`!netrc` --- netrc file processing

- netrc Objects

netrc

Source code: `Lib/netrc.py`

The `~netrc.netrc` class parses and encapsulates the netrc file format used by the Unix `ftp` program and other FTP clients.

A `~netrc.netrc` instance or subclass instance encapsulates data from a netrc file. The initialization argument, if present, specifies the file to parse. If no argument is given, the file `.netrc` in the user's home directory -- as determined by `os.path.expanduser` -- will be read. Otherwise, a `FileNotFoundError` exception will be raised. Parse errors will raise `NetrcParseError` with diagnostic information including the file name, line number, and terminating token.

If no argument is specified on a POSIX system, the presence of passwords in the `.netrc` file will raise a `NetrcParseError` if the file ownership or permissions are insecure (owned by a user other than the user running the process, or accessible for read or write by any other user). This implements security behavior equivalent to that of ftp and other programs that use `.netrc`. Such security checks are not available on platforms that do not support `os.getuid`.

3.4 Added the POSIX permission check.

3.7 `os.path.expanduser` is used to find the location of the `.netrc` file when *file* is not passed as argument.

3.10 `~netrc.netrc` try UTF-8 encoding before using locale specific encoding. The entry in the netrc file no longer needs to contain all tokens. The missing tokens' value default to an empty string. All the tokens and their values now can contain arbitrary characters, like whitespace and non-ASCII characters. If the login name is anonymous, it won't trigger the security check.

NetrcParseError

Exception raised by the `~netrc.netrc` class when syntactical errors are encountered in source text. Instances of this exception provide three interesting attributes:

msg

Textual explanation of the error.

filename

The name of the source file.

lineno

The line number on which the error was found.

netrc Objects

A `~netrc.netrc` instance has the following methods:

netrc.authenticators(host)

Return a 3-tuple `(login, account, password)` of authenticators for *host*. If the netrc file did not contain an entry for the given host, return the tuple associated with the 'default' entry. If neither matching host nor default entry is available, return `None`.

netrc.__repr__()

Dump the class data as a string in the format of a netrc file. (This discards comments and may reorder the entries.)

Instances of `~netrc.netrc` have public instance variables:

netrc.hosts

Dictionary mapping host names to `(login, account, password)` tuples. The 'default' entry, if any, is represented as a pseudo-host by that name.

netrc.macros

Dictionary mapping macro names to string lists.

□□□ #2
□ □□□ □□□ 2026-10-31 17:15:38 UTC
□ □□□ □□□ 2026-10-31 17:15:40 UTC