

# Double-Byte Character Support

In most East Asian languages, language-specific characters in code page strings (that is, Natural format A) are represented by 2 bytes (the so-called double-byte characters) and ASCII characters (EBCDIC on mainframes) are represented by 1 byte. Thus, a code pages string consists of characters with different lengths: some have 1 byte and others have 2 bytes.

Natural provides a basic support for double-byte characters. On Windows, this support is activated when both the Natural default code page and the Windows system code page are defined as double-byte code pages. If Natural does not define a specific code page, it is sufficient when a double-byte Windows system code page has been defined. On UNIX and OpenVMS, the support for double-byte characters is activated when the Natural default code page is a double-byte code page. On mainframes, the profile parameter CP must be set to an EBCDIC MBCS code page, for example IBM-942.

When double-byte character support is enabled, Natural assures for all string manipulations that a double-byte character is treated as a unit. This is essential for keeping the meaning of a string.

If a single leading or trailing byte of a double-byte character is left over after the manipulation of a variable of format A (for example, after extracting a substring with the SUBSTRING option), this byte is replaced with a blank character.

For the example below, the code page Shift\_JIS is selected. Variable #A contains a string which consists of four characters. The first and last character is the double-byte character "FULL WIDTH LATIN SMALL LETTER B" which is represented in code page Shift\_JIS by the byte sequence H' 8282'. The second and third character is the single byte character "LATIN SMALL LETTER A" which is represented by one byte H' 61'. Thus, the hexadecimal representation of the full string is H' 828261618282'.

```
DEFINE DATA LOCAL
  1  #A  (A10)
END-DEFINE

#A := ' b aa b '

WRITE #A #A (EM=H(6))
EXAMINE #A FOR PATTERN ' B ' REPLACE 'a'
WRITE #A #A (EM=H(6))

END
```

Without double-byte character support the output of the above program is as follows:

```
Page          1                      07-02-07    17:22:09

 b aa b      828261618282
 B a b      826161828220
```

This is the result of not having treated the character " b " (H' 8282' in code page Shift\_JIS) as one unit. The trailing byte of this character and the following character "a" (H' 61') are falsely interpreted as the double-byte character " B " (H' 8261' in code page Shift\_JIS).

With double-byte character support, the output of the program is as expected:

Page 1

07-02-07 17:22:09

b aa b 828261618282  
b aa b 828261618282