILE_00_XLS]

SkipRows, *SkipCols* - the number of rows and columns in the source file to be skipped on import.

[FILE_00_Access]

SourceType

This option indicates the Access source type - Table or query, 0 stands for table, and 1 for query.

TableName and *query* - these options set the source Access table name to import from if source type is 0 and the source query text if the source type is 1.

[FILE_00_TXT]

SkipRows - the number of rows in the source file to be skipped on import.

[FILE_00_CSV]

SkipRows - the number of rows in the source file to be skipped on import.

CSVDelimiter, *CSVRightQuot*, *CSVLeftQuot* - these options are used on reading the source CSV file, they store options for delimiting columns and the quotation marks.

The next section is **[FILE_XX_MAP]**.

It is also specific to the file type, but is obligatory for all file types. In this section correspondence between the source file columns and the destination table fields is set.

If the source file is an *Excel* file, then the map is set in the following format:

<DB_FieldName>=CellIndex.

Use semicolon to separate multiple cells. You can define an Excel row or column as a range of cells, e.g.

Field1=A1;A4;A6-A9;B1-F1.

If the source file is an *Access* or *DBF* file, then the mapping is set in the following format:

<DB_FieldName>=<Source_Field_Name>.

If the source file is a *TXT* file, the mapping looks in the following way:

<DB_FieldName>=<Position>;<Size>.

This means that you should provide the starting position and the size of the source file column for each destination field.

To set mapping for a *CSV* file, provide a column number for each destination field, e.g.

Field1=1

Field2=4

etc.

The properties that are set in sections **[FILE_XX_BASE_FORMATS]** and **[FILE_XX_IMPORT_OPTIONS]** correspond to those set on the 'Base Formats' and the 'Options' tabs.

The values that stand for *BOOLEAN TRUE* and *BOOLEAN FALSE* values are set in sections **[FILE_XX_BOOLEAN_TRUE]** and **[FILE_XX_BOOLEAN_FALSE]**.

The *NULL* values are set in section **[FILE_XX_NULL_VALUES]**

Data formats that are set for each destination field separately are stored in sections named in the following way:

[FILE_XX_DATA_FORMATS_<FIELD_NAME>],

e.g. *FILE_00_DATA_FORMATS_FIELD1.*

The properties within these sections also correspond to the properties set on the 'Data Formats' tab of the [Settings data formats](#)^[55] step for each field.

[FILE_00_IMPORT_OPTIONS]

This section contains parameters that define import options specified on Steps 6-9 of the wizard.

CommitAfterDone

1 - enabled, 0 - disabled

CommitRecCount

The number of records in each block to be supplemented with the COMMIT statement.

ImportRecCount

1 - enabled, 0 - disabled

ImportAllRows

1 - enabled, 0 - disabled

SaveResultSQL

1 - enabled, 0 - disabled

ResultSQLFile

The path to the result *.sql file.

ImportKind

0 - Universal Mode, 1 - Native Mode

TruncateLongString

1 - enabled, 0 - disabled

RollbackIfError

1 - enabled, 0 - disabled

ImportMode

0 - Insert All, 1 - Insert New, 2 - Update, 3 - Update or Insert, 4 - Delete, 5 - Delete or Insert

ImportAddType

1 - enabled, 0 - disabled

AllowDuplicates

1 - enabled, 0 - disabled

NativeModeKind

0 - Single Commands, 1 - Batch insert

[#General#]

This section stores information about the product name and its major version.

See also:

[Working with console application](#)^[77]

Part



4 Appendix

4.1 Supported file formats

• **MS Excel 97-2003**

The most popular e-table format used by Microsoft® Excel (*.xls). The result files are fully compatible with Microsoft® Excel versions 97-2000, 2003 and XP.

• **MS Access**

File of Microsoft® Access format (*.mdb, *.accdb) with an ADO connection used.

• **HTML**

Hyper Text Markup Language file format (*.html, *.htm), complete compatibility with HTML 4.0 specification.

• **Text file**

Plain text file format (*.txt).

• **CSV file**

Comma-Separated Value file format (*.csv).

Note: all the text formats including *Text file*, *CSV* are usually used as working or interchange formats.

• **XML**

A markup language for documents containing structured information (*.xml).

• **DBF**

Database file format (*.dbf) used by dBASE and a number of xBASE applications.

• **MS Excel**

The contemporary e-table format used by Microsoft® Excel (*.xlsx). The result files are fully compatible with Microsoft® Excel 2007.

• **MS Word**

The contemporary text processing format used by Microsoft® Word (*.docx). The result files are fully compatible with Microsoft® Word 2007.

• **ODF Spreadsheets**

OASIS Open Document Format for Office Applications - open document file format for spreadsheets (*.ods) used by a number of applications including OpenOffice.org and KOffice.

• **ODF Text**

OASIS Open Document Format for Office Applications - open document file format for word processing (*.odt) documents used by a number of applications including OpenOffice.org and KOffice.

4.2 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 connected with the fact that a number of Linux-like OS's (for example FreeBSD) include SSH server in their standard integration. To learn more information on this issue, please, visit <https://www.openssh.com>. SSH tunneling feature of SQL Manager 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 the default port is closed for external connections due to some reasons. The connection over SSH tunnel works in the following way. First, a connection is established and the process of authentication between SSH client built in SQL Manager and remote PostgreSQL server is performed. Then all incoming and outgoing information between the program and PostgreSQL server is transmitted through SSH server with the help of a communication port (usually it is 22), and SSH server transfers this information directly to PostgreSQL server.

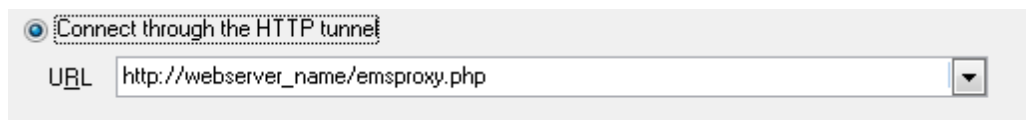
4.3 HTTP tunneling options

HTTP tunneling is a method that allows 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 because of the security reasons. The HTTP tunnel works 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. Also you need to upload the special `emsproxy.php` script to your web-server to access it remotely (you can place it to the folder with your other PHP scripts). If your web-server complies with the requirements and the script is installed correctly, you'll see the message "EmsProxy v 2.3" (version number can be different) in your browser when opening the `https://<your_webserver_name>/emsproxy.php` page.

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 used by Data Import for PostgreSQL.

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 Import for PostgreSQL are available.

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

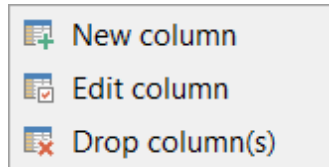


The screenshot shows a user interface for selecting a connection method. A radio button is selected next to the text "Connect through the HTTP tunnel". Below this, there is a label "URL" followed by a text input field containing the value "http://webserver_name/emsproxy.php". A small downward-pointing arrow is visible at the end of the input field.

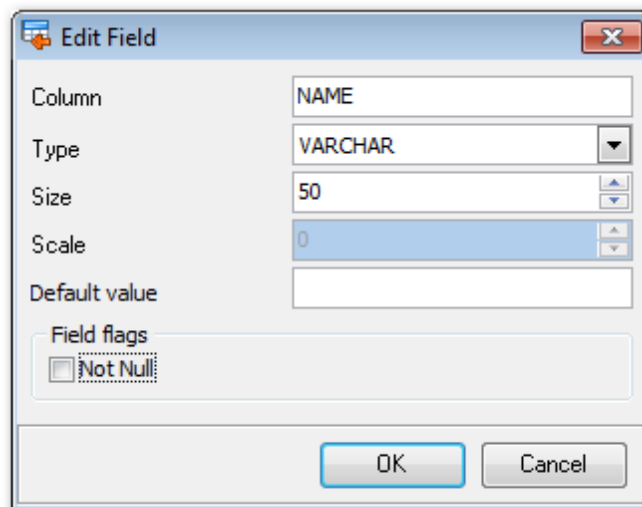
4.4 Add/Edit field

When you import data with destination table creation, you can manage this table's fields.

Use the context menu to New/Edit/Drop a column.



Selecting **New column** or **Edit column** context menu item opens the field editor that allows you to view/edit the properties of the destination table fields.



Use the **Column** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

The **Type** tab defines the type of the field data.

Size

Defines the size of the field value.

Scale

For *numeric* and *decimal* types you need to define the number of decimal to the right of the decimal point.


Default value

Define this option value if you need inserted records to get specified value.

Not NULL

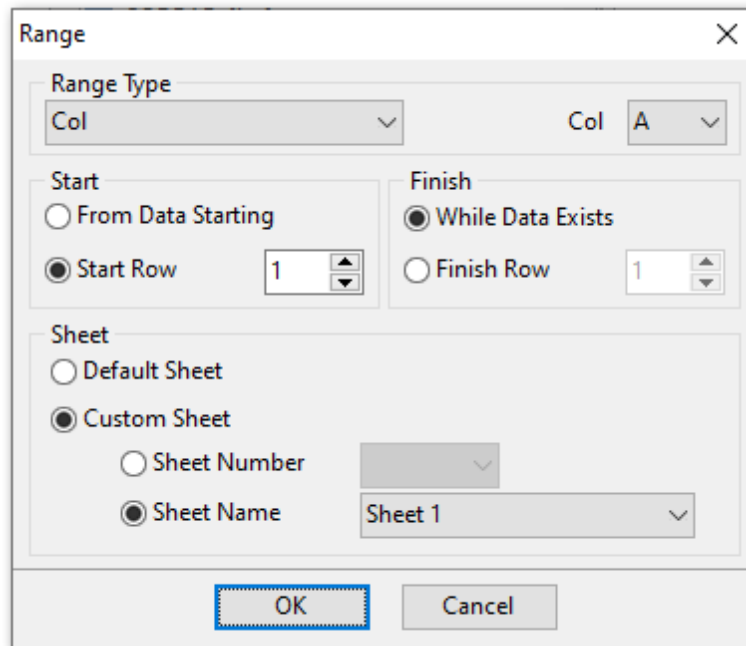
Check this option to prevent the entry of NULL or unknown values in column.

4.5 Add/Edit Range

To set a range of data to be imported from the file, use the **Add range**  button.

To remove a range, use the **Delete range**  button.

To edit an existing range, double-click it in the **Ranges** list. The **Range** dialog allows you to set a number of options for the range being edited.



Select **Range Type** from the corresponding drop-down list. Then specify the *column*, *row* or *cell* number whose range is to be defined.

Start and **Finish** section allows you to set the range of the values to be imported from the specified row or column.

Use the **From Data Starting** and **While Data Exist** options to automatically define the range of the values.

Use **Sheet** section to define the location of the row/column for which the row was specified.

You can select either **Default sheet**, or **Custom Sheet**.

For custom sheet you need to select **Sheet Number** or **Sheet Name** from the appropriate drop-down list.

Note: If Cell was set as Range Type, the only thing you can specify at this dialog is its number.

Credits

Software Developers:

Alex Paclin

Alexey Butalov

Michael Kuzevanov

Alexey Saybel

Technical Writers:

Semyon Slobodenyuk

Dmitry Doni

Olga Ryabova

Cover Designer:

Tatyana Makurova

Translators:

Anna Shulkina

Serge Fominikh

Team Coordinators:

Alexey Butalov

Alexander Chelyadin

Roman Tkachenko