

# FORMAT

**FORMAT** [(*rep*)] *parameter* ...

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Parameter
- Beispiel

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

Verwandte Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET SAME | GET TRANSACTION | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Erstellen von Ausgabe-Reports*

---

## Funktion

Das Statement `FORMAT` dient dazu, Werte für Eingabe- und Ausgabeparameter festzusetzen.

Die Gültigkeit der mit einem `FORMAT`-Statement festgesetzten Werte hat zur Kompilierungszeit Vorrang vor den auf Session-Ebene mit einem `GLOBALS`-Kommando, `SET GLOBALS`-Statement oder vom Natural-Administrator gesetzten Parameterwerten.

Die mit dem `FORMAT`-Statement festgesetzten Werte können ihrerseits auf Statement- oder Elementebene (Feldebene) von den mit den Statements `DISPLAY`, `INPUT`, `PRINT`, `WRITE`, `WRITE TITLE` oder `WRITE TRAILER` gesetzten Parameterwerten überschrieben werden.

Die Einstellungen gelten Sie bis zum Ende des betreffenden Programms, oder bis sie mit einem weiteren `FORMAT`-Statement geändert werden.

Das `FORMAT`-Statement generiert keinen ausführbaren Code im Natural-Programm. Seine Ausführung hängt nicht vom logischen Ablauf des Programms ab. Es wird während der Kompilierung ausgewertet, um die Parameter für die Kompilierung der betroffenen `DISPLAY`-, `WRITE`-, `PRINT`- und `INPUT`-Statements zu setzen. Das `FORMAT`-Statement wirkt sich auf alle nachfolgenden `DISPLAY`-, `WRITE`-, `PRINT`- und `INPUT`-Statements aus.

# Syntax-Beschreibung

<i>(rep)</i>	<p><b>Report-Spezifikation:</b></p> <p>Mit der Notation (<i>rep</i>) kann ein bestimmter anderer Report angegeben werden, auf den sich das FORMAT-Statement beziehen soll.</p> <p>Es kann ein Wert von 0 bis 31 oder ein logischer Name, der mit einem DEFINE PRINTER-Statement zugewiesen wurde, angegeben werden.</p> <p>Falls nichts anderes angegeben wird, bezieht sich das FORMAT-Statement auf den ersten Report (Report 0).</p> <p>Informationen zum Steuern des Formats eines mit Natural erzeugten Ausgabe-Reports siehe <i>Steuerung der Ausgabe von Daten im Leitfaden zur Programmierung</i>.</p>
<i>parameter</i>	<p><b>Parameter:</b></p> <p>Die Parameter können in beliebiger Reihenfolge angegeben werden und müssen jeweils durch ein oder mehr Leerzeichen voneinander getrennt werden. Der Eintrag für einen Parameter darf nicht über das Ende einer Sourcecode-Zeile hinausgehen.</p> <p>Die hier gültigen feldsensitiven Parameter-Einstellungen kommen nur für Variablenfelder in Betracht, die in einem INPUT-, WRITE-, DISPLAY- oder PRINT-Statement des ausgewählten Reports verwendet werden. Sie haben aber keine Auswirkung auf in einem der erwähnten Statements verwendete Text-Konstanten.</p> <p>Siehe auch <i>Parameter</i> weiter unten.</p>

## Parameter

Die Beschreibungen der Parameter, die Sie beim FORMAT-Statement verwenden können, finden Sie in der *Parameter-Referenz*.

Parameter	Beschreibung
AD	Attribute Definition
AL	Alphanumeric Length for Output
CD	Color Definition
DF	Date Format
DL	Display Length for Output
EM	Edit Mask
ES	Empty Line Suppression
FC	Filler Character
FL	Floating Point Mantissa Length
GC	Filler Character for Group Heading

Parameter	Beschreibung
HC	Header Centering
HW	Heading Width
IC	Insertion Character
IP	Input Prompting Text
IS	Identical Suppress
KD	Key Definition
LC	Leading Characters
LS	Line Size
MC	Multiple-Value Field Count (Kann nur im Reporting Mode verwendet werden.)
MP	Maximum Number of Pages of a Report
MS	Manual Skip
NL	Numeric Length for Output
PC	Periodic Group Count (Kann nur im Reporting Mode verwendet werden.)
PM	Print Mode
PS	Page Size
SF	Spacing Factor
SG	Sign Position
TC	Trailing Characters
UC	Underlining Character
ZP	Zero Printing

Siehe auch *Unterstreichungszeichen für Überschriften – der UC-Parameter im Leitfaden zur Programmierung.*

## Beispiel

```

** Example 'FMTEX1': FORMAT
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 POST-CODE
  2 COUNTRY
END-DEFINE
*
FORMAT AL=7      /* Alpha-numeric field output length
FC=+            /* Filler character for field header
GC=*           /* Filler character for group header
HC=L           /* Header left justified
IC=<<          /* Insert characters
IS=ON         /* Identical suppress on

```

```

TC=>> /* Trailing character
UC== /* Underline character
ZP=OFF /* Zero print off
*
LIMIT 5
READ EMPLOY-VIEW BY NAME
  DISPLAY NOTITLE
    NAME 3X CITY 3X POST-CODE 3X COUNTRY
END-READ
*
END

```

Ausgabe des Programms FMTEX1:

```

NAME+++++++  CITY+++++++  POSTAL+++++  COUNTRY++++
              ADDRESS++++
=====
<<ABELLAN>>  <<MADRID >>  <<28014 >>  <<E >>
<<ACHIESO>>  <<DERBY >>  <<DE3 4TR>>  <<UK >>
<<ADAM >>    <<JOIGNY >>  <<89300 >>  <<F >>
<<ADKINSO>>  <<BROOKLY>>  <<11201 >>  <<USA>>
              <<BEVERLE>>  <<90211 >>

```