Object Linking and Embedding - OLE

Object Linking and Embedding - OLE

This document covers the following topics:

What is OLE in the Natural Context?

Natural supports the following OLE technologies:

- OLE Documents
- OLE Visual Editing (In-place Activation)
- ActiveX Controls

If you are new to OLE, it is highly recommended that you first get a basic overview by referring to one of the various sources available. One such source, for example, is the Microsoft Win32 software development kit documentation.

OLE Documents Support

OLE documents is a technology that integrates different Windows applications seamlessly so that the end user can concentrate on the data rather than on handling the different applications. With OLE you can, for example, embed a Word for Windows document in a Natural dialog. Whenever the end user enters the text container to edit the document, the entire Word functionality is available. Thus, the end user does not have to invoke Word.

OLE Documents Support is provided by the Natural dialog element OLE container control.

The OLE documents technology defines container and server applications. A container application is an application that is able to use objects created by a server application. These objects are used by linking or embedding them. In this context, Natural is the container application because the dialog editor provides an OLE container control. A typical server application is Microsoft Word; the Word documents would then be the objects used by Natural.

Embedding and Linking

- Linking means that the content of a document is accessed via a link to an external file. This file is stored in the server's format (for example, a file in *.rtf* format would be stored in a file system outside Natural; the server residing in this external file system would be Microsoft Word).
- Embedding means that the content of a document is maintained in the container application and is stored in the container's internal format. Embedded documents are created
  - either by building them from scratch in the container application;
  - or by loading an external document.

Embedded objects are edited by visual editing ("in-place activation"), whereas linked objects must be opened in an extra server window for editing.

Natural provides the dialog element OLE container control for embedding and linking documents. Furthermore, Natural provides actions to save and load embedded documents in internal Natural format. By default, these embedded objects in internal format are stored and retrieved in the %NATGUI_BMP% directory with a default extension of *.neo* (Natural Embedded Object).

To display an embedded object with the OLE container control when the dialog starts

1. Invoke the container control's attribute window.
2. Set the **Type** entry to "Existing OLE Object".
3. Select a file specification in the Name field.

To display an embedded object dynamically at runtime

- Use the `PROCESS GUI` statement action `OLE-READ-FROM-FILE`.

To display a linked object with the OLE container control when the dialog starts

1. Invoke the container control's attribute window.
2. Set the **Type** entry to "OLE Server".
3. In the Select OLE Server or Document dialog that comes up, select **Create From File** and select a file specification.

To display a linked object dynamically at runtime

- Assign the file specification of the external document to the attribute `SERVER-OBJECT`.

Visual Editing - In-place Activation

In-place activation means that the end user is able to activate a server application in the container application's window. Such a server application is related to an object embedded in a Natural dialog's OLE container control. The server application is activated by double-clicking on the OLE container control. The Natural dialog's toolbar and menu-bar control are then merged with the server application's

menu and toolbar. The dialog now contains toolbar items and menu items that enable you to edit the object with the help of the server's functionality.

ActiveX Controls Support

ActiveX controls support enables the Natural programmer to use the many third-party ActiveX controls inside a Natural dialog. Natural enables you to access the ActiveX controls properties and methods direct and to program the ActiveX controls events.

ActiveX controls support is provided by the Natural dialog element "ActiveX control". For more information, see *Working with ActiveX Controls*.

OLE Container Control

The following topics are covered below:

- Creating an OLE Container Control
- Creating an OLE Container Control in the Dialog Editor
- Creating an OLE Container Dynamically At Runtime
- Clearing or Deleting an OLE Container At Runtime
- OLE Container Controls And The Dialog's Menu Bar
- Other OLE Container Control Functionality

Creating an OLE Container Control

You can create an OLE container control either statically in the dialog editor or dynamically at runtime.

Creating an OLE Container Control in the Dialog Editor

The OLE container control enables you to integrate server applications. You can integrate server applications in the following three ways, as indicated by the Object Information group frame, **Type** entry of the OLE container control's attributes window.

- Type: New OLE object. You create an OLE container control that acts as a placeholder for the insertable object. At runtime, your end user can create the embedded object by starting the server application. The embedded object can then be saved as Natural embedded object (*.neo* file).
- Type: Existing OLE object. Your end user changes an existing embedded object in the OLE container control. The embedded object is saved as Natural embedded object (*.neo* file).
- Type: OLE server. You create a native OLE object in your application or you create a link to an external object.

To create an OLE container control in the dialog editor

1. In the dialog editor main menu, choose **Insert**, then **OLE Container**.
2. Draw a rectangle by holding down the right mouse button, dragging the mouse vertically/horizontally and releasing the mouse button.

   An empty OLE container is created.

To display a document in the OLE container when starting the dialog

1. Double-click the OLE container control to invoke the attribute window.
2. In the **Type** selection box, choose **OLE server** for linking an external document. Or choose **Existing OLE object** for reading in an embedded object.
3. Choose the **...** button to select the external or embedded object file.

Creating an OLE Container Dynamically At Runtime

Before you enter the examples in an event-handler section, declare a handle variable for the OLE container control in the local data area of the dialog:

```
01 #OCT-1 HANDLE OF OLECONTAINER
```

Example for creating an OLE container control at runtime and linking an external document:

```
PROCESS GUI ACTION ADD WITH
PARAMETERS
   HANDLE-VARIABLE = #OCT-1
   TYPE = OLECONTAINER
   SERVER-OBJECT = 'PICTURE.BMP'
   RECTANGLE-X = 56
   RECTANGLE-Y = 32
   RECTANGLE-W = 336
   RECTANGLE-H = 160
   PARENT = #DLG$WINDOW
   SUPPRESS-CLICK-EVENT = SUPPRESSED
   SUPPRESS-DBL-CLICK-EVENT = SUPPRESSED
   SUPPRESS-CLOSE-EVENT = SUPPRESSED
   SUPPRESS-ACTIVATE-EVENT = SUPPRESSED
   SUPPRESS-CHANGE-EVENT = SUPPRESSED
END-PARAMETERS GIVING *ERROR
```

Example for creating an OLE container control at runtime and embedding a Natural embedded object:

```
PROCESS GUI ACTION ADD WITH
PARAMETERS
   HANDLE-VARIABLE = #OCT-1
   TYPE = OLECONTAINER
   EMBEDDED-OBJECT = 'SLIDE.NEO'
   RECTANGLE-X = 56
   RECTANGLE-Y = 32
   RECTANGLE-W = 336
   RECTANGLE-H = 160
   PARENT = #DLG$WINDOW
   SUPPRESS-CLICK-EVENT = SUPPRESSED
   SUPPRESS-DBL-CLICK-EVENT = SUPPRESSED
```

```
SUPPRESS-CLOSE-EVENT = SUPPRESSED
SUPPRESS-ACTIVATE-EVENT = SUPPRESSED
SUPPRESS-CHANGE-EVENT = SUPPRESSED
END-PARAMETERS GIVING *ERROR
```

Clearing or Deleting an OLE Container At Runtime

This section contains examples for clearing and deleting an OLE container at runtime.

Before you enter the examples in an event-handler section, declare a handle variable for the OLE container control in the local data area of the dialog:

```
01 #OCT-1 HANDLE OF OLECONTAINER
```

Example for clearing (removing the document of) the OLE container control:

```
PROCESS GUI ACTION CLEAR WITH #OCT-1
```

Example for deleting the OLE container control:

```
PROCESS GUI ACTION DELETE WITH #OCT-1
```

OLE Container Controls And The Dialog's Menu Bar

The menu item attribute `MENU-ITEM-OLE` can have four different values which detemine if and where the menu item in question is displayed during in-place activation of a server.

The menu item attribute `MENU-ITEM-TYPE` also has the value `MT-OBJECTVERBS`. This enables you to have the OLE container control display the available server actions (command verbs) in this menu item.

Other OLE Container Control Functionality

While a document is displayed in an OLE container control, the end user has the possibility to activate the default command verb of the server by double-clicking inside the OLE container control's rectangle. This is equivalent to executing the `PROCESS GUI` statement action `OLE-ACTIVATE`. Furthermore, the end user can select a server command verb by displaying a popup menu. You display this popup menu by holding down the right mouse button inside the OLE container. Then you select the desired command verb and release the mouse button.

If the `MODIFIABLE` attribute of an OLE container control is set to `FALSE`, a double-click on the container does not start the default command verb of the server and holding down the right mouse button does not show the popup menu with the available server command verbs (see also *Executing Standardized Procedures*).

During visual editing (in-place activation), the server uses the Natural dialog for the editing of the document. The server does its work as a task on its own and the Natural processing continues. Thus, it is possible to execute event code and, for example, to limit the visual editing to a certain time by specifying `PROCESS GUI ACTION OLE-DEACTIVATE, WITH #OCT-1` in a timer's event section (see also *Executing Standardized Procedures*).

Attributes, Events and PROCESS GUI Statement Actions

The following sections list all the attributes, events and `PROCESS GUI` statement actions that apply specifically to the OLE container control.

Attributes

The OLE-specific attributes provided with the OLE container control are:

- `EMBEDDED-OBJECT`
- `ICONIZED`
- `OBJECT-SIZE`
- `SERVER-OBJECT`
- `SERVER-PROGID`
- `SUPPRESS-ACTIVATE-EVENT`
- `SUPPRESS-CLOSE-EVENT`
- `ZOOM-FACTOR`

Event

This OLE-specific event occurs when a server application is activated:

- Activate event

PROCESS GUI Statement Actions

The OLE-specific `PROCESS GUI` statement actions provided with the OLE container control are:

- `OLE-ACTIVATE`
- `OLE-DEACTIVATE`
- `OLE-GET-DATA`
- `OLE-INSERT-OBJECT`
- `OLE-READ-FROM-FILE`
- `OLE-SAVE-TO-FILE`
- `OLE-SET-DATA`