

The Creation of a Command Line Tool that can View Minecraft Data

This week's blog post is about developing an command line tool to display Minecraft NBT files. You will be taught:

Named Binary Tag (NBT) files store information about Minecraft in a basic binary file format. To know more about the details of the NBT format, read these documents:

"NBT format" via [Minecraft Wiki](#). "NBT" via [Minecraft.vg](#). "Named Binary Tag specification" via [Minecraft.net \(WebArchive\)](#).

Script source code

```
view-nbt
```

The script can be used to view an NBT file

I have an NBT file, level.dat, that contains information about a Minecraft level I play on:

To look at the data in a readable format I use view-nbt.

(level.dat files are compressed, so I use the --gzip flag.)

How the script functions

NBT files may contain tags of many different types. I used an Enum.Enum Tag , to represent them.

To assist in parsing process, I designed several data structures to aid in parsing.

Scalar_fmt is an fmt dictionary that maps certain Tag objects to format strings usable by the struct module. This is for tags that are simple such as Tag.Int and Tag.Byte. array_fmt is an array which converts Tag objects into format strings that can be used by the struct module (but missing a length prefix). This is for simple array tags, such as Tag.Int_Array. byte_tag is an array of byte-sized data that maps bytes to the appropriate tags. I learned this from

studying the NBT specification for file formats.

The struct module is used by the parser to break down binary data into different numbers (like short double, short, and int).

The parser is recursive.

It directly parses Tag.String as well as tags in array_fmt and scalar_fmt. - It recursively parses Tag.List and Tag.Compound.

The parsing creates a Node object, which is a collections.namedtuple that holds:

- The tag type. Optional: The tag name. The value of the tag.

To make it easier to read the data, I developed an algorithm called json_friendly that converts Node objects into data structures.

To support NBT files that are compressed using Python's gzip module, I use it.

I employ Python's json module to convert the JSON friendly data into a string.

Let's conclude...

This week's blog post will show you how to create a command line tool to view Minecraft NBT files. You learned:

How to decode binary data with the struct module. How to write a recursive parser for the NBT file format. How to convert the parsed data into the JSON compatible file format.

My challenge to you is:

Create a tool, view-region which converts Region files into JSON that is readable.

If you liked this week's blog, please share it with your friends and keep an eye out for next week's post. We'll see you next time!

Pyramids