

# INPUT-Syntax 2 — Verwendung einer vordefinierten Eingabemaske

Diese Form des INPUT-Statements wird benutzt, wenn bei der Eingabeverarbeitung eine mit dem Natural-Map-Editor erstellte Eingabemaske (Map) verwendet werden soll.

Hierbei gibt es zwei Möglichkeiten:

- das Programm enthält keine Parameterliste
- das Programm enthält eine Parameterliste (*operand1*).

```
INPUT [WINDOW= 'window-name'] [WITH-TEXT-option]
[MARK-option]
[ALARM-option]
[USING] MAP map-name [NO ERASE]
[
  operand1 ...
  NO PARAMETER
]
```

Dieses Kapitel behandelt folgende Themen:

- INPUT USING MAP ohne Parameterliste
- Im Programm definierte Eingabefelder
- INPUT Syntax 2 — Beschreibung
- INPUT-Statement unter Nicht-Screen-Modi
- Eingabedaten aus dem Natural-Stack

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

## INPUT USING MAP ohne Parameterliste

Die folgenden Anforderungen müssen bei der Benutzung von INPUT USING MAP ohne Parameterliste erfüllt werden:

- Der *map-name* muss als alphanumerische Konstante (maximal 8 Zeichen lang) angegeben werden.
- Die verwendete Map muss bereits erstellt sein, bevor das Programm, das sie referenziert, kompiliert werden kann.

- Die Feldnamen werden bei der Kompilierung dynamisch von der Map-Source übernommen. Die Feldnamen müssen in Map und Programm identisch sein.
- Zu diesem Zeitpunkt muss auf alle im INPUT-Statement referenzierten Felder zugegriffen werden können.
- Im Structured Mode müssen die Felder vorher definiert werden, und Datenbankfelder müssen durch Referenzierung der betreffenden Verarbeitungsschleife bzw. des betreffenden Views korrekt referenziert werden.
- Im Reporting Mode müssen Benutzervariablen in der Map neu definiert werden.
- Wird das Layout der Map verändert, müssen die die Map verwendenden Programme nicht neu katalogisiert werden. Wenn aber Array-Strukturen oder -Namen, Format/Länge von Feldern geändert oder Felder zur Map hinzugefügt bzw. aus ihr gelöscht werden, müssen die die Map verwendenden Programme neu katalogisiert werden.
- Die Map-Source muss bei der Programm-Kompilierung zur Verfügung stehen; sonst kann das INPUT USING MAP-Statement nicht kompiliert werden.

**Anmerkung:**

Wollen Sie das Programm kompilieren, obwohl noch keine Map zur Verfügung steht, geben Sie NO PARAMETER an: das INPUT USING MAP-Statement kann dann kompiliert werden, auch wenn die Map noch nicht vorhanden ist.

## Im Programm definierte Eingabefelder

Wenn Sie Namen der Eingabefelder (*operand1*) im Programm definieren, müssen diese nicht mit den für die Map verwendeten Feldnamen übereinstimmen.

Die Reihenfolge der Felder im Programm muss allerdings zur Reihenfolge der Felder in der Map passen. Hierbei ist zu beachten, dass der Map-Editor die in der Map definierten Felder in alphabetischer Reihenfolge der Feldnamen sortiert. Näheres hierzu finden Sie in der Map-Editor-Beschreibung in der *Editors*-Dokumentation.

Das Programm-Editor-Zeilenskommando `.I (mapname)` kann dazu verwendet werden, ein vollständiges INPUT USING MAP-Statement mit einer Parameterliste, die anhand der in der angegebenen Map definierten Felder generiert wird, zu erstellen.

Wird das Layout der Map verändert, muss das Programm nicht neu katalogisiert werden, es sei denn, in der Map werden Felder gelöscht, hinzugefügt oder umbenannt, Feldformate/-längen geändert oder Array-Strukturen modifiziert.

Bei der Ausführung prüft Natural, ob Format und Länge der Map-Felder in Einklang mit denen der Programm-Felder stehen. Ist dies nicht der Fall, wird eine entsprechende Fehlermeldung ausgegeben.

## INPUT Syntax 2 — Beschreibung

Operanden-Definitionstabelle:

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

Syntax-Element-Beschreibung:

<b>INPUT WINDOW='window-name'</b>	<p><b>Eingabe-Fenster:</b></p> <p>Siehe <i>Syntax 1</i> des INPUT-Statements.</p>
<b>WITH TEXT/MARK/ALARM-options</b>	<p><b>Optionen:</b></p> <p>Diese Optionen sind unter <i>Syntax 1</i> des INPUT-Statements beschrieben; siehe <i>WITH TEXT-Option</i>, <i>MARK-Option</i>, <i>ALARM-Option</i>.</p>
<b>USING MAP map-name</b>	<p><b>Name der Eingabemaske:</b></p> <p>Mit der USING MAP-Klausel wird eine Map-Definition aufgerufen, die vorher mit dem Map-Editor in einer Natural-Systemdatei gespeichert worden ist.</p> <p>Der <i>map-name</i> kann als 1 bis 8 Zeichen lange alphanumerische Konstante oder in Form einer Benutzervariablen angegeben werden. Wird eine Variable verwendet, muss diese vorher definiert worden sein. Der Map-Name darf ein Und-Zeichen (&amp;) enthalten; dies wird dann zur Ausführungszeit den aus einem Zeichen bestehenden Code ersetzt, der dem aktuellen Wert der Systemvariablen *LANGUAGE entspricht. Dadurch ist es möglich, verschiedensprachige Maps aufzurufen.</p> <p>Die Ausführung des INPUT-Statements löscht den bisherigen Bildschirminhalt, es sei denn, Sie verwenden eine NO ERASE-Klausel (siehe unten), wobei dann die Map den aktuellen Inhalt des Bildschirms überlagert.</p>
<b>NO ERASE</b>	<p><b>Überlagerte Anzeige:</b></p> <p>Siehe <i>Syntax 1</i> des INPUT-Statements.</p>
<b>operand1</b>	<p><b>Zu verwendende Felder:</b></p> <p>Es können Datenbankfelder oder Benutzervariablen angegeben werden, die jedoch alle vorher definiert worden sein müssen. Die Felder müssen in Anzahl, Reihenfolge Format und Länge und für (Arrays) in der Anzahl der Ausprägungen mit den referenzierten Map-Feldern übereinstimmen, da andernfalls ein Fehler generiert wird.</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>

# INPUT-Statement unter Nicht-Screen-Modi

Sie können den Eingabemodus mit dem Session-Parameter `IM` ändern.

## Forms-Modus

Im Forms-Modus (Profil/Session-Parameter `IM=F`) zeigt Natural den gesamten Ausgabertext des Map-Layouts Feld für Feld an, und zwar entsprechend der Positionierungsparameter. Dadurch kann der Benutzer Daten Feld für Feld eingeben. Wenn alle Daten eingegeben sind, wird eine Hardcopy-Ausgabe erzeugt, die genau dem Abbild auf dem Bildschirm entspricht.

Im Falle eines Fehlers kann der Benutzer das Terminalkommando `%R` eingeben und sodann das gesamte Formular erneut eingeben. Die Eingaben werden dann wie bei der ersten Ausführung des INPUT-Statements verarbeitet.

## Keyword/Delimiter-Modus

In diesem Modus (Profil/Session-Parameter `IM=D`) können Daten 1.) unter *Verwendung von Schlüsselwörtern* oder 2.) *in Abhängigkeit von der Position/Reihenfolge der Felder* eingegeben werden, wie auf der folgenden Seite beschrieben.

### 1. Dateneingabe unter Verwendung von Schlüsselwörtern

Der Text, der im Forms-Modus vor einem Feld ausgegeben würde, wird im Keyword/Delimiter-Modus als Schlüsselwort (Keyword) verwendet, um das betreffende Feld zu identifizieren. Nach dem Schlüsselwort muss das Input-Assign-Zeichen stehen (siehe Session-Parameter `IA`) und unmittelbar danach die Eingabedaten.

Die Eingabedaten werden durch das mit dem Session-Parameter `ID` definierte Input-Delimiterzeichen voneinander getrennt. Auf ein Input-Assign-Zeichen folgende Leerzeichen werden als Eingabedaten interpretiert. Nach dem letzten Datenelement ist kein Input-Delimiterzeichen erforderlich.

Die Reihenfolge der Eingabedaten ist beliebig. Wird ein nicht im INPUT-Statement definiertes Schlüsselwort eingegeben, gibt Natural eine entsprechende Fehlermeldung aus. Es brauchen nicht alle Felder mit Eingabedaten gefüllt zu werden. Felder, in die nichts eingegeben wird, werden auf Leerzeichen (alphanumerische Felder) bzw. Null (numerische und hexadezimale Felder) gesetzt.

### 2. Dateneingabe in Abhängigkeit von der Position/Reihenfolge der Felder

Es können auch lediglich Eingabedaten, jeweils durch das gültige Input-Delimiter-Zeichen (`ID`-Parameter) voneinander getrennt, eingegeben werden. In diesem Fall muss die Reihenfolge der Eingabedaten der Reihenfolge der Eingabefelder im INPUT-Statement entsprechen.

Beide Eingabearten (1 und 2) können auch miteinander kombiniert werden. Solange keine Schlüsselwörter angegeben werden, ist die im INPUT-Statement angegebene Reihenfolge der Felder maßgeblich; wird ein Schlüsselwort angegeben, so werden die auf das Schlüsselwort folgenden Eingabedaten dem durch das Schlüsselwort identifizierten Feld zugeordnet. Folgen hierauf wieder Eingabedaten ohne Schlüsselwörter, so werden diese Daten den Feldern zugeordnet, die im INPUT-Statement auf das mit Schlüsselwort identifizierte Feld folgen.

**Anmerkung:**

Ein Schlüsselwort und das dazugehörige Eingabefeld müssen sich auf derselben logischen Zeile befinden. Wenn die Länge beider zusammen die Zeilenlänge überschreitet, vergrößern Sie die Zeilenlänge (Session-Parameter LS) entsprechend, so dass Schlüsselwort und Feld in eine Zeile passen.

Im Keyword/Delimiter-Modus eingegebene Eingabedaten werden entsprechend der auch im Screen-Modus gültigen Regeln auf ihre Gültigkeit überprüft. Wird versucht, einen Wert einzugeben, der länger als das betreffende Feld ist, so gibt Natural eine entsprechende Fehlermeldung aus.

Werden Daten für ein INPUT-Statement im Keyword/Delimiter-Modus von einem gepufferten Terminal (Typ 3270) oder einer Workstation aus eingegeben, so müssen alle von einem bestimmten INPUT-Statement zu verarbeitenden Eingabedaten auf einem einzigen Schirm eingegeben werden; erst wenn alle Daten eingegeben sind, darf FREIG gedrückt werden.

## Eingabedaten aus dem Natural-Stack

Daten, die mittels eines FETCH-, RUN- oder STACK-Statements im Natural-Stack abgelegt wurden, werden bei der Ausführung des nächsten INPUT-Statements als Eingabedaten verarbeitet.

Das INPUT-Statement verarbeitet die Daten in dem oben beschriebenen Keyword/Delimiter-Modus.

Felder, für die keine Eingabedaten zur Verfügung stehen, werden je nach Format mit Leerzeichen bzw. Nullen gefüllt. Sind mehr Daten als Eingabefelder vorhanden, so werden überschüssige Eingabedaten ignoriert.

Wenn ein Feld mit Daten aus dem Stack gefüllt wird, gelten die Feldattribute nicht für die Daten.

Über die Natural-Systemvariable \*DATA können Sie erfahren, wieviele Datenelemente gegenwärtig im Natural-Stack zur Verfügung stehen.