

HULFT改行コード変換のしくみ

HULFTには、改行コード(CRLF や LF)を明示的に指定する設定がありません。
 では、改行コードを変換しないのかというと、そうではなく各ホスト種に合わせた変換を行います。

1. 改行コードの種類

Windows系 CRLF CR(Carriage Return:行頭復帰、16進数 0D)とLF(Line Feed:改行、16進数 0A)を合わせたもの
 UNIX/Linux LF Line Feed:改行、16進数 0A
 詳細は、<http://www.esector.co.jp/product/hulft.html> の“#02改行コードとは”を参照

2. HULFTにおける改行コードに関する設定-配信側

詳細ホスト情報

配信管理情報

3. 変換前、変換後の改行コード

配信側と集信側のホスト種に依存し、下記の改行コード変換がおこなわれます。

# 配信側ホスト種と改行コード	→	集信側:Windows	集信側:UNIX/Linux	集信側:メインフレーム(汎用機)
1 WindowsのCRLF (0x0D0A)	→	変換無	CRLF (0x0D0A) → LF (0x0A) 変換	CRLF (0x0D0A) を削除
2 WindowsのLF (0x0A)	→	LF (0x0A) → CRLF (0x0D0A) 変換	変換無	LF (0x0A) を削除
3 UNIX/LinuxのLF (0x0A)	→	LF (0x0A) → CRLF (0x0D0A) 変換	変換無	LF (0x0A) を削除

4. HULFT使用時における改行コード変換に関するヒント

1) HULFT for Windows の改行コードの扱い

HULFT for Windows では LF (0x0A) も CRLF (0x0D0A) も両方 “改行コード” とみなします。

ただし、HULFT for Windows が集信側となる場合、受け取ったテキストファイルの改行コードとして、CRLF (0x0D0A) を採用します。

2) HULFT for UNIX/Linuxのファイルで “^M”が入ってしまう。

“^M”は、CR (0x0D) を意味します。

配信側のHULFT for Windowsの配信管理情報→転送タイプで誤って“TEXT”以外を指定した可能性があります。

また、utlbreakで展開されたファイルに“^M”がある場合、配信側のHULFT for Windowsでutljoin.exe実行時、“-b”を誤って指定した可能性があります。

3) Windowsなのに誤って改行コードがLFになった場合

[3. 変換前、変換後の改行コード]の表にあるように、HULFT for Windowsの配信時は、LFでも改行とみなします。

また、集信時はCRLFとするため、自ホスト→自ホストの転送を行うと、LF→CRLFの変換が可能です。

4) UNIX/Linux上で、改行コードの変換を行いたい場合

下記コマンドで、変換可能です。

○ CRLF→LFの変換コマンド

perl -pe 's/¥r¥n/¥n/' 入力_CRLFファイル > 出力_LFファイル

○ LF→CRLFの変換コマンド

perl -pe 's/¥n/¥r¥n/' 入力_LFファイル > 出力_CRLFファイル

5) 改行コードを追加/削除する

HULFTでは、Windows、UNIX/Linux上の改行コードを追加/削除するユーティリティ (utlIf) が提供されています。

参照)

HULFT7 Windows オペレーション・マニュアル

5.2.1 ファイルレコード編集

HULFT7 UNIX/Linux オペレーション・マニュアル

5.2.1 ファイルレコード編集