

Benutzerkonstanten

Konstanten können überall in Natural-Programmen benutzt werden. Dieses Dokument behandelt die Arten von Konstanten, die unterstützt werden, und erläutert wie sie benutzt werden.

Dieses Kapitel behandelt folgende Themen:

- Numerische Konstanten
 - Alphanumerische Konstanten
 - Unicode-Konstanten
 - Datums- und Zeitkonstanten
 - Hexadezimale Konstanten
 - Logische Konstanten
 - Gleitkomma-Konstanten
 - Attribut-Konstanten
 - Handle-Konstanten
 - Namens-Konstanten definieren
-

Numerische Konstanten

Folgende Themen werden behandelt:

- Numerische Konstanten
- Verarbeitung von numerischen Konstanten

Numerische Konstanten

Eine numerische Konstante kann 1 bis 29 numerische Ziffern enthalten.

Eine mit einem COMPUTE-, MOVE- oder arithmetischen Statement benutzte numerische Konstante kann einen Dezimalpunkt und Vorzeichen enthalten.

Beispiele:

```
1234    +1234    -1234
```

```
12.34   +12.34   -12.34
```

```
MOVE 3 TO #XYZ  
COMPUTE #PRICE = 23.34  
COMPUTE #XYZ = -103  
COMPUTE #A = #B * 6074
```

Intern werden numerische Konstanten in gepackter Form (Format P) dargestellt. Ausnahme: Wenn eine numerische Konstante in einer arithmetischen Operation benutzt wird, in der der andere Operand eine Ganzzahl-Variable (Format I) ist, wird die numerische Konstante in Ganzzahl-Form (Format I) dargestellt.

Verarbeitung von numerischen Konstanten

Wenn eine numerische Konstante innerhalb eines der Statements COMPUTE, MOVE oder DEFINE DATA mit der INIT-Option benutzt werden, überprüft Natural zur Kompilierungszeit, ob ein Konstanten-Wert in das entsprechende Feld passt. Dadurch werden Laufzeitfehler in Situationen vermieden, in denen eine solche Fehlerbedingung bereits bei der Kompilierung entdeckt werden kann.

Alphanumerische Konstanten

Folgende Themen werden behandelt:

- Struktur einer alphanumerischen Konstanten
- Verwendung von Apostrophen in alphanumerischen Konstanten
- Verkettung von alphanumerischen Konstanten

Struktur einer alphanumerischen Konstanten

Eine alphanumerische Konstante kann 1 bis 1.073.741.824 Bytes (1 GB) alphanumerische Zeichen enthalten. Eine alphanumerische

Eine alphanumerische Konstante muss entweder in Apostrophen (')

'text'

oder Anführungszeichen (") stehen.

"text"

Beispiele:

```
MOVE 'ABC' TO #FIELDX
MOVE '% INCREASE' TO #TITLE
DISPLAY "LAST-NAME" NAME
```

Anmerkung:

Eine alphanumerische Konstante, die zur Zuweisung eines Wertes zu einer Benutzervariable verwendet wird, darf nicht auf mehrere Statement-Zeilen aufgeteilt werden.

Verwendung von Apostrophen in alphanumerischen Konstanten

Möchten Sie, dass ein Apostroph (') Teil einer in Apostrophen stehenden alphanumerischen Konstante wird, müssen Sie dies durch 2 Apostrophen (')') oder ein einzelnes Anführungszeichen (')') ausdrücken.

Möchten Sie, dass ein Apostroph (') Teil einer in Anführungszeichen (") stehenden alphanumerischen Konstante wird, drücken Sie dies durch einen einzelnen Apostroph (') aus.

Beispiel:

Wenn Sie Folgendes ausgeben möchten

```
HE SAID, 'HELLO'
```

können Sie eine der folgenden Notationen benutzen:

```
WRITE 'HE SAID, ''HELLO'' '
WRITE 'HE SAID, "HELLO" '
WRITE "HE SAID, ""HELLO"" "
WRITE "HE SAID, 'HELLO' "
```

Anmerkung:

Wenn Anführungszeichen nicht in Apostrophen konvertiert werden (siehe oben), dann ist der Grund dafür die Einstellung des Profilparameters TQ (Anführungszeichen konvertieren). Einzelheiten dazu erfahren Sie von Ihrem Natural-Administrator.

Verketteten von alphanumerischen Konstanten

Alphanumerische Konstanten können mittels eines Bindestrichs miteinander verkettet werden, so dass sie einen einzelnen Wert bilden.

Beispiele:

```
MOVE 'XXXXXX' - 'YYYYYY' TO #FIELD
MOVE "ABC" - 'DEF' TO #FIELD
```

Auf diese Art können alphanumerische Konstanten auch mit hexadezimalen Konstanten verkettet werden.

Unicode-Konstanten

Folgende Themen werden behandelt:

- Unicode-Textkonstanten
- Apostroph innerhalb von Unicode-Textkonstanten
- Unicode-Hexadezimalkonstanten
- Verketteten von Unicode-Konstanten

Unicode-Textkonstanten

Einer Unicode-Textkonstante muss das Zeichen U vorausgehen, und sie muss entweder in Apostrophen (') stehen

```
U'text'
```

oder in Anführungszeichen ("):

```
U"text"
```

Beispiel:

```
U'HELLO'
```

Der Compiler speichert diese Textkonstante in dem generierten Programm im Unicode-Format (UTF-16).

Apostroph innerhalb von Unicode-Textkonstanten

Wenn Sie möchten, dass ein Apostroph (') Teil einer Unicode-Textkonstante wird, die in Apostrophen steht, müssen Sie dies als zwei Apostrophe (' ') oder als ein einzelnes Anführungszeichen (") kodieren.

Wenn Sie möchten, dass ein Apostroph Teil einer Unicode-Textkonstante wird, die in Anführungszeichen steht, müssen Sie dies als einen einzelnen Apostroph (') kodieren.

Beispiel:

Wenn Sie Folgendes ausgeben möchten

```
HE SAID, 'HELLO'
```

können Sie eine der folgenden Notationen benutzen:

```
WRITE U'HE SAID, ''HELLO''
WRITE U'HE SAID, "HELLO"
WRITE U"HE SAID, ""HELLO""
WRITE U"HE SAID, 'HELLO' "
```

Anmerkung:

Wenn Anführungszeichen nicht in Apostrophe umgesetzt werden (siehe oben), dann liegt dies an der Einstellung des Profilparameters TQ (Anführungszeichen umsetzen). Einzelheiten erfahren Sie von Ihrem Natural-Administrator.

Unicode-Hexadezimalkonstanten

Die folgende Syntax wird benutzt, um ein Unicode-Zeichen oder eine Unicode-Zeichenkette in seiner hexadezimale Notation anzugeben:

```
UH' hhhh...'
```

wobei *h* eine hexadezimale Ziffer (0–9, A–F) ist.

Da ein UTF-16-Unicode-Zeichen aus einem Doppelbyte besteht, muss die Anzahl der angegebenen hexadezimalen Zeichen ein Mehrfaches von vier sein.

Beispiel:

Dieses Beispiel definiert die Zeichenkette 45.

```
UH'00340035'
```

Verketteten von Unicode-Konstanten

Die Verkettung von Unicode-Textkonstanten (U) und Unicode-Hexadezimalkonstanten (UH) ist zulässig.

Gültiges Beispiel:

```
MOVE U'XXXXXX' - UH'00340035' TO #FIELD
```

Unicode-Textkonstanten oder Unicode-Hexadezimalkonstanten können nicht mit alphanumerischen Codepage-Konstanten oder H-Konstanten verkettet werden.

Ungültiges Beispiel:

```
MOVE U'ABC' - 'DEF' TO #FIELD
MOVE UH'00340035' - H'414243' TO #FIELD
```

Weiteres gültiges Beispiel:

```
DEFINE DATA LOCAL
1 #U10 (U10)                /* Unicode variable with 10 (UTF-16) characters,
                             /* total byte length = 20
1 #UD (U) DYNAMIC          /* Unicode variable with dynamic length
END-DEFINE
*
#U10 := U'ABC'              /* Constant is created as X'004100420043' in the object,
                             /* the UTF-16 representation for string 'ABC'.

#U10 := UH'004100420043'   /* Constant supplied in hexadecimal format only,
                             /* corresponds to U'ABC'

#U10 := U'A'-UH'0042'-U'C' /* Constant supplied in mixed formats, corresponds to U'ABC'.
END
```

Datums- und Zeitkonstanten

Folgende Themen werden behandelt:

- Datumskonstante
- Zeitkonstante
- Erweiterte Zeitkonstante

Datumskonstante

Eine Datumskonstante kann in Verbindung mit einer Variablen des Formats D benutzt werden.

Datumskonstanten können folgende Formate haben:

D' <i>yyyy-mm-dd</i> '	Internationales Datumsformat
D' <i>dd.mm.yyyy</i> '	Deutsches Datumsformat
D' <i>dd/mm/yyyy</i> '	Europäisches Datumsformat
D' <i>mm/dd/yyyy</i> '	USA-Datumsformat

Dabei bezeichnet *dd* den Tag, *mm* den Monat und *yyyy* das Jahr.

Beispiel:

```
DEFINE DATA LOCAL
  1 #DATE (D)
END-DEFINE
...
MOVE D'2004-03-08' TO #DATE
...
```

Das standardmäßige Datumsformat wird von dem vom Natural-Administrator gesetzten Profilparameter DTFORM (Datumsformat) gesteuert.

Zeitkonstante

Eine Zeitkonstante kann in Verbindung mit einer Variablen des Formats T benutzt werden.

Eine Zeitkonstante hat das folgende Format:

```
T'hh:ii:ss'
```

Dabei bezeichnet *hh* Stunden, *ii* Minuten und *ss* Sekunden.

Beispiel:

```
DEFINE DATA LOCAL
  1 #TIME (T)
END-DEFINE
...
MOVE T'11:33:00' TO #TIME
...
```

Erweiterte Zeitkonstante

Eine Zeitvariable (Format T) kann Datums- und Zeit-Informationen enthalten, wobei die Datumsinformationen eine Untermenge der Zeitinformationen sind. Allerdings können bei einer "normalen" Zeitkonstante (Präfix T) nur die Zeit-Informationen einer Zeitvariablen verarbeitet werden:

```
T'hh:ii:ss'
```

Bei einer erweiterten Zeitkonstante (Präfix E) ist es möglich, den vollständigen Inhalt einer Zeitvariablen einschließlich der Datums-Informationen zu verarbeiten:

```
E'yyyy-mm-dd hh:ii:ss'
```

Einmal abgesehen davon ist die Benutzung einer erweiterten Zeitkonstanten in Verbindung mit einer Zeitvariablen identisch mit der für eine normale Zeitkonstante.

Anmerkung:

Das Format, in dem Datums-Informationen angegeben werden müssen, ist bei einer erweiterten Zeitkonstante abhängig von der Einstellung des Profilparameters DTFORM. Bei der oben angegebenen erweiterten Zeitkonstante wird von DTFORM=I (internationales Datumsformat) ausgegangen.

Hexadezimale Konstanten

Folgende Themen werden behandelt:

- Verwundung und Verarbeitung von hexadezimalen Konstanten
- Verketteten von hexadezimalen Konstanten

Verwundung und Verarbeitung von hexadezimalen Konstanten

Eine hexadezimale Konstante kann benutzt werden, um einen Wert einzugeben, der nicht als ein standardmäßiges Tastaturzeichen eingegeben werden kann.

Eine hexadezimale Konstante kann 1 bis 1.073.741.824 bytes (1 GB) alphanumerische Zeichen enthalten.

Einer hexadezimalen Konstante geht ein Präfix H voraus. Die Konstante selbst muss in Apostrophen (') stehen und kann aus den hexadezimalen Zeichen 0 – 9, A – F bestehen. Zwei hexadezimale Zeichen sind erforderlich, um ein Daten-Byte darzustellen.

Die hexadezimale Darstellung eines Zeichens variiert, je nachdem, ob Ihr Computer einen ASCII- oder EBCDIC-Zeichensatz verwendet. Wenn Sie hexadezimale Konstanten auf einen anderen Computer übertragen, müssen Sie deshalb vielleicht die Zeichen konvertieren.

ASCII-Beispiele:

```
H'313233'      (equivalent to the alphanumeric constant '123')
H'414243'      (equivalent to the alphanumeric constant 'ABC')
```

EBCDIC-Beispiele:

```
H'F1F2F3'      (equivalent to the alphanumeric constant '123')
H'C1C2C3'      (equivalent to the alphanumeric constant 'ABC')
```

Wenn eine hexadezimale Konstante in ein anderes Feld übertragen wird, wird sie als ein alphanumerischer Wert behandelt (Format A).

Die Datenübertragung eines alphanumerischen Werts (Format A) in ein Feld, das nicht in einem der Formate A, U oder B definiert ist, ist nicht zulässig. Deshalb wird eine hexadezimale Konstante als Ausgangswert in einem DEFINE DATA-Statement mit einem Syntaxfehler NAT0094 zurückgewiesen, wenn die entsprechende Variable nicht vom Format A, U oder B ist.

Beispiel:

```
DEFINE DATA LOCAL
1 #I(I2) INIT <H'000F'>          /* causes a NAT0094 syntax error
END-DEFINE
```

Verketteten von hexadezimalen Konstanten

Hexadezimale Konstanten können mittels eines Bindestrichs zwischen den Konstanten miteinander verkettet werden.

ASCII-Beispiel:

```
H'414243' - H'444546' (equivalent to 'ABCDEF')
```

EBCDIC-Beispiel:

```
H'C1C2C3' - H'C4C5C6' (equivalent to 'ABCDEF')
```

Auf diese Weise können hexadezimale Konstanten auch mit alphanumerischen Konstanten verkettet werden.

Logische Konstanten

Die logischen Konstanten TRUE (wahr) und FALSE (falsch) können benutzt werden, um einen logischen Wert einem mit Format L definierten Feld zuzuweisen.

Beispiel:

```
DEFINE DATA LOCAL
  1 #FLAG (L)
END-DEFINE
...
MOVE TRUE TO #FLAG
...
IF #FLAG ...
  statement ...
  MOVE FALSE TO #FLAG
END-IF
...
```

Gleitkomma-Konstanten

Gleitkomma-Konstanten können mit im Format F definierten Variablen benutzt werden.

Beispiel:

```
DEFINE DATA LOCAL
  1 #FLT1 (F4)
END-DEFINE
...
COMPUTE #FLT1 = -5.34E+2
...
```

Attribut-Konstanten

Attribut-Konstanten können mit im Format C (Kontroll-Variablen) definierten Variablen benutzt werden. Diese Art von Konstante muss in Klammern stehen.

Die folgenden Attribute können benutzt werden:

Attribut	Beschreibung
AD=D	Standard
AD=B	blinkend
AD=I	hell hervorgehoben
AD=N	keine Anzeige
AD=V	invers
AD=U	unterstrichen
AD=C	kursiv
AD=Y	dynamisches Attribut
AD=P	geschützt
CD=BL	blau
CD=GR	grün
CD=NE	neutral
CD=PI	rosa
CD=RE	rot
CD=TU	türkis
CD=YE	gelb

Weitere Informationen zu diesen Attributen finden Sie bei den Session-Parametern AD und CD.

Beispiel:

```

DEFINE DATA LOCAL
  1 #ATTR (C)
  1 #FIELD (A10)
END-DEFINE
...
MOVE (AD=I CD=BL) TO #ATTR
...
INPUT #FIELD (CV=#ATTR)
...

```

Handle-Konstanten

Die Handle-Konstante `NULL-HANDLE` kann mit Objekt-Handles benutzt werden.

Weitere Informationen zu Objekt-Handles siehe *NaturalX*.

Namens-Konstanten definieren

Wenn Sie denselben Konstanten-Wert mehrmals in einem Programm benutzen müssen, können Sie den Pflegeaufwand durch Definition einer Namens-Konstante reduzieren:

- Definieren Sie ein Feld im `DEFINE DATA`-Statement,
- weisen Sie ihm einen Konstanten-Wert zu und
- benutzen Sie im Programm den Feldnamen anstatt des Konstanten-Werts.

Wenn der Wert geändert werden muss, müssen Sie ihn somit nur einmal im `DEFINE DATA`-Statement und nicht überall in dem Programm ändern, in dem er vorkommt.

Geben Sie den Konstanten-Wert in viereckigen Klammern mit dem Schlüsselwort `CONSTANT` hinter der Feld-Definition im `DEFINE DATA`-Statement an.

- Wenn der Wert alphanumerisch ist, muss er in Apostrophen (') stehen.
- Wenn der Wert Text im Unicode-Format ist, muss ihm das Zeichen `U` vorangehen, und er muss in Apostrophen (') stehen.
- Wenn der Wert im hexadezimalen Unicode-Format ist, müssen ihm die Zeichen `UH` vorangehen, und er muss in Apostrophen (') stehen.

Beispiel:

```
DEFINE DATA LOCAL
  1 #FIELD A (N3) CONSTANT <100>
  1 #FIELD B (A5) CONSTANT <'ABCDE'>
  1 #FIELD C (U5) CONSTANT <U'ABCDE'>
  1 #FIELD D (U5) CONSTANT <UH'00410042004300440045'>
END-DEFINE
...
```

Während der Ausführung des Programms kann der Wert einer solchen Namens-Konstante nicht geändert werden.