

# Dynamische Variablen

Dieses Kapitel behandelt folgende Themen:

- Sinn und Zweck dynamischer Variablen
  - Definition dynamischer Variablen
  - Zurzeit für eine dynamische Variable benutzter Wertespeicher
  - Größenbeschränkungsprüfung
  - Hauptspeicherplatz für eine dynamische Variable zuweisen/freigeben
- 

## Sinn und Zweck dynamischer Variablen

Insofern als die maximale Größe der großen Datenstrukturen (zum Beispiel Bilder, Audiosignale, Videos) zur Anwendungsentwicklungszeit nicht genau bekannt sein kann, bietet Natural zusätzlich die Definition alphanumerischer und binärer Variablen mit dem Attribut `DYNAMIC`.

Der Wertebereich von Variablen, die mit diesem Attribut definiert sind, wird zur Ausführungszeit dynamisch erweitert, wenn es notwendig wird (zum Beispiel bei einer Zuweisungsoperation: `#picture1 := #picture2`). Dies bedeutet, dass große binäre und alphanumerische Datenstrukturen in Natural verarbeitet werden können, ohne dass Sie eine Beschränkung zur Entwicklungszeit definieren müssen.

Die Zuweisung von dynamischen Variablen zur Ausführungszeit unterliegt natürlich den Beschränkungen des verfügbaren Hauptspeichers. Wenn die Zuweisung dynamischer Variablen dazu führt, dass eine Meldung wegen unzureichenden Hauptspeichers vom zugrundeliegenden Betriebssystem zurückgegeben wird, kann das `ON ERROR`-Statement benutzt werden, um diese Fehlerbedingung abzufangen, sonst wird eine Fehlermeldung von Natural zurückgegeben.

Die Natural-Systemvariable `*LENGTH` kann benutzt werden, um die Länge (in Code- Einheiten) des Wertespeichers zu erhalten, der zur Zeit für eine gegebene dynamische Variable benutzt wird. Für Format A und B ist die Größe einer Code-Einheit 1 Byte. Für Format U ist die Größe einer Code-Einheit 2 Bytes (UTF-16). Natural setzt `*LENGTH` automatisch auf die Länge des Source-Operanden bei Zuweisungen, in denen die dynamische Variable betroffen ist. `*LENGTH(feld)` gibt deshalb die aktuell für ein/e dynamische/s Natural-Feld oder eine Variable in Code-Einheiten benutzte Größe zurück.

Wenn der Speicherplatz für die dynamische Variable nicht mehr erforderlich ist, kann das `REDUCE`- oder `RESIZE`-Statement benutzt werden, um den für die dynamische Variable benutzten Speicherplatz auf Null (oder eine andere gewünschte Größe) zu reduzieren. Wenn die obere Grenze der Hauptspeicher-Benutzung für eine spezifische dynamische Variable bekannt ist, kann das `EXPAND`-Statement benutzt werden, um den für die dynamische Variable benutzten Speicherplatz auf diese bestimmte Größe zu setzen.

Große Variablen für alphanumerische und binäre Daten basieren auf den bekannten Natural-Formaten A und B. Die Beschränkungen von 253 Bytes für Format A und 126 für Format B sind nicht mehr gültig. Die neue Größenbeschränkung ist 1 GB. Diese großen statischen Variablen und Felder werden genauso verarbeitet wie traditionelle alphanumerische und binäre Variablen und Felder in Bezug auf Definition,

Redefinition, Wertespeicherzuweisung, Konvertierungen, Referenzierungen in Statements, usw. Alle Regeln zu alphanumerischen und binären Formaten gelten für diese großen Formate.

Soll eine dynamische Variable initialisiert werden, sollte das MOVE ALL UNTIL-Statement für diesen Zweck benutzt werden.

## Definition dynamischer Variablen

Da die wirkliche Größe von großen alphanumerischen und binären Datenstrukturen zur Anwendungsentwicklungszeit nicht genau bekannt sein mag, kann die Definition *dynamischer* Variablen des Formats A, B oder U zur Verwaltung dieser Strukturen benutzt werden. Die dynamische Zuweisung und Erweiterung (Neuzuweisung) großer Variablen ist gegenüber der Anwendungsprogrammierungslogik transparent. Dynamische Variablen werden ohne Längenangabe definiert. Der Hauptspeicher wird entweder implizit zur Ausführungszeit zugewiesen, wenn die dynamische Variable als ein Ziel-Operand benutzt wird, oder explizit mit einem EXPAND- oder RESIZE-Statement.

Dynamische Variablen können nur in einem DEFINE DATA-Statement mittels der folgenden Syntax definiert werden:

```
level variable-name (format) DYNAMIC
```

Dabei ist:

- *level* die Stufennummer
- *variable-name* der Name der großen Variablen
- *format* das Format der Variablen (A, U oder B)
- DYNAMIC das Schlüsselwort, durch das die Variable als dynamische Variable definiert wird.

Die folgenden Beschränkungen gelten für eine dynamische Variable:

- Eine Redefinition einer dynamischen Variable ist nicht zulässig.
- Eine dynamische Variable darf nicht in einer REDEFINE-Klausel enthalten sein.

## Zurzeit für eine dynamische Variable benutzter Wertespeicher

Die Größe (in Code-Einheiten) des zurzeit benutzten Wertespeichers einer dynamischen Variable kann aus der Systemvariable \*LENGTH übernommen werden. \*LENGTH wird bei Zuweisungen automatisch auf die (verwendete) Länge des Source-Operanden gesetzt.

**Warnung:**

Aus Performance-Gründen kann der zur Aufnahme des Wertes der dynamischen Variablen zugewiesene Speicherbereich größer sein als der Wert von `*LENGTH` (benutzte Größe steht dem Entwickler zur Verfügung). Sie sollten sich nicht auf den Speicherplatz verlassen, der über die benutzte Länge (wie durch `*LENGTH` angegeben) hinaus zugewiesen wird. Dieser Platz kann jederzeit freigegeben werden, auch wenn auf die betreffende dynamische Variable nicht zugegriffen wird. Es ist für den Natural-Programmierer nicht möglich, Informationen über die aktuell zugewiesene Größe zu erhalten. Diese Größe ist ein interner Wert.

Die Systemvariable `*LENGTH(field)` gibt die benutzte Länge eines dynamischen Natural-Feldes oder einer solchen Variable in Code-Einheiten zurück. `*LENGTH` kann nur benutzt werden, um die aktuell verwendete Länge für dynamische Variablen zu erhalten.

## Größenbeschränkungsprüfung

### Profilparameter `USIZE`

Bei dynamischen Variablen ist eine Größenbeschränkungsprüfung während der Kompilierung nicht möglich, da bei dynamischen Variablen keine Länge definiert ist. Der Profilparameter `USIZE` gibt die Größe des Benutzerpuffers im virtuellen Speicher an. Der Benutzerpuffer enthält alle Daten, die von Natural dynamisch zugewiesen wurden. Wenn eine dynamische Variable während der Ausführung zugewiesen oder erweitert wird und die `USIZE`-Grenze überschritten wird, wird eine Fehlermeldung ausgegeben.

## Hauptspeicherplatz für eine dynamische Variable zuweisen/freigeben

Die Statements `EXPAND`, `REDUCE` und `RESIZE` werden benutzt, um Hauptspeicherplatz für eine dynamische Variable explizit zuzuweisen und freizugeben.

### Syntax:

```
EXPAND [SIZE OF] DYNAMIC [VARIABLE] operand1 TO operand2
REDUCE [SIZE OF] DYNAMIC [VARIABLE] operand1 TO operand2
RESIZE [SIZE OF] DYNAMIC [VARIABLE] operand1 TO operand2
```

dabei ist *operand1* eine dynamische Variable und *operand2* ein nicht-negativer, numerischer Längenwert ist.

## EXPAND

Das EXPAND-Statement wird benutzt, um den zurzeit *zugewiesenen* Speicherplatz der dynamischen Variable (*operand1*) auf die angegebene Größe (*operand2*) zu erweitern.

Die aktuell benutzte Größe (siehe Natural-Systemvariable \*LENGTH weiter oben) für die dynamische Variable wird nicht geändert.

Wenn die angegebene Größe (*operand2*) kleiner ist als die Größe des aktuell zugewiesenen Speicherplatzes der dynamischen Variable, wird das Statement ignoriert.

## REDUCE

Das REDUCE-Statement wird benutzt, um die Größe des zurzeit *zugewiesenen* Speicherplatzes der dynamischen Variable (*operand1*) auf die angegebene Größe (*operand2*) zu reduzieren.

Der über die angegebene Größe (*operand2*) hinausgehende, für die dynamische Variable (*operand1*) zugewiesene Speicherplatz kann jederzeit freigegeben werden, wenn das Statement ausgeführt wird, oder auch später.

Wenn die zurzeit *benutzte* Speichergröße (siehe Natural-Systemvariable \*LENGTH weiter oben) für die dynamische Variable größer ist als die angegebene Speichergröße (*operand2*), wird \*LENGTH dieser dynamischen Variable auf die angegebene Größe gesetzt. Der Inhalt der Variablen wird abgeschnitten, aber nicht geändert.

Wenn die angegebene Größe größer ist als der zurzeit zugewiesene Speicherplatz der dynamischen Variable, wird das Statement ignoriert.

## RESIZE

Das RESIZE-Statement passt die Größe des zurzeit zugewiesenen Speicherplatzes der dynamischen Variable (*operand1*) an die angegebene Größe (*operand2*) an.

Wenn die angegebene Größe der dynamischen Variable kleiner ist als die benutzte Größe (siehe Natural-Systemvariable \*LENGTH weiter oben), wird die benutzte Größe dementsprechend reduziert.

Wenn die *angegebene* Größe größer ist als die Größe des zurzeit zugewiesenen Speicherplatzes der dynamischen Variable, wird die Größe des zugewiesenen Speicherplatzes der dynamischen Variable erhöht. Die aktuell benutzte Größe (siehe \*LENGTH) der dynamischen Variable ist davon nicht betroffen und bleibt unverändert.

Wenn die angegebene Größe identisch ist mit der Größe des zurzeit zugewiesenen Speicherplatzes der dynamischen Variable, hat die Ausführung des RESIZE-Statements keine Auswirkungen.