How To Define Dialog Elements

This document covers the following topics:

- Introduction
- HANDLE OF GUI
- NULL-HANDLE

Introduction

Dialog elements are uniquely identified by a handle. A handle is a binary value that is returned when a dialog element is created. A handle must be defined in a `DEFINE DATA` statement of the dialog.

You can define a handle

- by creating a dialog or a dialog element with the dialog editor; in this case, the handle definition is generated;
- by explicitly entering the definition in a global, local, or parameter data area of the dialog;
- by explicitly entering the definition in a subprogram or a subroutine.

Anmerkung:
Handles of ActiveX controls are defined in a slightly different way than the standard handle definition described below. This is described in *Working with ActiveX Controls*.

A handle is defined inside a `DEFINE DATA` statement in the following way:

level handle-name [(*array-defintion*)] **`HANDLE OF`** *dialog-element-type*

Handles may be defined on any *level*.

Handle-name is the name to be assigned to the handle; the naming conventions for user-defined variables apply.

Dialog-element-type is the type of dialog element. Its possible values are the values of the `TYPE` attribute. It may not be redefined and not be contained in a redefinition of a group.

Examples:

```
1 #SAVEAS-MENUITEM HANDLE OF MENUITEM
1 #OK-BUTTON (1:10) HANDLE OF PUSHBUTTON
```

When you have defined a handle, you can use the *handle-name* with handle attribute operands in those Natural statements where an operand may be specified. With handle attribute operands, you can, for example, dynamically query, set, or modify attribute values for the defined *dialog-element-type*. This is the most important programming technique in the dialog editor. For details, see the section *How To Manipulate Dialog Elements*.

If there is a dialog element handle of the same name in two different dialogs, the PARENT attribute ensures that Natural knows the difference between the two handles (two different PARENT values). Handles may be passed as parameters or may be assigned from one handle variable to another.

HANDLE OF GUI

In addition to the handle types referring to one dialog element, the generic handle type HANDLE OF GUI is available. In event-handler code, you can use HANDLE OF GUI to refer to the handle of any type of dialog element.

This can be useful, for example, if you are querying an attribute value in all dialog elements on one level: you go through the dialog elements one after the other; in the course of this query, it is not clear which type of dialog element is going to be queried next. Then a GUI handle makes it possible to query the next dialog element regardless of its type. This saves a lot of coding, because otherwise, you would have to query the attribute's value of each dialog element separately.

Example:

```
...
1 #CONTROL HANDLE OF GUI
...
#CONTROL := #DLG$WINDOW.FIRST-CHILD
REPEAT UNTIL #CONTROL = NULL-HANDLE
  ...
  #CONTROL := #CONTROL.SUCCESSOR
END-REPEAT
```

NULL-HANDLE

The HANDLE constant NULL-HANDLE may be used to query, set or modify a NULL value of a HANDLE. Such a NULL value means that the dialog element is nonexistent (even if it has been created explicitly).

Example:

```
DEFINE DATA PARAMETER
  1 #PUSH HANDLE OF PUSHBUTTON
END-DEFINE
...
IF #PUSH = NULL-HANDLE
...
```

The HANDLE constant NULL-HANDLE represents the NULL value of a HANDLE variable or of an attribute with format HANDLE. For handle variables, the value indicates that the expression *handle.attribute* refers to the global attribute list. For attributes, this value indicates that no value is currently set.