

# Toys2Life Character Creation Tutorial 05/10/2017

## Welcome to Toys2Life!

This tutorial will help you make your own Toys2Life character. Then, your character and any other Toys2Life character will be able to talk with each other!

Before you begin, take a moment to think about who your character is, and how they talk. Ask yourself:

- How does my character spend their time?
- What makes them happy, and what makes them sad?
- Is there anything unusual about them that you might not expect based on how they look?
- How do they talk? Do they like to use lots of fancy words, or do they keep it simple? Do they have any favorite phrases?

Now that you have a better idea of who your character is, make sure that you have extracted the Toys2Life folder to your computer. We'll begin with the StarterCharacter.JSON file, which you can find in the CharacterJSON folder.

This file contains nine sample phrases. Right now, these are phrases that Hoshi Kelvinson would say. (She's an astronaut who runs a food truck in space.) Change these nine phrases to things that *your* character would say, and you'll have enough dialogue to bring them to life!

## Creating Your Character File

Go ahead and open up the StarterCharacter.JSON file using notepad. (Sometimes, other word processors can cause problems.) At the top, you'll see this:

```
"CharacterName": "Hoshi Kelvinson",  
"CharacterPrefix": "HK",
```

**Replace Hoshi Kelvinson with your character's name.** (*Important:* throughout this tutorial, make sure to leave quotation marks, and any material outside of the quotation marks, as they are.) Then, **replace HK with two letters that stand for your**

## character's name.

Next, you'll see nine phrases. A “phrase” is something that your character will say, and each phrase in Toys2Life uses the same format. Here's an example of a phrase:

```
{
  "DialogStr": "My name's Hoshi Kelvinson.",
  "FileName": "MyNamesHoshi",
  "PhraseWeights": {
    "Greeting": 5
  },
  "PhraseRating": "G"
},
```

First is the **dialogue**. This is something that your character will actually say.

Second is the **file name**. When you make an audio recording of the dialogue, you'll use this to name the file.

Next is the **phrase type**. This determines what situations your character will say the dialogue in. For example, your character will use a “Greeting” when they want to say hello or introduce themselves.

Then, we have the **phrase weight**, which is a number. This determines how often the character will say this phrase, compared to other phrases of the same type. The bigger the number, the more often they will say this phrase.

Last, we have the **phrase rating**. Phrases that are appropriate for kids can be rated “G” or “PG”, just like a movie.

For this tutorial, we will only need to change the **dialogue** and **file names**. We can leave the rest of the material alone.

Let's go through each of the nine phrases, starting with a **Greeting**. (In the example below, notice that the phrase type is listed at “Greeting”.)

```
{
```

```

    "DialogStr": "My name's Hoshi Kelvinson.",
    "FileName": "MyNamesHoshi",
    "PhraseWeights": {
        "Greeting": 5
    },

    "PhraseRating": "G"
},

```

To write a Greeting, replace the **dialogue** here with the way your character says “hello”! Maybe they like to introduce themselves, like Hoshi does. If your character is a cowgirl or cowboy, maybe they say “Howdy!”

Then, change the **file name** to match what you just wrote for the **dialogue**. It doesn’t have to match exactly—you can just use the first few words of the dialogue to name your file. (*Important:* only use letters and numbers in the file name. Don’t use spaces, punctuation marks, or special characters.)

Second, we have a **YesNoQuestion**.

```

{
    "DialogStr": "Have you ever seen the sun rise from the moon?",
    "FileName": "HaveYouEverSeenTheSun",
    "PhraseWeights": {
        "YesNoQuestion": 5
    },

    "PhraseRating": "G"
},

```

To write a YesNoQuestion, replace the **dialogue** here with any question your character might ask that could be answered with a “yes” or a “no.” If your character is an elephant, they might say, “Do you like my trunk?” Then, replace the **file name** to match the start of the dialogue, just like the last phrase.

Third, we have a **GiveCredit**.

```

{
    "DialogStr": "You fixed my laser skillet!",

```

```

    "FileName": "YouFixedMy",
    "PhraseWeights": {
        "GiveCredit": 5
    },

    "PhraseRating": "G"
},

```

To write a GiveCredit, imagine that your character is talking to someone who did something nice for them. Your character is saying what that kind person did. Here, Hoshi is happy that someone fixed her laser skillet. (It would be hard to make space pancakes without it.) If your character is a basketball fan, maybe they would say “You got me playoff tickets!” Replace the **dialogue** and the **file name**. Make sure that the dialogue starts with “You”.

Fourth, we have a **RequestAffirmation**.

```

{
    "DialogStr": "Sometimes, when I'm flipping pancakes out past Pluto and there aren't any customers in line, I get a little lonely.",
    "FileName": "SometimesWhenImFlipping",
    "PhraseWeights": {
        "RequestAffirmation": 5
    },

    "PhraseRating": "G"
},

```

To write a RequestAffirmation, imagine what makes your character feel sad, or sick, or unconfident. How would they talk about that? It sounds like Hoshi feels blue when she doesn’t have any customers. If your character is a vampire, they might say “I don’t feel quite right. I’ve spent too long away from my coffin.” Replace the **dialogue** and the **file name**.

Fifth, we have a **GiveAffirmation**.

```

{
    "DialogStr": "Pay me a visit the next time you're in space. I'll keep a glass of orange juice cold, just for you.",

```

```

    "FileName": "PayMeAVisit",
    "PhraseWeights": {
        "GiveAffirmation": 5
    },

    "PhraseRating": "G"
},

```

To write a GiveAffirmation, think about what your character would say to cheer somebody else up. Hoshi is offering a nice glass of OJ. Your character might give a compliment. If they like animals, they might say “You’re sweeter than a basket full of kittens.” Replace the **dialogue** and the **file name**.

Last, we have a **Yes**, a **No**, a **ThankYou**, and a **YoureWelcome**. As you’ve probably guessed, these are the ways that your character will say “Yes,” “No,” “Thank you,” and “You’re welcome.” You can leave these four phrases as they are, if you like. If your character has a special way of saying any of these things, you can also change them, just like the previous phrases. For example, if your character knows Spanish, they might say “Gracias” instead of “Thank you.” If your character is a pirate, they might say “Aye, matey!” instead of “Yes.”

Make sure to save before closing the file. Then, **rename the file to the character’s name**. Great job—you’ve just written your first Toys2Life character! There are just a few more steps before you get started.

## Making Audio for Your Character

Of course, to make your character actually talk, you’ll need audio of them talking. You can use your own voice for this—or maybe you’d rather record a friend! There were nine phrases in the character file you just finished, and you will need to make a separate mp3 audio file for each one.

You will name each file based on the character prefix and file names we discussed earlier. The format is CharacterPrefix\_FileName . Remember that you listed the CharacterPrefix at the start of the JSON file. For example:

```

"CharacterName": "Hoshi Kelvinson",
"CharacterPrefix": "HK",
{
  "DialogStr": "My name's Hoshi Kelvinson.",
  "FileName": "MyNamesHoshi",
  "PhraseWeights": {
    "Greeting": 5
  },
  "PhraseRating": "G"
},

```

For this phrase and character, you would name the mp3 this way:

HK\_MyNamesHoshi

The free program Audacity is very useful for recording and editing mp3 files. After you have made and named all nine mp3s, put them all in the DialogAudio folder.

## Last Steps

To finish up, you will need to open the DialogEngine.exe CONFIG file using notepad. You can find this in the Toys2Life folder. (The extension of this file is .config. Be careful not to confuse it with the .exe application which is also called DialogEngine.)

You will need to make a couple of changes to this file, so that Toys2Life will know where to find the right files on your computer. Towards the end of the file, you will see the keys “CharactersDirectory”, “DialogsDirectory”, “LogsDirectory” and “AudioDirectory”:

```

<add key="CharactersDirectory"
value="C:\Isaac\Toys2Life\DlgEng\CharacterJSON\"/>

<add key="DialogsDirectory"
value="C:\Isaac\Toys2Life\DlgEng\DialogJSON\"/>

<add key="LogsDirectory" value="C:\Isaac\Toys2Life\DlgEng\Logs\"/>

<add key="AudioDirectory"

```

```
value="C:\Isaac\Toys2Life\DlgEng\DialogAudio\"/>
```

You will need to replace the highlighted portions with wherever you put the Toys2Life folder on your computer. For example, if you put the folder on C:\Users\Hoshi\Desktop\, You would want to change the keys as follows:

```
<add key="CharactersDirectory"
value="C:\Users\Hoshi\Desktop\Toys2Life\DlgEng\CharacterJSON\"/>

<add key="DialogsDirectory"
value="C:\Users\Hoshi\Desktop\Toys2Life\DlgEng\DialogJSON\"/>

<add key="LogsDirectory"
value="C:\Users\Hoshi\Desktop\Toys2Life\DlgEng\Logs\"/>

<add key="AudioDirectory"
value="C:\Users\Hoshi\Desktop\Toys2Life\DlgEng\DialogAudio\"/>
```

Make sure to save the file before exiting.

Congratulations—that's it! You're now ready to bring your character to life. If you double click on the DialogEngine.exe file you may need to press enter a few times before you see and hear the character's dialog.

## Troubleshooting

*If your character is not working, there may be a mistake on your character's JSON file. To find out if this is the case, go to <http://www.jsoneditoronline.org/>. Click "open," and then "open from disk." Then, select the JSON file with your character's name. If there is a problem with the file, you will see an error message. In the window on the left, you will find a red "x." This will give you a clue where the problem may be. You may have a missing punctuation mark, like a comma or quotation mark. Or, you may have an extra punctuation mark where there should not be any. After fixing this in the JSON file using notepad, you can open it in the editor again to check if the problem is solved.*