

# How To Manipulate Dialog Elements

To manipulate dialog elements, Natural provides you with handle attribute operands. You use handle attribute operands wherever an operand may be specified in a Natural statement. This is the most important programming technique in event-handler code.

Important: You must have defined a handle.

**Note:** ActiveX controls are manipulated in a slightly different way than the standard way described below. This is described in Working with ActiveX Controls.

Handle attribute operands may be specified as follows:

```
handle.name - attribute.name [(index-specification)]
```

The *handle-name* is the handle of the *dialog-element-type* as defined in the HANDLE definition of the DEFINE DATA statement.

The *attribute-name* is the name of an attribute which has to be valid for the *dialog-element-type* of the handle.

**Examples:**

```

1 #PB-1 HANDLE OF PUSHBUTTON /* #PB-1 is a handle-name of the
                             /* dialog-element-type PUSHBUTTON
RESET #PB-1.STRING...      /* #PB-1.STRING is the handle attribute operand
                             /* where STRING is a valid attribute-name of the
                             /* dialog-element-type PUSHBUTTON

1 #RB-1(1:5) HANDLE OF RADIOBUTTON /* #RB-1 is an array of five RADIOBUTTONs
IF #RB-1.CHECKED(3) = CHECKED /* If the third radio-button control is
THEN... /* checked ...

```

**Querying, Setting and Modifying Attribute Values**

In most applications, it will be necessary

- to set an attribute value before creating the dialog element,
- to modify the value after creating the dialog element, and
- to query an attribute value.

In some cases, it may be necessary to modify and query some attributes during processing, for example to query the checked/not checked state of a radio-button control or to disable (= modify) a menu item.

You can do that, for example, in the ASSIGN, MOVE or CALLNAT statements.

**Examples:**

```

1 #PB-1 HANDLE OF PUSHBUTTON /* #PB-1 is a handle-name of the
... /* dialog-element-type PUSHBUTTON
#PB-1.STRING:= 'MY BUTTON' /* Set or modify the value of the STRING
                             /* attribute to 'MY BUTTON'
#TEXT:= #PB-1.STRING /* Query the value of the STRING attribute
                             /* and assign the value to #TEXT
CALLNAT 'SUBPGM1' #PB-1.STRING /* Query the value of the STRING attribute
                             /* and pass it on to the subprogram

```

When you use the *handle-name* variable only on the left side of the statements, as in the first of the three examples above, the attribute value is set or modified, that is, it is assigned the value of the specified *operand*.

When you use the *handle-name* variable on the right side of the statements, as in the second example, the attribute value is queried, that is, the value is assigned to the *operand*.

Once a handle has been defined (either explicitly in specified Natural code, or implicitly with the Dialog Editor), it can be used with most Natural statements. However, only a specific set of attributes can be queried, set or modified for a particular dialog element. To find out which values an attribute can have, see the chapter Attributes in the Dialog Component Reference documentation.

Although an exact data type is specified for the values of most attributes, it is sufficient to supply move-compatible values to a handle attribute operand. The rules are the same as those for Natural variables.

**Restrictions**

Handle attribute operands must not be used in the following statements:

AT BREAK, FIND, HISTOGRAM, INPUT, READ, READ WORK FILE.

User-defined variables can be used instead.

## **Numeric/Alphanumeric Assignment**

If you assign numeric operands to alphanumeric attributes, the values of these attributes will be in a non-displayable format. The Natural arithmetic assignment rules apply.

If you need a displayable format, you can use MOVE EDITED.

**Examples:**

```
#PB-1.STRING:= -12.34 /* Non-displayable format
MOVE EDITED #I4 (EM = -Z(9)9) TO #PB-1.STRING /* Displayable format
```

The following edit masks may be used for the various format/length definitions of numeric operands:

<b>Format/Length</b>	<b>Edit Mask</b>
I1	-ZZ9
I2	-Z(5)9
I4	-Z(9)9
<i>Nn.m/Pn.m</i>	<i>-Z(n).9(m)</i>

Back to Event-Driven Programming Techniques.