

# Helproutinen

Helproutinen haben besondere Eigenschaften, die die Verarbeitung von Hilfe-Aufrufen erleichtern. Sie ermöglichen den Aufbau komplexer interaktiver Hilfe-Systeme. Helproutinen werden mit dem Programm-Editor erstellt.

Dieses Kapitel behandelt folgende Themen:

- Helproutinen aufrufen
  - Helproutinen spezifizieren
  - Programmierhinweise für Helproutinen
  - Parameter an Helproutinen übergeben
  - Gleichheitszeichen-Option
  - Array-Felder
  - Hilfe als eingeblendetes Fenster
- 

## Helproutinen aufrufen

In einer Natural-Anwendung fordern die Benutzer Hilfe an, indem Sie das Hilfe-Zeichen — standardmäßig ein Fragezeichen (?) — in einem Feld eingeben oder die Hilfe-Taste (normalerweise PF1) drücken. Dadurch rufen Sie eine Helproutine auf. Dabei gilt Folgendes:

- Das Hilfe-Zeichen ist nur einmal einzugeben.
- Das Hilfe-Zeichen muss das einzige geänderte Zeichen in der Eingabe-Zeichenkette sein.
- Das Hilfe-Zeichen muss das erste Zeichen in der Eingabe-Zeichenkette sein.

Auch in numerischen Feldern kann das Hilfe-Zeichen verwendet werden, wenn für das betreffende Feld eine Helproutine definiert ist. (Ungeachtet dessen überprüft Natural, ob es sich bei Eingaben in das Feld um gültige numerische Eingaben handelt.)

Die Hilfe-Taste kann - falls sie nicht bereits festgelegt ist - mit einem `SET KEY`-Statement bestimmt werden:

```
SET KEY PF1=HELP
```

Sie können eine Helproutine nur dann aufrufen, wenn sie in dem Programm oder der Map, von dem/der sie aufgerufen wird, spezifiziert ist.

## Helproutinen spezifizieren

Eine Helproutine kann folgendermaßen angegeben werden:

- in einem Programm: auf Statement-Ebene und auf Feldebene,
- in einer Map: auf Maskenebene und auf Feldebene.

Wenn Sie Hilfe für ein Feld anfordern, dem keine Helproutine zugeordnet ist, oder wenn Sie Hilfe anfordern, ohne dass ein bestimmtes Feld referenziert wird, dann wird die auf der jeweiligen Statement- oder Maskenebene angegebene Helproutine aufgerufen.

Eine Helproutine kann auch über ein `REINPUT USING HELP`-Statement aufgerufen werden — entweder im Programm selbst oder in einer Verarbeitungsregel (Processing Rule). Falls das `REINPUT USING HELP`-Statement eine `MARK`-Option enthält, wird die Helproutine für das markierte Feld aufgerufen. Ist keine feldspezifische Helproutine angegeben, wird die Helproutine für die betreffende Map aufgerufen.

Ein `REINPUT`-Statement in einer Helproutine kann sich nur auf `INPUT`-Statements innerhalb derselben Helproutine beziehen.

Es gibt zwei Möglichkeiten, eine Helproutine anzugeben, entweder mit dem Session-Parameter `HE` in einem `INPUT`-Statement:

```
INPUT (HE='HELP2112')
```

oder im "Extended Field Editing"-Bereich des Map-Editors (wie in *Maps erstellen* und in der *Editors*-Dokumentation beschrieben).

Der Name einer Helproutine kann entweder eine alphanumerische Konstante sein oder eine alphanumerische Variable, die den Namen enthält. Falls es sich um eine Konstante handelt, muss der Name der Helproutine in Apostrophen angegeben werden.

## Programmierhinweise für Helproutinen

Die Verarbeitung einer Helproutine kann mit einem `ESCAPE ROUTINE`-Statement gestoppt werden.

Bitte beachten Sie, dass ein `END OF TRANSACTION`- bzw. `BACKOUT TRANSACTION`-Statement in einer Helproutine die Transaktionslogik des Hauptprogramms beeinflusst.

## Parameter an Helproutinen übergeben

Eine Helproutine kann auf die gerade aktive Global Data Area zugreifen (aber keine eigene Global Data Area haben). Außerdem kann sie eine eigene Local Data Area haben.

Darüber hinaus können Daten von der/an die Helproutine über Parameter übergeben werden. Eine Helproutine kann bis zu 20 explizite Parameter und einen impliziten Parameter haben. Die expliziten Parameter werden mit dem Operanden `HE` hinter dem Namen der Helproutine ausgegeben:

```
HE='MYHELP','001'
```

Der implizite Parameter ist das Feld, für das die Helprountine aufgerufen wurde:

```
INPUT #A (A5) (HE='YOURHELP','001')
```

wobei 001 der explizite Parameter ist und #A der implizite Parameter, also das Feld.

Dies wird im DEFINE DATA PARAMETER-Statement der Helprountine wie folgt definiert:

```
DEFINE DATA PARAMETER
1 #PARM1 (A3)          /* explicit parameter
1 #PARM2 (A5)          /* implicit parameter
END-DEFINE
```

Bitte beachten Sie, dass im obigen Beispiel der implizite Parameter #PARM2 auch weggelassen werden kann. Der implizite Parameter dient dazu, auf das Feld zuzugreifen, für das Hilfe angefordert wurde, sowie dazu, Daten von der Helprountine an das Feld zu übergeben. Es wäre beispielsweise denkbar, als Helprountine ein Rechenprogramm zu haben und das Rechenergebnis an das Feld zu übergeben.

Wenn Hilfe angefordert wird, so wird die Helprountine aufgerufen, bevor Daten vom Bildschirm an die Datenbereiche des Programms weitergegeben werden. Das bedeutet, dass Helprountinen nicht auf Daten zugreifen können, die während derselben Bildschirmtransaktion eingegeben wurden.

Nach Beenden der Helprountine werden auf dem Bildschirm die Feldwerte aktualisiert, die durch die Helprountine verändert wurden — mit Ausnahme der Felder, deren Inhalte bereits vorher vom Benutzer verändert worden waren, aber inklusive des Feldes, für das Hilfe angefordert wurde.

*Ausnahme:* Wenn das Feld, für das Hilfe angefordert wurde, durch dynamische Attribute (Session-Parameter DY) in mehrere Teile unterteilt wird und der Teil, in den das Fragezeichen eingegeben wurde, sich *nach* einem vom Benutzer veränderten Teil befindet, wird eine durch die Helprountine bewirkte Veränderung des Feldinhalts nicht wirksam.

Kontrollvariablen werden nach Beenden der Helprountine nicht noch einmal ausgewertet, selbst wenn sie innerhalb der Helprountine verändert wurden.

## Gleichheitszeichen-Option

Es ist möglich, das Gleichheitszeichen (=) als expliziten Parameter anzugeben:

```
INPUT PERSONNEL-NUMBER (HE='HELPROUT',=)
```

Dieser Parameter wird dann als internes Feld (A65) verarbeitet, welches den Namen des Feldes bzw. der Map enthält. Die entsprechende Helprountine würde beispielsweise folgendermaßen beginnen:

```
DEFINE DATA PARAMETER
1 FNAME (A65)          /* contains 'PERSONNEL-NUMBER'
1 FVALUE (N8)          /* value of field (optional)
END-DEFINE
```

Diese Möglichkeit kann dazu eingesetzt werden, auf eine übergreifende Helproutine zuzugreifen, die den Feldnamen liest und feldspezifische Hilfe liefert, indem sie auf die Online-Dokumentation der Anwendung oder auf das Predict-Data-Dictionary zugreift.

## Array-Felder

Ist das Feld, für das Hilfe aufgerufen wird, Teil eines Arrays, dann werden seine Indexangaben als implizite Parameter (1–3 je nach Rang ungeachtet der expliziten Parameter) angegeben.

Diese Parameter haben das Format I2.

```
INPUT A(*,*) (HE='HELPROUT',=)
```

Die entsprechende Helproutine würde wie folgt beginnen:

```
DEFINE DATA PARAMETER
1 FNAME (A65) /* contains 'A'
1 FVALUE (N8) /* value of selected element
1 FINDEX1 (I2) /* 1st dimension index
1 FINDEX2 (I2) /* 2nd dimension index
END-DEFINE
...
```

## Hilfe als eingeblendetes Fenster

Sie können die Größe eines Hilfe-Schirms so festlegen, dass sie kleiner ist als die Größe Ihres Bildschirms. In diesem Fall wird der Hilfe-Schirm als eingerahmtes Fenster auf dem Bildschirm eingeblendet:

```
*****
                                PERSONNEL INFORMATION
PLEASE ENTER NAME: ?_____
PLEASE ENTER CITY: _____
                                +-----+
                                !
                                ! Type in the name of an      !
                                ! employee in the first         !
                                ! field and press ENTER.         !
                                ! You will then receive          !
                                ! a list of all employees         !
                                ! of that name.                   !
                                !
                                ! For a list of employees        !
                                ! of a certain name who          !
                                ! live in a certain city,         !
                                ! type in a name in the           !
                                ! first field and a city           !
                                ! in the second field             !
                                ! and press ENTER.                 !
*****! *****
                                +-----+
```

Innerhalb einer Helproutine haben Sie folgende Möglichkeiten, die Größe eines Fensters zu bestimmen:

- in einem `FORMAT`-Statement (z.B. um die Zeilen- und Seitenlänge anzugeben: `FORMAT PS=15 LS=30`)
- über ein `INPUT USING MAP`-Statement; in diesem Fall gilt die für die verwendete Map (in den **Map Settings**) festgelegte Größe
- durch ein `DEFINE WINDOW`-Statement; mit diesem Statement können Sie ein Fenster entweder explizit definieren oder dies Natural überlassen (Natural wird dann die Größe des Fensters je nach Inhalt festlegen).

Die Position des eingeblendeten Fensters wird automatisch in Abhängigkeit von der Position des Feldes, für das Hilfe angefordert wurde, bestimmt. Natural plaziert das Fenster automatisch möglichst nahe an das Feld, ohne es zu überdecken. Mit dem `DEFINE WINDOW`-Statement können Sie die Position des Fensters auch selbst bestimmen.

Weitere Informationen über Bildschirmfenster finden Sie beim `DEFINE WINDOW`-Statement in der *Statements*-Dokumentation und beim Terminalkommando `%W` in der *Terminalkommando*-Dokumentation.