

1バイト文字2バイト文字が混在する日本語で、QuarkXPressのAppleScript上でテキストを扱うことは、難しいことを以前お話ししました(97年3月号)。そのため、テキスト処理の自動化については、特に触れることをしませんでした。今回、Perlと連携

Perlって何?

Perl(パール)について、ご存知な方も多いと思いますので、簡単に説明します。元来はUNIXで使用されていたプログラミング言語です。UNIXでは、ちょっとした文字列処理をCでプログラムを書かずに、sedやawkといった言語を用いた処理を行っていました。テキストファイル内にある、あるパターンの文字列をsedやawkを通すことで、別の文字列に置換することができます。Perlはその兄貴分と考えていいでしょう。単純な文字列置換はもちろん、複雑なプログラムも可能にしています。また、UNIXから移植され多くのOSで実行可能ですし、その配付も無料で行われています。最近では、インターネットのホームページでCGIを作成するのにPerlで行われるのが一般的になっていますので、この方面からPerlを知った方も多いのではないのでしょうか。

どうしてPerlなのか?

高速な動作! Macで使用する範囲では同じ作業でもawkより速く処理します。

大きなファイルでも楽々処理! 3,4MBのテキストファイルでも数分もかからずに(一般的な置換作業ならば)問題なく処理します。

もちろん、その配付が無料であり誰でも使用して構わないことは大きな理由の一つになります。

MacPerl & MacJPerl

Macintosh上で利用可能なのがMacPerlです。ただし、このままですとテキストが1バイトの欧文を前提としています。2バイトの日本語も処理できるようにするには、パッチを当てる必要があります。パッチを当てたPerlをJPerlと呼びます。MacPerlにパッチを当てたものがMacJPerlとなります。

MacJPerlを用意する

Perlのバージョンは最新では5となっています。ただし、JPerlではパッチが用意されていないので、MacJPerlはバージョン4を利用します。また、MacJPerlでは、PowerMac版では動作が安定しない部分がありますので、68K版を使用します。

CD-ROMには、以下のプログラムを収録しました。

- ・ MacPerl 4.1.8 (68K版)
- ・ MJP418 (J14) キット

する方法が分かりましたので、今回はその報告をしたいと思います。Perlは非常に強力なテキスト処理が可能ですから、テキスト内の文字列の一括置換など、様々なPerlプログラムと連携することで生産性がグッと上がることと思います。

・ MJP418 (J142) patch

MacPerlにMJP418 (J14) キットのバッチを当て、MacJPerl418 (J14)にし、MJP418 (J142) patchのバッチでMacJPerl418 (J142)にしてください。以下、記事ではMacPerlと記載しますが、MacJPerl418(J142)を指し、スクリプトの実行もMacJPerl418(J142)を前提とします。

QuarkXPressとの連携を考える

では、QuarkXPressとはどのように連携させるか考えてみます。AppleScriptを用いずに、手作業で行う場合は次のようになります。

- (1) テキストボックス内のテキストを「テキスト保存」する
- (2) 保存したファイルをMacPerlの入力ファイルとして、ある処理をさせる
- (3) 処理後の出力ファイルを、QuarkXPress側でテキストとして「テキスト取り込み」を行う



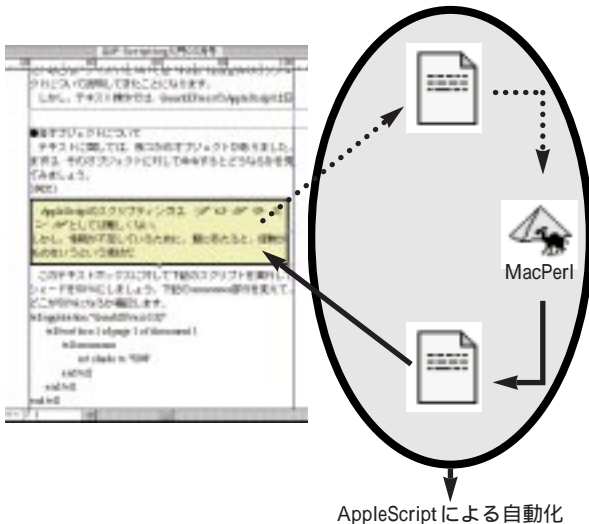
これをAppleScriptで自動化する場合を考えてみます。

- (1) 処理文字列を選択する
文字列を選択するか(「編集ツール」で文字列を反転させる)、かテキストボックスを選択状態にします。
- (2) 処理するMacPerlプログラムを選択する
処理するMacPerlプログラムによって、テキストをプレーンなテキストで処理するか、XpressTagとして処理するか選択できるようにします、
- (3) MacPerlによる処理を行う
(1)の選択した文字列をテキストファイルとしてテンポラリーファイルとして書きだし、MacPerlの入力ファイルとします。選択したMacPerlプログラムを実行します。その際、必要があれば、QuarkXPress側のフォント名称やフォントサイズを

Perlに引き渡します。

(4) 処理後のテキストファイルを読み込む

処理後のテキストファイルを、選択した範囲に読み込ませます



テキスト保存

テキストを選択して保存するにはどういったスクリプトを書いたらよいでしょうか？

(1) 選択したテキストボックス内の全テキスト

(a) プレーンなテキストとして保存する

```
tell application "QuarkXPress3.3J"
  tell document 1
    save story 1 of current box in "保存ファイル"
  end tell
end tell
```

(b) XPressTagとして保存する

```
tell application "QuarkXPress3.3J"
  tell document 1
    save story 1 of current box in "保存ファイル" as "TEXT"
  end tell
end tell
```

(2) 選択した範囲のテキスト

(a) プレーンなテキストとして保存する

```
tell application "QuarkXPress3.3J"
  tell document 1
    set tool mode to contents mode
    save contents of selection in "保存ファイル"
  end tell
end tell
```

(b) XPressTagとして保存する

```
tell application "QuarkXPress3.3J"
  tell document 1
    set tool mode to contents mode
    save story 1 of current box in "保存ファイル" as "TEXT"
  end tell
end tell
```

のようになります。XPressTagの場合は、as "TEXT" をsave文の末尾につけるだけで大丈夫です。

テキスト取り込み

逆にテキストを取り込む場合です。

(1) 選択したテキストボックス内の全テキスト

```
tell application "QuarkXPress3.3J"
  set import styles to true
  tell document 1
    set story 1 of current box to alias {"保存ファイル"}
  end tell
end tell
```

読み込む場所がstory 1 of current box となります。

(2) 選択した範囲のテキスト

```
tell application "QuarkXPress3.3J"
  set import styles to true
  tell document 1
    set contents of selection to alias {"保存ファイル"}
  end tell
end tell
```

読み込む場所がcontents of selection となります。

import stylesは「スタイルシートを含む」かどうかです。プレーンなテキストならば、この行は不要ですが、あっても取り込み内容に差はありません。しかし、XPressTagならば、絶対に必要になります。

MacPerlに値を引き渡す

テキストの保存と取り込み方法が分かったので残るは、MacPerlの部分です。問題なのは、

(1) 入力ファイルをどのように指定するか

(2) QuarkXPressでの設定値（例：フォント名称、フォントサイズなど）をMacPerlに引き渡す方法はあるのかの2点でした。

(1) 入力ファイルの指定方法

```
tell application "MacPerl"
  Do Script {"実行プログラム","保存ファイル"} mode Batch
end tell
```

このように、実行プログラム名称の後に、保存ファイル名称を記入します。もちろん、両者ともパスを含んだ形式で記入します。

(2) QuarkXPressでの設定値の引渡し方法

(1) のスクリプトにenvironmentパラメータを付与します

```
tell application "MacPerl"
  Do Script {perlScript, tempFileName} mode Batch ♪
    environment {"FONTNAME",FTNAME,"FONTSIZE",FTSIZE}
  end tell
```

environment {"FONTNAME",FTNAME,"FONTSIZE",FTSIZE}の意味は、MacPerlでの環境変数 FONTNAMEeに、FTNAMEの値を引渡

し、環境変数 FONTSIZE には、FTSIZE の値を引渡すことを表しています。

つまり、QuarkXPress 側の処理で、変数 FTNAME と FTSIZE に値をセットしておきます。そして、environment パラメータに引き渡します。MacPerl 側では、\$ENV{"FONTNAME"} によって変数 FTNAME の値を、\$ENV{"FTSIZE"} によって変数 FTSIZE の値を得ることが出来ます。

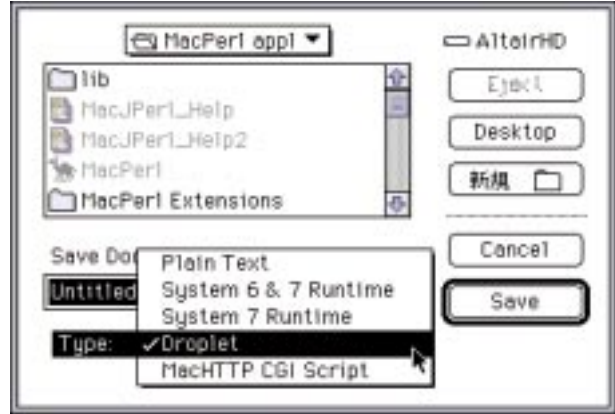
これによって、QuarkXPress での値を MacPerl に引き渡すことが出来るようになりました。

(なお、environment パラメータの値を利用するしないは MacPerl の実行スクリプトに依ります。MacPerl 側で利用しないケースで environment が記入されていても問題はありませぬ)

MacPerl プログラム

MacPerl プログラムをどのように書くかは、Perl 本が数多く出ていますので、それを参考して下さい。ただし、保存する方法

は、Droplet 形式を選んで下さい。Text 形式でも実行は可能ですが、問題が発生する場合があります。また、入力ファイルを上書きすることを前提としていますので、MacPerl でコマンドラインスイッチ -i オプションが必要になります。



```

- QuarkXPress & MacPerl Scripts

set tempFilePath to path to temporary items folder as text
set tempFileName to tempFilePath & "tempXPressText"
set PerlScript to choose file with prompt ♪
    "実行する Perl プログラムを選択して下さい" of type {"APPL"}
set textOrtag to button returned of ♪
    (display dialog "テキスト形式と XPressTags 形式、どちらで処理しましょうか?" buttons {"キャンセル", "テキスト", "XPressTags"})

--選択しているテキストボックスのテキストを保存

set selectChoice to button returned of (display dialog "処理範囲は?" buttons {"キャンセル", "ストーリー", "選択範囲"})
tell application "QuarkXPress3.3J"
    if selectChoice is "選択範囲" then
        tell document 1
            set tool mode to contents mode
            if contents of selection is "" then
                beep
                display dialog "テキストが選択されていません" buttons {"OK"} default button 1
                return
            end if
            set FTNAME to font of character 1 of selection of current box
            set FTSIZEN to round (((size of character 1 of selection of current box) as points) as real) * 1000) rounding to nearest
            set FTSIZE to FTSIZEN / 1000 as text
            if textOrtag is "テキスト" then
                save contents of selection in tempFileName
            else
                save contents of selection in tempFileName as "TEXT"
            end if
        end tell
    else
        tell document 1
            try
                get box type of current box
            on error
                beep
                display dialog "テキストボックスが選択されていません" buttons {"OK"} default button 1
                return
            end try
        end tell
    end if
end tell

```

```

end try
set FTNAME to font of character 1 of story 1 of current box
set FTSIZEN to round (((size of character 1 of selection of current box) as points) as real) * 1000) rounding to nearest
set FTSIZE to FTSIZEN / 1000 as text
if textOrtag is "テキスト" then
    save story 1 of current box in tempFileName
else
    save story 1 of current box in tempFileName as "TEXT"
end if
end tell

--Perl で処理

with timeout of 300 seconds
    tell application "MacPerl"
        activate
        Do Script {PerlScript, tempFileName} mode Batch ♪
            environment {"FONTNAME", FTNAME, "FTSIZE", FTSIZE}
        end tell
    end timeout

--処理後のファイルを選択しているテキストボックスに読み込む

tell application "QuarkXPress3.3J"
    activate
    set import styles to true
    if selectChoice is "選択範囲" then
        tell document 1
            set contents of selection to alias {tempFileName}
        end tell
    else
        tell document 1
            set story 1 of current box to alias {tempFileName}
        end tell
        do updates
    end if
end tell

```

QuarkXPress & MacPerl スクリプト

次の MacPerl プログラムをサンプルとして CD-ROM に収録しました。前ページのスクリプトを起動し、任意の MacPerl プログラムを選択して実行してみましょう。

1: 全角欧文->半角一括置換 (テキスト形式を選択)

全角の英数字を半角に置換します

```
#!/usr/local/bin/perl -pi.bak
tr/A-Z a-z /A-Za-z/;
tr/0-9 /0-9/;
```

この記事は、1997年8月7日PM5時32分に書いています。QuarkXPressとMacPerlを連携させているいろいろなTEXT処理が可能になりそうです。Y

↓ 処理後

この記事は、1997年8月7日PM5時32分に書いています。QuarkXPressとMacPerlを連携させているいろいろなTEXT処理が可能になりそうです。Y

2: 縦組中の文字回転 (テキスト形式を選択)

0~9, さらに2文字数字が連続した場合とA~Zの場合, 文字を回転させます。

プログラム内容は、CD-ROMを参照して下さい

この記事は、1997年8月7日PM5時32分に書いています。QuarkXPressとMacPerlを連携させているいろいろなTEXT処理が可能になりそうです。	↓ 処理後	この記事は、1997年8月7日PM5時32分に書いています。QuarkXPressとMacPerlを連携させているいろいろなTEXT処理が可能になりそうです。Y
---	-------	--

3: 総ルビ付与 (テキスト形式を選択)

総ルビを振ります。辞書RUBIS.DICはMacPerlプログラムと同じ階層において下さい。

プログラム内容は、CD-ROMを参照して下さい

単語が漢字で連続するものはルビが振れませんが、その他は愛する、愛でる、愛しいなど、漢字に続くひらがなから判断してルビを振ります。ただし、処理の時間は数分かかります。

↓ 処理後

単語が漢字で連続するものはルビが振れませんが、その他は愛する、愛でる、愛しいなど、漢字に続くひらがなから判断してルビを振ります。ただし、処理の時間は数分かかります。

このように、MacPerl プログラムをいろいろ作るだけで、様々なテキスト処理が可能になります。QuarkXPress側ではテキストを選択してからAppleScriptを実行するだけでテキストが入れ替わりますので、MacPerlを意識せずに処理できます。

最後に

このスクリプトは、NiftyServe上で青山和光氏の助言をいただいて、ここまでに至りました。改めて御礼申し上げます。また、「文字情報と印刷・DTPフォーラム」(FPRINT)の15番会議室

「せどおうくばー (SED, AWK, PERL)」発言番号1343番以降でいろいろなMacPerlプログラムや、このAppleScriptの改善が行われています。収録したプログラムは開発中のものが多いです(8/7現在)。本番の使用に当たっては思わぬ障害が出るかもしれません。最新版の使用をお勧めしますので、是非、FPRINTの15番会議室を覗いてみてください。

収録内容

スクリプト

- QuarkXPress & MacPerl Scripts

MacPerl 関連

- MacPerl 4.1.8 (68K版)
- MJP418 (J14) キット
- MJP418 (J142) patch
- MacPerl サンプルプログラム
 - 縦中横文字
 - 全角欧文->半角一括置換
 - 総ルビ付与
 - xtag.pl ver0.13 (ライブラリ)
 - ビプロフォントの一括指定 ver0.11
 - 英数字の全角半角統一 ver0.11
 - 縦組中の半角数字回転(桁無制限) ver0.11
 - 単独の英大文字を全角に
 - 約物の一括つめ ver0.12
 - 行頭約物前に半角アキ