

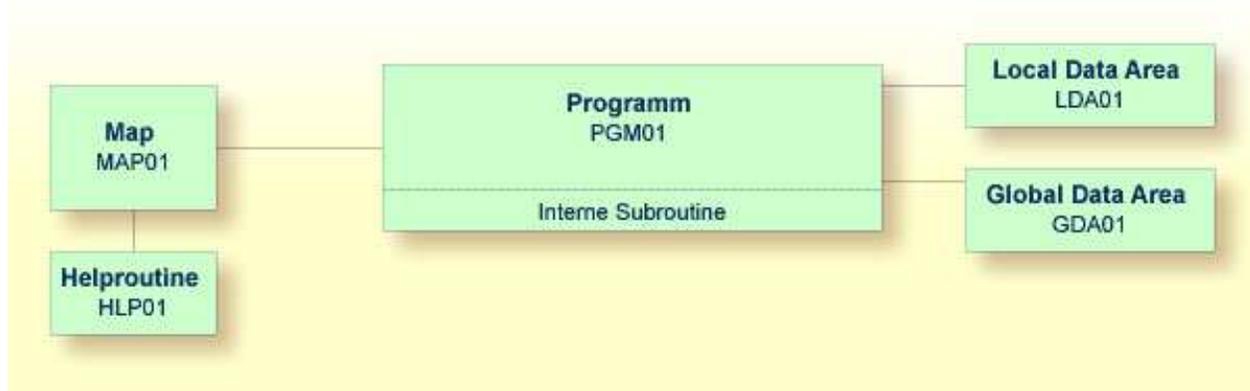
Global Data Areas

Die Daten in einer Global Data Area (GDA) können von mehreren Programmen, externen Subroutinen und Helprountinen benutzt werden.

Jede Änderung eines Datenelements in einer Global Data Area hat Auswirkungen auf alle Natural-Objekte, die diese Global Data Area referenzieren. Aus diesem Grund müssen Sie nach der Änderung einer Global Data Area alle zuvor erstellten Natural-Objekte, die diese Global Data Area referenzieren, noch einmal mit STOW speichern. Die Reihenfolge, in der die Objekte mit STOW gespeichert werden, ist wichtig. Sie müssen zuerst die Global Data Area mit STOW speichern und danach das Programm. Wenn Sie das Programm zuerst mit STOW speichern wollen und dann erst die Global Data Area, dann kann das Programm nicht gespeichert werden, weil die neuen Elemente in der Global Data Area nicht gefunden werden.

Sie werden jetzt eine Global Data Area erstellen, die von Ihrem Programm benutzt wird und auch von einer externen Subroutine, die Sie später erstellen werden. Als Basis für Ihre Global Data Area übernehmen Sie einige Informationen aus der Local Data Area, die Sie soeben erstellt haben.

Wenn Sie mit den Übungen in diesem Kapitel fertig sind, wird Ihre Beispielanwendung aus den folgenden Modulen bestehen:



Dieses Kapitel enthält die folgenden Übungen:

- Global Data Area mit Hilfe einer bestehenden Local Data Area erstellen
- Local Data Area anpassen
- Global Data Area aus dem Programm aufrufen

Global Data Area mit Hilfe einer bestehenden Local Data Area erstellen

Sie können eine neue Data Area erstellen, indem Sie eine bestehende Data Area editieren und sie dann unter einem anderen Namen und einem anderen Typ abspeichern. Die ursprüngliche Data Area bleibt hierbei unverändert und die neue Data Area kann sofort editiert werden. Da die Felder #NAME-START und #NAME-END nicht in der Global Data Area benötigt werden, werden Sie sie entfernen.

Anmerkung:

Es ist auch möglich, eine Global Data Area zu erstellen, indem Sie aus dem Menü **Object** den Befehl **New > Global Data Area** wählen.

▶ **Global Data Area erstellen**

1. Kehren Sie zur Local Data Area zurück.
2. Wählen Sie aus dem Menü **Object** den Befehl **Save As**.

Das Dialogfeld **Save As** erscheint.

3. Geben Sie "GDA01" als Name für die Global Data Area ein.
4. Achten Sie darauf, dass im Library-Workspace die Library markiert ist, die auch Ihr Programm enthält (d.h. der Knoten **TUTORIAL**).
5. Wählen Sie aus dem Dropdown-Listenfeld **Type** den Eintrag **Global**.
6. Wählen Sie die Befehlsschaltfläche **OK**.

Der neue Name und Typ werden nun in der Titelleiste des Editorfensters angezeigt. Im Library-Workspace wird die neue Global Data Area im Knoten **Global Data Areas** angezeigt.

7. Drücken Sie STRG und markieren Sie die folgenden Felder:

```
#NAME-START
#NAME-END
```

8. Wählen Sie aus dem Kontextmenü den Befehl **Delete**.

Oder:

Drücken Sie ENTF.

Die Global Data Area sollte nun folgendermaßen aussehen:

GDA01 [TUTORIAL] - Global Data Area *						
	Type	Level	Name	Format	Length	Handle
		1	#MARK	A	1	
	V	1	EMPLOYEES-VIEW			
		2	PERSONNEL-ID	A	8	
	G	2	FULL-NAME			
		3	NAME	A	20	
		2	DEPT	A	6	
	G	2	LEAVE-DATA			
		3	LEAVE-DUE	N	2.0	

- Speichern Sie die Global Data Area mit STOW.

Die Global Data Area kann jetzt von Ihrem Programm gefunden werden, und auch von der externen Subroutine, die Sie später anlegen werden.

- Schließen Sie das Editorfenster für die Global Data Area.

Local Data Area anpassen

Die Felder, die jetzt in der Global Data Area enthalten sind, werden nicht mehr in der Local Data Area benötigt. Deshalb werden Sie jetzt alle Felder außer #NAME-START und #NAME-END aus der Local Data Area entfernen.

► Felder entfernen

- Markieren Sie die Local Data Area LDA01 im Library-Workspace und wählen Sie aus dem Kontextmenü den Befehl **Open**.

Oder:

Klicken Sie die Local Data Area LDA01 im Library-Workspace doppelt an.

- Markieren Sie im daraufhin erscheinenden Data-Area-Editor alle Felder außer #NAME-START und #NAME-END.

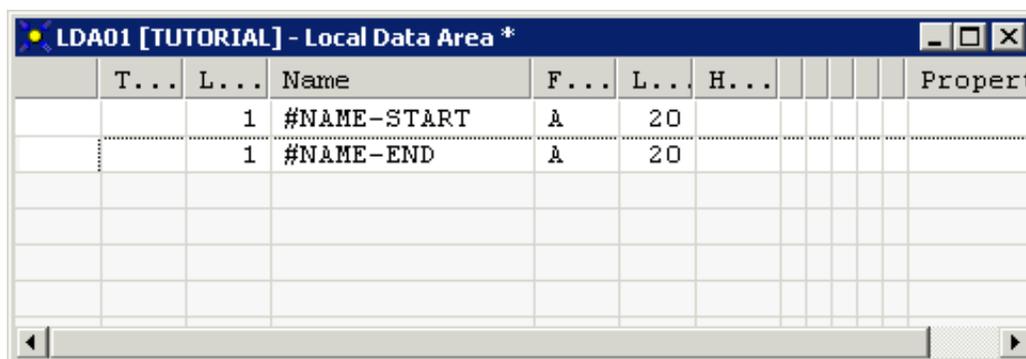
- Wählen Sie aus dem Kontextmenü den Befehl **Delete**.

Oder:

Drücken Sie ENTF.

- Speichern Sie die geänderte Local Data Area mit STOW.

Die Local Data Area sollte nun folgendermaßen aussehen:



T...	L...	Name	F...	L...	H...	Propert
	1	#NAME-START	A	20		
	1	#NAME-END	A	20		

Global Data Area aus dem Programm aufrufen

Sobald eine Global Data Area mit STOW gespeichert wurde, kann sie aus einem Natural-Programm heraus aufgerufen werden.

Sie werden jetzt das DEFINE DATA-Statement in Ihrem Programm so ändern, dass die eben von Ihnen definierte Global Data Area benutzt wird.

Lassen Sie den Data-Area-Editor im Hintergrund geöffnet.

► Global Data Area in Ihrem Programm benutzen

1. Kehren Sie zum Programmmeditor zurück.
2. Geben Sie Folgendes in der Zeile über LOCAL USING LDA01 ein:

```
GLOBAL USING GDA01
```

Eine Global Data Area muss immer vor einer Local Data Area angegeben werden. Andernfalls tritt ein Fehler auf.

Ihr Programm sollte nun folgendermaßen aussehen:

```
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL USING LDA01
END-DEFINE
*
RP1. REPEAT
*
  INPUT USING MAP 'MAP01'
*
  IF #NAME-START = '.' THEN
    ESCAPE BOTTOM (RP1.)
  END-IF
*
  IF #NAME-END = ' ' THEN
    MOVE #NAME-START TO #NAME-END
  END-IF
*
  RD1. READ EMPLOYEES-VIEW BY NAME
    STARTING FROM #NAME-START
    ENDING AT #NAME-END
*
    IF LEAVE-DUE >= 20 THEN
      PERFORM MARK-SPECIAL-EMPLOYEES
    ELSE
      RESET #MARK
    END-IF
*
    DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK
*
  END-READ
*
  IF *COUNTER (RD1.) = 0 THEN
    REINPUT 'No employees meet your criteria.'
  END-IF
*
END-REPEAT
*
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
*
END
```

3. Führen Sie das Programm mit RUN aus.
4. Um zu überprüfen, ob das Ergebnis immer noch dasselbe ist wie vorher (als noch keine Global Data Area mit `DEFINE DATA` referenziert wurde), geben Sie "JONES" als Startname ein und drücken Sie EINGABE.
5. Drücken Sie ESC, um das Ausgabefenster zu schließen.
6. Speichern Sie das Programm mit STOW.

Sie können nun mit den nächsten Übungen fortfahren: *Externe Subroutinen*.