Bidirectional Language Support

Some languages, for example Arabic and Hebrew, are written from right-to-left (RTL), whereas the majority of the languages, for example English and German, are written from left-to-right (LTR). Text which contains both left-to-right and right-to-left characters is called bidirectional text.

Natural provides a basic support for bidirectional languages. On Windows, this support is activated when both the Natural default code page and the Windows system code page are defined as bidirectional code pages. If Natural does not define a specific code page, it is sufficient when a bidirectional Windows system code page has been defined. On UNIX and OpenVMS, the support for bidirectional languages is activated when the Natural default code page is a bidirectional code page.

The output of Natural programs can be controlled using the profile parameter PM, the terminal command %V, and the session parameter PM.

On UNIX and OpenVMS, the profile parameter DO (Display Order) is additionally used to support applications that have been originally written for terminals which support inverse (right-to-left) print mode, but no bidirectional data. These applications create the display order of bidirectional data in the application code. With the parameter DO, these applications are enabled to run compatibly also with I/O devices that support bidirectional data. This is for instance the case if an application runs in a browser with the Natural Web I/O Interface.

The profile parameter PM defines the default screen direction. When PM is set to R (reset), the default screen direction is left-to-right. When PM is set to I (inverse), the default screen direction is right-to-left. All non-alphanumeric fields and system variables are automatically inverted by Natural so that they are displayed correctly from right-to-left if the screen direction is right-to-left. PF key lines (UNIX and OpenVMS) are not inverted; they are always shown from left-to-right.

The terminal command %V can be used to change the screen direction. If the screen direction is right-to-left, the layout of the current window is mirrored, which means that the origin of all window components or fields is the upper right corner. The screen direction is changed to right-to-left using %VON and is reverted to left-to-right using %VOFF.

The session parameter PM reverses the direction of a field. The effect of "reversing the direction of a field" depends on the statement in which the PM parameter is used and the platform. If the PM parameter is used in a MOVE statement, the content of the field is simply reversed (that is, the first character will become the last character, and so on); the result does not depend on the characters of the field. Trailing blanks are removed before the field is reversed.

For example, the following program

```
DEFINE DATA LOCAL
1  TEST1  (A10)
1  TEST2  (A10)
END-DEFINE
TEST1 := 'program'

MOVE TEST1 (PM=I) TO TEST2
INPUT TEST1 (AD=O) TEST2 (AD=O)

END
```

produces the following output:

```
TEST1 program    TEST2 margorp
```

where "margorp" is the reversed version of "program".

When the PM parameter is used for IO statements such as INPUT or DISPLAY, its effect is even more complex. In this case, the field direction is based on the screen direction:

- If the screen direction is left-to-right and PM=I is applied to a field, the field direction changes to right-to-left.
- If the screen direction is right-to-left and PM=I is applied to a field, the field direction changes to left-to-right.

On Windows and browser terminals (Natural Web I/O Interface), "reversing the field direction" does not mean that the characters of the field are simply reversed. Instead, the complex bidirectional algorithm is applied (for more information, see the Microsoft Windows documentation). On character-oriented terminals, however, the characters of a field are not resorted; they are simply reversed.

In the following example, the characters assigned to the variable TEST have been entered in the following sequence:

```
a b c  ש נ ב  1 2 3
```

The following is an example program for Windows. The characters of the constant are already resorted when entering them in the program editor.

```
DEFINE DATA LOCAL
1  TEST  (A20)
END-DEFINE
TEST := 'abc 123 בנש '

SET CONTROL 'voff'

INPUT TEST (AD=O)  /
      TEST (AD=O PM=I)

SET CONTROL 'von'

INPUT TEST (AD=O)  /
      TEST (AD=O PM=I)
END
```

This program produces the following two screens on Windows:

```
TEST abc 123 בנש
TEST          123 בנש   abc
```

and

```
                                        123 בנש   abc TEST
                                  abc 123 בנש         TEST
```

The following is an example program for UNIX and OpenVMS. If the characters are entered in the sequence as described above, the program is displayed in the following way, because the characters are simply displayed in the keying sequence.

```
DEFINE DATA LOCAL
1  TEST  (A20)
END-DEFINE
TEST := 'abc שלום  123'

SET CONTROL 'voff'

INPUT TEST (AD=O)  /
      TEST (AD=O PM=I)

SET CONTROL 'von'

INPUT TEST (AD=O)  /
      TEST (AD=O PM=I)
END
```

On UNIX and OpenVMS, this program produces the following two screens:

```
TEST abc שלום  123
TEST          321 ולש   cba
```

and

```
                                        321 ולש   cba TSET
                                  abc שלום  123           TSET
```

On Windows, UNIX and OpenVMS, the map editor simplifies the handling of maps with bidirectional fields by offering the **Reverse Map** command. This command changes the display direction of the current map. The position of the fields is not changed; only the view is changed. On Windows, this command applies only to the current map. On UNIX and OpenVMS, a flag is set so that all following maps are displayed reversed; a following **Reverse Map** command will restore the original situation.

On Windows, the output of dialogs can be controlled in a similar way: both the dialog itself and most of the dialog controls offer an RTL attribute. If the RTL attribute of the dialog is checked, the screen direction of the dialog is right-to-left. If the RTL attribute of other controls is checked, the direction of these controls is right-to-left.

The profile parameter PM defines the default setting of the RTL attribute for new dialogs. When PM is set to R (reset), the RTL attribute is unchecked by default. When PM is set to I (inverse), the RTL attribute is checked by default. The default setting of the RTL attribute for newly created controls of a dialog is derived from the RTL attribute setting of the dialog.

If the RTL attribute of a dialog is changed when the dialog already contains controls, a dialog appears asking whether the RTL attributes of the controls should also be changed.

When working with bidirectional languages on Windows, "GUI" is the preferred print method. With the print method "GUI", the printed page will show the same layout as the window displayed on the screen. The sorting of the field characters is identical. If the print method "TTY" is used, the printed layout will most probably differ from the layout of the screen window because the field characters are printed in logical sequence. For fields with right-to-left direction, all characters are simply reversed (that is, the first character will become the last character, and so on).