

# J601におけるボックス化された日本語表示

## On the Displaystyle of Boxed Japanese Phrases in J601

慶応義塾大学理工学部  
竹内寿一郎

### 1. はじめに

Jで日本語が使用できることが明らかになったのは、J60シリーズの直後にリリースされた初めてのウィンドウズ版、JFWであった。このバージョンはその昔、USBなどが広く知られていなかった頃、丁度フロッピー1枚に収まる大きさで(現在のJはおよそ4.7メガ)、非常に便利であった。当時、JのデフォルトのフォントはISIJで、これでないとは綺麗にボックス表示がなされなかった。日本語についての発見は、いつだったか私の知る限りでは研究室の学生の一人が、「先生、フォントにFixedSysを使うと日本語が表示できますよ。」と言ったのに始まる。試してみると本当に日本語が表示されたので便利に使用してきたのである。

ところがこれはJ402迄で、J403以降はフォントにFixedSysを使用しても日本語が表示出来ないようになってしまった。しかしながらそれは文字化けしているためijxの画面にうまく表示されていないだけで、それをテキストファイルにコピーすると、きちんと元の日本語に戻り、多少の慣れを必要とするが日本語をデータとして使用することが出来た。日本語は2バイトを用いて表されたので、表示された文をボックスで囲んだとき、FixedSysでは文字化けした(枠)を9!:'7'+++++++---'を使うことにより、曲りなりにもテキストファイル上でうまく表すようにしていた。しかし画面上で日本語を表示しなければならない、会社のシステム的设计などにJを使うことが出来なかったのは当然であった。

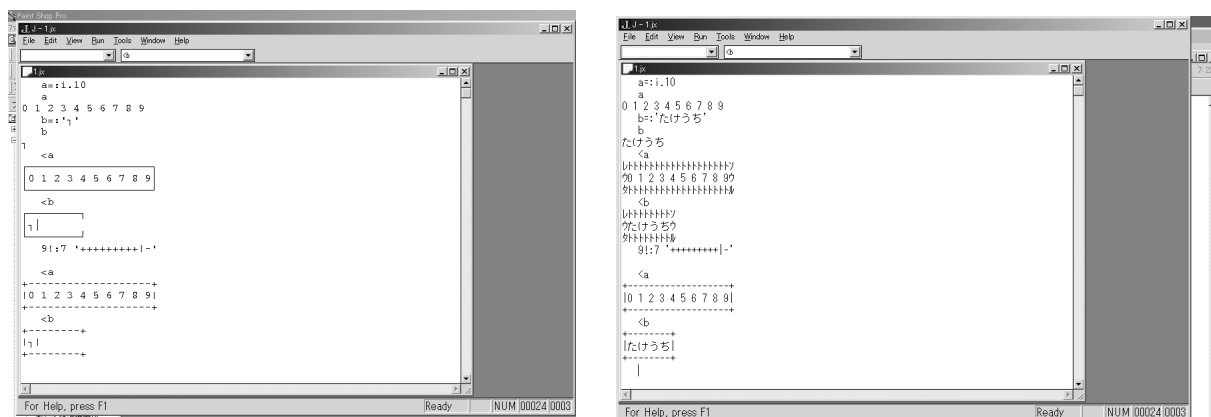


図1. 初めてのウィンドウズJFWのときの表示  
左がフォントISIJ、右がフォントFixedSysを使用した例である

今回、J601が使用出来るようになり、デフォルトフォントはCourier Newで、日本語を試して見ることにした。確かにFixedSysを用いていたときのように快適に日本語が使えて喜んでいたのだが、getexcelのcharinでエクセルファイルの中の漢字を読み込むと、ボックス化された文字は枠がやたらに長くなってしまったことが分かった。また、ijx画面でもボックス化された文字も同じく枠が長くなってしまっていた。そこで本稿ではijx画面からテキスト

ファイルにコピーしたとき、ボックス化された日本語の枠が最適な長さになるような方法を考えることにした。

## 2. 各バージョンでのボックス化された日本語表示の例

前ページにあるように最初のウィンドウズバージョン JFW では、漢字データは2バイトコードを用いて表していて、ISIJ フォント表示でも見られるように、単にフォントによって表示の仕方が変わっていることが分かる。当時コメント (NB.) に日本語を多用したのでフォントは FixedSys を使い、ボックスの文字化けを避けるため、9!:'+++++++'を実行していたのである。

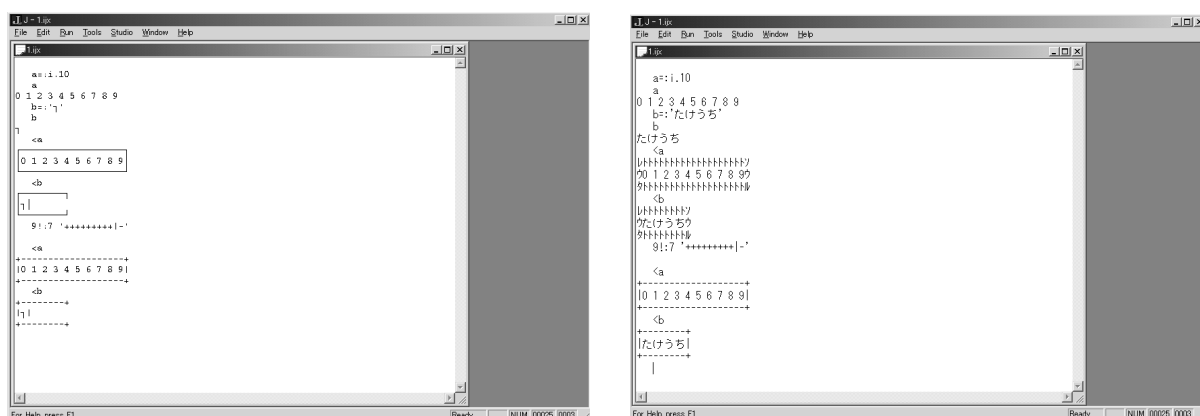


図2. J402バージョンでの表示で図1全く同じである  
左がフォント ISIJ、右がフォント FixedSys を使用した例である

J402 までは日本語に関しては全く変更がなかったようであるが、J403 から事情ががらりと変わってしまった。

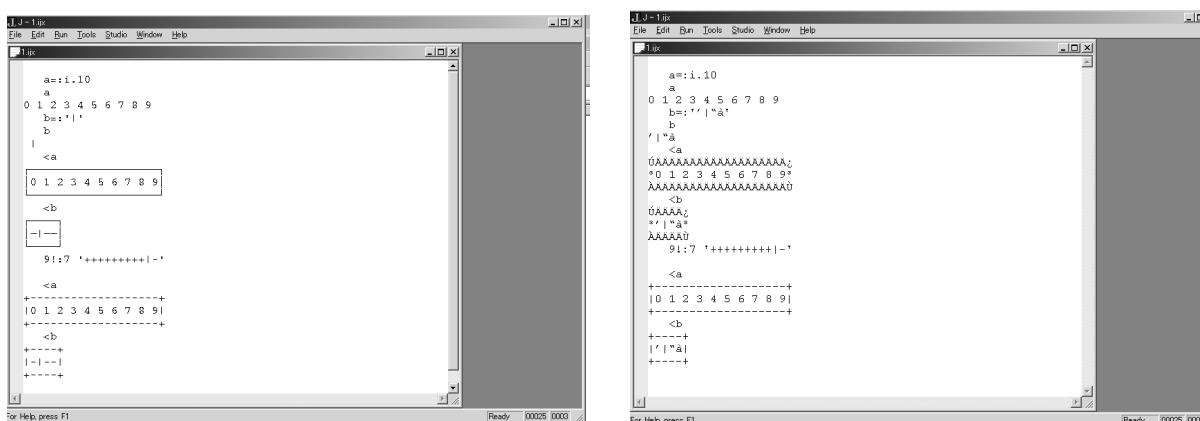


図3. J403バージョンでの表示はこれまでと異なっている  
左がフォント ISIJ、右がフォント FixedSys を使用した例である

J403 になると ISIJ では、漢字は文字化けするものの文字コードの関係からか、ボックスしたときの枠は変な文字列になっていない。FixedSys では残念ながら、バージョン J403 から日本語表示が出来なくなってしまったが、慣れれば文字コードや文字数から若干漢字など

が読み取ることが出来るが、これは至難の業である。ともかく日本語表示も、ボックス表示もメチャメチャである。つまり、J403からはFixedSysは使えないということになった。

そこで J601 直前の J503 での日本語環境を見て見る。

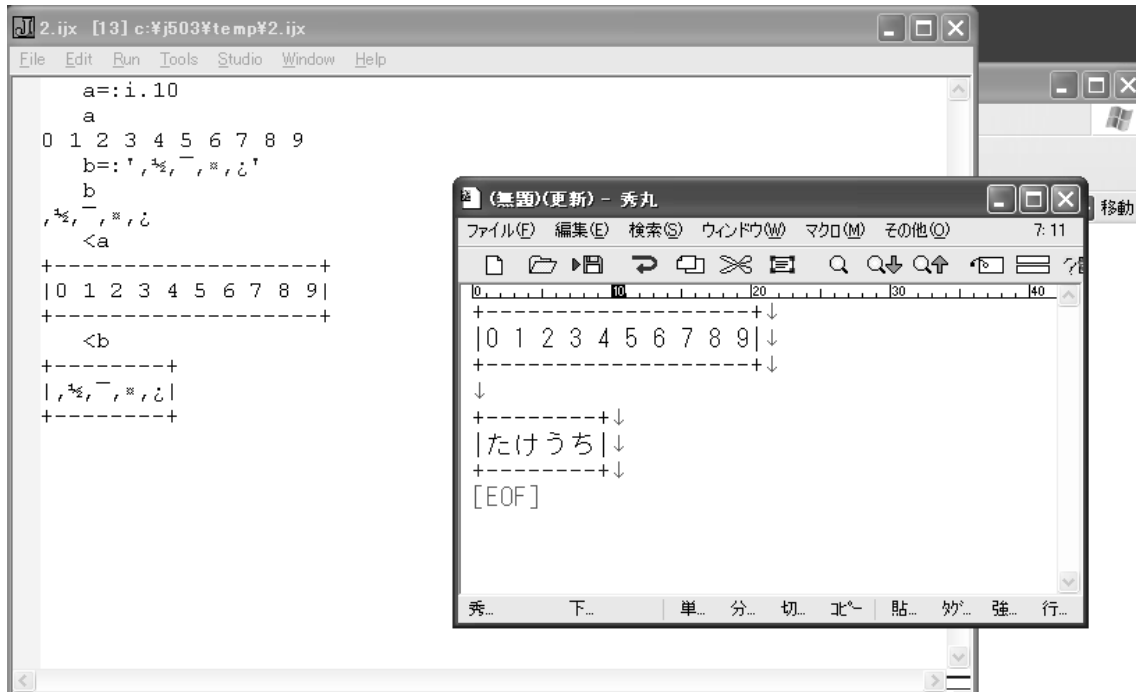


図 4. J503 バージョンでの表示、フォントは Courier New である  
日本語は文字化けしているが、ボックスはきちんとしている

ここでの例ではフォントが Courier New で、ボックス仕様が、9!:'+++++++'が実行されたかのようになっていて、ボックス記号が図 1、図 2 の FixedSys みたいに文字化けしていない。しかしながら日本語が 2 バイト表示であるから、それをテキストファイルにコピーしても、図 4 のようにしっかり写されていることが分かる。その点は便利ではあるがやはり画面に日本語が出ないという不満が残り、何とも不便である。

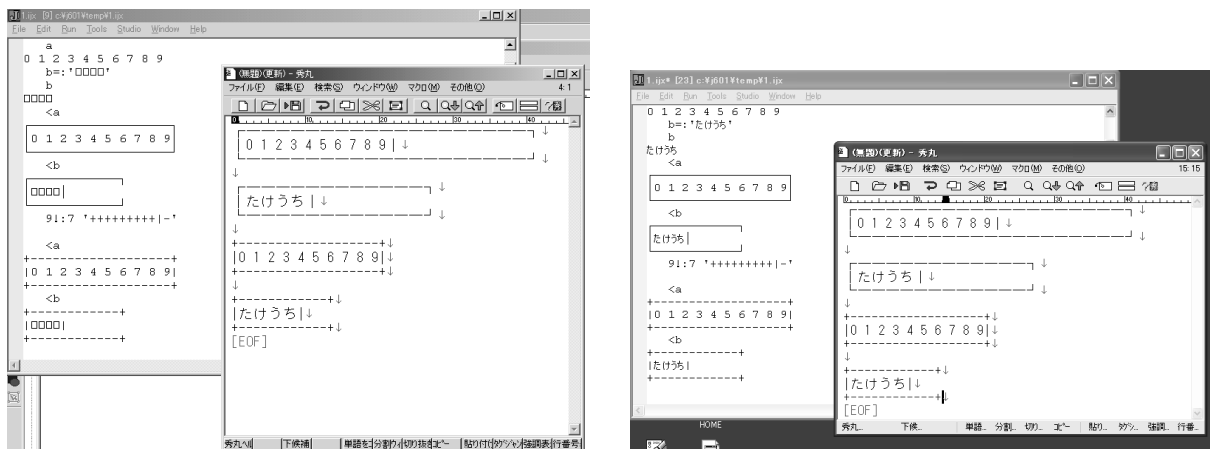


図 5. J601 で日本語が使えるのは Windows xp でのお話  
左が Windows2000、右が Windows xp、フォントは Courier New である

そこで最新の J601 を使用してみた。まず、驚いたことに Windows2000 の J601 では日本語が表示されないということである。さらに今度は Windows xp で J601 を詳細に使ってみると、確かに日本語が扱えて便利ではあるが、ボックスで囲むといろいろ不便なことが出てくる。その例が図5に載せてある。日本語どころか半角文字も含めてボックスで囲んだ ijx 画面をコピーすると、大半は枠の長さが長くなってしまっているのである。日本語が表示可能に成ったと喜んでいたら残念なことに、これまで安心して使っていた半角文字にまで悪い影響が出て来てしまったのである。

以下は試みにフォント ISIJ を用いて Windows2000 と Windows xp で J601 を使ってみたが、やはり枠線が長くなり、ijx 画面で日本語も表示できず不便であった。

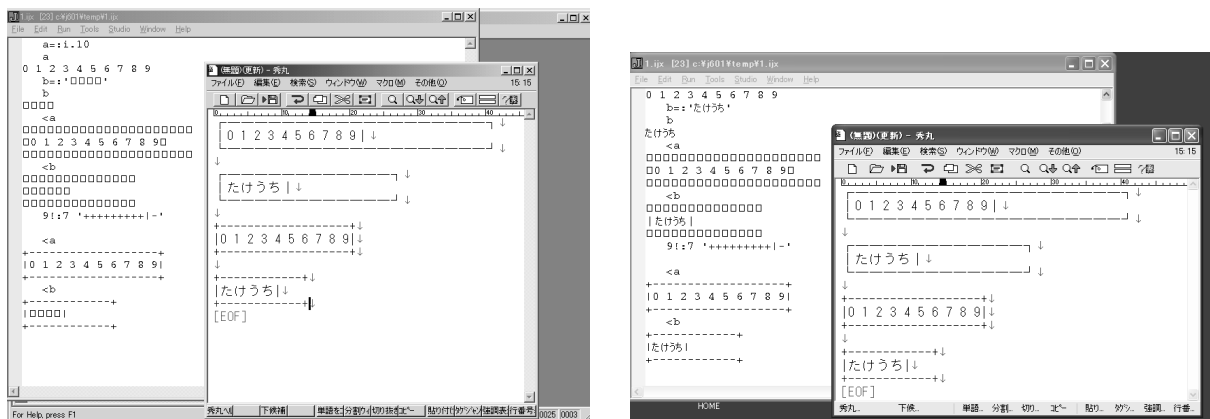


図6. フォント ISIJ で J601 を同様に試して見た  
左が Windows2000、右が Windows xp である

### 3. J601 での日本語表示

J601 での日本語表示の不便さの原因を調べてみたのが次の図である。

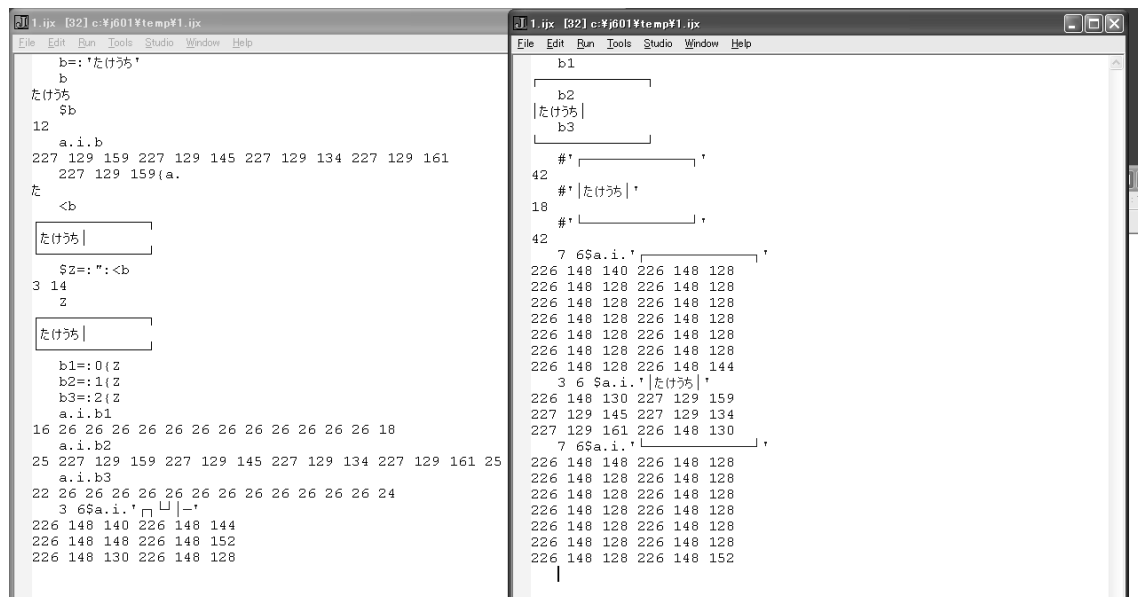


図7. J601 バージョンでのボックス表示、フォントは Courier New である  
右図は左図の続きである。基本的には全ての文字が3バイトからなっている

ボックス化した文字列をカウントすると、左図中ほどより、囲いの部分は1バイト、すなわち16,18,22,24の4隅、縦棒25、横棒26の合計6種類からなる。左図下で4隅と縦横を調べると3バイト・6種類の文字からなることが分かる。つまり枠の文字コードは2バイトで、枠の漢字コードは3バイトであることが分かる。枠の文字数は1バイト2単位(テキストでは1バイトが2単位、つまり2コマ)を使い、表示内容のバイト数に応じて枠がつくられている。すなわち、「たけうち」は4文字であるから3バイト×4文字=12バイト。これが横の長さで、始めの縦棒と終わりの横棒の一つずつが加わり合計14バイト、1バイトが2単位だから28単位使うので、枠がとてつもなく長ったらしくなるのである。

#### 4. J601でボックス表示をどう適正化するか

ボックス表示をテキストファイルにコピーしたとき、枠の文字はすべて全角、すなわち1バイト2単位で出来ているので、囲まれる内容に半角文字が含まれていると、綺麗な囲みを作ることはできない。そこで、ここでは全ての文字列が日本語で書かれている(全角文字から成っている)ことを仮定する。そのスクリプトが以下のものである。

NB. ボックスで囲まれた日本語をテキストファイルに最適表示する関数

NB. 使い方

NB. Kbox " :<' 使い方の例題'

NB. 

NB. | 使い方の例題 |

NB. 

Kbox=: 3 : 0

mm=.>:3%~2-~zz=.#0{yy[nn=.#yy=.y NB. mmが求める長さ

if. '┌'-:{.0{y do.goto\_ERR. end.

print X=. '┌',(-mm){.X=.{.y

print }:.y

print Y=. '└',(-mm){.Y=.{.y

goto\_owari.

label\_ERR.

print 'ERORR'

label\_owari.

'END'


)

【例題】

Kbox " :<' 使い方の例題'



| 使い方の例題 |



END

Kbox " :<' たけうち'



| たけうち |  
└──────────┘

END

Kbox " :<' 使い方の例題,'

| 使い方の例題 |  
└──────────┘

END

Kbox " :<' たけうち,'

| たけうち |  
└──────────┘

END

Kbox " :<' 竹内,'

| 竹内 |  
└────────┘

END

Kbox " :<' J 研究会で発表した内容です。,'

| J 研究会で発表した内容です。 |  
└──────────────────────────┘

END