SET KEY

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Tasten programm-sensitiv machen und deaktivieren
- Kommandos/Programme einer Taste zuweisen
- Eingabedaten einer Taste zuweisen (DATA)
- Tastenfunktion vorübergehend deaktivieren
- Helproutine zuweisen (HELP)
- Dynamische Funktionszuweisung (DYNAMIC)
- GUI-Element-Zuweisung deaktivieren (DISABLED)
- SET KEY-Statements auf verschiedenen Programmebenen
- Namen zuweisen
- Beispiel

Funktion

Das SET KEY-Statement dient dazu, den folgenden Tasten-Arten Funktionen zuzuweisen:

- Videoterminal PA-Tasten (Programmabruftasten)
- PF-Tasten (Programmfunktionstasten)
- CLEAR- bzw. LÖSCH-Taste.

Wird ein SET KEY-Statement ausgeführt, erhält Natural während der Programmausführung die Kontrolle über diese Tasten, und zwar unter Verwendung der Werte, die den Tasten zugewiesen sind.

Über die Natural-Systemvariable *PF-KEY kann ermittelt werden, welche Taste zuletzt gedrückt wurde.

Anmerkung:
Wird eine Taste gedrückt, der keine Funktion zugewiesen ist, wird der Benutzer entweder aufgefordert, eine andere Taste zu drücken, oder der Wert ENTR wird in die Natural-Systemvariable *PF-KEY gestellt, d.h. Natural reagiert, als ob die ENTER- bzw. FREIG-Taste gedrückt worden wäre (je nachdem, wie Ihr Natural-Administrator den Profilparameter IKEY gesetzt hat). Auf Großrechnern hängt die Verarbeitung von PA- und PF-Tasten auch davon ab, wie Ihr Natural-Administrator den Natural-Profilparameter KEY gesetzt hat.

Siehe auch *Verarbeitung aufgrund von Funktionstasten* im *Leitfaden zur Programmierung*.

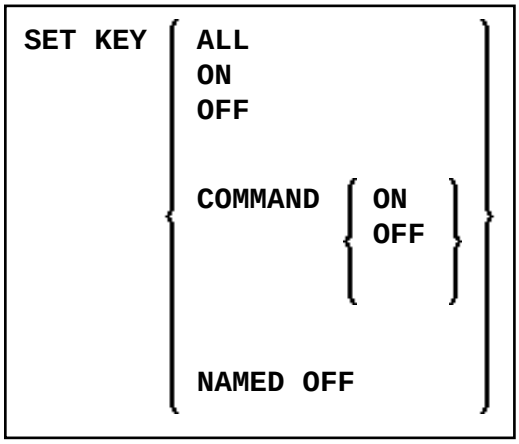
Programmierschnittstelle (API): `USR4005N`. Siehe auch *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

Syntax-Beschreibung

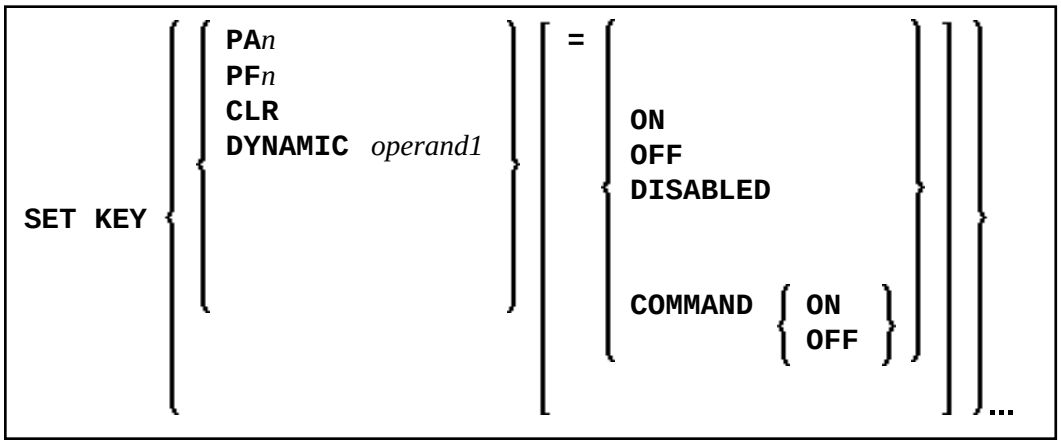
Mehrere Strukturen sind bei diesem Statement möglich.

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

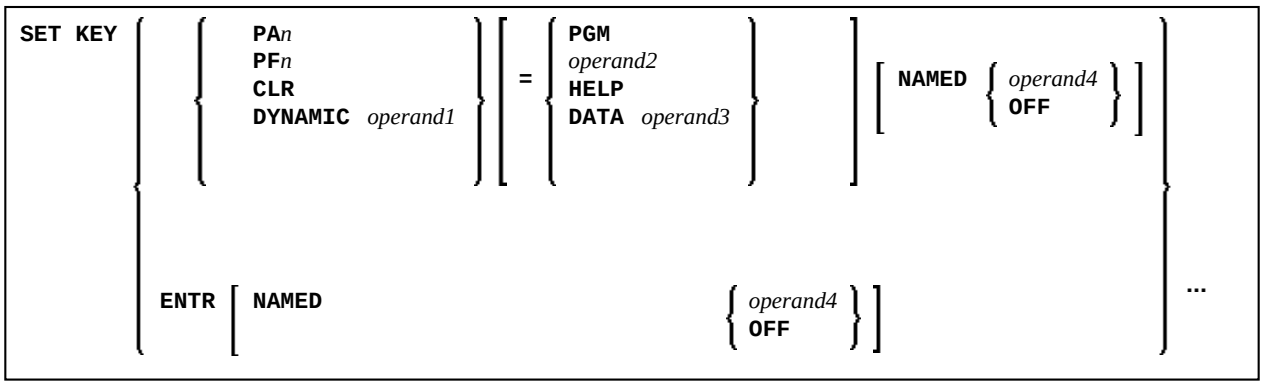
Syntax 1 – für alle Tasten



Syntax 2 – für einzelne Tasten



Syntax 3 – für einzelne Tasten



Operanden-Definitionstabelle:

Operand	Mögliche Struktur					Mögliche Formate											Referenzierung erlaubt	Dynam. Definition
operand1		S				A											ja	nein
operand2	C	S				A	U										ja	nein
operand3	C	S				A	U										ja	nein
operand4	C	S				A	U										ja	nein

Tasten programm-sensitiv machen und deaktivieren

Wenn eine Taste programm-sensitiv gemacht ist, kann Sie vom gerade aktiven Programm abgefragt werden. Das Drücken einer programm-sensitiven Taste hat die gleiche Wirkung wie das Drücken der FREIG-Taste. Alle auf dem Bildschirm eingegebenen Daten werden an das Programm übergeben.

Anmerkung:
Beim Drücken einer programm-sensitiven PA- oder CLEAR- bzw. LÖSCH-Taste werden keine Daten vom Bildschirm an das Programm übergeben.

Die Programm-Sensitivität ist nur während der Ausführung des aktuellen Programms wirksam. Vgl. Abschnitt *SET KEY-Statements auf verschiedenen Programmebenen*.

Beispiele:

SET KEY ALL	Dieses Statement bewirkt, dass alle Tasten programm-sensitiv gemacht werden. Alle Funktionszuweisungen zu Tasten werden damit überschrieben.
SET KEY PF2 **SET KEY PF2=PGM**	Jedes dieser beiden Statements bewirkt, dass PF2 programm-sensitiv gemacht wird.
SET KEY OFF	Dieses Statement deaktiviert alle Funktionstastenbelegungen. Die Natural-Systemvariable `*PF-KEY` enthält `ENTR`, nachdem `SET KEY OFF` ausgeführt worden ist.
SET KEY ON	Dieses Statement reaktiviert die Funktionen aller Tasten, denen eine Funktion zugewiesen war, und macht alle Tasten, die vor der Deaktivierung programm-sensitiv waren, wieder programm-sensitiv.
SET KEY PF2=OFF	Dieses Statement deaktiviert PF2. Nach Ausführung von `SET KEY PF2=OFF` enthält die Natural-Systemvariable `*PF-KEY ENTR`, wenn sie vorher `PF2` enthalten hatte.
SET KEY PF2=ON	Dieses Statement reaktiviert die Funktion, die PF2 zugewiesen wurde, bevor die Taste deaktiviert oder programm-sensitiv gemacht wurde. War PF2 keine Funktion zugewiesen, wird die Taste wieder programm-sensitiv gemacht.

Programm-Sensivität einer Taste und Inhalt von *PF-KEY

Das folgende Beispiel zeigt die Beziehung zwischen der Programm-Sensitivität einer Taste und dem Inhalt der Systemvariablen `*PF-KEY`.

Gehen wir davon aus, dass PF2 mittels `SET KEY PF2=PGM` programm-sensitiv gemacht worden ist und dass ein `INPUT`-Statement im Anschluss daran ausgeführt wird. Die folgende Tabelle zeigt, wie Aktionen durch den Benutzer und ausgeführte Natural-Statements den Inhalt von `*PF-KEY` beeinflussen.

Reihenfolge	Aktion durch Benutzer / Natural-Statement ausgeführt	Inhalt der Systemvariablen *PF-KEY
1	Benutzer drückt PF2.	`PF2`
2	`SET KEY OFF`	`ENTR`
3	`SET KEY ON`	`PF2`
4	`SET KEY PF2=OFF`	`ENTR`
5	`SET KEY PF2=ON`	`PF2`
6	`SET KEY PF3=OFF`	`PF2`

Kommandos/Programme einer Taste zuweisen

Sie können einer Taste ein Kommando oder einen Programmnamen zuweisen. Wenn die Taste gedrückt wird, wird das aktuelle Programm unterbrochen und das der Taste zugewiesene Kommando/Programm über den Natural-Stack aufgerufen. Wenn Sie einer Taste ein Kommando/Programm zuweisen, können Sie auch Parameter an das Kommando/Programm übergeben (siehe drittes Beispiel unten).

Sie können einer Taste auch ein Terminalkommando zuweisen. Wenn die Taste gedrückt wird, wird das ihr zugewiesene Terminalkommando ausgeführt.

Wenn *operand2* als Konstante angegeben wird, muss diese in Apostrophen stehen.

Beispiele:

SET KEY PF4='SAVE'	Das Kommando SAVE wird PF4 zugewiesen.
SET KEY PF4=#XYX	Der in der Variablen #XYZ enthaltene Wert wird PF4 zugewiesen.
SET KEY PF6='LIST MAP *'	Das Kommando LIST, einschließlich der LIST-Kommando-Parameter MAP und * wird PF6 zugewiesen.
SET KEY PF2='%%'	Das Terminalkommando %% wird PF2 zugewiesen.

Die Zuweisungen sind solange wirksam, bis sie mit einem anderen SET KEY-Statement überschrieben werden, der Benutzer in eine andere Anwendung wechselt oder die Natural-Session beendet wird. Vgl. Abschnitt *SET KEY-Statements auf verschiedenen Programmebenen*.

Anmerkung:
Bevor ein über eine Taste aufgerufenes Programm ausgeführt wird, führt Natural intern ein BACKOUT TRANSACTION-Statement aus.

Eingabedaten einer Taste zuweisen (DATA)

Sie können einer Taste eine Datenkette (*operand3*) zuweisen. Wenn die Taste gedrückt wird, werden die Daten in das Eingabefeld übertragen, in dem der Cursor gerade steht, und werden dann an das ausführende Programm übergeben (als ob FREIG gedrückt worden wäre).

Wenn *operand3* als Konstante angegeben wird, muss diese in Apostrophen stehen.

Beispiel:

```
SET KEY PF12=DATA 'YES'
```

Für eine DATA-Zuweisung gilt dasselbe wie für eine Kommando-Zuweisung: sie ist solange wirksam, bis sie mit einem anderen SET KEY-Statement überschrieben wird, der Benutzer in eine andere Anwendung wechselt oder die Natural-Session beendet wird. Vgl. Abschnitt *SET KEY-Statements auf verschiedenen Programmebenen*.

Tastenfunktion vorübergehend deaktivieren

Mit COMMAND OFF können Sie eine Funktion (Kommando, Programm oder Daten), die einer Taste zugewiesen ist, vorübergehend außer Kraft setzen. Wenn die Taste vor der Zuweisung der Funktion programm-sensitiv war, macht COMMAND OFF sie wieder programm-sensitiv.

Mit einem anschließenden COMMAND ON können Sie die zugewiesene Funktion später wieder aktivieren.

Beispiele:

SET KEY PF4=COMMAND OFF	Die der Taste PF4 zugewiesene Funktion wird vorübergehend deaktiviert; war PF4 vor der Zuweisung der Funktion programm-sensitiv, wird die Taste jetzt wieder programm-sensitiv.
SET KEY PF4=COMMAND ON	Die der Taste PF4 zugewiesene Funktion wird wieder reaktiviert.
SET KEY COMMAND OFF	Die allen Tasten zugewiesenen Funktionen werden vorübergehend deaktiviert; waren Tasten vor der Zuweisung der jeweiligen Funktion programm-sensitiv, werden sie jetzt wieder programm-sensitiv.
SET KEY COMMAND ON	Die allen Tasten zugewiesenen Funktionen werden wieder reaktiviert.

Mit `SET KEY PF`*nn*`=''` können Sie die einer Taste zugewiesene Funktion löschen und gleichzeitig die Programm-Sensitivität der Taste deaktivieren.

Helproutine zuweisen (HELP)

Sie können einer Taste `HELP` zuweisen. Wenn die Taste gedrückt wird, wird die Helproutine aufgerufen, die dem Feld, in dem der Cursor gerade steht, zugewiesen ist.

Der Effekt ist derselbe wie beim Aufrufen von Hilfe durch Eingabe des Hilfe-Zeichens in das Feld. (Das Hilfe-Zeichen — standardmäßig ein Fragezeichen (?) — wird vom Natural-Administrator mit dem Natural-Profilparameter `HI` bestimmt.)

Beispiel:

```
SET KEY PF1=HELP
```

Für die `HELP`-Zuweisung gilt dasselbe wie für Programm-Sensitivität: sie gilt nur für die Ausführung des aktuellen Programms. Vgl. Abschnitt *SET KEY-Statements auf verschiedenen Programmebenen*.

Dynamische Funktionszuweisung (DYNAMIC)

Anstatt im `SET KEY`-Statement eine bestimmte Taste anzugeben, können Sie die Option `DYNAMIC` mit Angabe einer Variablen (*operand1*) verwenden und dieser Variablen im Programm einen Wert (PF*n*, PA*n*, `CLR`) zuweisen. Dadurch haben Sie die Möglichkeit, eine Funktion anzugeben und es von der Programmlogik abhängig zu machen, welcher Taste diese Funktion zugewiesen wird.

Beispiel:

```
...
IF ...
   MOVE 'PF4' TO #KEY
ELSE
   MOVE 'PF7' TO #KEY
END-IF
...
SET KEY DYNAMIC #KEY = 'SAVE'
...
```

GUI-Element-Zuweisung deaktivieren (DISABLED)

Elemente graphischer Benutzeroberflächen (GUI) wie z.B. Drucktasten, Menüs und Bitmaps sind als PF-Tasten implementiert. Mit der Option DISABLED können Sie das einer PF-Taste zugewiesene GUI-Element deaktivieren. Das betreffende GUI-Element wird dann grau dargestellt und kann nicht ausgewählt werden.

Mit SET KEY PF*nn*=ON können Sie SET KEY PF*nn*=DISABLED wieder rückgängig machen.

Beispiel:

SET KEY PF10=DISABLED	Deaktiviert das PF10 zugewiesene GUI-Element.

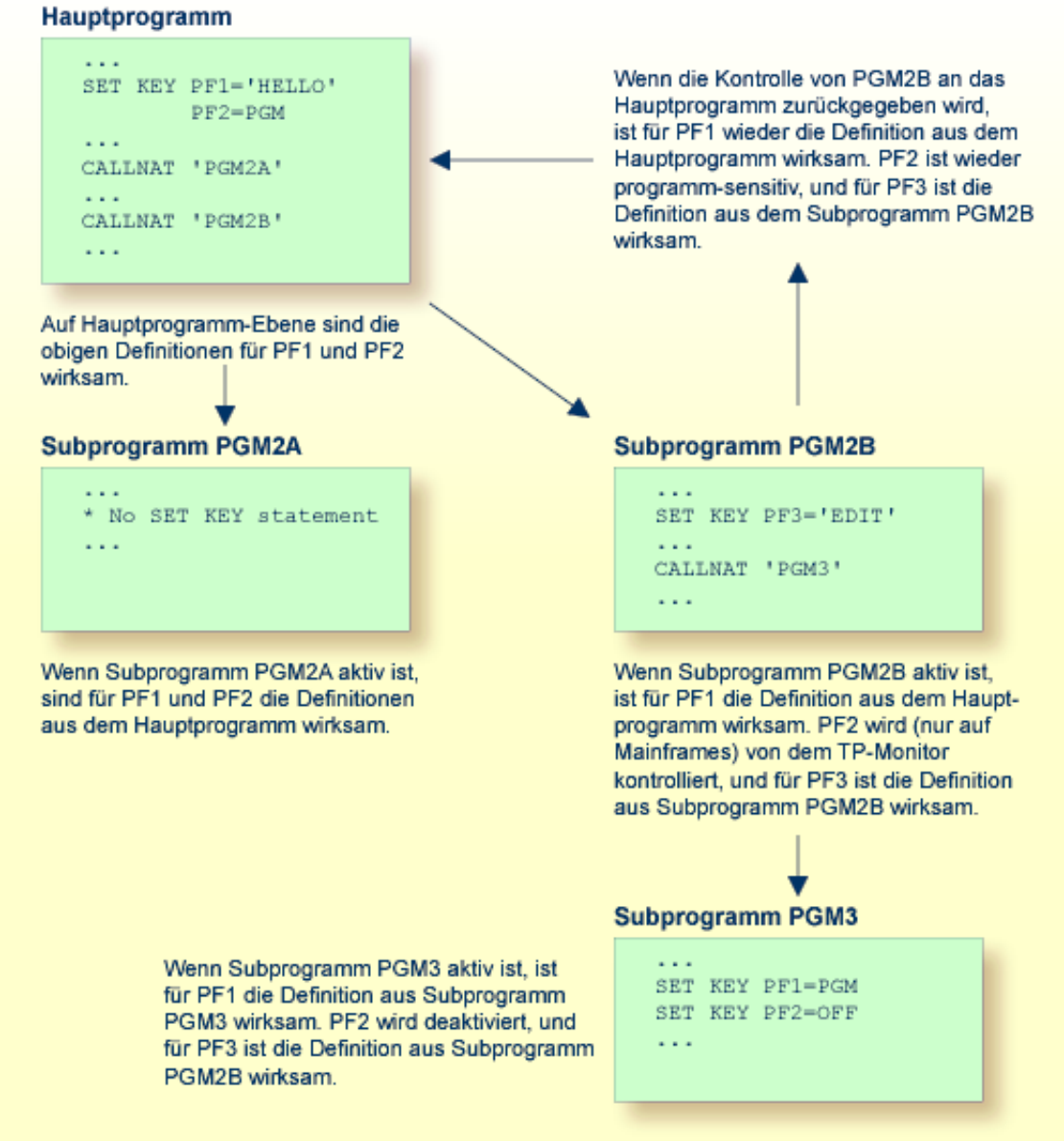
Die DISABLED-Option kann nur in Processing Rules verwendet werden.

SET KEY-Statements auf verschiedenen Programmebenen

Wenn eine Anwendung SET KEY-Statements auf verschiedenen Ebenen enthält, gilt folgendes:

- Wenn Tasten programm-sensitiv gemacht werden, gilt die Programm-Sensitivität auch für alle (aufgerufenen) Programme (bzw. Unterprogramme) auf untergeordneten Ebenen, es sei denn, diese Programme enthalten eigene SET KEY-Statements. Wenn die Kontrolle an ein übergeordnetes Programm zurückgegeben wird, gelten wieder die auf dieser übergeordneten Ebene gemachten SET KEY-Zuweisungen.
- Für Tasten, die als HELP-Tasten definiert sind, gilt das gleiche wie für programm-sensitive Tasten.
- Wenn einer Taste eine Funktion (Programm, Kommando, Terminalkommando oder Daten) zugewiesen wird, gilt diese Zuweisung für alle über- und untergeordneten Ebenen — ganz gleich, auf welcher Ebene die Zuweisung erfolgt —, und zwar solange, bis der Taste eine andere Funktion zugewiesen wird oder sie programm-sensitiv gemacht wird, oder bis der Benutzer in eine andere Anwendung wechselt oder die Natural-Session beendet wird.

Beispiel für SET KEY-Statements auf verschiedenen Programmebenen



Namen zuweisen

Mit der NAMED-Klausel können Sie einer Taste einen Namen (*operand4*) zuweisen. Dieser Name wird dann in der PF-Tastenleiste auf dem Bildschirm angezeigt, was es dem Benutzer ermöglicht, die den Tasten zugewiesenen Funktionen zu erkennen:

```
                ?   Help
              .   Exit
             ---- ------------------------------------------------
      Code ..: ?   Library ..: *_______
                   Object ...: *_____________________________________
                   DBID .....: 0__    FILENR ...: 0__


  Command ===>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help        Exit  Last        Flip                                 Canc
```

Die Anzeige der PF-Tastenleiste wird mit dem Session-Parameter KD aktiviert. Die Form der Anzeige der PF-Tastenleiste können Sie mit dem Terminalkommando %Y beeinflussen.

Der Name, den Sie einer Taste zuweisen, darf bis zu 10 Stellen lang sein. Im normalen tabellarischen PF-Tastenleistenformat (%YN) werden jeweils nur die ersten 5 Stellen angezeigt.

Wenn Sie *operand4* als Konstante angeben, muss diese in Apostrophen stehen (siehe Beispiele weiter unten).

Sie können einer Taste keinen Namen geben, ohne ihr eine Funktion zuzuweisen oder sie programm-sensitiv zu machen. Der ENTER-Taste können Sie allerdings nur einen Namen (z.B. EINGABE) zuweisen, aber keine Funktion.

Mit NAMED OFF löschen Sie den Namen einer programm-sensitiven Taste.

Beispiele:

SET KEY ENTR NAMED 'EXEC'	Der ENTER-Taste wird der Name EXEC zugewiesen.
SET KEY PF3 NAMED 'EXIT'	PF3 wird programm-sensitiv gemacht und erhält den Namen EXIT.
SET KEY PF3 NAMED OFF	PF3 wird programm-sensitiv gemacht, und der PF3 zugewiesene Name wird gelöscht.
SET KEY NAMED OFF	Alle programm-sensitiven Tasten zugewiesenen Namen werden gelöscht.
SET KEY PF4='AP1' NAMED 'APPL1'	PF4 werden das Programm AP1 und der Name APPL1 zugewiesen.

Wenn Sie normales tabellarisches PF-Tastenleistenformat (%YN) verwenden, gilt folgendes:

- Wenn Sie einer Taste ein Kommando/Programm zuweisen und die NAMED-Klausel weglassen, wird der Name dieses Kommandos/Programms in der PF-Tastenleiste angezeigt; ist der Name länger als 5 Stellen, wird stattdessen CMND angezeigt.
- Wenn Sie einer Taste Eingabedaten zuweisen und die NAMED-Klausel weglassen, wird als Name DATA in der PF-Tastenleiste angezeigt.
- Wenn Sie einer PF-Taste (per NAMED-Klausel) einen Namen im Unicode-Format zuweisen, kann es sein, dass der Name nicht richtig unter den entsprechenden Überschriften positioniert wird. Dieses Problem kann allerdings nur bei Verwendung des *Natural Web I/O Interface* und nur bei Zeichen

vom Typ "wide" auftreten. In diesem Fall wird empfohlen, das sequenzielle PF-Tastenformat zu verwenden (%YS oder %YP).

Wenn Sie sequentielles PF-Tastenleistenformat (%YS oder %YP) verwenden, werden nur die Tasten in der PF-Tastenleiste angezeigt, denen sie Namen zugewiesen haben; d.h. wenn Sie einer Taste ein Kommando/Programm/Daten zuweisen und die NAMED-Klausel weglassen, erscheint die Taste nicht in der PF-Tastenleiste.

Siehe auch *Verarbeitung aufgrund der Namen von Funktionstasten* im *Leitfaden zur Programmierung*.

Beispiel

```
** Example 'SKYEX1': SET KEY
************************************************************************
DEFINE DATA LOCAL
1 #PF4 (A56)
END-DEFINE
*
MOVE 'LIST VIEW' TO #PF4
*
SET KEY PF1 PF2
SET KEY PF3 = 'MENU'
        PF4 = #PF4
        PF5 = 'LIST VIEW EMPLOYEES' NAMED 'Empl'
*
FORMAT KD=ON
INPUT ////
      10X 'The following function keys are assigned:' //
      10X 'PF1: Funktion for PF1      ' /
      10X 'PF2: Funktion for PF2      ' /
      10X 'PF3: Return to MENU program' /
      10X 'PF4: LIST VIEW             ' /
      10X 'PF5: LIST VIEW EMPLOYEES   ' ///
*
IF *PF-KEY = 'PF1'
  WRITE 'Funktion for PF1 executed.'
END-IF
IF *PF-KEY = 'PF2'
  WRITE 'Funktion for PF2 executed.'
END-IF
*
END
```

Ausgabe des Programms SKYEX1:

```
The following function keys are assigned:

PF1: Funktion for PF1
PF2: Funktion for PF2
PF3: Return to MENU program
PF4: LIST VIEW
PF5: LIST VIEW EMPLOYEES
```