

How to use Neovim Arabic in Windows

Fatima Zohra Mezerreg

14th April 2024

1 Introduction

In our roles as developers, we rely on a diverse array of tools, from code editors to documentation platforms. Consequently, our workflow often involves installing numerous applications to facilitate various tasks. But what if there was a single tool capable of handling all our needs proficiently? Wouldn't that be great?

Enter Neovim - a versatile solution that functions not only as a code editor but also as a text editor, eliminating the need for heavyweight applications like Microsoft Word or Google Docs. It allows for the creation of visually appealing documents effortlessly.

Now, you might wonder: Can Neovim support Arabic language for document writing? The answer is a resounding yes! Neovim boasts robust support for Arabic writing. However, the question remains: Does it match the effectiveness of other text editors that offer Arabic support? And is it equally user-friendly for Windows users like myself?

In this article, I will delve into Neovim Arabic, putting its capabilities to the test and comparing it with other text editors to provide answers to these questions.

2 Install and configure Neovim on windows

2.1 Installation

To install Neovim on Windows, there are two different methods.

2.1.1 Using WSL

If you are familiar with Linux, you can install Neovim in a Linux distribution via WSL. To do this:

- Ensure that WSL is enabled and functioning on your windows system.
- Execute the following command to install Neovim:

```
sudo apt-get install neovim
```

- Verify that Neovim is successfully installed by running:

```
nvim --help
```

2.1.2 Using the Windows version of Neovim

- The simplest way to install Neovim is by visiting the official website and downloading the Neovim '.msi' file.
- Double-click the downloaded file to initiate the installation process.
- Follow the on-screen instructions to complete the installation. This process will only take a few minutes.
- Once the installation is complete, verify that Neovim is functioning properly by performing one of the following:
 - Open Neovim from the shortcut icon on your desktop, if available.
 - Navigate to the folder where Neovim is installed and open 'nvim-qt.exe' located in the 'bin' folder.
- This will open a terminal window displaying the Vim editor interface, confirming that Neovim is installed and operational on your Windows system.

Congratulations!! you have successfully installed Neovim on your Windows system.

2.2 Configuration

After installing Neovim, you can customize its behavior in two ways: either by configuring the initial settings such as line numbering and indentation, or by adding plugins to enhance its appearance and functionality.

All of these configurations are stored in the `init.vim` file. Here's where you can locate this file in both WSL and Windows:

2.2.1 For WSL:

You'll find the `init.vim` file in the `~/.config/nvim` directory. If it doesn't exist, you can create it by following these steps:

1. Open a terminal.
2. Run the following commands:

```
mkdir -p ~/.config/nvim
touch ~/.config/nvim/init.vim
```

3. Then, open the `init.vim` file in Neovim to make your configuration changes by running:

```
nvim ~/.config/nvim/init.vim
```

2.2.2 For Windows:

You can find the `init.vim` file in the `'Users\Your User Name\AppData\Local\nvim'` directory. Similarly, if it doesn't exist, you can create it manually.

Once you have the `init.vim` file open in Neovim, you can start customizing your editor's settings and adding plugins to tailor it to your preferences.

I understand that you may encounter similar steps in various articles and videos. However, I'm providing them here because many existing resources often fail to clearly explain for beginners (like myself) where to locate the `init.vim` file and the `.config` folder in both WSL and Windows environments. It can be frustrating if these directories don't exist, leaving you unsure of what to do next. You may end up, like me, wasting precious time searching for them. Therefore, I hope these detailed instructions can help alleviate any confusion and save you time in the process.

3 Using arabic in Neovim

It's widely known that programming and displaying Arabic text correctly can be challenging. This is due to its unique features, such as right-to-left directionality and the complex shaping of letters, which vary depending on their position within a word. Consequently, typing Arabic often presents significant difficulties, particularly in code editors or terminals editors like Vim. However, Vim has made notable advancements in supporting Arabic text, which we'll explore next.

3.1 Enable arabic setting

The first step you need to take to use Arabic in Neovim is to enable the appropriate character encoding by typing: `:set encoding=utf-8`.

Then, To enable Arabic support in Neovim, you have three methods:

1. First method:

To start typing in Arabic with all its characteristics, simply enter the `":set arabic"` command. This command triggers several changes:

- It sets the appropriate keymap for Arabic input.
- It enables the deletion of single combined pair characters.
- It activates right-to-left (RTL) mode.
- It enables RTL command mode.
- It enables Arabic shaping mode, allowing visual character alterations.

You can disable Arabic support by using the `":set noarabic"` command, which resets all settings related to Arabic, including the keymap, direction, and shape.

2. Second method:

Alternatively, you can enable Arabic settings using a combination of commands:

- `":set keymap=arabic"` activates the Arabic keymap. You can toggle between Arabic and English keymaps using `'Ctrl-^'`. The Neovim terminal displays the current mapping as "insert arabic" for Arabic and "insert" for Latin.
- `":set rl"` or `":set rightleft"` changes the direction to RTL.

- `":set arabicshape"` activates the necessary visual character alterations required for Arabic language support, including shaping, composing, and combining.

3. Third method:

You can simply enable arabic setting in nvim by just opening your file in arabic mode using this command: `"nvim -R your-arabic-filename"`

3.2 Setting the Arabic fonts

You can set any Arabic font you prefer when writing in Neovim by typing:

```
:set guifont=your_ARABIC_FONT
```

For example, if you wish to use Amiri with a size of 11, enter the following command:

```
:set guifont=Amiri:h11
```

However, this command may not work if you are using nvim-qt. In this case, use the following command to resolve the issue:

```
:Guifont! Amiri:h11
```

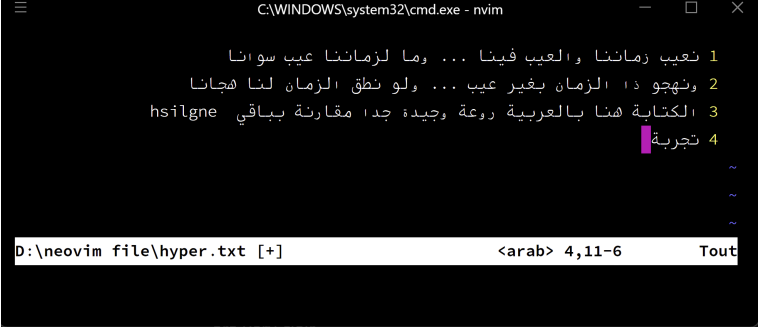
Before utilizing any font, ensure that it is successfully installed on your Windows system.

You can save all these settings in your `init.vim` file if you want Neovim to always open with these configurations.

4 My Observations on Arabic in Windows Terminals

When I initially used Neovim on Windows, I opened it in the Windows Command Prompt. The outcome was disappointing as it displayed question mark characters instead of Arabic letters. Subsequently, I installed other terminals to identify the optimal one for Arabic writing.

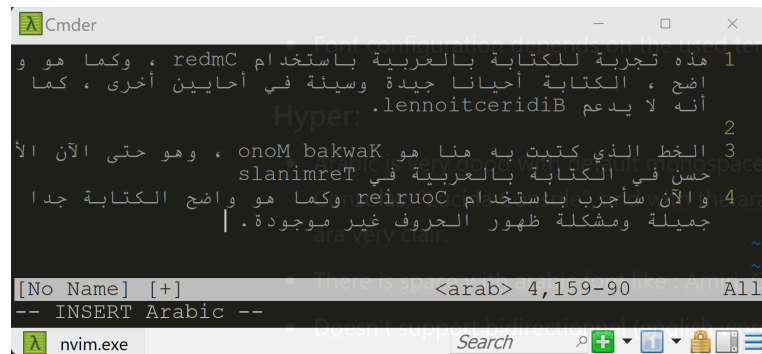
I have observed various behaviors of Arabic text rendering in different terminals on Windows. Below is a summary of my findings:

Terminal	Observations
Hyper	<ul style="list-style-type: none"> • Arabic displays well with default monospace fonts like Menlo, "DejaVu Sans Mono", Consolas, and "Lucida Console", as well as with the Arabic monospace font Kwakab Mono. • Some Arabic fonts like Amiri and Cairo introduce spacing issues. • Lacks bidirectional support (English words may appear reversed). • Configuration is straightforward: open the configuration file using 'CTRL-, '. <p>This picture shows the results of writing arabic in hyper with 'Courier' monospace font:</p> 

Cmdr

- Arabic appears perfectly in the terminal itself, but when using nvim, it displays question marks instead of Arabic letters. Even when Arabic letters are displayed, the shape may be distorted, resembling cmd and PowerShell behavior.
- Changing the font to Courier and setting the font charset to Arabic improves Arabic display significantly.
- Configuration is simple: right-click on the title bar at the top of the Cmdr window, select "Settings" from the context menu.

Arabic in Cmdr with 'Courier' font:

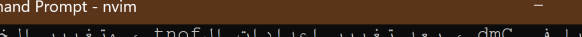


Arabic in Cmdr with 'Kawkab Mono' arabic font:



Cmd & PowerShell

- Only a few fonts display Arabic characters in cmd: Courier, which is the best option, Simplified Arabic Fixed, though some letters may appear as rectangular symbols, and Kawkab Mono.
- Configuration is straightforward, with limited options available.
- Only monospace fonts are supported.
- Bidirectional support is lacking.



Command Prompt - nvim

1 مرحبا في dmC ، بعد تغيير إعدادات الtnof ، وتغيير الخط إلى reiruoC ، أصبحت الكتابة جيدة بالعربية في .dmC.

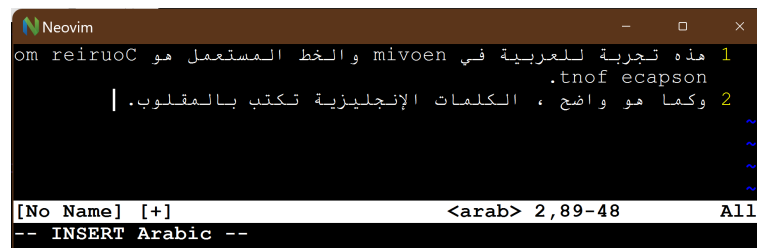
[No Name] [+] 1,169-103 All

-- INSERT --

Neovim QT

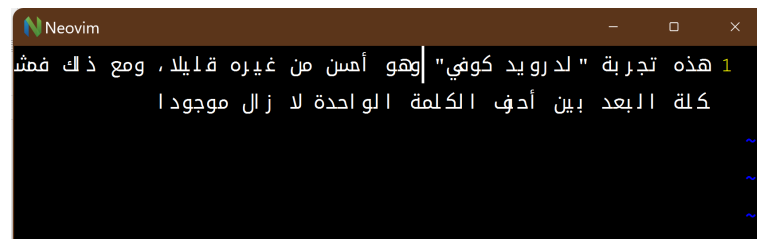
- Arabic characters display well with monospace fonts, particularly with Courier, showcasing a good shape.
- However, when using Arabic fonts, there are noticeable spaces between letters. Among the Arabic fonts tested, only Kwakab Mono, a monospace font, renders perfectly.
- Bidirectional text support is lacking.
- Configuration is straightforward, primarily relying on commands for customization.

Arabic in nvim-qt using 'Courier' font:



The screenshot shows a Neovim QT window with a dark theme. The text is displayed in a monospace font (Courier). The text is in Arabic and includes some English words. The text is: "1 هذه تجربة للعربية في mivoen والخط المستعمل هو reiruoC .tnof ecapson 2 وكما هو واضح ، الكلمات الإنجليزية تكتب بالمقلوب." The text is right-aligned. The status bar at the bottom shows "[No Name] [+] <arab> 2,89-48 All" and "-- INSERT Arabic --".

Arabic in nvim-qt using 'Droid arabic Kufi' font:



The screenshot shows a Neovim QT window with a dark theme. The text is displayed in a monospace font (Droid arabic Kufi). The text is in Arabic and includes some English words. The text is: "1 هذه تجربة "لارويد كوفي" وهو أحسن من غيره قليلا ، ومع ذلك فمش كلمة البعد بين أهدف الكلمة الواحدة لا زال موجودا". The text is right-aligned. The status bar at the bottom shows "[No Name] [+] <arab> 2,89-48 All" and "-- INSERT Arabic --".

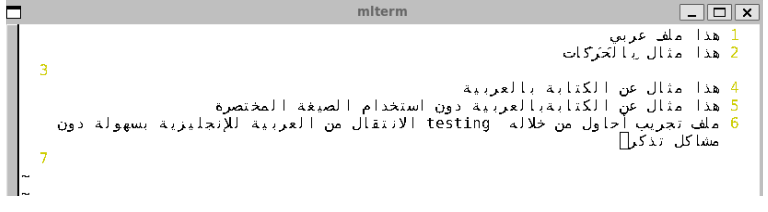
mlterm	<ul style="list-style-type: none"> • Arabic characters are rendered elegantly with monospace fonts, notably with Kawkab Mono and Droid Arabic Kufi. • One standout feature of mlterm is its bidirectional text support, allowing seamless integration of English text within Arabic text without any word inversions, unlike other terminals. • Configuration is easy and intuitive. Accessing the configuration menu is as simple as typing 'CTRL' and clicking the right button of the mouse. 
--------	--

Table 1: Summary of Observations on Arabic in Windows Terminals

In summary, after examining how different terminals handle Arabic text, we found that Neovim QT and mlterm have their strengths and weaknesses. Neovim QT displays Arabic text decently with certain fonts but struggles with spacing and lacks bidirectional support. On the other hand, mlterm excels in rendering Arabic text accurately, especially with monospace fonts, and it supports bidirectional text well. Overall, mlterm seems to be the better choice for users looking for reliable Arabic text support and easy configuration.

5 Conclusion

In conclusion, Neovim can support Arabic language for document writing, but the user experience may vary depending on the terminal used. Users seeking reliable Arabic text support may find mlterm to be a better choice due to its accurate ren-

dering and bidirectional text support. However, advancements in technology may lead to improvements in Arabic text handling across terminals in the future.