

KAPITEL 2



Lotus Script V4

Mit Lotus Notes R5 wird in Lotus Notes die Version 4 der Programmiersprache Lotus Script implementiert. Derzeit ist die Version 4 von Lotus Script in Lotus SmartSuite 98, Lotus Notes R5 und in LSCube, einem IBM-Produkt (Solution Suite für Lotus Script) implementiert.

Die Version 4 von Lotus Script bietet eine Reihe von Erweiterungen und sprengt viele Limitierungen, die in den Vorgängerversionen von Lotus Script bei größeren Projekten oftmals größeres Kopfzerbrechen verursachten. Lotus-Script-Code in der Version 3 kann jedoch unverändert übernommen werden. Umgekehrt kann eine in der Version 4 (Lotus Notes R5) codierte Routine in einer Version-3-Umgebung von Lotus Script (beispielsweise Lotus Notes Version 4.5x beziehungsweise 4.6x) ausgeführt werden, sofern die Implementierung sich an die Limits von Lotus Script Version 3 hält.

Obwohl bei der Programmierung mit Lotus Script in Lotus-Produkten immer Produktklassen zum Einsatz kommen, um die Funktionalität des Produktes nutzen zu können, enthält Lotus Script selbst kein Objektmodell irgendeiner Anwendung. Es wird immer auf das Objektmodell der umgebenden Anwendung zurückgegriffen. Dennoch sind die Limits generell bei der Programmierung relevant.

In diesem Kapitel wird kurz auf die Änderungen eingegangen, um Ihnen einen Überblick darüber zu geben, mit welchen Gegebenheiten man bei der Programmierung konfrontiert wird beziehungsweise welche Erweiterungen man bei der Programmierung verwenden kann.

2.1 Limits

- ◆ Das Limit bezüglich der gesamten Programmgröße wurde erhöht. Das alte 64-Kbyte-Limit wurde in verschiedenen Bereichen beseitigt und auf eine Gesamtgröße des kompilierten Codes von zwei Gigabyte erhöht. Dadurch dürfte die Fehlermeldung »Maximum allowable code size exceeded« der Vergangenheit angehören.
- ◆ Die Menge der möglichen Funktionen und Variablen wurde von 64 Kbyte Gesamtgröße (zirka 1000 bis 2000 Symbole) auf 64 Kbyte Symbole erhöht. Auch hier dürfte der Fehler »Maximum allowable symbol table size exceeded« von den Bildschirmen verschwinden.
- ◆ Die Größe der Daten auf Modul-Level ((Declarations)-Sektion) wurde für Variablen erhöht. Möglich werden dadurch folgende Deklarationen:



```
Dim a1(32000) as integer
Dim a2(32000) as integer
```

Der zugehörige Fehler war »Size of data cannot exceed 64K in this scope«.

- ◆ Die mögliche Größe für Arrays wurde von 64 Kbyte auf acht Dimensionen erhöht, wobei jede Dimension innerhalb der Grenzen von -32768 bis 32767 liegen muß. Der dadurch hoffentlich für immer eliminierte Fehler war »Array size exceeds maximum«.
- ◆ Die Größe einer Zeichenfolge zur Laufzeit wurde zur Vermeidung des Fehlers »String Too Large« von 64 Kbyte auf zwei Gigabyte erhöht. Zeichenfolgen werden in Lotus Script V4 immer als Unicode gespeichert.

2.2 Verbesserungen

2.2.1 Vorwärts-Referenzen

Klassen, Funktionen und Methoden können nun verwendet werden, bevor sie deklariert wurden. Dies bedingt jedoch einen zusätzlichen Compilerlauf.

Beispiel:

Vorwärts-Referenz einer Methode:

```
class c
  sub X()
    call Y()
  end sub

  sub Y()
    call X()
  end sub
end class
```

Vorwärts-Referenz einer Klasse:

```
class P
  child as C
end class
```

```
class C
  parent as P
  next as C
end class
```

2.2.2 Funktion »Evaluate«

Bis zur Version 3 von Lotus Script konnte der *Evaluate*-Funktion nur ein Konstantenausdruck übergeben werden. Das bedeutete für den Programmierer zusätzlichen Codierungsaufwand, da die zu evaluierende Funktion erst einer Stringvariablen zugewiesen werden mußte. In Lotus Notes Version 4.5x beziehungsweise 4.6x mußte wie folgt codiert werden:

```
...
myName$ = "Raimund Mann/Musik/DE"
sFormula$ = |@Name([CN]; "|" & myName$ & "|") |
vRc = Evaluate( sFormula$ )
Print vRc(0)
...
```

In Lotus Script Version 4 kann kürzer codiert werden:

```
...
vRc = Evaluate(|@Name([CN]; "Raimund Mann/Musik/DE")|)
Print "Common Name", vRc(0)
...
```