

# How Dialogs, Controls and Items Are Related Hierarchically

Dialogs and their dialog elements are organized hierarchically. Typically, the dialog window contains a number of controls. The controls are children of the window or of other controls which are capable of acting as containers. A control may contain a number of items. For example, a List Box Control may contain several list box items. The control is the parent of the items.

The dialogs themselves are also organized hierarchically. Every time the OPEN DIALOG statement is specified, the parent of the newly created dialog must be provided as a parameter. This parameter may be NULL-HANDLE or the handle of an existing dialog. If NULL-HANDLE is provided, the dialog belongs to the desktop rather than to any other dialog. This means that the dialog can be closed and minimized independently of any other dialog in the application. A dialog having an existing dialog as parent is closed or minimized when the parent dialog is closed or minimized.

The first dialog in an application plays a special role and is sometimes called the base dialog. When the base dialog is closed, all other dialogs in the application are also closed, whether they are children of the base dialog or not.

All children on one hierarchical level are sorted in the sequence of their creation. Each dialog element therefore always "knows" its parent, its predecessor and successor (on the same hierarchical level), and its first and last child (if present). You can retrieve this information by using the following attributes:

- PARENT
- PREDECESSOR
- SUCCESSOR
- FIRST-CHILD
- LAST-CHILD

These attributes contain handle values of dialog elements. If their value is NULL, the dialog element has no parent, successor, or child. The following example demonstrates how to go through all dialog elements of a dialog.

## Example 1:

```
1 #CONTROL HANDLE OF GUI

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

List box controls and list box items contain an additional attribute:

SELECTED-SUCCESSOR can be set for either the list box control itself or for any of its items. It points to the next selected item in a list box control. For the list box control itself, it points to the first selected item.

**Example 2:**

```
1 #ITEM HANDLE OF LISTBOXITEM  
  
#ITEM := #LISTBOX.SELECTED-SUCCESSOR  
REPEAT UNTIL #ITEM = NULL-HANDLE  
  ...  
  #ITEM := #ITEM.SELECTED-SUCCESSOR  
END-REPEAT
```

The above example is the query necessary to find all selected items in a List Box Control where multiple selection is allowed (MULTI-SELECTION attribute).

Back to Event-Driven Programming Techniques.