Writing flat files with implicit substructure and Asian data from Unicode systems

June 7th 2006

1 Problem and solution

With the introduction of Unicode the correct handling of code pages in the file interface becomes important.

Whereas in non-Unicode systems a “memcopy” approach in the outside communication was sufficient now code page conversions between the system format (UTF-16 little endian or UTF-16 big endian) and the desired outside format is necessary.

While this is common for all kind of desired outside formats there are special things to consider, if the outside code page is one of the non-Unicode multibyte code pages (simplified and traditional Chinese, Japanese and Korean), as the number of character units occupied to represent a character is changing while converting from UTF-16 to an old multibyte format:

‘한’	Number of character units in the application server	Number of character units in the multibyte file (KSC5601 format)
Non-Unicode application server	2	2
Unicode application server	1	2

This length change is in general no problem, but if the data to be written into a flat file has an implicit substructure, the intended layout in the file will not be as expected if no special measures are taken to keep the layout compatible.

As an example, consider the following structure:

Table 1 Definition of a structure SAM:

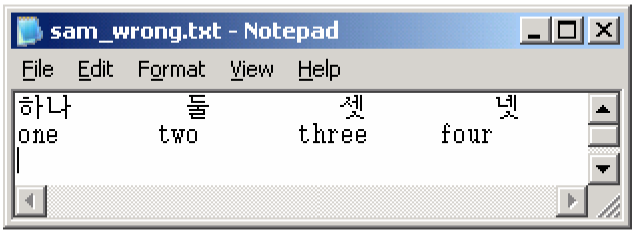
Field	Data type	Length
F1	CHAR	10
F2	CHAR	10
F3	CHAR	10
F4	CHAR	10

Let an internal table contain the following entries:

	F1	F2	F3	F4
Row 1	one	Two	three	four
Row2	하나	둘	셋	넷

If you output the table content field by field everything is fine, but when you start to write everything as one single line with implicit substructure you will see the following result when running the report sam_file_wrong in a Unicode system and opening the resulting file with notepad using a Korean font:

```
report sam_file_wrong.

…

data: l_sam_line(40) type c,
      l_sam like table of l_sam_line.

l_sam_line+0(10)  = '하나'.
l_sam_line+10(10) = '둘'.
l_sam_line+20(10) = '셋'.
l_sam_line+30(10) = '넷'.
append l_sam_line to l_sam.

l_sam_line+0(10) = 'one'.
l_sam_line+10(10) = 'two'.
l_sam_line+20(10) = 'three'.
l_sam_line+30(10) = 'four'.
append l_sam_line to l_sam.


call function 'GUI_DOWNLOAD'
  EXPORTING
    filename                = l_full_dataset_name
    filetype                = 'ASC'
    write_field_separator   = ' '
    codepage                = '8500'
  TABLES
    data_tab                = l_sam
  EXCEPTIONS
    file_write_error        = 1
….
```



```
하나          둘          셋          넷
one         two         three       four
```

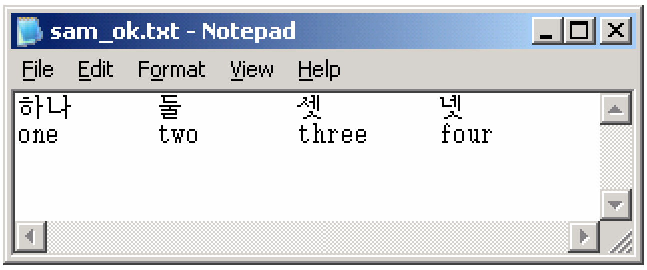
Offset length access to the implicit subfields does not work any more.

The solution is in general to write the data in a format where you have an clear distinction between content and layout (as XML for example). Nevertheless if the container approach must not be changed you can prepare an outside format with intended alignment using the class CL_NLS_STRUC_CONTAINER as follows:

```
report sam_file_ok.

…
data: l_sam_line(40) type c,
      l_sam like table of l_sam_line,
      begin of l_sam_struc,
          f1(10) type c,
          f2(10) type c,
          f3(10) type c,
          f4(10) type c,
      end of l_sam_struc.

data: l_struc_cont type ref to cl_nls_struc_container,
      l_cptab      type nls_langu_cp_tab,
      l_cptab_wa   type nls_langu_cp_line,
      l_cont       type string.


l_sam_line+0(10) = '하나'.
l_sam_line+10(10) = '둘'.
```

```
l_sam_line+20(10) = '셋'.
l_sam_line+30(10) = '넷'.
append l_sam_line to l_sam.

l_sam_line+0(10) = 'one'.
l_sam_line+10(10) = 'two'.
l_sam_line+20(10) = 'three'.
l_sam_line+30(10) = 'four'.
append l_sam_line to l_sam.



* adapt lengths:
" In principle you should have used an XML approach,
" but if you insist in sending one line with implicit substructure
" you have to adjust the layout.
clear l_cptab.
l_cptab_wa-langu = '3'. l_cptab_wa-codepage = '8500'. append l_cptab_wa to l_cptab.

l_struc_cont = cl_nls_struc_container=>create( l_cptab ).
loop at l_sam into l_sam_line.
  l_sam_struc = l_sam_line.
  try.
      call method l_struc_cont->struc_to_cont
        EXPORTING
          struc = l_sam_struc
          langu = '3'    "3 is the internal language value for Korean
        IMPORTING
          cont = l_sam_line.
    catch cx_sy_conversion_codepage .
    catch cx_parameter_invalid_range .
  endtry.
  modify l_sam from l_sam_line.
endloop.

call function 'GUI_DOWNLOAD'
  EXPORTING
    filename                = l_full_dataset_name
    filetype                = 'ASC'
    write_field_separator   = ' '
    codepage                = '8500'
  TABLES
    data_tab                = l_sam
  EXCEPTIONS
    file_write_error        = 1
….
```

Now the alignment is as intended and you can process the file with your old offset length values to access the subfields:



```
하나          둘          셋          넷
one         two         three       four
```