
libfv Documentation

Release v0.7.0

Christopher Simpkins

Jul 04, 2018

Contents:

```
class fontv.libfv.FontVersion (font, develop='DEV', release='RELEASE', sha1_develop='-dev',
                               sha1_release='-release')
```

FontVersion is a ttf and otf font version string class that provides support for font version string reads, reporting, modification, & writes. It provides full support for the OpenFV font versioning specification (<https://github.com/openfv/openfv>). Support is provided for instantiation from ttf and otf fonts, as well as from fontTools.ttLib.ttf objects (<https://github.com/fonttools/fonttools>).

The class works on Python “strings”. String types indicated below refer to the Python2 unicode type and Python3 string type.

PUBLIC ATTRIBUTES:

`contains_metadata`: (boolean) boolean for presence of metadata in version string

`contains_state`: (boolean) boolean for presence of state substring metadata in the version string

`contains_status`: (boolean) boolean for presence of development/release status substring in the version string

`develop_string`: (string) The string to use for development builds in the absence of git commit SHA1 string

`fontpath`: (string) The path to the font file

`is_development`: (boolean) boolean for presence of development status substring at `version_string_parts[1]`

`is_release`: (boolean) boolean for presence of release status status substring at `version_string_parts[1]`

`metadata`: (list) A list of metadata substrings in the version string. Either `version_string_parts[1:]` or empty list

`release_string`: (string) The string to use for release builds in the absence of git commit SHA1 string

`sha1_develop`: (string) The string to append to the git SHA1 hash string for development builds

`sha1_release`: (string) The string to append to the git SHA1 hash string for release builds

`state`: (string) The state metadata substring

`ttf`: (fontTools.ttLib.TTFont) for font file

`version_string_parts`: (list) List that maintains in memory semicolon parsed substrings of font version string

`version`: (string) The version number substring formatted as “Version X.XXX”

PRIVATE ATTRIBUTES

`_nameID_5_dict`: (dictionary) {(platformID, platEncID, langID) : fontTools.ttLib.TTFont name record ID 5 object } map

Parameters

- **font** – (string) file path to the .otf or .ttf font file OR (ttLib.TTFont) object for appropriate font file
- **develop** – (string) the string to use for development builds in the absence of git commit SHA1 string
- **release** – (string) the string to use for release builds in the absence of a git commit SHA1 string
- **sha1_develop** – (string) the string to append to the git SHA1 hash string for development builds
- **sha1_release** – (string) the string to append to the git SHA1 hash string for release builds

Raises fontTools.ttLib.TTLibError if fontpath is not a ttf or otf font

Raises IndexError if there are no nameID 5 records in the font name table

Raises IOError if fontpath does not exist

`_get_repo_commit()`

Private method that makes a system git call via the GitPython library and returns a short git commit SHA1 hash string for the commit at HEAD using *git rev-list*.

Returns (string) short git commit SHA1 hash string

`_is_development_substring(needle)`

Private method that returns a boolean that indicates whether the needle string meets the OpenFV specification definition of a Development Status metadata substring.

See OpenFV specification for Status substring definition details (<https://github.com/openfv/openfv>)

Parameters **needle** – (string) test string

Returns boolean True = is development substring and False = is not a development substring

`_is_release_substring(needle)`

Private method that returns a boolean that indicates whether the needle string meets the OpenFV specification definition of a Release Status metadata substring.

See OpenFV specification for Status substring definition details (<https://github.com/openfv/openfv>)

Parameters **needle** – (string) test string

Returns boolean True = is release substring and False = is not a release substring

`_is_state_substring_return_state_match(needle)`

Private method that returns a tuple of boolean, string. The boolean value reflects the truth test needle is a State substring. The match value is defined as the contents inside [and] delimiters as defined by the regex pattern. If there is no match, the string item in the tuple is an empty string.

See OpenFV specification for State substring definition details (<https://github.com/openfv/openfv>)

Parameters **needle** – (string) test string to attempt match for state substring

Returns (boolean, string) see full docstring for details re: interpretation of returned values

`_parse()`

Private method that parses version string data to set FontVersion object attributes. Called on FontVersion object instantiation and at the completion of setter methods in the library in order to update object attributes with new data.

Returns None

`_parse_metadata()`

Private method that parses a font version string for semicolon delimited font version string metadata. Metadata are defined as anything beyond the first substring item of a version string.

See OpenFV specification for version substring definition details (<https://github.com/openfv/openfv>)

Returns None

`_parse_state()`

Private method that parses a font version string for [...] delimited data that represents the State substring as defined by the OpenFV specification. The result of this test is used to define State data in the FontVersion object.

See OpenFV specification for the state substring metadata definition (<https://github.com/openfv/openfv>)

Returns None

`_parse_status()`

Private method that parses a font version string to determine if it contains development/release Status

substring metadata as defined by the OpenFV specification. The result of this test is used to define Status data in the FontVersion object.

See OpenFV specification for the Status substring metadata definition (<https://github.com/openfv/openfv>)

Returns None

`_parse_version_substrings` (*version_string*)

Private method that splits a full semicolon delimited version string on semicolon characters to a Python list.

Parameters **version_string** – (string) the semicolon delimited version string to split

Returns None

`_read_version_string` ()

Private method that reads OpenType name ID 5 and head.fontRevision record data from a fontTools.ttLib.ttFont object and sets FontVersion object properties. The method is called on instantiation of a FontVersion object

Returns None

`_set_state_status_substring` (*state_status_string*)

Private method that sets the State/Status substring in the FontVersion.version_string_parts[1] list position. The method preserves Other metadata when present in the version string.

See OpenFV specification for State/Status substring and Other metadata definition details (<https://github.com/openfv/openfv>)

Parameters **state_status_string** – (string) the string value to insert at the status substring position of the self.version_string_parts list

Returns None

`clear_metadata` ()

Public method that clears all version string metadata in memory. This results in a version string that ONLY includes the version number substring. The intent is to support removal of unnecessary version string data that are included in a font binary.

See OpenFV specification for Version number substring and Metadata definition details (<https://github.com/openfv/openfv>)

Returns None

`get_head_fontrevision_version_number` ()

Public method that returns the version number that is parsed from head.fontRevision record as a float value.

Returns float

`get_metadata_list` ()

Public method that returns a Python list containing metadata substring items generated by splitting the string on a semicolon delimiter. Metadata are defined according to the OpenFV specification. The version number string (i.e. “Version X.XXX”) is not present in this list.

See OpenFV specification for the version string Metadata definition (<https://github.com/openfv/openfv>)

Returns list of string (Python 3) or list of unicode (Python 2)

`get_name_id5_version_string` ()

Public method that returns the full version string as the semicolon delimiter joined contents of the FontVersion.version_string_parts Python list.

Returns string (Python 3) or unicode (Python 2)

get_state_status_substring()

Public method that returns the State and/or Status substring at position 2 of the semicolon delimited version string. This substring may include any of the following metadata according to the OpenFV specification:

- “DEV”
- “RELEASE”
- “[state]-dev”
- “[state]-release”

See OpenFV specification for State and Status substring definitions (<https://github.com/openfv/openfv>)

Returns string (Python 3) or unicode (Python 2), empty string if this substring is not set in the font

get_version_number_string()

Public method that returns a string of the version number in XXX.XXX format. A version number match is defined according to the OpenFV specification with up to three digits on either side of the period.

See OpenFV specification for the font version number format definition and semantics (<https://github.com/openfv/openfv>)

Returns string (Python 3) or unicode (Python 2). Empty string if unable to parse version number format

get_version_number_tuple()

Public method that returns a tuple of integer values with the following definition:

(major version, minor version position 1, minor version position 2, minor version position 3)

where position is the decimal position of the integer in the minor version string. The version number format is defined by the OpenFV specification.

See OpenFV specification for the font version number format definition and semantics (<https://github.com/openfv/openfv>)

Returns tuple of integers or None if the version number substring is inappropriately formatted

set_development_status()

Public method that sets the in memory Development Status metadata substring for the font version string.

See OpenFV specification for Status substring and Development status definition (<https://github.com/openfv/openfv>)

Returns None

set_release_status()

Public method that sets the in memory Release Status metadata substring for the font version string.

See OpenFV specification for Status substring and Release status definition details (<https://github.com/openfv/openfv>)

Returns None

set_state_git_commit_sha1(development=False, release=False)

Public method that adds a git commit sha1 hash label to the font version string at the State metadata position as defined by the OpenFV specification. This can be combined with a Development/Release Status metadata substring if the calling code defines either the development or release parameter to a value of True. Note that development and release are mutually exclusive. ValueError is raised if both are set to True. The font source must be under git version control in order to use this method. If the font source is not under git version control, an IOError is raised during the attempt to locate the .git directory in the project.

See OpenFV specification for State substring definition details (<https://github.com/openfv/openfv>)

Parameters

- **development** – (boolean) False (default) = do not add development status indicator; True = add indicator
- **release** – (boolean) False (default) = do not add release status indicator; True = add indicator

Raises IOError when the git repository root cannot be identified using the directory traversal in the fontv.utilities.get_git_root_path() function

Raises ValueError when calling code sets both development and release parameters to True as these are mutually exclusive requests

Returns None

set_version_number (*version_number*)

Public method that sets the version number substring with the version_number parameter. The version_number parameter should follow the OpenFV specification for the font version number format.

See OpenFV specification for the font version number definition and semantics (<https://github.com/openfv/openfv>)

The method will raise ValueError if the version_string cannot be cast to a float type. This is mandatory for the definition of the head table fontRevision record definition in the font binary. Attempts to add metadata strings to the version_number violate the OpenFV specification and are intentionally not permitted.

Parameters **version_number** – (string) version number in X.XXX format where X are integers

Returns None

set_version_string (*version_string*)

Public method that sets the entire version string (including metadata if desired) with a version_string parameter. The version_string parameter should be formatted according to the OpenFV font versioning specification (<https://github.com/openfv/openfv>) for the OpenType name table ID 5 record version string.

The method will raise a ValueError if the version number used in the version_string cannot be cast to a float type. This is mandatory for the definition of the head table fontRevision record definition in the font binary. Attempts to add metadata strings to the version_number violate the OpenFV specification and are intentionally not permitted.

Parameters **version_string** – (string) The version string with semicolon delimited metadata (if metadata are included)

Returns None

write_version_string (*fontpath=None*)

Public method that writes the in memory version data to:

1. each OpenType name table ID 5 record in original font file
2. OpenType head table fontRevision record

The name table ID 5 record(s) write is with a semicolon joined list of the items in FontVersion.version_string_parts

The head table fontRevision record write is with the version number float value in FontVersion.head_fontRevision

The write is to a .otf file if the FontVersion object was instantiated from a .otf binary and a .ttf file if the FontVersion object was instantiated from a .ttf binary. By default the write is to the same file path that was

used for instantiation of the FontVersion object. This write path default can be modified by passing a new file path in the fontpath parameter.

Parameters **fontpath** – (string) optional file path to write out the font version string to a font binary

Returns None

Symbols

`_get_repo_commit()` (fontv.libfv.FontVersion method), 2
`_is_development_substring()` (fontv.libfv.FontVersion method), 2
`_is_release_substring()` (fontv.libfv.FontVersion method), 2
`_is_state_substring_return_state_match()` (fontv.libfv.FontVersion method), 2
`_parse()` (fontv.libfv.FontVersion method), 2
`_parse_metadata()` (fontv.libfv.FontVersion method), 2
`_parse_state()` (fontv.libfv.FontVersion method), 2
`_parse_status()` (fontv.libfv.FontVersion method), 2
`_parse_version_substrings()` (fontv.libfv.FontVersion method), 3
`_read_version_string()` (fontv.libfv.FontVersion method), 3
`_set_state_status_substring()` (fontv.libfv.FontVersion method), 3

C

`clear_metadata()` (fontv.libfv.FontVersion method), 3

F

`FontVersion` (class in fontv.libfv), 1

G

`get_head_fontrevision_version_number()` (fontv.libfv.FontVersion method), 3
`get_metadata_list()` (fontv.libfv.FontVersion method), 3
`get_name_id5_version_string()` (fontv.libfv.FontVersion method), 3
`get_state_status_substring()` (fontv.libfv.FontVersion method), 3
`get_version_number_string()` (fontv.libfv.FontVersion method), 4
`get_version_number_tuple()` (fontv.libfv.FontVersion method), 4

S

`set_development_status()` (fontv.libfv.FontVersion method), 4
`set_release_status()` (fontv.libfv.FontVersion method), 4
`set_state_git_commit_sha1()` (fontv.libfv.FontVersion method), 4
`set_version_number()` (fontv.libfv.FontVersion method), 5
`set_version_string()` (fontv.libfv.FontVersion method), 5

W

`write_version_string()` (fontv.libfv.FontVersion method), 5