multiformats / **multicodec**

Code Issues 38 Pull requests 25 Actions Projects Security Ins

multicodec / README.md

erwin-kok Add Kotlin implementation 4 months ago

129 lines (81 loc) · 7.04 KB

Preview Code Blame Raw

multicodec

made by Protocol Labs | project multiformats | freenode #ipfs | readme style standard

> Canonical table of of codecs used by various multiformats

Table of Contents

Motivation

Multicodec is an agreed-upon codec table. It is designed for use in binary representations, such as keys or identifiers (i.e CID).

Description

The code of a multicodec is usually encoded as unsigned varint as defined by multiformats/unsigned-varint. It is then used as a prefix to identify the data that follows.

Examples

Multicodec is used in various Multiformats. In Multihash it is used to identify the hashes, in the machine-readable Multiaddr to identify components such as IP addresses, domain names, identities, etc.

Multicodec table

Find the canonical table of multicodecs at table.csv. There's also a sortable viewer.

Status

Each multicodec is marked with a status:

- draft - this codec has been reserved but may be reassigned if it doesn't gain wide adoption.
- permanent - this codec has been widely adopted and may not reassigned.
- deprecated - this codec has been deprecated.

NOTE: Just because a codec is marked draft, don't assume that it can be re-assigned. Check to see if it ever gained wide adoption and, if so, mark it as permanent.

Adding new multicodecs to the table

The process to add a new multicodec to the table is the following:

1. Fork this repo
2. Add your codecs to the table. Each newly proposed codec must have:
3. A unique codec.
4. A unique name.
5. A category.
6. A status of "draft".
7. Submit a Pull Request

This "first come, first assign" policy is a way to assign codes as they are most needed, without increasing the size of the table (and therefore the size of the multicodecs) too rapidly.

The first 127 bits are encoded as a single-byte varint, hence they are reserved for the most widely used multicodecs. So if you are adding your own codec to the table, you most likely would want to ask for a codec bigger than `0x80` .

Codec names should be easily convertible to constants in common programming languages using basic transformation rules (e.g. upper-case, conversion of `-` to `_` , etc.). Therefore they should contain alphanumeric characters, with the first character being alphabetic. The primary delimiter for multi-part names should be `-` , with `_` reserved for cases where a secondary delimiter is required. For example: `bls12_381-g1-pub` contains 3 parts: `bls_381` , `g1` and `pub` , where `bls_381` is "BLS 381" which is not commonly written as "BLS381" and therefore requires a secondary separator.

The `validate.py` script can be used to validate the table once it's edited.

Implementations

- go
- JavaScript
- Python
    - py-multicodec
    - `multicodec` sub-module of Python module multiformats
- Haskell
- Elixir
- Scala
- Ruby
- Java
- Kotlin
    - `multicodec` part of Kotlin project multiformat
- Add yours today!

Reserved Code Ranges

The following code ranges have special meaning and may only have meanings assigned to as specified in their description:

Private Use Area

Range: `0x300000 – 0x3FFFFF`

Codes in this range are reserved for internal use by applications and will never be assigned any meaning as part of the Multicodec specification.

FAQ

> Why varints?

So that we have no limitation on protocols.

> What kind of varints?

An Most Significant Bit unsigned varint, as defined by the multiformats/unsigned-varint.

> Don't we have to agree on a table of protocols?

Yes, but we already have to agree on what protocols themselves are, so this is not so hard. The table even leaves some room for custom protocol paths, or you can use your own tables. The standard table is only for common things.

> Where did multibase go?

For a period of time, the multibase prefixes lived in this table. However, multibase prefixes are *symbols* that may map to *multiple* underlying byte representations (that may overlap with byte sequences used for other multicodecs). Including them in a table for binary/byte identifiers lead to more confusion than it solved.

You can still find the table in multibase.csv.

> Can I use multicodec for my own purpose?

Sure, you can use multicodec whenever you have the need for self-identifiable data. Just prefix your own data with the corresponding varint encodec multicodec.

Contribute

Contributions welcome. Please check out the issues.

Check out our contributing document for more information on how we work, and about contributing in general. Please be aware that all interactions related to multiformats are subject to the IPFS Code of Conduct.

Small note: If editing the README, please conform to the standard-readme specification.

License