

Handle-Definition

Die Handle-Definition steht in den folgenden Statements zur Verfügung: DEFINE DATA LOCAL, DEFINE DATA OBJECT, DEFINE DATA PARAMETER, DEFINE DATA INDEPENDENT und DEFINE DATA CONTEXT.

Die *handle-definition* hat die folgende Syntax:

<i>handle-name</i>	HANDLE OF	<i>dialog-element-type</i>	CONSTANT	<i>init-definition</i>
		OBJECT	INIT	
<i>(array-definition)</i>	HANDLE OF	<i>dialog-element-type</i>	CONSTANT	<i>array-init-definition</i>
		OBJECT	INIT	

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Funktion

Eine Handle identifiziert ein Dialog-Element im Code und wird in Handle-Variablen gespeichert. Weitere Informationen siehe Abschnitt *NaturalX* im *Leitfaden zur Programmierung*.

Die Handle-Definition in dem DEFINE DATA-Statement wird automatisch bei der Erstellung eines Dialog-Elements oder Dialogs generiert.

Nachdem Sie eine Handle definiert haben, können Sie den Handle-Namen in einem beliebigen Statement benutzen, um Attributwerte für den definierten *dialog-element-type* abzufragen, zu setzen oder zu ändern.

Siehe auch *Event-Driven Programming Techniques* (Ereignisgesteuerte Programmieretechniken) im *Leitfaden zur Programmierung*.

Beispiele für Handle-Definition:

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

Syntax-Beschreibung

<i>handle-name</i>	Der der Handle zuzweisende Name. Es gelten die Namenskonventionen für Benutzervariablen. Siehe <i>Namen von Benutzervariablen</i> in der Dokumentation <i>Natural Studio benutzen</i> .
--------------------	---

HANDLE OF <i>dialog-element-type</i>	Der Typ des Dialogelements. Mögliche Werte wie beim TYPE-Attribut. Weitere Informationen siehe <i>Dialog Elements</i> und <i>Attributes</i> in der <i>Dialog Component Reference</i> -Dokumentation.
HANDLE OF OBJECT	Wird im Zusammenhang mit NaturalX benutzt, siehe Abschnitt <i>NaturalX</i> im <i>Leitfaden zur Programmierung</i> .
CONSTANT	<p>Die Variable bzw. das Array ist als Namens-Konstante zu behandeln. Der/Die zugewiesene(n) Konstanten-Wert(e) wird/werden jedesmal benutzt, wenn die Variable/das Array referenziert wird. Der/die zugewiesene(n) Wert(e) können nicht während der Ausführung des Programms geändert werden.</p> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Aus Gründen der internen Behandlung ist es nicht zulässig, Definitionen von Variablen und Definitionen von Konstanten innerhalb einer Gruppen-Definition miteinander zu vermischen, d.h. eine Gruppe kann entweder nur Variablen oder nur Konstanten enthalten. 2. Die CONSTANT-Klausel darf nicht bei den Statements DEFINE DATA INDEPENDENT und DEFINE DATA CONTEXT verwendet werden.
INIT	<p>Der Variable bzw. dem Array soll ein Ausgangswert zugewiesen werden. Dieser Wert wird auch benutzt, wenn diese Variable/dieses Array in einem RESET INITIAL-Statement referenziert wird.</p> <p>Anmerkung: Bei den Statements DEFINE DATA INDEPENDENT und DEFINE DATA CONTEXT wird die INIT-Klausel in jedem ausgeführten Programmierobjekt ausgewertet, das diese Klausel enthält (nicht nur im Programmierobjekt, das die Variable zuweist). Bei globalen Variablen funktioniert INIT anders.</p>
<i>init-definition</i>	Mit der Option <i>init-definition</i> definieren Sie die Ausgangswerte oder Konstanten-Werte für eine Variable. Siehe <i>Definition von Ausgangswerten</i> .
<i>array-definition</i>	Mit <i>array-definition</i> definieren Sie die Untergrenze und Obergrenze einer Dimension in einer Array-Definition. Siehe auch den Abschnitt <i>Definition von Array-Dimensionen</i> weiter unten.
<i>array-init-definition</i>	Dem Array soll ein Ausgangswert zugewiesen werden. Dieser Wert wird auch benutzt, wenn dieses Array in einem RESET INITIAL-Statement referenziert wird.