
When and how is the dos2unix command used in Aleph?

- **Product:** Aleph
 - **Product Version:** 20, 21, 22, 23
 - **Relevant for Installation Type:** Dedicated-Direct, Direct, Local, Total Care
-

Question

When and how is the dos2unix command used in Aleph?

Answer

This command is used when plain text files in DOS/MAC format need to be converted to UNIX format.

For example, this is needed when a unique input file for an Aleph service is needed, an input file that cannot be easily created with Aleph's "retrieve records" and/or "print records" services, and is then uploaded to the server via ftp without specifying the file is in "ascii".

When a non-ascii input file is used, an error like the following may occur:

```
"[error] The first non-empty line of the input file has an invalid structure. Processing is stopped."
```

To convert a plain text file already uploaded to the server to UNIX without changing the file's name, do the following:

1. Login to the Aleph server as "aleph" user.

2. Enter the following command:

```
dos2unix <xxxxxxxxxxxx>
```

"<xxxxxxxxxxxx>" above refers to the uploaded file name.

3. The following line appears:

```
dos2unix: converting file <xxxxxxxxxxxx> to UNIX format ...
```

When the UNIX prompt appears, the file has finished converting and may be used as intended.

Additional Information

To change a file to ascii format before ftp'ing it:

1. Login to the ftp server.

2. Type "ascii" on the ftp command line prior to executing the ftp.

NOTE: The dos2unix command can be used for DOS/MAC file formats and also when the file is in BINARY mode.

Two common use cases where this is applicable:

1. Using an Aleph service that requires a plain-text input file.

For example, running p_manage_36 service ("Check Input File Against Database" in the Cataloging module) fails to match an expected a BIB record number.

When the input file for this service was created it looked like the following (e.g.):

```
000000001 0359 L $$aSFXHA110978966558810
```

The Aleph service is looking for an exact match to the filing_text in the z11 record (e.g.):

```
03 filing_text ....sfxha110978966558810
```

By uploading it in binary mode instead of ascii, an invisible "^M" character appears at the end of each line within the file, so the text is actually being read by the process as:

```
$$aSFXHA110978966558810^M
```

Use the dos2unix command for the input file, and then rerun the service.

2. A change was made to an *.xsl file (e.g. hold-request-slip.xsl), then accessed via util/i/6 ("Create Print Templates Package for the PC Client").

The new form fails to load to the PCs, creating circulation errors.

Looking at the file created by the util/i/6 process (./xxx01/form.eng/html.pck) for the lines related to the hold-request-slip.xsl file, they appear as follows (e.g.):

```
PKC_FILE_BEGIN hold-request-slip-00.xsl
<xsl:stylesheet^M
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">^M
<xsl:include href="funcs.xsl"/>^M
<xsl:template match="/">^M
<xsl:call-template name="header"/>^M
^M
<etc.>
```

When placing the file on the server, it was not specified to be an ascii file, and so these "^M" line feed characters appear at every line break.

In another case (where dos2unix didn't seem to be working): "The problem seemed to be the encoding, when I re-encoded the file and ftped it back to Aleph the procedures worked correctly."

• **Article last edited:** 02-Mar-2016