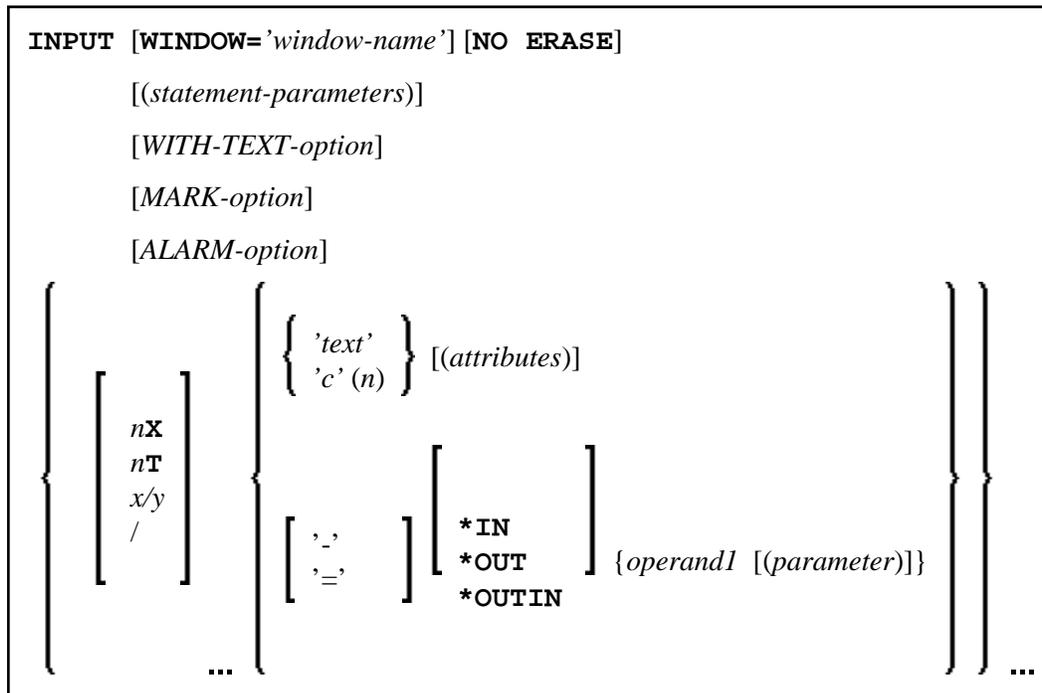


INPUT-Syntax 1 — Dynamisch generierter Eingabeschirm

Diese Form des INPUT-Statements wird dazu verwendet, entweder einen Eingabe-Schirm zu generieren oder ein Eingabedaten-Layout zu erstellen, das (auf Großrechnern) im Batch-Betrieb von einer sequentiellen Eingabedatei gelesen werden kann.



Dieses Kapitel behandelt folgende Themen:

- INPUT Syntax 1 — Beschreibung
- Beispiele — Verwendung von Syntax 1

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

INPUT Syntax 1 — Beschreibung

Operanden-Definitionstabelle:

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

Syntax-Element-Beschreibung:

<p>INPUT WINDOW='window-name'</p>	<p>Eingabe-Fenster:</p> <p>Mit der Option WINDOW=' <i>window-name</i> ' bewirken Sie, dass das INPUT-Statement für das angegebene Fenster (Window) ausgeführt werden soll. Das angegebene Fenster muss in einem DEFINE WINDOW-Statement definiert sein; siehe <i>Beispiel 2 — INPUT-Statement mit DEFINE WINDOW-Statement</i> weiter unten.</p> <p>Das angegebene Fenster ist nur für die Dauer des betreffenden INPUT-Statements aktiv und wird nach Ausführung des INPUT-Statements automatisch deaktiviert.</p> <p>Siehe auch die Statements DEFINE WINDOW und SET WINDOW.</p>
<p>NO ERASE</p>	<p>Überlagerte Anzeige:</p> <p>NO ERASE bewirkt, dass die vom INPUT-Statement ausgegebene Schirmanzeige eine bereits vorhandene Anzeige überlagern kann, ohne letztere zu löschen.</p> <p>Schirm bezieht sich in diesem Zusammenhang auf die logische Ausgabe und nicht auf den physischen Bildschirm.</p> <p>Alle ungeschützten Felder auf dem alten Schirm werden geschützt, so dass keine Eingaben mehr in sie gemacht werden können. Die alten Daten bleiben auf dem Schirm, bis der neue Schirm angezeigt wird. Überlagert ein Feld des neuen Schirms teilweise ein altes, so werden das Zeichen vor dem neuen Feld und das nächste Zeichen im alten Feld durch ein Leerzeichen ersetzt.</p>

<i>statement-parameters</i>	<p>Parameter auf Statement-/Feldebene:</p> <p>Unmittelbar nach dem Schlüsselwort <code>INPUT</code> oder nach einem der auszugebenden Felder können Sie in Klammern einen oder mehrere Session-Parameter setzen.</p> <p>Eine Liste der mit dem <code>INPUT</code>-Statement anzugebenden Parameter finden Sie im Abschnitt <i>Statement-Parameter</i> weiter unten.</p> <p>Diese Parameter haben dann für das jeweilige Statement oder Feld Gültigkeit statt der betreffenden mit einem <code>GLOBALS</code>-Kommando, <code>SET GLOBALS</code>- oder <code>FORMAT</code>-Statement gesetzten Parameter. Werden mehrere Parameter angegeben, müssen sie jeweils durch ein oder mehrere Leerzeichen voneinander getrennt werden. Die Angabe eines Parameters darf sich nicht über zwei Sourcecode-Zeilen erstrecken.</p> <p>Die hier gültigen Parameter-Einstellungen kommen nur für Variablen-Felder in Betracht, haben aber keine Auswirkungen auf Text-Konstanten. Wenn Sie Feldattribute für eine Text-Konstante setzen möchten, dann müssen Sie explizit für dieses Element gesetzt werden.</p> <p>Beispiel:</p> <pre> DEFINE DATA LOCAL 1 VARI (A4) INIT <'1234'> /* Output END-DEFINE /* Produced * /* ----- INPUT 'Text' VARI /* Text 1234 INPUT (AD=U) 'Text' VARI /* Text 1234 INPUT 'Text' (AD=U) VARI (AD=U) /* <u>Text</u> 1234 INPUT 'Text' (AD=U) VARI /* <u>Text</u> 1234 END </pre> <p>Beispiele für den Einsatz von Parametern auf Statement- und Element-Ebene finden Sie auf den folgenden Seiten.</p>
WITH TEXT-option	<p>Textausgabe in Meldungszeile:</p> <p>Diese Option dient dazu, Text anzugeben, der in der Meldungszeile ausgegeben werden soll (siehe den entsprechenden Abschnitt weiter unten).</p>
MARK-option	<p>Cursorposition im Feld:</p> <p>Siehe Abschnitt <i>MARK-Option</i> weiter unten.</p>
ALARM-option	<p>Warntonausgabe:</p> <p>Siehe Abschnitt <i>Alarm-Option</i> weiter unten.</p>
Other syntax elements (nX, nT, x/y, operand1 usw.)	<p>Sonstige Syntax-Elemente:</p> <p>Siehe Abschnitt <i>Feldpositionierung, Text, Attributzuweisung</i> weiter unten.</p>

Statement-Parameter

Parameter, die mit dem INPUT-Statement angegeben werden können:		Spezifikation
		S=auf Statement-Ebene
		E=auf Element-Ebene
AD	Attribute Definition	SE
AL	Alphanumeric Length for Output	SE
CD	Color Definition	SE
CV	Control Variable	SE
DF	Date Format	SE
DL	Display Length for Output	SE
DY	Dynamic Attributes	SE
EM	Edit Mask	SE
EMU	Unicode Edit Mask	E
FL	Floating Point Mantissa Length	SE
HE	Helproutine	SE
IP	Input Prompting Text	SE
LS	Line Size	S
MC	Multiple-Value Field Count	S
MS	Manual Skip	S
NL	Numeric Length for Output	SE
PC	Periodic Group Count	S
PM	Print Mode *	SE
PS	Page Size **	S
SB	Selection Box	E
SG	Sign Position	SE
ZP	Zero Printing	SE

* Der Session-Parameter PM darf nicht bei Textkonstanten angegeben werden.

** Die Einstellung des Session-Parameters PS bleibt unberücksichtigt, wenn die Anzahl der Ausprägungen eines Arrays den im Parameter PS angegebenen Wert überschreitet.

Beschreibungen der einzelnen Session-Parameter sind in der *Parameter-Referenz* enthalten.

WITH TEXT-Option

$[\text{WITH}] \text{ TEXT } \left\{ \begin{array}{l} * \text{ operand1} \\ \text{operand2} \end{array} \right\} [(\text{attributes})][, \text{operand3}] \dots 7$
--

Operanden-Definitionstabelle:

Operand	Mögliche Struktur		Mögliche Formate												Referenzierung erlaubt	Dynam. Definition				
<i>operand1</i>	C	S						N	P	I	B*								ja	ja
<i>operand2</i>	C	S				A													ja	ja
<i>operand3</i>	C	S				A	N	P	I	F	B	D	T	L					ja	ja

* Format B von *operand1* kann nur mit einer Länge kleiner gleich 4 benutzt werden.

Diese Option dient dazu, Text anzugeben, der in der Meldungszeile ausgegeben werden soll. In der Regel handelt es sich dabei um eine Meldung, was auf dem jeweiligen Schirm getan werden soll bzw. wie eine falsche Eingabe korrigiert werden soll.

Syntax-Element-Beschreibung:

<i>operand1</i>	<p>Meldungstext aus der Natural-Fehlermeldungsdatei:</p> <p>Als <i>operand1</i> geben Sie eine Natural-Fehlernummer an. Natural liest dann die entsprechende Fehlermeldung von der Natural- Fehlermeldungsdatei.</p> <p>Es können entweder benutzerdefinierte Meldungen oder Natural-Systemmeldungen gelesen werden.</p> <ul style="list-style-type: none"> • Wenn Sie einen positiven Wert von bis zu vier Ziffern (zum Beispiel: 0954) angeben, werden benutzerdefinierte Meldungen gelesen. • Wenn Sie einen negativen Wert von bis zu vier Ziffern (zum Beispiel –0954) angeben, werden Natural-Systemmeldungen gelesen. <p>Siehe auch REINPUT-Statement, <i>Beispiel 4 – WITH TEXT-Optionen</i>.</p> <p>Natural-Fehlermeldungsdateien werden mit der SYSERR-Utility erstellt und gepflegt.</p> <p>Die Natural-Fehlermeldungen sind in der Messages and Codes-Dokumentation enthalten.</p>
<i>operand2</i>	<p>Eigener Meldungstext:</p> <p>Als <i>operand2</i> geben Sie den Text an, der in der Meldungszeile ausgegeben werden soll.</p> <p>Siehe REINPUT-Statement, <i>Beispiel 4 – WITH TEXT-Optionen</i>.</p>
<i>attributes</i>	<p>Ausgabeattribute:</p> <p>Als <i>attributes</i> können Sie <i>operand1</i> oder <i>operand2</i> bestimmte Anzeige- und Farbattribute zuordnen.</p> <p>Diese Attribute und die benutzbare Syntax sind im Abschnitt <i>Ausgabeattribute</i> weiter unten beschrieben.</p>

operand3	<p>Dynamische Meldungstext-Komponente:</p> <p><i>operand3</i> kann in Form einer numerischen Konstanten oder Textkonstanten oder als Name einer Variablen angegeben werden.</p> <p>Der entweder mit <i>operand1</i> oder <i>operand2</i> angegebene Wert dient dazu, einen Teil der Meldung zu ersetzen und dynamisch zu generieren.</p> <p>Innerhalb der Fehlermeldung dient die Notation <i>:n</i>: zur Referenzierung von <i>operand3</i>, wobei <i>n</i> die Ausprägung (1 – 7) von <i>operand3</i> darstellt.</p> <p>Siehe REINPUT-Statement, <i>Beispiel 4 - WITH TEXT-Optionen</i>.</p> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Wird <i>operand3</i> mehrmals angegeben, müssen diese Operanden mit einem Komma voneinander getrennt werden. Falls das Komma als Dezimalzeichen verwendet wird (wie mit dem Session-Parameter DC definiert) und es sich bei <i>operand3</i> um numerische Konstanten handelt, setzen Sie Leerzeichen vor und nach dem Komma, damit es nicht als Dezimalkomma missinterpretiert wird. 2. Alternativ können mehrere Ausprägungen von <i>operand3</i> auch mit dem Input-Delimiterzeichen (wie mit dem Session-Parameter ID definiert) voneinander getrennt werden; dies geht jedoch nicht bei ID=/ (Schrägstrich), da der Schrägstrich in der Syntax des INPUT-Statements eine andere Bedeutung hat. 3. Nicht signifikante Nullen oder Leerzeichen werden aus dem Feldwert entfernt, bevor er in einer Meldung angezeigt wird.
-----------------	--

Ausgabeattribute

attributes gibt die für die Text-Anzeige zu benutzenden Ausgabe-Attribute an. Es gibt die folgenden Attribute:

<table> <tr> <td>{</td> <td>{</td> <td>AD=AD-value ...</td> <td>}</td> <td rowspan="3">}</td> </tr> <tr> <td></td> <td></td> <td>CD=CD-value ...</td> <td>}</td> </tr> <tr> <td></td> <td></td> <td>PM=PM-value ...</td> <td>}</td> </tr> <tr> <td></td> <td></td> <td>... ..</td> <td>}</td> <td></td> </tr> <tr> <td></td> <td>{</td> <td>AD-value ...</td> <td>}</td> <td rowspan="2">}</td> </tr> <tr> <td></td> <td></td> <td>CD-value ...</td> <td>}</td> </tr> <tr> <td></td> <td></td> <td>... ..</td> <td>}</td> <td></td> </tr> </table>	{	{	AD =AD-value ...	}	}			CD =CD-value ...	}			PM =PM-value ...	}			}			{	AD-value ...	}	}			CD-value ...	}			}	
{	{	AD =AD-value ...	}	}																												
		CD =CD-value ...	}																													
		PM =PM-value ...	}																													
		}																													
	{	AD-value ...	}	}																												
		CD-value ...	}																													
		}																													

Die möglichen Parameterwerte sind in der *Parameter-Referenz* aufgeführt.

- *AD* - Attribute Definition, Abschnitt *Feldanzeige*
- *CD* - Color Definition
- *PM* - Print Mode

Anmerkung:

Der Compiler akzeptiert tatsächlich mehr als einen Attributwert für ein Ausgabefeld. Zum Beispiel können Sie Folgendes angeben: AD=BDI. In solch einem Fall gilt allerdings nur der letzte Wert. Im hier gezeigten Beispiel erhält nur der Wert I Gültigkeit, und das Ausgabefeld wird intensiviert (hell hervorgehoben) angezeigt.

MARK-Option

Mit MARK POSITION können Sie bewirken, dass der Cursor in einem beliebigen, nicht geschützten Feld auf dem Bildschirm platziert wird. Zusätzlich können Sie die Position des Cursors innerhalb dieses Feldes bestimmen.

Standardmäßig, das heisst, wenn Sie die MARK POSITION weglassen, wird der Cursor an den Anfang des ersten, nicht geschützten Feldes positioniert.

```

MARK [POSITION operand4 [IN]] [FIELD] { operand1 }
                                     { *fieldname }
```

Operanden-Definitionstabelle:

Operand	Mögliche Struktur		Mögliche Formate										Referenzierung erlaubt	Dynam. Definition				
<i>operand4</i>	C	S															ja	ja
<i>operand1</i>	C	S	A														ja	ja

Syntax-Element-Beschreibung:

<i>operand1</i>	<p>Feldnummer für Referenzierung:</p> <p><i>operand1</i> gibt die Nummer des Feldes an, in dem der Cursor platziert werden soll:</p> <p>Jedem in einem INPUT-Statement angegebenen Feldattribut AD=A oder AD=M (d.h. ein ungeschütztes Feld) wird eine Feldreferenz- Nummer zugewiesen. Der Startwert ist 1.</p>
<i>*fieldname</i>	<p>Feldname für Referenzierung:</p> <p>Anstelle der Feldnummer können Sie den Feldnamen angeben, um den Cursor in ein bestimmtes Feld zu platzieren. Dazu verwenden Sie die Notation <i>*fieldname</i>.</p>
<i>operand4</i>	<p>Cursorposition im referenzierten Feld:</p> <p>Mit MARK POSITION können Sie den Cursor an eine bestimmte Stelle — die Sie mit <i>operand4</i> angeben — innerhalb des mit <i>operand1</i> oder <i>*fieldname</i> angegebenen Feldes platzieren.</p> <p><i>operand4</i> darf keine Dezimalstellen enthalten.</p>

Beispiele:

```

MARK #NUMBER           /* Field number
MARK 3                 /* Third map field
MARK *#FIELD1         /* Map field
MARK POSITION 3 IN #NUMBER /* Third character in field number

```

Siehe auch *Beispiel 3 — INPUT-Statement mit MARK POSITION-Option* weiter unten.

ALARM-Option

Diese Option bewirkt, dass der Warnton des Terminals ausgelöst wird, wenn das INPUT-Statement ausgeführt wird. Voraussetzung ist, dass die verwendete Terminal-Hardware dies ermöglicht.

[[AND] [SOUND] ALARM]

Text vor einem Feld

Ist der Session-Parameter IP nicht auf IP=OFF gesetzt, so wird der jeweilige Feldname vor dem Feldwert (Forms-Modus) oder als Aufforderung, ein Feld auszuwählen, (Keyword/Delimiter-Modus) angezeigt. Wenn Sie vor dem Feld ' - ' angeben, wird der Feldname nicht angezeigt; wenn Sie einen ' *text* ' angeben, wird dieser statt des Feldnamens angezeigt.

Feldpositionierung, Text, Attributzuweisung

$\left[\begin{array}{c} n\mathbf{X} \\ n\mathbf{T} \\ x/y \end{array} \right]$	$\left[\begin{array}{c} 'text' [(attributes)] \\ 'c' (n) [(attributes)] \\ ' - ' \\ '= ' \\ / \dots \end{array} \right]$	$\left[\begin{array}{c} *IN \\ *OUT \\ *OUTIN \end{array} \right] \{operand1 [(parameter(s))]\}$
---	---	---

Verschiedene Notationen stehen zur Feldpositionierung, Texterstellung und Attributzuweisung zur Verfügung.

<i>nX</i>	Leerstellen zwischen Spalten: Fügt zwischen den Feldern <i>n</i> Leerstellen ein.
<i>nT</i>	Tabulator für Feldausgabe: Setzt einen Tabulator, d.h. die Ausgabe eines Feldes beginnt ab Spalte <i>n</i> .
<i>x/y</i>	Ausgabeposition für nächstes Feld: Gibt das nachfolgende Element in Zeile <i>x</i> ab Spalte <i>y</i> aus. <i>y</i> darf nicht 0 sein. Rückwärtspositionierung in derselben Zeile ist nicht möglich.

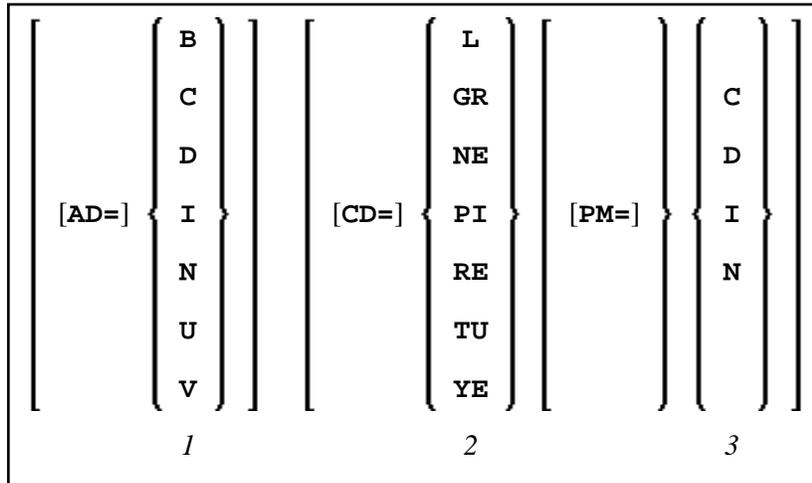
'text'	<p>Textausgabe:</p> <p>In Apostrophen angegebener Text wird schreibgeschützt ausgegeben. Siehe auch den Abschnitt <i>Text-Notation</i>, Unterabschnitt <i>Mit einem Statement zu benutzenden Text definieren</i>.</p>
'c' (n)	<p>Wiederholungszeichen:</p> <p>Wie 'text'. Aber das Zeichen <i>c</i> (character) wird <i>n</i>-mal ausgegeben. <i>n</i> darf 1 – 132 sein. Siehe auch den Abschnitt <i>Text-Notation</i>, Unterabschnitt <i>Vor einem Feldwert n mal anzuzeigendes Zeichen definieren</i>.</p>
attributes	<p>Ausgabeattribute:</p> <p>Dient zum Setzen der Anzeigeattribute. Siehe <i>Attribute</i> weiter unten.</p>
'_'	<p>Unterdrückung des Feldnamens:</p> <p>Die Notation ' _ ' unmittelbar vor einem Feld bewirkt, dass die Anzeige des Feldnamens vor dem Feld unterdrückt wird.</p> <p>Anmerkung: Eine Textkette vor einem Feld ersetzt den Feldnamen als Eingabeaufforderungstext.</p>
'='	<p>Feldüberschrift-Ausgabe:</p> <p>Die Notation ' = ' unmittelbar vor einem Feld bewirkt, dass unmittelbar vor dem Feldwert die Feldüberschrift ausgegeben wird.</p>

'/'	<p>Zeilenvorschub:</p> <p>Ein Schrägstrich ' / ' zwischen zwei Feldern oder Textelementen bewirkt einen Zeilenvorschub, d.h. das nachfolgende Element wird in der nächsten Zeile ausgegeben.</p> <p>Felder können als reine Eingabefelder (Session-Parameter AD=A), reine Ausgabefelder (AD=O) oder als modifizierbare Ausgabefelder (AD=M) definiert werden. Standardmäßig gilt AD=A. Felder, die mit AD=A oder AD=M definiert sind, werden als ungeschützte Felder ausgegeben, d.h. der Benutzer hat die Möglichkeit, Daten in diese Felder einzugeben.</p> <p>Bei TTY-Geräten beansprucht die Ausgabe modifizierbarer Felder die doppelte Feldgröße (einmal für Ausgabe und einmal für Eingabe), damit ein neuer Wert eingegeben werden kann. Bei TTY-Bildschirmen beginnt ein Eingabefeld (AD=A oder AD=M), dessen Wert bei der Eingabe nicht angezeigt wird (Feldattribut N), immer in einer neuen Zeile.</p> <p>Beispiel:</p> <pre>INPUT #A (AD=A) #B (AD=O) #C (AD=M)</pre> <p>#A ist ein Eingabefeld (ungeschützt), in das ein Wert eingegeben werden kann.</p> <p>#B ist ein Ausgabefeld (schreibgeschützt), dessen angezeigter Wert nicht überschrieben werden kann.</p> <p>#C ist ein Feld, dessen ausgegebener Wert verändert werden kann, indem er durch einen neuen Wert überschrieben wird.</p>
*IN, *OUT und *OUTIN	<p>Feldattribute:</p> <p>Entspricht den Feldattributen, die mit dem Session-Parameter AD gesetzt werden können: AD=A, AD=O bzw. AD=M.</p> <p>Anmerkung:</p> <p>Wenn eine nicht änderbare Systemvariable in einem INPUT-Statement benutzt wird, wird der Wert als ein reines Ausgabefeld AD=O oder *OUT-Attribut angezeigt.</p>

<i>operand1</i>	<p>Zu verwendende Felder:</p> <p>Als <i>operand1</i> geben Sie das zu verwendende Feld an. Sie können Datenbankfelder oder Benutzervariablen angeben.</p> <p>Natural überträgt den Inhalt eines Feldes direkt vom Datenbereich an das INPUT-Statement; eine MOVE-Operation ist hierzu nicht erforderlich.</p> <p>Ändert sich der Wert eines Datenbankfeldes aufgrund einer INPUT-Verarbeitung, so betrifft dies nur den Feldwert im Datenbereich. Um den Wert eines Feldes auf der Datenbank zu ändern, sind entsprechende UPDATE- bzw. STORE-Statements erforderlich.</p> <p>Wird in einem INPUT-Statement der Name einer Gruppe von Datenbankfeldern referenziert, so werden alle in der Gruppe enthaltenen Felder einzeln als Eingabefelder verwendet.</p> <p>Wird ein Bereich von Ausprägungen eines Arrays referenziert, so wird jede Ausprägung einzeln als Eingabefeld verarbeitet; allerdings wird nur der ersten Ausprägung ein Text oder der Feldname vorangestellt.</p> <p>Auf Großrechnern können keine Arrays mit Bereichen angegeben werden, die es ermöglichen, zur Ausführungszeit die Anzahl der Ausprägungen zu variieren.</p>
<i>parameter(s)</i>	<p>Statement-Parameter:</p> <p>Unmittelbar nach <i>operand1</i> können Sie in Klammern einen oder mehrere Parameter angeben (siehe Tabelle <i>Statement-Parameter</i> und folgendes Beispiel).</p> <p>Diese Parameter haben dann für das jeweilige Feld Gültigkeit statt der betreffenden mit einem GLOBALS-Kommando, SET GLOBALS- (im Reporting Mode) oder FORMAT-Statement gesetzten Parameter. Werden mehrere Parameter angegeben, müssen sie jeweils durch ein oder mehrere Leerzeichen voneinander getrennt werden. Die Angabe eines Parameters darf sich nicht über zwei Sourcecode-Zeilen erstrecken.</p> <p>Die hier gültigen Parameter-Einstellungen kommen nur für Variablenfelder in Betracht, haben aber keine Auswirkung auf Textkonstanten. Wenn Sie Feldattribute für eine Textkonstante setzen möchten, müssen Sie explizit für dieses Element gesetzt werden.</p> <p>Informationen zu den einzelnen Parametern entnehmen Sie der Tabelle im Abschnitt <i>Statement-Parameter</i>.</p> <p>Anmerkung: Ist für ein Datenbankfeld eine Editiermaske definiert, so wird der Editiermasken-Parameter EM dynamisch im entsprechenden DDM referenziert. Editiermasken können für Eingabe- wie für Ausgabefelder angegeben werden. Wird für ein Eingabefeld eine Editiermaske definiert, müssen die Daten in Einklang mit der Editiermasken-Definition eingegeben werden.</p>

Ausgabeattribute

Die folgenden Attribute können verwendet werden:



1. Anzeigeattribute - siehe Session-Parameter AD in der *Parameter-Referenz*.
2. Farbattribute - siehe Session-Parameter CD in der *Parameter-Referenz*.
3. Attribute für Druck-Modus - siehe Session-Parameter PM in der *Parameter-Referenz*.

Beispiele — Verwendung von Syntax 1

- Beispiel 1 — INPUT-Statement
- Beispiel 2 — INPUT-Statement mit DEFINE WINDOW-Statement
- Beispiel 3 — INPUT-Statement mit MARK POSITION-Option

Beispiel 1 — INPUT-Statement

```

** Example 'IPTEX1': INPUT
*****
DEFINE DATA LOCAL
1 #FNC (A1)
END-DEFINE
*
INPUT 10X 'SELECTION MENU FOR EMPLOYEES SYSTEM' /
      10X '-' (35) //
      10X 'ADD      (A)' /
      10X 'UPDATE   (U)' /
      10X 'DELETE   (D)' /
      10X 'STOP     (.)' //
      10X 'PLEASE ENTER FUNCTION: ' #FNC
*
DECIDE ON EVERY VALUE OF #FNC
  VALUE 'A' /* invoke the object containing the add function here
    WRITE 'Add function selected.'
  VALUE 'U' /* invoke the object containing the update function here
    WRITE 'Update function selected.'
  VALUE 'D' /* invoke the object containing the delete function here
    WRITE 'Delete function selected.'

```

```

VALUE '.'
STOP
NONE
REINPUT 'Please enter a valid function.' MARK *#FNC
END-DECIDE
*
END

```

Ausgabe des Programms IPTEX1:

```

SELECTION MENU FOR EMPLOYEES SYSTEM
-----

ADD      (A)
UPDATE   (U)
DELETE   (D)
STOP     (.)

PLEASE ENTER FUNCTION:

```

Beispiel 2 — INPUT-Statement mit DEFINE WINDOW-Statement

```

** Example 'INPEX1': INPUT (with DEFINE WINDOW statement)
*****
DEFINE DATA LOCAL
1 #STRING (A15)
END-DEFINE
*
DEFINE WINDOW WIND1
SIZE 10 * 40
BASE 5 / 10
FRAMED ON POSITION TEXT
*
INPUT WINDOW='WIND1'
'PLEASE ENTER HERE:' / #STRING
*
END

```

Ausgabe des Programms INPEX1:

```

+-----Top+
! PLEASE ENTER HERE:      !
! #STRING                  !
!                          !
!                          !
!                          !
!                          !
!                          !
!                          !
+-----Bottom+

```

Beispiel 3 — INPUT-Statement mit MARK POSITION-Option

```

** Example 'INPEX2': INPUT (with POSITION)
*****
DEFINE DATA LOCAL
1 #START (A30)
END-DEFINE
*
ASSIGN #START = 'EXAM_'
*

```

```
INPUT (AD=M) MARK POSITION 5 IN *#START
  / 'PLEASE COMPLETE START VALUE FOR SEARCH'
  / 5X #START
END
```

Ausgabe des Programms INPEX2:

```
PLEASE COMPLETE START VALUE FOR SEARCH
#START EXAM[ ]
```