19.10. uu — Encode and decode uuencode files

Source code: Lib/uu.py

This module encodes and decodes files in uuencode format, allowing arbitrary binary data to be transferred over ASCII-only connections. Wherever a file argument is expected, the methods accept a file-like object. For backwards compatibility, a string containing a pathname is also accepted, and the corresponding file will be opened for reading and writing; the pathname `'-'` is understood to mean the standard input or output. However, this interface is deprecated; it's better for the caller to open the file itself, and be sure that, when required, the mode is `'rb'` or `'wb'` on Windows.

This code was contributed by Lance Ellinghouse, and modified by Jack Jansen.

The uu module defines the following functions:

uu. **encode**(*in_file*, *out_file*, *name=None*, *mode=None*)

> Uuencode file *in_file* into file *out_file*. The uuencoded file will have the header specifying *name* and *mode* as the defaults for the results of decoding the file. The default defaults are taken from *in_file*, or `'-'` and 0o666 respectively.

uu. **decode**(*in_file*, *out_file=None*, *mode=None*, *quiet=False*)

> This call decodes uuencoded file *in_file* placing the result on file *out_file*. If *out_file* is a pathname, *mode* is used to set the permission bits if the file must be created. Defaults for *out_file* and *mode* are taken from the uuencode header. However, if the file specified in the header already exists, a uu.Error is raised.

> decode() may print a warning to standard error if the input was produced by an incorrect uuencoder and Python could recover from that error. Setting *quiet* to a true value silences this warning.

exception uu. **Error**

> Subclass of Exception, this can be raised by uu.decode() under various situations, such as described above, but also including a badly formatted header, or truncated input file.

> **See also:**
>
> **Module** binascii
>
> > Support module containing ASCII-to-binary and binary-to-ASCII conversions.