

Importing Table Data

Only in DbVisualizer Pro

 This feature is only available in the DbVisualizer Pro edition.

You can import data using the Import Table Data wizard.

- [Input File Format and Other Options](#)
 - [CSV format page](#)
 - [Excel format page](#)
- [Data Formats and Data Type Per Column](#)
- [Matching Columns and Data Types for an Existing Table](#)
- [Adjusting Table Declaration for a New Table](#)
- [Importing Binary/BLOB and CLOB Data \(CSV Only\)](#)
- [Saving And Loading Settings](#)
- [Other Ways to Import Table Data](#)
- [Known limitations](#)

You can import data from a file into an existing table or to a new table. The import source can be either a CSV file or an Excel file (*.xls* or *.xlsx*). The steps are almost identical:

1. Select the table node for the table you want to import to, or the **Tables** node if you are importing to a new table, in the **Databases** tab tree,
2. Open the **Import Table Data** wizard from the right-click menu,
3. Specify the input file on the first wizard page (CSV or Excel file),
4. [Excel only] : If the input file is an Excel file, you are asked to choose the Excel sheet to import on the next page.
5. Specify file format and other options,
6. Specify data formats and the data type per column,
7. Adjust details about the destination table,
8. Click **Import** on the last page.

 Instead of choosing Import Table Data from the right-click menu, you can **drag and drop a file** from the operating system's file manager on the Tables node or a table node.

How many INSERT statements to execute during the import process before committing the changes can be specified in the **Properties** tab for the connection, in the **Transaction** category.

Input File Format and Other Options

On the File Format page, you specify what and how the data in the source file should be imported. This includes specifying what row to start the import from and if empty rows should be skipped.

Currently DbVisualizer supports import of CSV files and the Excel file formats ".xls" and ".xlsx".

CSV format page

Import File

Delimiters

Column Delimiter: Auto Detect String TAB

Options

Row of Header:

Start Row Of Data:

Skip Empty Row(s):

Skip Rows Starting With:

Text Quoted Between:

Data

Grid | File

*	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
9	107	"Diana"	"Lorentz"	"DLORENTZ"	"590.423.5567"	"2007-02-07 00:00:00"
10	108	"Nancy"	"Greenberg"	"NGREENBE"	"515.124.4569"	"2002-08-17 00:00:00"
11	109	"Daniel"	"Faviet"	"DFAVIET"	"515.124.4169"	"2002-08-16 00:00:00"
12	110	"John"	"Chen"	"JCHEN"	"515.124.4269"	"2005-09-28 00:00:00"
13	111	"Ismael"	"Sciarra"	"ISCIARRA"	"515.124.4369"	"2005-09-30 00:00:00"
14	112	"Jose Manuel"	"Urman"	"JMIURMAN"	"515.124.4469"	"2006-03-07 00:00:00"

Preview Rows: Fit Column Widths

Settings > < Back Next > Cancel

In the **Delimiters** section, define the character that separates the columns in the file. If you enable **Auto Detect**, DbVisualizer tries the following characters:

- comma ","
- tab "TAB"
- semicolon ";"
- percent "%"

You can specify any character sequence as a delimiter, but it must not contain more than four characters.

You can use the **Options** area to further specify how to read the input file, for instance if certain rows should be skipped and how text data is quoted.

The **Data** section at the bottom of the page shows a preview of the parsed data in the **Grid** tab and the original source file in the **File** tab. If a row in the Grid tab is red, it indicates that the row will be ignored during the import process. This happens if any of the **Options** settings result in rows not being qualified.

Excel format page

The Excel format page is very much like the CSV format page.

As Excel is from start organized in columns and rows the **Column delimiter** setting is not applicable to Excel files. The **Skip Rows Starting With** and the **Text Quoted Between** options are also not supported for Excel.

As shown in the snapshot below there is no **File** tab for Excel files.

The **Grid** tab shows a preview of the data, just as in the CSV case.

Import File [Close]

Options

Row of Header:

Start Row Of Data:

Skip Empty Row(s):

Data

Grid

*	EMPLOYEE...	FIRST_NAME	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY
2	100	Steven	King	SKING	515.123.4567	2003-06-17 ...	AD_PRES	24000
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21 ...	AD_VP	17000
4	102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13 ...	AD_VP	17000
5	103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03 ...	IT_PROG	9000
6	104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21 ...	IT_PROG	6000
7	105	David	Austin	DAUSTIN	590.423.4569	2005-06-25 ...	IT_PROG	4800
8	106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05 ...	IT_PROG	4800
9	107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07 ...	IT_PROG	4200
10	108	Nancy	Greenberg	NGREENBE	515.124.4569	2002-08-17 ...	FI_MGR	12008
11	109	Daniel	Faviet	DFAVIET	515.124.4169	2002-08-16 ...	FI_ACCOUNT	9000
12	110	John	Chen	JCHEN	515.124.4269	2005-09-28 ...	FI_ACCOUNT	8200
13	111	Ismael	Sciarra	ISCIARRA	515.124.4369	2005-09-30 ...	FI_ACCOUNT	7700

Preview Rows: Fit Column Widths

Settings [v] < Back Next > Cancel

Data Formats and Data Type Per Column

The Data Formats page is used to define formats for some data types. The first row in the preview grid contains a data type drop-down lists. DbVisualizer tries to determine the data type for each column by looking at the value for the number of rows specified as **Preview Rows**. If this data type is incorrect for a column, use the drop-down lists to select the appropriate type.

Import File
✕

Data Formats

Date: ▾ Example: 2016-06-03

Time: ▾ Example: 12:52:35

Timestamp: ▾ Example: 2016-06-03 12:52:35

Thousand Separator:

Decimal Separator:

Null Values: Example: (null), NULL, nada

Boolean True: Example: true, 1, yes, on

Boolean False: Example: false, 0, no, off

Data

Grid
File

EMPLOYEE...	FIRST_NAME	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY	C
100	"Steven"	"King"	"SKING"	"515.123.45...	"2003-06-17...	"AD_PRES"	24000	(r)
101	"Neena"	"Kochhar"	"NKOCHHAR"	"515.123.45...	"2005-09-21...	"AD_VP"	17000	(r)
102	"Lex"	"De Haan"	"LDEHAAN"	"515.123.45...	"2001-01-13...	"AD_VP"	17000	(r)
103	"Alexander"	"Hunold"	"AHUNOLD"	"590.423.45...	"2006-01-03...	"IT_PROG"	9000	(r)
104	"Bruce"	"Ernst"	"BERNST"	"590.423.45...	"2007-05-21...	"IT_PROG"	6000	(r)

Preview Rows:
 Fit Column Widths

Settings ▾

< Back
Next >
Cancel

If you need to change the data type for a number of columns, e.g. set them all to String, you can Copy/Paste the data type. First change it for one of the columns using the drop-down, select and copy that new data type value and then select the data type for all other columns and use paste to change them all at once. If you make a mistake, you can change the **Preview Rows** value to let DbVisualizer determine the types again.

If you import to an existing table, there is yet another way to adjust the data types for the file columns, described in the next section.

Matching Columns and Data Types for an Existing Table

When you are importing to an existing table, the Import Destination page provides two options: **Grid** and **Current Database Table**. You can use the **Grid** choice to import the data into a grid that is presented in its own window in DbVisualizer if you just want to just process the data in some way without saving it in the database.

When the **Current Database Table** choice is selected, the page shows information about the table into which the data will be imported in the **Map Table Columns with File Columns** grid shows the columns in the selected database table and the columns in the source file.

Import File [Close]

Import Into
 Grid Current Database Table New Database Table

Database Table
 Database Connection: Ora12C_HR
 Database:
 Schema: HR
 Table: EMPLOYEES

Map Table Columns with File Columns

Key	Table Column Name	Table Data Type	File Column Name	File Data Type
	EMPLOYEE_ID	NUMBER		
	FIRST_NAME	VARCHAR2		
	LAST_NAME	VARCHAR2		
	EMAIL	VARCHAR2	EMAIL	String
	PHONE_NUMBER	VARCHAR2	PHONE_NUMBER	String
	HIRE_DATE	DATE	HIRE_DATE	String
	JOB_ID	VARCHAR2	JOB_ID	String

Map Columns ▾

- Map by Column Name
- Map by Column Index
- Map File Data Type = Table Data Type
- Clear Mappings

Use Delimited Identifiers

Settings ▾ < Back Next > Cancel

DbVisualizer automatically associates the columns in the source file with the columns in the target table in the order they appear. If the columns appear in a different order in the file than in the table, but they are named the same, you can use the auto-mapping menu in the upper right corner of the **Map Table Columns with File Columns** grid to automatically map the columns by name. **Map by Column Name** and **Map by Column Index** do exactly what it sounds like. **Map File Data Type = Table Data Type** sets the **File Data Type** for each column to the type of the corresponding table column.

If the column names are different between the file and the table and also appear in different order, you can manually map them using the drop-down lists in the File Column Name field. Choose the empty choice in the columns drop-down to ignore the column during import.

Import File [Close]

Import Into
 Grid Current Database Table New Database Table

Database Table
 Database Connection: Ora12C_HR
 Database:
 Schema: HR
 Table: EMPLOYEES

Map Table Columns with File Columns Map Columns ▾

Key	Table Column Name	Table Data Type	File Column Name	File Data Type
	EMPLOYEE_ID	NUMBER	EMPLOYEE_ID	Number
	FIRST_NAME	VARCHAR2	FIRST_NAME	String
	LAST_NAME	VARCHAR2	LAST_NAME ▾	String
	EMAIL	VARCHAR2		String
	PHONE_NUMBER	VARCHAR2	EMPLOYEE_ID	String
	HIRE_DATE	DATE	FIRST_NAME	String
	JOB_ID	VARCHAR2	LAST_NAME	String
	SALARY	NUMBER	EMAIL	Number
	COMMISSION_PCT	NUMBER	PHONE_NUMBER	String
	MANAGER_ID	NUMBER	HIRE_DATE	Number
	DEPARTMENT_ID	NUMBER	JOB_ID	Number
			SALARY	Number
			COMMISSION_PCT	
			MANAGER_ID	
			DEPARTMENT_ID	

Use Delimited Identifiers

Settings ▾ < Back Next > Cancel

You can use copy/paste of the values in the **File Column Name** and **File Data Type** fields to quickly fill the selection of cells instead of manually selecting the correct data in the drop-downs.

There is also a **Use delimited identifiers** checkbox. Check this box if you want the SQL statements for importing the table to use delimited identifiers; in other words, if you want to use table and column names with special characters, mixed case, or anything else that requires delimited (quoted) identifiers.

Adjusting Table Declaration for a New Table

When you are importing to a new table, the Import Destination page provides two options: **Grid** and **New Database Table**. You can use the Grid choice to import the data into a grid that is presented in its own window in DbVisualizer if you just want to just process the data in some way without saving it in the database.

When the New Database Table choice is selected, you are presented with a field for the table name and a number of tabs for column and constraint declarations. The Columns tab is filled out based on the source data and the data types from the Data Formats page.

Import File
✕

Import Into

Grid
 Current Database Table
 New Database Table

New Table Details

Database Connection: Ora12C_HR

Database:

Schema: HR

Table: newTable

Columns

Primary Key

Foreign Keys

Unique Constraints

Check Constraints

Organize On

Name	Data Type	Size	Scale	Nullable	Default
EMPLOYEE_ID	INTEGER			<input type="checkbox"/>	
FIRST_NAME	VARCHAR2	20		<input type="checkbox"/>	
LAST_NAME	VARCHAR2	20		<input type="checkbox"/>	
EMAIL	VARCHAR2	20		<input type="checkbox"/>	
PHONE_NUMBER	VARCHAR2	20		<input type="checkbox"/>	
HIRE_DATE	VARCHAR2	30		<input type="checkbox"/>	
JOB_ID	VARCHAR2	20		<input type="checkbox"/>	
SALARY	INTEGER			<input type="checkbox"/>	
COMMISSION_PCT	VARCHAR2	10		<input checked="" type="checkbox"/>	
MANAGER_ID	INTEGER			<input checked="" type="checkbox"/>	
DEPARTMENT_ID	INTEGER			<input type="checkbox"/>	

Settings
▼

< Back

Next >

Cancel

Note that it is not always possible to find a database specific type for the data format specified on the Data Format page. You must then pick the correct type from the **Data Type** drop-down menu. The size for string column types may also need to be adjusted. By default, the size is set to the maximum number of characters found for the column in the number of rows specified as **Preview Rows**, adjusted up to the next power of ten. You can ignore certain columns by removing them in the **Columns** tab. **Keys** and other constraints can be created using the other tabs.

You can go back to the Data Format page and increase the **Preview Rows** value if you believe that it will help DbVisualizer to pick better defaults. If you do so, you need to click the **Reload** button when you come back to this page to rescan the source data and get new default values.

If you make a mistake, or if the import fails, so you have to go back and make adjustments before you import again, make sure you enable **Drop existing table, if any**. It is disabled by default to prevent you from accidentally dropping an existing table when you intend to import to a new table, but if the import fails, the new table may already have been created so it needs to be dropped before a new table with your adjusted input can be created.

There is also a **Use delimited identifiers** checkbox. Check this box if you want the SQL statements for importing the table to use delimited identifiers; in other words, if you want to use table and column names with special characters, mixed case, or anything else that requires delimited (quoted) identifiers.

Importing Binary/BLOB and CLOB Data (CSV Only)

If you have exported data to a CSV file using DbVisualizer, use the Import Table Data feature to import it. On the Data Format page, ensure that the format for the source file column is set to **BLOB** or **CLOB**.

Data	
ID	PHOTO
\$ Number	\$ BLOB
1	`\${data 1-1} C:\Users\ulf\Documents\tmp\blobs\dbvis-3503876270486780709.bin BinaryData noshow vl=file} \$

If you have exported Binary/BLOB and CLOB data as an SQL script, you just run the script in the SQL Commander to import it. When the SQL Commander encounters a variable that refers to a file, it reads the file and inserts the content as the column value.

Saving And Loading Settings

If you often use the same settings, you can save them as the default settings for this assistant. If you use a number of common settings, you can save them to individual files that you can load as needed. Use the Settings button menu to accomplish this:

- **Save as Default Settings**
Saves all format settings as default. These are then loaded automatically when open an Export Schema dialog
- **Use Default Settings**
Use this choice to initialize the settings with default values
- **Remove Default Settings**
Removes the saved defaults and restores the regular defaults
- **Load**
Use this choice to open the file chooser dialog, in which you can select a settings file
- **Save As**
Use this choice to save the settings to a file

Other Ways to Import Table Data

If you have a script containing INSERT statements for all data, you can execute it in the [SQL Commander](#).

Known limitations

- Excel files cannot contain CLOB/BLOB type of data (e.g. images etc). Cells with this kind of data are imported as empty.
- There is a size limitation when importing Excel files with the `.xls` filename extension. The size limitation is roughly 20 megabytes, depending on your configuration and how much memory is used for other things. [Increasing DbVisualizer max memory](#) may allow you to import larger files.