DIALOGUEDESIGNER

DOCUMENTATION

About x

Dialogue Designer is a powerful and easy to use branching dialogue editor designed for story-driven games like RPGs and Visual Novels. You can create complex non-linear data structures in minutes, no programming experience required.

The editor was created for non-programmers. It is easy to understand and navigate. It is also fast and lightweight - it runs smoothly even on less powerful PCs.

It includes a characters database and local variables database (strings, integers, booleans). They are incredibly helpful in organizing the dialogue and keeping track of player's progress. You can easily change the dialogue language by selecting it from a drop-down menu. There are no limits for the amount of languages you can use. Export your dialogue with one click to a JSON text file, which can be imported into most modern game engines and frameworks.

Software versions and compatibility x

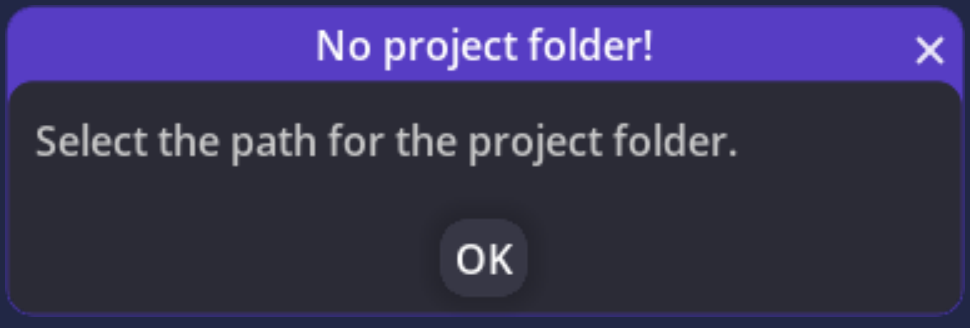
There have been 3 releases of DD so far: 1.0 , 2.0, and 3.0. This documentation focuses on the newest version 3.0 with updated interface, so in order to get the most out of it, it is recommended to get download the newest version (it's free for those who have bought any of the older versions).

The new releases are backwards compatible with dialogue files created using older versions. If you find any compatibility issues, please email me at radmattsoftware@gmail.com with a description of the issue.

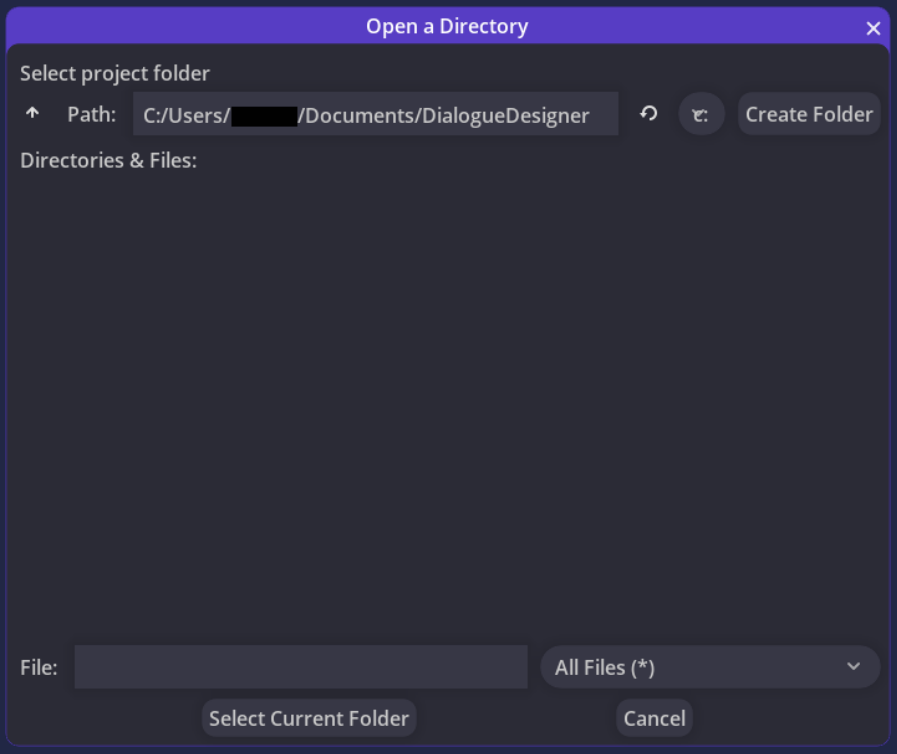
Getting started x

Dialogue Designer is available to download for Windows, Linux and Mac. You pay once and get access to ALL versions, so don't worry, if you ever switch operating systems, you will still be able to use the software and files created with it. With Linux and Windows versions, there come two types of .exe files (x64, x32 etc.). Please make sure you use the right version for your operating system. If you're not sure which one to use, it doesn't hurt to try running both of them and seeing which one works best for you.

The first time you launch Dialogue Designer you will be greeted with a very friendly message that says "No project folder!". Don't worry, it's how it's supposed to be. The software doesn't yet know where it has to save the dialogues created.

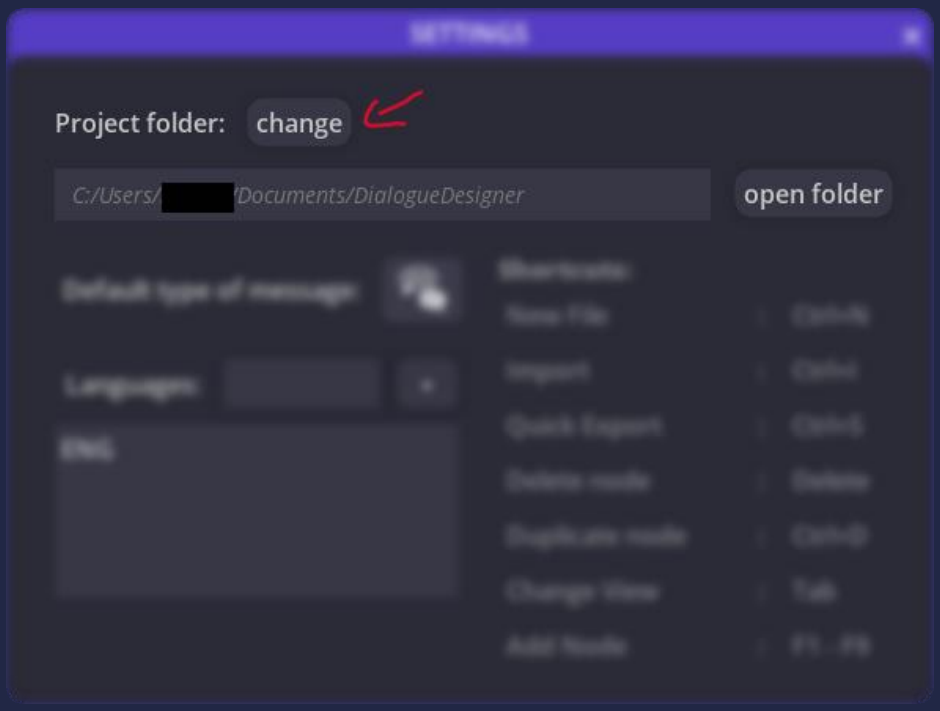


Click OK and then select the folder where you want your dialogues to be saved.



Important! If for some reason you don't set the project folder, the dialogues will be saved in: *C:/Users/[your name]/Documents/DialogueDesigner*.

You can change the projects folder at any time in the settings (see Interface section).



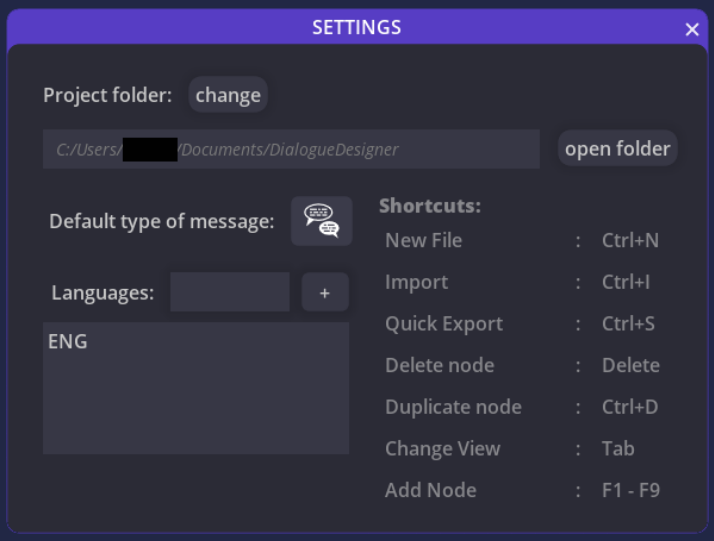
Press open folder button to open up the folder in your OS's file explorer.

Interface x

1. **Top bar** - you can acces all functions of the software from here.
2. **Import** - import JSON files created by Dialogue Designer.
3. **File name** - set the name of the file.
4. **Export** - export the file to the project folder (see previous section).
5. **Nodes** - used to create dialogues.
6. **Local variables** database.
7. **Characters** database.
8. **Language** - set the language of the text.
9. **Zoom** - change the view zoom.
10. **Scroll bars** - drag them to navigate around the editor.

Basic navigation:

- Press **Left Mouse Button** to select; hold to select more than one node.
- Hold **Middle Mouse Button** and move the mouse to pan the view (you can also use **SPACE** instead).
- Press **Right Mouse Button** to change node's name (text at the top).
- Use **mouse wheel** to scroll view vertically.
- While holding **SHIFT**, use with **mouse wheel** to scroll view horizontally.



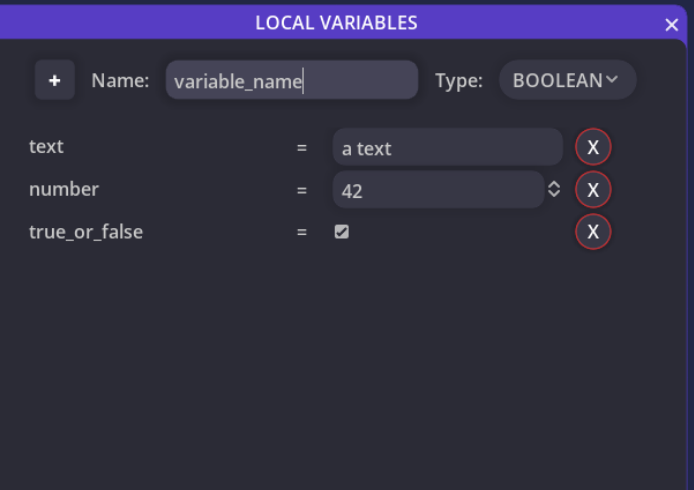
Settings:

Project folder - change the folder where the files will be saved. *Open folder* - open the project folder in your OS's file explorer.

Default type of message - change if the dialogue will appear in bubbles or boxes.

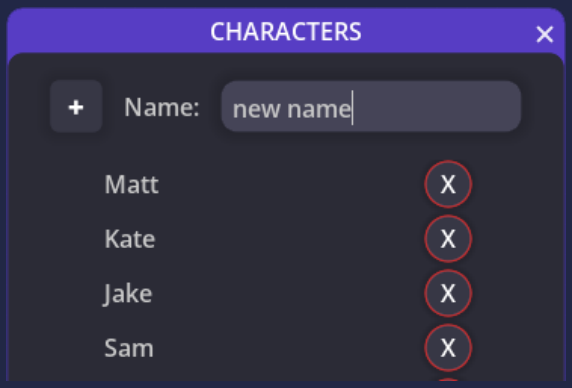
Languages - Add new languages to the dialogues. Write language's name in the box and then click the **+** button to add it.

Shortcuts - shortcuts that can help you use the editor easier and quicker (they can't be changed).



Variables database:

Type - type of the variable: **STRING** - text, **INTEGER** - number, **BOOLEAN** - true or false

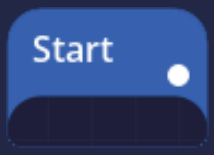


Characters database:

Shows characters appearing in the dialogue.

Creating dialogues

Once you have the editor ready, you will see the START node.



This is the first node you should find first in the file when creating your dialogue system.

Click on any of the colored circles above to add new nodes. They are color-coded:

- **RED** - dialogue creation
- **GREEN** - logic
- **BLUE** - variables
- **PURPLE** - comments

Exporting / Importing x

Exporting

Before you export the dialogue you must make sure that there is at least one node added (besides START) and that the file is named:

Then, press the export icon:

Only one file will be exported. That file can be used both for dialogues in games as well as editing later in the editor.

Importing

Before you import a dialogue make sure you saved the current one because it will overwrite the current dialogue tree.

File structure x

The files exported by the editor are in JSON format.
More: https://en.wikipedia.org/wiki/JSON

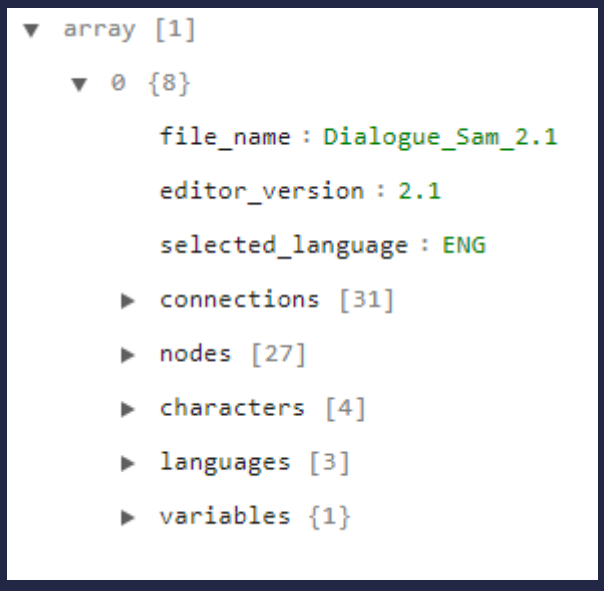
Example file: https://jsonblob.com/bb42fdcc-f807-11e9-b936-f52daeb23252

There are 4 main categories in the file:

- **nodes** - nodes used when constructing the dialogue
- **languages** - languages available
- **variables** - stores characters defined in ***Variables Database***
- **characters** - stores characters defined in ***Characters Database***

Data stored in those categories are used **for games** (you use them when writing your dialogue system).

(There are also categories used only by the editor. You don't need to care about them, they are just there so the editor can import them properly: editor_version, file_name, connections (connections between nodes), selected_language).



In code:

* variable types:

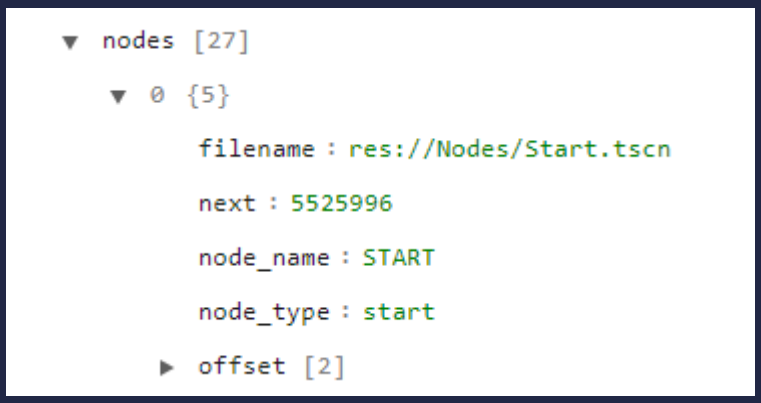
0 - *string* ;

1 - *integer* ;

2 - *boolean*.

Nodes:

Your dialogue will always start from the ***start node***;



The most important keys (categories) for you are ***next*** and ***node_type***.

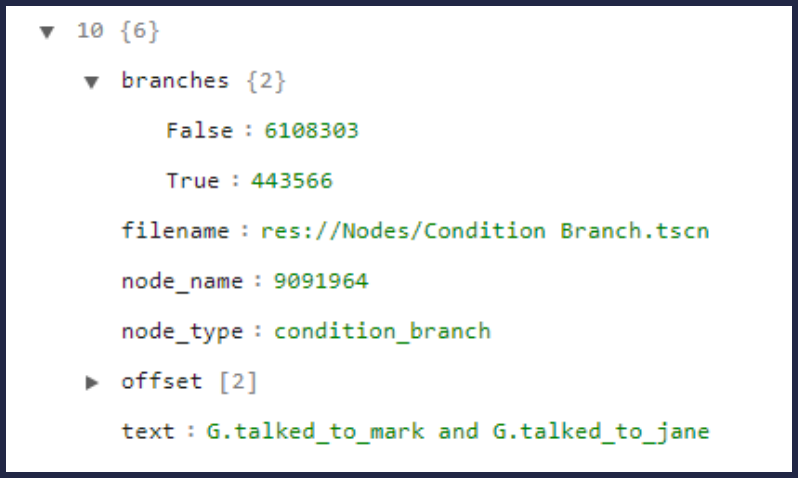
next - shows the ID (***node_name***) of the next node. If **next** is *null* then that means that the dialogue has ended.

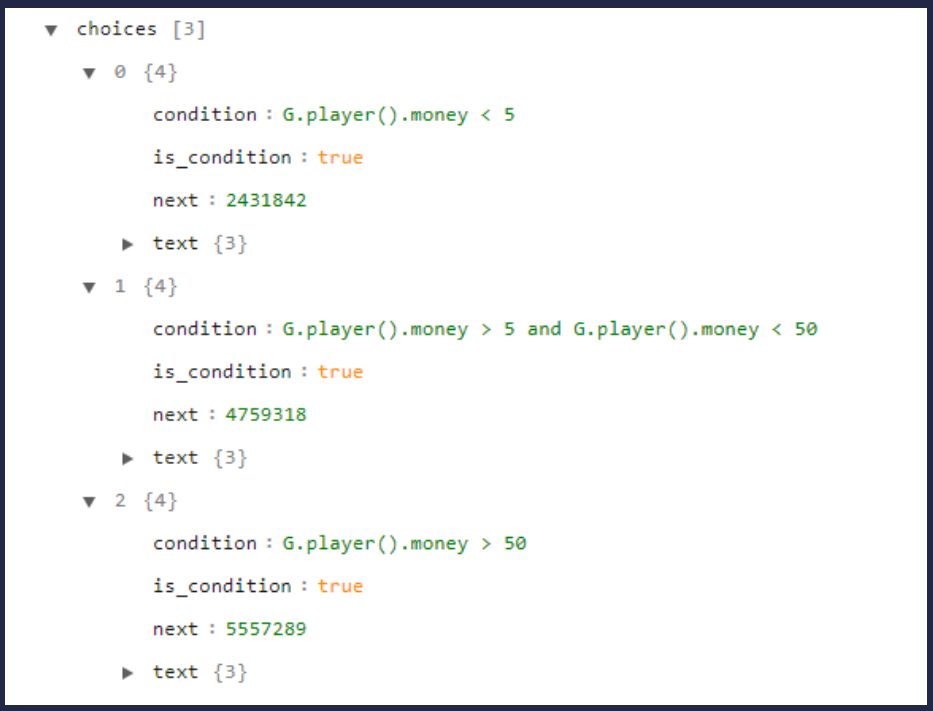
node_type - the type of node; this is useful because each node has its own set of keys and values that need to be read in a different way.

Branches and choices:

Sometimes you need more ***next*** values, for example when you want to introduce a little bit of non-linearity or give the player choices.

That's where ***branches*** and ***choices*** keys come in:





They work in the same way as standard next values but need to be checked with an *if statement* or a *switch* to get the ***next*** value the dialogue will use.

*Always use only one **next** value!*

Show dialogue node:

character - character which says the dialogue lines:

0 : [character's name]

1 : [character's numer]

- Speaker type:
  - **0** - character / **1** - custom path / **2** - Interacted character; this is used simply to check which value will be retrieved when selecting the speaker.

Using in game engines x

A few resources helpful in importing into games engines:

Godot - https://radmatt.itch.io/godot-dialogue-system

Unity - https://docs.unity3d.com/Manual/JSONSerialization.html

Unreal 4 - https://docs.unrealengine.com/en-US/API/Runtime/Json/Dom/FJsonObject/index.html

Construct - https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/jso...

Game Maker Studio - https://docs.yoyogames.com/source/dadiospice/002_reference/file handling/json_en...

Fix for **MacOS** users x

Some users may have problems running the editor because not all security permissions needed for MacOS are enabled in the executable file. The files are obviously 100% safe, there are no risks, so there's no need to worry.

SOLUTION 1:

To make the application executable:

1. Open a terminal window (CMD + Space -> terminal);
2. Using the cd command, navigate to the place where the application is stored: cd <path_to_application>
3. Run chmod +x <application_file> to make it executable. If it doesn't let you, sudo it: sudo chmod +x <application>.
(by Fanatique)

The full command should look similar to this:

chmod +x DialogueDesigner.app/Contents/MacOS/DialogueDesigner

SOLUTION 2:

You should be able to right-click on the app and select Open. Doing so will bring up the same warning dialog but should show an extra button that lets you continue on and open the app.

SOLUTION 3:

Adjust your security preferences in order to allow running unknown software.

Check these links for more info or if the above doesn't work:

https://superuser.com/questions/1345755/how-to-fix-the-application-cant-be-opene... or https://discussions.apple.com/thread/7586240

That's it! x

Created by **radmatt**.

radmattsoftware@gmail.com

YouTube

Twitter

Got any questions? Should something else be explained? Don't hesitate to contact me.