

# Internationalization and Localization (i18N and L10N)

DbVisualizer is currently available in US English and supports user data in more or less any language or character encoding; DbVisualizer leverages the character sets supported by the Java VM, the fonts supported by the operating system, and the character encoding capabilities of the database to read, write, and render user data. You can also set your own preferences for date and time formats (see [Changing the Data Display Format](#)).

## Fonts and Character Sets

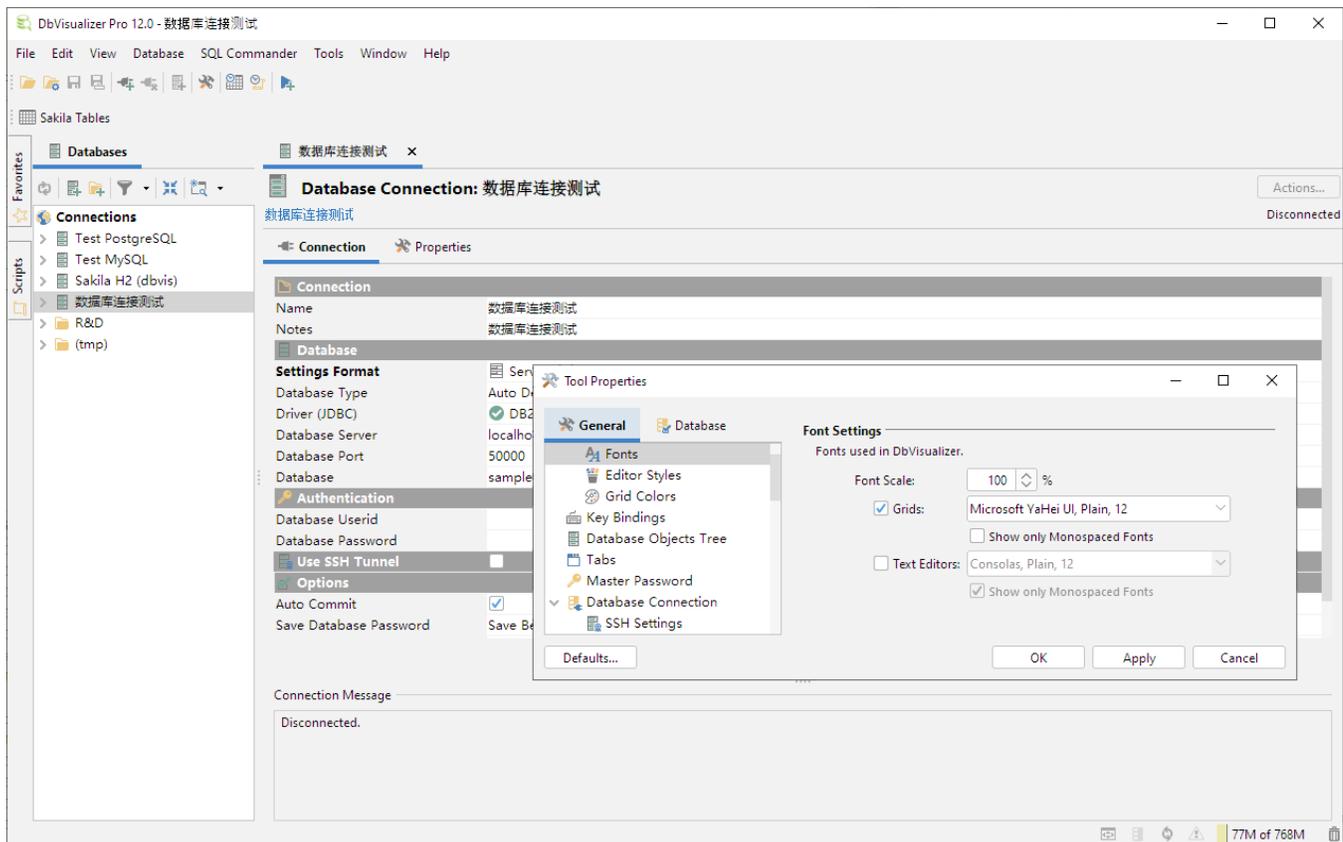
Depending on which platform you run (Windows, Unix/Linux, macOS) and what character set you want to use, you may have to change the font settings.

DbVisualizer offers two settings for fonts:

- **Grids:** controls the font for GUI components where a proportional font is usually preferred (see [Changing the GUI Appearance](#)).
- **Text Editors:** controls the font for editors, log files, and other components where a monospaced font is usually preferred (see [Editing SQL Scripts - Font Settings](#)).

### Example:

If you run Microsoft Windows and want to name your database connections using simplified Chinese, you will have to change the default font for **Grids** to one that supports this character set. In this example, we use Microsoft YaHei UI:



## Encoding

Even if the font is capable of rendering the characters, data may be scrambled in DbVisualizer if the encoding between the JDBC driver and the database is wrong.

When text results fetched from the database are scrambled there are a few areas that needs to be verified:



- The font selected in DbVisualizer must be capable to show the characters.
- The encoding used to pass information from the database to the JDBC driver must match. Encoding depends on the capabilities of the JDBC driver. Check the driver documentation what options are available. Once determined, encoding changes are most often applied as [driver properties](#) in DbVisualizer.

You may also need to specify the encoding of text in binary data types, either for a specific connection or for all all connections of a specific database type. In this example we specify **UTF-8** encoding for all **SQL Server** connections:

