

Anwendungsbezogene Systemvariablen

Dieses Kapitel behandelt folgende Systemvariablen:

- *APPLIC-ID
- *APPLIC-NAME
- *COM
- *CONTROL
- *CONVID
- *COUNTER (r)
- *CPU-TIME
- *CURRENT-UNIT
- *DATA
- *DIALOG-ID
- *ERROR-LINE
- *ERROR-NR
- *ERROR-TA
- *ETID
- *EVENT
- *ISN (r)
- *LBOUND
- *LENGTH (field)
- *LEVEL
- *LIBRARY-ID
- *LINE
- *NUMBER (r)
- *OCCURRENCE
- *PAGE-EVENT

- *PAGE-LEVEL
 - *PROGRAM
 - *ROWCOUNT
 - *STARTUP
 - *STEPLIB
 - *SUBROUTINE
 - *THIS-OBJECT
 - *TYPE
 - *UBOUND
-

*APPLIC-ID

Format/Länge: A8
Inhalt modifizierbar Nein

Diese Systemvariable enthält die ID der Library, in der der Benutzer zurzeit angemeldet ist.

*APPLIC-NAME

Format/Länge: A32
Inhalt modifizierbar Nein

Unter Natural Security

Wenn Natural Security installiert ist, enthält diese Systemvariable den Namen der Library , in der der Benutzer zurzeit angemeldet ist, oder, wenn der Benutzer über einen speziellen Link angemeldet ist, den Link-Namen. Wenn Natural Security nicht installiert ist, enthält diese Systemvariable den Namen SYSTEM.

Die allgemeine Option `Set *APPLIC-NAME always to library name` kann so gesetzt werden, dass *APPLIC-NAME immer den Namen der Library enthält, und zwar unabhängig davon, ob der Benutzer über einen speziellen Link angemeldet ist oder nicht. Siehe `Set *APPLIC-NAME always to library name` in der *Natural Security*-Dokumentation.

*COM

Format/Länge: A128
Inhalt modifizierbar Ja

Diese Systemvariable stellt einen Kommunikationsbereich dar, der es ermöglicht, Daten von außerhalb eines Bildschirmfensters zu verarbeiten.

Wenn ein Fenster aktiv ist, können normalerweise keine Daten außerhalb des Fensters auf dem Bildschirm eingegeben werden. Wenn jedoch eine Map *COM als modifizierbares Feld enthält, kann ein Benutzer in dieses Feld auch dann Daten eingeben, wenn gerade ein Fenster auf dem Bildschirm aktiv ist.

Die weitere Verarbeitung kann dann vom Inhalt von *COM abhängig gemacht werden. Auf diese Weise können Sie Benutzeroberflächen implementieren, bei denen ein Benutzer auch dann Daten in der Kommandozeile eingeben kann, wenn ein Fenster mit eigenen Eingabefeldern aktiv ist.

Anmerkung:

Obwohl *COM als modifizierbares Feld in einem INPUT-Statement verwendet werden kann, wird es nicht als Eingabefeld, sondern als Systemvariable behandelt; d.h. Eingaben, die in *COM gemacht werden, werden genommen, wie sie sind, ohne dass eine Eingabeverarbeitung (z.B. Umsetzung in Großbuchstaben) erfolgt. Sobald *COM über ein INPUT-Statement auf dem Bildschirm angezeigt wurde, wird mit jedem anschließenden INPUT- oder REINPUT-Statement der jeweils aktuelle Inhalt von *COM ausgegeben.

Siehe auch die folgenden Abschnitte im *Leitfaden zur Programmierung*:

- *Dialog-Gestaltung*
 - *Verarbeitung von Daten außerhalb des aktiven Fensters*
 - *Positionierung des Cursors auf *COM – Terminalkommando %T**
 - *Daten vom Bildschirm kopieren*

***CONTROL**

Format/Länge:	Handle
Inhalt modifizierbar	Nein

Diese Systemvariable enthält das Handle des Dialogelements für welches das aktuelle Event ausgelöst wurde.

Ausführliche Informationen über Events finden Sie unter *Event-Driven Programming Techniques* im *Leitfaden zur Programmierung*.

***CONVID**

Format/Länge:	I4
Inhalt modifizierbar	Ja

Diese Systemvariable enthält die Konversations-ID des aktuellen konversationalen Remote Procedure Calls (RPC). Diese ID wird von einem OPEN CONVERSATION-Statement gesetzt.

Mit einem OPEN CONVERSATION-Statement kann ein Client einen Server zur alleinigen Benutzung erhalten, um eine Reihe von Services (Subprogrammen) innerhalb eines Server-Prozesses auszuführen. Diese alleinige Benutzung wird als Konversation bezeichnet. Das OPEN CONVERSATION-Statement dient dazu, eine Konversation zu eröffnen und die an ihr beteiligten Subprogramme anzugeben. Wenn ein OPEN CONVERSATION-Statement ausgeführt wird, weist es der Systemvariablen *CONVID eine eindeutige ID zu, die die Konversation identifiziert.

Es können mehrere Konversationen gleichzeitig offen sein. Um von einer offenen Konversation zu einer anderen zu wechseln, weisen Sie *CONVID die entsprechende Konversations-ID zu.

Weitere Informationen zu Natural RPC finden Sie in der *Natural Remote Procedure Call (RPC)*-Dokumentation.

*COUNTER (r)

Format/Länge: P10
 Inhalt modifizierbar Ja

Diese Systemvariable enthält die Anzahl, wie oft eine mit einem FIND-, READ-, HISTOGRAM- oder PARSE-Statement initiierte Verarbeitungsschleife durchlaufen wurde.

Durch Angabe von (r) in Klammern hinter *COUNTER können Sie eine bestimmte Schleife referenzieren, wobei r das Statement-Label bzw. die Sourcecode-Zeilenummer des betreffenden FIND-, READ-, HISTOGRAM- oder PARSE-Statements ist. Wenn Sie keine bestimmte Schleife referenzieren, bezieht sich *COUNTER auf die gerade aktive Verarbeitungsschleife.

Datensätze, die aufgrund einer WHERE-Klausel nicht weiterverarbeitet werden, werden im *COUNTER nicht mitgezählt. Datensätze, die aufgrund eines ACCEPT/REJECT-Statements nicht weiterverarbeitet werden, werden mitgezählt.

*CPU-TIME

Format/Länge: I4
 Inhalt modifizierbar Nein

Diese Systemvariable enthält die aktuell vom Natural-Prozess benutzte CPU-Zeit in Einheiten von 10 ms.

*CURRENT-UNIT

Format/Länge: A32
 Inhalt modifizierbar Nein

Diese Systemvariable enthält den Namen der aktuell ausgeführten Einheit. Dies ist

- im Falle des Objekttyps "Function" der Name der Function,

- der Name der internen Subroutine, falls eine interne Subroutine ausgeführt wird,
- der Name der externen Subroutine im Falle des Objekttyps Subroutine (siehe auch Systemvariable *SUBROUTINE),
- im Falle aller anderen Objekttypen (Programm, Subprogramm, Map, Dialog usw.) der Name des Objekts (siehe auch Systemvariable *PROGRAM).

Der Inhalt von *CURRENT-UNIT liegt immer in Großbuchstaben vor.

***DATA**

Format/Länge: N3
Inhalt modifizierbar Nein

Diese Systemvariable enthält die Anzahl der im Natural-Stack gelagerten Datenelemente, die dem nächsten INPUT-Statement als Eingabedaten zur Verfügung stehen. Ist der Stack leer, enthält *DATA den Wert null (0). Ein Wert von -1 bedeutet, dass das zuoberst im Stack gelagerte Element ein Kommando oder der Name einer Natural-Transaktion ist.

Die Werte der Profil/Session-Parameter IA und ID zum Zeitpunkt der Ausführung des STACK-Statements dienen dazu, den Wert von *DATA zu bestimmen.

***DIALOG-ID**

Format/Länge: I4
Inhalt modifizierbar Nein

Diese Systemvariable enthält die ID der aktuellen Dialoginstanz.

Ausführliche Informationen über Dialogen finden Sie unter *Event-Driven Programming Techniques* im *Leitfaden zur Programmierung*.

***ERROR-LINE**

Format/Länge: N4
Inhalt modifizierbar Nein

Diese Systemvariable enthält die Sourcecode-Zeilenummer des Statements, das einen Fehler verursacht hat.

***ERROR-NR**

Als alternative Schreibweise können Sie auch *ERROR verwenden.

Format/Länge: N7

Inhalt modifizierbar Ja

Diese Systemvariable enthält die Fehlernummer des Fehlers, der die Verzweigung zu einer ON ERROR-Bedingung bewirkt hat.

Es werden nur Fehlernummern im Bereich von 0 bis 9999 unterstützt.

Normalerweise enthält *ERROR-NR die Natural-System-Fehlernummer, die das Eintreten der Fehlerbedingung verursacht hat; wenn allerdings ein Statement REINPUT WITH TEXT *nnnn ausgeführt wird, wird *ERROR-NR mit der betreffenden anwendungsspezifischen Fehlernummer nnnn gefüllt.

Sie können dieser Systemvariablen in einem Natural-Programm einen anderen Wert zuweisen, allerdings nicht innerhalb eines ON ERROR-Statement-Blocks.

*ERROR-NR wird wieder auf null (0) gesetzt, sobald ein neues Level-1-Programm ausgeführt wird.

*ERROR-TA

Format/Länge: A8

Inhalt modifizierbar Ja

Diese Systemvariable enthält den Namen des Programms, an das die Kontrolle im Falle einer Fehlerbedingung übergeben wird.

Im Falle eines Fehlers führt Natural ein STACK TOP DATA-Statement aus und legt folgende Informationen, welche von einer Error-Transaktion als INPUT-Daten verwendet werden können, oben auf dem Stack ab:

- Fehlernummer (N4 bei SG=OFF; N5 bei SG=ON)
- Zeilennummer (N4)
- Status (A1)
- Programmname (A8)
- Level (N2)

Status kann sein:

C	Kommandoverarbeitungsfehler.
L	Logon-Fehler. Bei Status L ist die Zeilennummer 0.
O	Objektzeitfehler.
S	Nicht korrigierbarer Syntaxfehler.
R	Fehler auf Remote-Server (im Zusammenhang mit Natural RPC).

***ETID**

Format/Länge: A8
 Inhalt modifizierbar Nein

Diese Systemvariable enthält den Wert zur Identifizierung von Transaktionsdaten (End of Transaction ID) für Adabas. Dieser Wert kann folgendes sein:

- der Wert des Natural-Profilparameters ETID,
- die im Security-Profil des gerade aktiven Benutzers definierte ETID (falls Natural Security eingesetzt wird).

***EVENT**

Format/Länge: A32
 Inhalt modifizierbar Nein

Diese Systemvariable enthält den Namen des aktuellen Events.

Ausführliche Informationen über Events finden Sie unter *Event-Driven Programming Techniques im Leitfaden zur Programmierung*.

***ISN (*r*)**

Format/Länge: P10
 Inhalt modifizierbar Ja

Diese Systemvariable enthält die Adabas-ISN des Datensatzes, der gerade in einer mit einem FIND- oder READ-Statement initiierten Verarbeitungsschleife verarbeitet wird.

Mit *r* in Klammern hinter *ISN können Sie eine bestimmte Schleife referenzieren, wobei *r* das Label bzw. die Sourcecode-Zeilenummer des betreffenden FIND- oder READ-Statements ist. Wenn Sie keine bestimmte Schleife referenzieren, bezieht sich *ISN auf die gerade aktive Verarbeitungsschleife.

Bei einer mit HISTOGRAM initiierten Verarbeitungsschleife enthält *ISN die Nummer der Ausprägung, in der der zuletzt gelesene Wert des gerade verarbeiteten Deskriptors enthalten ist. Ist der Deskriptor nicht Teil einer Periodengruppe, enthält *ISN den Wert null (0).

Datenbank-spezifische Anmerkungen:

SQL-Datenbanken	*ISN kann nicht benutzt werden.
Tamino	*ISN enthält die XML-Objekt-ID.

*LBOUND

Format/Länge: I4
 Inhalt modifizierbar Nein

LBOUND enthält die aktuelle Untergrenze (Indexwert) eines Arrays für die angegebene(n) Dimension(en) (1, 2 oder 3) oder für alle Dimensionen (Stern-Notation ()).

Syntax:

*LBOUND (<i>operand1</i> [, <i>dim</i>])

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	A	A U N P I F B D T L C G O	ja	nein

operand1 ist das Array, für das die Untergrenze angegeben wird. Die Index-Notation des Arrays ist optional. Als Index-Notation ist nur die Stern-Notation (*) für den vollständigen Bereich für jede Dimension zulässig.

dim ist die Dimensionsnummer, für die die aktuelle Untergrenze zurückgegeben wird:

$dim = \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ * \end{array} \right\}$
--

Wenn keine Dimension angegeben wird, wird die Untergrenze der ersten Dimension zurückgegeben.

Wenn 1, 2 oder 3 angegeben wird, wird die Untergrenze der ersten, zweiten oder dritten Dimension zurückgegeben.

Wird Stern-Notation (*) verwendet, wird die Untergrenze aller definierten Dimensionen zurückgegeben, d.h.

1	im Falle eines eindimensionalen Arrays
2	im Falle eines zweidimensionalen Arrays
3	im Falle eines dreidimensionalen Arrays

Wenn ein X-Array nicht zugewiesen wird und die Untergrenze der angegebenen Dimension dieses X-Arrays die variable Indexgrenze ist, d.h., wenn sie durch einen Stern (*) in der Indexdefinition dargestellt wird, dann ist die Untergrenze dieses X-Arrays nicht definiert, und der Zugriff auf *LBOUND

führt zu einem Laufzeitfehler. Um den Laufzeitfehler zu vermeiden, kann die Systemvariable *OCCURRENCE benutzt werden, um auf Null-Ausprägungen abzuprüfen:

```
DEFINE DATA LOCAL
  1 #XA(A5/1:*)
END-DEFINE
IF *OCCURRENCE (#XA) NE 0 AND *LBOUND(#XA) > 10
  THEN ...
```

Beispiele:

```
DEFINE DATA LOCAL
  1 #I (I4)
  1 #J (I4/1:3)
  1 #XA (A5/10:*,20:*)
END-DEFINE
#I := *LBOUND(#XA) /* lower bound of 1st dimension is 10
#I := *LBOUND(#XA,1) /* lower bound of 1st dimension is 20
#I := *LBOUND(#XA,2) /* lower bound of 2nd dimension is 20
#J(1:2):= *LBOUND(#XA,*) /* lower bound of all dimensions
/* (1st and 2nd)
/* #J(1) is 10 and #J(2) is 20
```

Siehe auch Systemvariablen *UBOUND und *OCCURRENCE.

***LENGTH (field)**

Format/Länge: I4
 Inhalt modifizierbar Nein

Diese Systemvariable gibt die gerade benutzte Länge (in Bytes) eines Feldes zurück, das als dynamische Variable in Form von Code-Einheiten definiert ist. Für Format A und B ist die Länge einer Code-Einheit 1 Byte, und für Format U ist die Länge einer Code-Einheit 2 Bytes (UTF-16).

*LENGTH(*field*) gilt nur für dynamische Variablen.

Siehe auch *Aktuell für eine dynamische Variable benutzter Wertespeicher im Leitfaden zur Programmierung*.

***LEVEL**

Format/Länge: N2
 Inhalt modifizierbar Nein

Diese Systemvariable enthält die Stufennummer (Level Number) des Objekts (Programm, Subprogramm, externe Subroutine, Map, Helpoutine oder Dialog), das gerade ausgeführt wird. Stufennummer 1 bezeichnet jeweils ein Hauptprogramm.

Interne Subroutinen werden von *LEVEL nicht berücksichtigt.

*LIBRARY-ID

Format/Länge: A8
Inhalt modifizierbar Nein

Diese Systemvariable enthält die (mit dem LOGON-Kommando angegebene) Kennung der Bibliothek (Library-ID), in der der Benutzer gerade arbeitet.

Diese Systemvariable entspricht der Systemvariablen *APPLIC-ID.

*LINE

Format/Länge: I4
Inhalt modifizierbar Nein

Diese Systemvariable enthält die Nummer der gerade in einem Natural-Objekt ausgeführten Zeile.

*NUMBER (*r*)

Format/Länge: P10
Inhalt modifizierbar Ja

Diese Systemvariable enthält die Anzahl der Datensätze, die mit einem FIND-Statement (unter Erfüllung der WITH-Bedingung) gelesen wurden, oder die Anzahl der ISNs, die mit einem HISTOGRAM-Statement ausgewählt wurden.

Mit der Notation (*r*) hinter *NUMBER können Sie ein bestimmtes FIND- oder HISTOGRAM-Statement referenzieren, wobei *r* das Label bzw. die Sourcecode-Zeilenummer des Statements ist. Wenn Sie kein bestimmtes Statement referenzieren, bezieht sich *NUMBER auf die gerade aktive FIND- bzw. HISTOGRAM-Schleife.

Anmerkung:

Wenn die aufgerufene Adabas-Datei durch die Adabas-Funktion *Security By Value* geschützt ist, enthält *NUMBER den Wert 9999999999, wenn mehr als 1 Datensatz gefunden wurde. Wenn 1 Datensatz gefunden wurde, enthält *NUMBER den Wert 1. Wenn kein Datensatz gefunden wurde, enthält *NUMBER den Wert 0.

Datenbank-spezifische Anmerkungen:

SQL-Datenbanken	Bei SQL-Datenbanken enthält *NUMBER nur die Anzahl der gefundenen Zeilen, wenn *NUMBER mit einem FIND NUMBER- oder einem HISTOGRAM-Statement ohne WHERE-Klausel benutzt wird. In allen anderen Fällen enthält *NUMBER nicht die Anzahl der gefundenen Zeilen: Werden keine Zeilen gefunden, enthält *NUMBER den Wert null (0); jeder andere Wert bedeutet, dass Zeilen gefunden wurden, aber der Wert steht in keinem Zusammenhang mit der tatsächlichen Anzahl der gefundenen Zeilen.
Tamino	<p>Wenn *NUMBER in Verbindung mit einem FIND NUMBER-Statement ohne WHERE-Klausel verwendet wird, enthält sie die Anzahl der gefundenen Zeilen.</p> <p>Wenn *NUMBER anderweitig bei einer XML-Datenbank verwendet wird, enthält sie nicht die Anzahl der gefundenen Zeilen: Werden keine Zeilen gefunden, enthält *NUMBER den Wert null (0); jeder andere Wert bedeutet, dass Zeilen gefunden wurden, aber der Wert steht in keinem Zusammenhang mit der tatsächlichen Anzahl der gefundenen Zeilen.</p> <p>Wenn FIND NUMBER bei einer WHERE-Klausel benutzt wird, wird die Anzahl der gefundenen Zeilen in *COUNTER zurückgegeben.</p>

*OCCURRENCE

Format/Länge: I4

Inhalt modifizierbar Nein

Diese Systemvariable enthält die aktuelle Anzahl der Ausprägungen eines variablen Arrays für die angegebene(n) Dimension(en).

Syntax:

*OCCURRENCE (<i>operand1</i> [, <i>dim</i>])

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	A	A U N P I F B D T L C G O	ja	nein

Operand1 ist das Array, für das die Anzahl der Ausprägungen zurückgegeben wird. Die Index-Notation des Arrays ist optional. Als Index-Notation ist nur die Stern-Notation (*) für den vollständigen Bereich für jede Dimension zulässig.

dim ist die Dimensions-Nummer, für die die aktuelle Anzahl der Ausprägungen zurückgegeben wird:

$$dim = \begin{pmatrix} 1 \\ 2 \\ 3 \\ * \end{pmatrix}$$

Erklärung:

1	Eindimensionales Array. Dies ist die Voreinstellung, wenn <i>dim</i> nicht angegeben wird.
2	Zweidimensionales Array.
3	Dreidimensionales Array.
*	Alle für das entsprechende Array definierten Dimensionen sind gültig.

In einer Parameter Data Area können Sie mit der Index-Notation 1:V ein Array mit einer variablen Anzahl von Ausprägungen definieren (vgl. DEFINE DATA-Statement). Die tatsächliche Anzahl der Ausprägungen eines solchen variablen Arrays wird erst zur Laufzeit bestimmt. Mit *OCCURRENCE können Sie die tatsächliche Anzahl der Array-Ausprägungen ermitteln.

Beispiele:

```
DEFINE DATA
  PARAMETER
    1 #ARRAY (A5/1:V)
  LOCAL
    1 #I (I4)
    ...
END-DEFINE
...
FOR #I = 1 TO *OCCURRENCE(#ARRAY)
  ...
END-FOR
...
```

Siehe auch Beispiel-Programme OCC1P und OCC2P.

In Zusammenhang mit X-Arrays enthält *OCCURRENCE die aktuelle Anzahl der Ausprägungen:

```
DEFINE DATA LOCAL
  1 #I (I4)
  1 #J (I4/1:3)
  1 #XA (A5/1:*,1:*)
END-DEFINE
EXPAND ARRAY #XA TO (1:10,1:20)
#I := *OCCURRENCE(#XA) /* number of occurrences of 1st dimension is 10
#I := *OCCURRENCE(#XA,1) /* number of occurrences of 1st dimension is 10
#I := *OCCURRENCE(#XA,2) /* number of occurrences of 2nd dimension is 20
#J(1:2) := *OCCURRENCE(#XA,*) /* number of occurrences of all dimensions
/* (1st and 2nd)
/* #J(1) is 10 and #J(2) is 20
END
```

*PAGE-EVENT

Format/Länge: U (dynamisch)
Inhalt modifizierbar Nein

Diese Systemvariable enthält den von Natural for Ajax gelieferten Namen des aktuellen Ereignisses.

Diese Systemvariable wird mit dem PROCESS PAGE-Statement zur Programmierung von Rich GUI-Anwendungen verwendet.

Weitere Informationen finden Sie in der *Natural for Ajax*-Dokumentation.

*PAGE-LEVEL

Format/Länge: I4
Inhalt modifizierbar Nein

Diese Systemvariable enthält den Level des aktiven PROCESS PAGE MODAL-Statement-Blocks.

Ist kein PROCESS PAGE MODAL-Statement-Block aktiv ist, dann ist der Wert von *PAGE-LEVEL Null (0).

Anmerkung:

Ist der Wert von *PAGE-LEVEL größer als Null, dann ist keine Ausgabe an Report 0 über ein INPUT-, PRINT-, WRITE- oder DISPLAY-Statement möglich.

*PROGRAM

Format/Länge: A8
Inhalt modifizierbar Nein

Diese Systemvariable enthält den Namen des Natural-Objekts, das gerade ausgeführt wird.

*ROWCOUNT

Format/Länge: I4
Inhalt modifizierbar Nein

Diese Systemvariable enthält die Anzahl der Zeilen, die mit einem der Natural-SQL-Statements "searched" DELETE, "searched" UPDATE oder INSERT (mit *select-expression*) in einer Datenbank-Tabelle gelöscht, geändert bzw. hinzugefügt wurden. *ROWCOUNT bezieht sich jeweils auf das zuletzt ausgeführte dieser Statements.

*STARTUP

Format/Länge: A8
 Inhalt modifizierbar Ja

Das Programm, dessen Name in dieser Systemvariable steht, wird immer dann ausgeführt, wenn Natural normalerweise die Kommandoingabeaufforderung (NEXT-Prompt bzw. Direktkommando-Zeile/-Fenster) anzeigen würde.

*STARTUP enthält den Namen des Programms, das in Natural Security als Startup-Transaktion im Security-Profil der betreffenden Library eingetragen ist (außer im Batch-Betrieb; vgl. *Natural Security*-Dokumentation).

Wenn keine Startup-Transaktion eingetragen ist oder wenn Natural Security nicht verwendet wird, enthält *STARTUP den Wert des Profilparameters STARTUP.

Über ein Natural-Programm können Sie der Systemvariablen *STARTUP einen Wert zuweisen, der dann ihren jeweils vorherigen Inhalt überschreibt.

Anmerkung:

Ein im Batch-Betrieb verwendetes Startup-Programm muss ein FETCH- oder STACK COMMAND-Statement enthalten; sonst kann ein Fehler NAT9969 auftreten.

Falls Sie die Kommandoingabeaufforderung durch Eingabe von %% (oder einem gleichwertigen Kommando) aufrufen — entweder in einer Nicht-Security-Umgebung oder in einer Security-Umgebung, in der der Kommando-Modus für die aktuelle Library nicht verboten ist — wird dadurch der Startup-Mechanismus deaktiviert. Um ihn wieder zu aktivieren, müssen Sie sich entweder erneut in die Library begeben oder ein Programm ausführen, das s.o. *STARTUP wieder einen Wert zuweist.

Wichtig:

Um das Programm zu deaktivieren, das in *STARTUP enthalten ist, setzen Sie z.B. mittels eines Statements RESET *STARTUP den Wert der Systemvariable auf "leer".

Unter Natural Security:

In einer Natural-Security-Umgebung, in der der Kommando-Modus für die aktuelle Library verboten ist, bewirkt %, dass das Programm, dessen Name in *STARTUP steht, aufgerufen wird.

Wenn ein durch eine Startup-Transaktion (*STARTUP) verursachter Natural-Laufzeitfehler auftritt, kann Naturals Fehlerverarbeitung dazu führen, dass die Startup-Transaktion erneut ausgeführt wird. Dieses würde zu einer Fehlerschleifensituation führen. Um eine solche Schleife zu verhindern, steht die allgemeine Option Logoff in error case if *STARTUP is active zu Verfügung. Siehe *Logoff in Error Case if *STARTUP is Active* in der *Natural Security*-Dokumentation.

*STEPLIB

Format/Länge: A8
 Inhalt modifizierbar Nein

Diese Systemvariable enthält den Namen der Steplib-Library, die mit der Natural-Library, in der der Benutzer gerade arbeitet, verknüpft ist.

Wenn Natural Security nicht installiert ist, enthält *STEPLIB den mit dem Profilparameter STEPLIB angegebenen Namen.

Wenn Natural Security installiert ist, kann die Steplib im Security-Profil der betreffenden Library eingetragen werden.

Anmerkung:

Bei der Steplib-Library wird immer davon ausgegangen, dass sie dieselbe Datenbank-ID und Dateinummer hat wie die aktuelle Library des Benutzers. Es wird davon ausgegangen, dass außer der Library SYSTEM die Libraries mit dem Namen SYSxxx sich in der Systemdatei FNAT und andere Libraries sich in FUSER befinden.

***SUBROUTINE**

Format/Länge: A32

Inhalt modifizierbar Nein

Diese Systemvariable enthält den Namen der externen Subroutine, die gerade ausgeführt wird. Der Inhalt von *SUBROUTINE steht stets in Großbuchstaben.

***THIS-OBJECT**

Format/Länge: HANDLE OF OBJECT

Inhalt modifizierbar Nein

Diese Systemvariable enthält eine Handle zum gerade aktiven Objekt. Das gerade aktive Objekt benutzt *THIS-OBJECT, um entweder seine eigenen Methoden auszuführen oder eine Referenz auf sich selbst an ein anderes Objekt zu übergeben.

*THIS-OBJECT enthält nur einen tatsächlichen Wert, wenn eine Methode ausgeführt wird. Sonst enthält es NULL-HANDLE.

***TYPE**

Format/Länge: A32

Inhalt modifizierbar Nein

Diese Systemvariable enthält den Typ des Natural-Objekts, das gerade ausgeführt wird.

Gültige Werte für *TYPE:

Wert	Objekttyp
PROGRAM	Programm
FUNCTION	Function
SUBPROGRAM	Subprogramm
SUBROUTINE	Subroutine
HELPROUTINE	Helproutine
MAP	Map
ADAPTER	Adapter
DIALOG	Dialog

*UBOUND

Format/Länge: I4

Inhalt modifizierbar Nein

UBOUND enthält die aktuelle Obergrenze (Indexwert) eines Arrays für die angegebene(n) Dimension(en) (1, 2 oder 3) oder für alle Dimensionen (Stern-Notation ()).

Syntax:

*UBOUND (<i>operand1</i> [, <i>dim</i>])

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	A	A U N P I F B D T L C G O	ja	nein

operand1 ist das Array, für das die Obergrenze angegeben wird. Die Index-Notation des Arrays ist optional. Als Index-Notation ist nur die *-Notation für den vollständigen Bereich für jede Dimension zulässig.

dim ist die Dimensionsnummer, für die die aktuelle Obergrenze zurückgegeben wird:

$dim = \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ * \end{array} \right\}$
--

Wenn keine Dimension angegeben wird, wird die Obergrenze der ersten Dimension zurückgegeben.

Wenn 1, 2 oder 3 angegeben wird, wird die Obergrenze der ersten, zweiten oder dritten Dimension zurückgegeben.

Wenn * angegeben wird, wird die Obergrenze aller definierten Dimensionen zurückgegeben, d.h.

1	im Falle eines eindimensionalen Arrays
2	im Falle eines zweidimensionalen Arrays
3	im Falle eines dreidimensionalen Arrays

Wenn ein X-Array nicht zugewiesen wird und die Obergrenze der angegebenen Dimension dieses X-Arrays die variable Indexgrenze ist, d.h., wenn sie durch einen Stern (*) in der Indexdefinition dargestellt wird, dann ist die Obergrenze dieses X-Arrays nicht definiert, und der Zugriff auf *UBOUND führt zu einem Laufzeitfehler. Um den Laufzeitfehler zu vermeiden, kann die Systemvariable *OCCURRENCE benutzt werden, um auf Null-Ausprägungen abzufragen:

```
DEFINE DATA LOCAL
  1 #XA(A5/1:*)
END-DEFINE
IF *OCCURRENCE (#XA) NE 0 AND *UBOUND(#XA) > 10
  THEN ...
```

Beispiele:

```
DEFINE DATA LOCAL
  1 #I (I4)
  1 #J (I4/1:3)
  1 #XA (A5/*:10,*:20)
END-DEFINE
#i := *UBOUND(#XA)          /* upper bound of 1st dimension is 10
#i := *UBOUND(#XA,1)       /* upper bound of 1st dimension is 10
#i := *UBOUND(#XA,2)       /* upper bound of 2nd dimension is 20
#j(1:2):= *UBOUND(#XA,*)   /* upper bound of all dimensions
                               /* (1st and 2nd)
                               /* #J(1) is 10 and #J(2) is 20
```

Siehe auch die Systemvariablen *LBOUND und *OCCURRENCE.