IEEE Standards Interpretation for IEEE Std 1003.1™-2001 IEEE Standard Standard for Information Technology -- Portable Operating System Interface (POSIX®)

Interpretation Request #109
Topic: man standard error **Relevant Sections:** XCU man

The 'pax' format states that extended header records have to be encoded in UTF-8. This in particular concerns the file names in the 'path' and 'linkpath' fields. Since file names are usually encoded as byte sequences on the host system, the description of these fields states that those sequences have to be converted to UTF-8 when being stored. This, however, leads to two major problems which make the 'pax' format effectively unusable for a general purpose archiver on traditional Unix implementations:

1. Traditional Unix implementations allow an application to pass arbitrary byte sequences to open() and related calls. They just handle the two bytes (not characters!) '/' and '\0' specially and otherwise accept anything. In particular, they do not require these byte sequences to have a character representation in any character encoding. Thus they also do not require these sequences to be in any relation to sequences representing characters in the current locale.

Moreover, many different locales may be used in parallel by different users on the same machine or even by one user in different terminals on the same machine. Even if the file names used for a single open() call are representable in the respective current locale, the Unix systems does not keep track of this relation anywhere. Thus if the super-user creates an archive of the whole file system, or if a user who has used multiple locales creates an archive of his entire data, there is no method for him to determine which character encoding has been used for a single file name.

It has sometimes been proposed as a workaround to create such an archive in a locale which assigns a separate character to every single byte, such as ISO-8859-1. This is not

a good workaround, however, as 1) it is against the purpose of UTF-8 conversion, since it effectively destroys the character representation of file names not in the locale used for archiving; 2) it forces the user to manually keep track of the locale used, such as on the tape label; 3) there is no guarantee that an equivalent locale is available on another system, including future revisions of the same implementation. (In fact, there are ISO-8859-1 locales which treat the range 0200 to 0237 as illegal.)

In effect, the 'pax' format is unable to hold a complete Unix file hierarchy in a sane way.

2. File names do not only occur in archive headers; the may also occur in file data stored inside the archive. An example would be a Makefile. But this may also effect files which mostly contain binary data, such as an executable or an office document. Such files cannot be subject to character conversion when the archive is created.

Now if an archive is created which e. g. contains an office document and some external image files with 8-bit file names to which links inside the office document point, the byte representation in the 'pax' archive headers differs from that in the file data within the archive. This will lead to broken links if the archive is extracted in another locale than the one it was created in. (The portability restrictions concerning locales mentioned above apply again here.)

In effect, the 'pax' format breaks links between the files stored.

Action:
Add fields to store file and link names as byte sequences, either replacing or supplementing the existing 'path' and 'linkpath' fields.

It might then be advisable to do the same for the 'uname' and 'gname' fields too.

Interpretation Response
The standards states the requirements for pax, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

Rationale for Interpretation
None.