NAME

gitformat-bundle - The bundle file format

SYNOPSIS

*.bundle
*.bdl

DESCRIPTION

The Git bundle format is a format that represents both refs and Git objects. A bundle is a header in a format similar to **git-show-ref**(1) followed by a pack in *.pack format.

The format is created and read by the **git-bundle**(1) command, and supported by e.g. **git-fetch**(1) and **git-clone**(1).

FORMAT

We will use ABNF notation to define the Git bundle format. See **gitprotocol-common**(5) for the details.

A v2 bundle looks like this:

```
bundle    = signature *prerequisite *reference LF pack
signature = "# v2 git bundle" LF

prerequisite = "-" obj-id SP comment LF
comment      = *CHAR
reference    = obj-id SP refname LF

pack         = ... ; packfile
```

A v3 bundle looks like this:

```
bundle    = signature *capability *prerequisite *reference LF pack
signature = "# v3 git bundle" LF

capability   = "@" key ["=" value] LF
prerequisite = "-" obj-id SP comment LF
comment      = *CHAR
reference    = obj-id SP refname LF
key          = 1*(ALPHA / DIGIT / "-")
```

value = *(%01-09 / %0b-FF)

pack = ... ; packfile

SEMANTICS

A Git bundle consists of several parts.

- which are only in the v3 format, indicate functionality that the bundle requires to be read properly.

- list the objects that are NOT included in the bundle and the reader of the bundle MUST already have, in order to use the data in the bundle. The objects stored in the bundle may refer to prerequisite objects and anything reachable from them (e.g. a tree object in the bundle can reference a blob that is reachable from a prerequisite) and/or expressed as a delta against prerequisite objects.

- record the tips of the history graph, iow, what the reader of the bundle CAN "git fetch" from it.

- is the pack data stream "git fetch" would send, if you fetch from a repository that has the references recorded in the "References" above into a repository that has references pointing at the objects listed in "Prerequisites" above.

In the bundle format, there can be a comment following a prerequisite obj-id. This is a comment and it has no specific meaning. The writer of the bundle MAY put any string here. The reader of the bundle MUST ignore the comment.

Note on shallow clones and Git bundles

Note that the prerequisites do not represent a shallow-clone boundary. The semantics of the prerequisites and the shallow-clone boundaries are different, and the Git bundle v2 format cannot represent a shallow clone repository.

CAPABILITIES

Because there is no opportunity for negotiation, unknown capabilities cause *git bundle* to abort.

- specifies the hash algorithm in use, and can take the same values as the **extensions.objectFormat** configuration value.

specifies an object filter as in the **--filter** option in **git-rev-list**(1). The resulting pack-file must be marked as a **.promisor** pack-file after it is unbundled.

GIT

Part of the **git**(1) suite