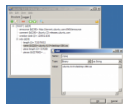


Index

µTorrent Adder
µTorrent MiniUI
BEncode Editor
History
pSX Frontend

BEncode Editor

Introduction



BEncoding is a data encoding scheme used primarily in the BitTorrent world. Because BEncoded files can contain binary data, and because of some of the intricacies involved in the way binary strings are stored, it is often not safe to edit such files in text editors. Many people have previously asked about where one might find an editor that can safely edit BEncoded files, but none have really existed in the general sense. The only ones I've seen were .torrent file editors, which (although they technically do edit BEncoded files) don't edit just ANY type of BEncoded file. Out of a bit of boredom and free time, this editor was born.

Example Uses

- **Edit .torrent files**
 - the announce/announce-list keys (binary/list) store the tracker(s)
 - the url-list key (binary/list) stores the webseed(s)
 - the nodes key (list) stores the DHT bootstrap node(s)
 - **WARNING:** When editing .torrent files, any changes to the info dictionary will change the .torrent file's infohash. Unless you know what you're doing, you should refrain from doing this. If you aren't familiar with the .torrent file metadata structure, read this: <http://wiki.theory.org/BitTorrentSpecification> (in particular, the "Metainfo File Structure" section).
- **Edit µTorrent's .dat files**
 - corrupt resume.dat files can sometimes be salvaged simply by loading and saving the resume.dat file in this editor
 - the paths stored in resume.dat can be edited en masse with Find/Replace
 - **WARNING:** Make sure you exit µTorrent before editing these files, as µTorrent rewrites/updates the files on exit. Additionally, the .fileguard key should be removed, since µTorrent will consider its .dat file to be damaged if it is edited and no longer matches the stored .fileguard hash.

Final Warning

This utility is for advanced users. Read the warnings above, and read them again until you understand well what you're getting yourself into by using this editor. Its relatively intuitive UI can beguile most uninitiated/beginning users, as it has an inherent ability to invalidate many files through *wrong* and *incorrect* edits.

Consider this editor to be experimental. As such, you should exercise caution by backing up any files you plan on editing with this utility.

Download

[v0.7.1.0] [x64, Unicode](#)
[v0.6.1.0] [Unicode, ANSI](#)
[v0.5.1.0] [Unicode, ANSI](#)
[v0.4.1.0] [Unicode, ANSI](#)
[v0.3.0.0] [Unicode, ANSI](#)
[v0.2.0.0] [Unicode](#)
[v0.1.0.0] [Unicode](#)

x64 builds will run only on 64-bit versions of Windows. Unicode builds can display Unicode characters (like Asian characters) properly, but work only with Windows 2000 and above. ANSI builds can't display Unicode characters properly, but should work on Windows 9x and above.

Notes

- Binary data and integers are exported as raw data rather than BEncoded data
- Dictionary merging does not sort keys or resolve duplicate keys
- File recovery recovers only decodable parts of input files
- Finding "Value by Key" finds exact key names unless RegEx is used
- Holding Ctrl while reloading reloads data from disk
- Holding Shift while moving moves an item to top/bottom (direction-dependent)
- Holding Shift while pasting pastes the item below the currently focused item
- Holding Shift while sorting will recursively search for dictionaries to sort
- Only dictionaries can be sorted
- Only files containing dictionaries and lists can be edited directly
- Switching an item between dictionary/list will cause the item to be cleared
- The root of a dictionary (not its children) must be selected to sort it

Final Thoughts

Bug reports, suggestions, and the like are always welcome... Just be aware that there are no guarantees that I'll ever get around to doing anything about them. I might not do anything about them for a period of time, and then I might suddenly work on them in a "spur of the moment" kind of... uh, whatever.

Subpages (1): [History](#)