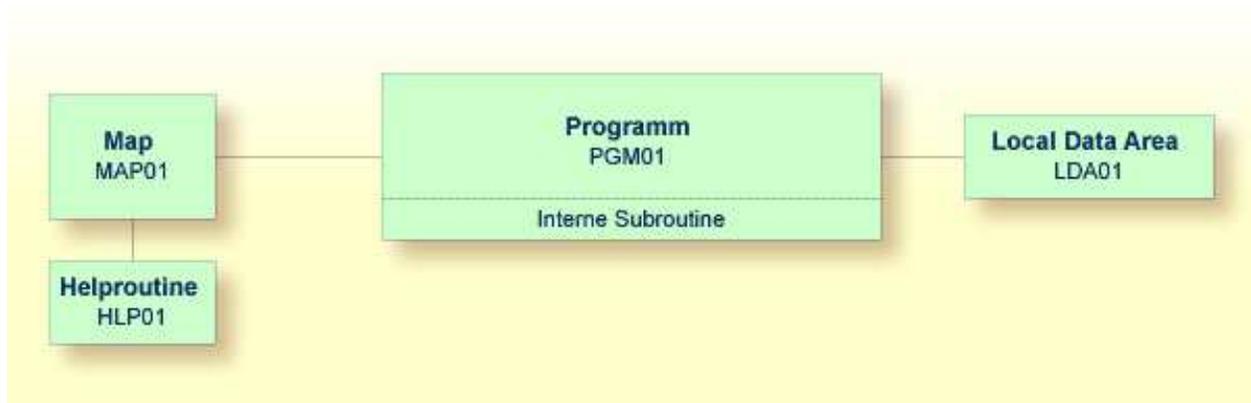


# Local Data Areas

Zurzeit werden die Felder, die in Ihrem Programm benutzt werden, mit `DEFINE DATA` im Programm selbst definiert. Es ist jedoch auch möglich, die Felddefinitionen in einer Local Data Area (LDA) außerhalb des Programms zu definieren und diese Local Data Area im `DEFINE DATA`-Statement des Programms lediglich namentlich zu referenzieren. Für die Wiederverwendbarkeit und zu Gunsten einer klaren Anwendungsstruktur ist es in der Regel besser, die Felder außerhalb eines Programms in einer Data Area zu definieren.

Sie werden jetzt die Informationen aus dem `DEFINE DATA`-Statement in eine Local Data Area verlagern. Wenn Sie mit den Übungen in diesem Kapitel fertig sind, wird Ihre Beispielanwendung aus den folgenden Modulen bestehen:



Dieses Kapitel enthält die folgenden Übungen:

- Local Data Area erstellen
- Datenfelder definieren
- Datenfelder aus einem DDM importieren
- Local Data Area aus dem Programm aufrufen

---

## Local Data Area erstellen

Sie werden jetzt den Data-Area-Editor aufrufen, in dem Sie die erforderlichen Felder definieren.

### ▶ Local Data Area erstellen

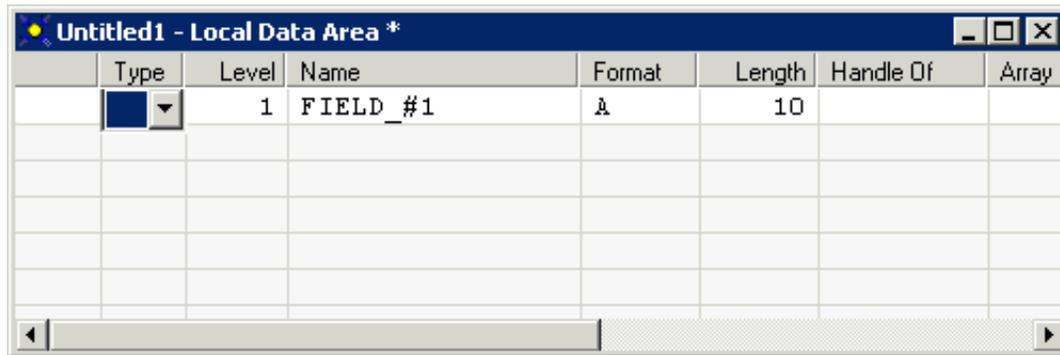
1. Markieren Sie im Library-Workspace die Library, die auch Ihr Programm enthält (d.h. markieren Sie den Knoten **TUTORIAL**).
2. Wählen Sie aus dem Kontextmenü den Befehl **New Source > Local Data Area**.

Oder:

Wählen Sie die folgende Schaltfläche in der Symbolleiste:



Ein Editorfenster erscheint.



Level, Name, Format und Länge (Length) sind in der ersten Zeile automatisch voreingestellt.

## Datenfelder definieren

Sie werden jetzt die folgenden Felder definieren:

| Level | Name        | Format | Länge |
|-------|-------------|--------|-------|
| 1     | #NAME-START | A      | 20    |
| 1     | #NAME-END   | A      | 20    |
| 1     | #MARK       | A      | 1     |

Dies sind die Benutzervariablen, die Sie zuvor im `DEFINE DATA`-Statement definiert haben.

Sie können die Datenfelder auf zwei verschiedene Arten definieren: entweder manuell im Editorfenster, wobei es in Ihrer Verantwortung liegt, das korrekte Format und die korrekte Länge des Datenfeldes zu definieren, oder mit dem Befehl **Import**, wobei Sie die Datenfelder einfach aus einer Liste auswählen und das korrekte Format und die korrekte Länge automatisch benutzt wird. Diese beiden Möglichkeiten sind unten beschrieben.

Der Zustand der folgenden Schaltfläche in der Insert-Symbolleiste des Data-Area-Editors weist auf die Einfügeposition hin. Wenn die Schaltfläche gedrückt aussieht, wird das neue Feld hinter dem markierten Feld eingefügt (dieser Zustand wird in diesem Tutorial vorausgesetzt); andernfalls wird das neue Feld vor dem markierten Feld eingefügt.



### ▶ Datenfeld manuell im Editorfenster definieren

1. Achten Sie darauf, dass die Spalte **Level** in der ersten Reihe den voreingestellten Wert "1" enthält.
2. Ändern Sie den voreingestellten Wert in der Spalte **Name** der ersten Reihe in "#NAME-START" ab.

3. Achten Sie darauf, dass die Spalte **Format** in der ersten Reihe den voreingestellten Wert "A" enthält.
4. Ändern Sie den voreingestellten Wert in der Spalte **Length** der ersten Reihe in "20" ab.

#### ▶ Datenfelder aus einem Programm importieren

1. Markieren Sie im Editor die Zeile, die für #NAME-START erstellt wurde.

**Anmerkung:**

Falls erforderlich, drücken Sie EINGABE oder ESC um vom Selektionsmodus für einzelne Zellen in den Selektionsmodus für komplette Reihen umzuschalten.

2. Wählen Sie aus dem Kontextmenü den Befehl **Import**.

Oder:

Wählen Sie die folgende Schaltfläche in der Symbolleiste:



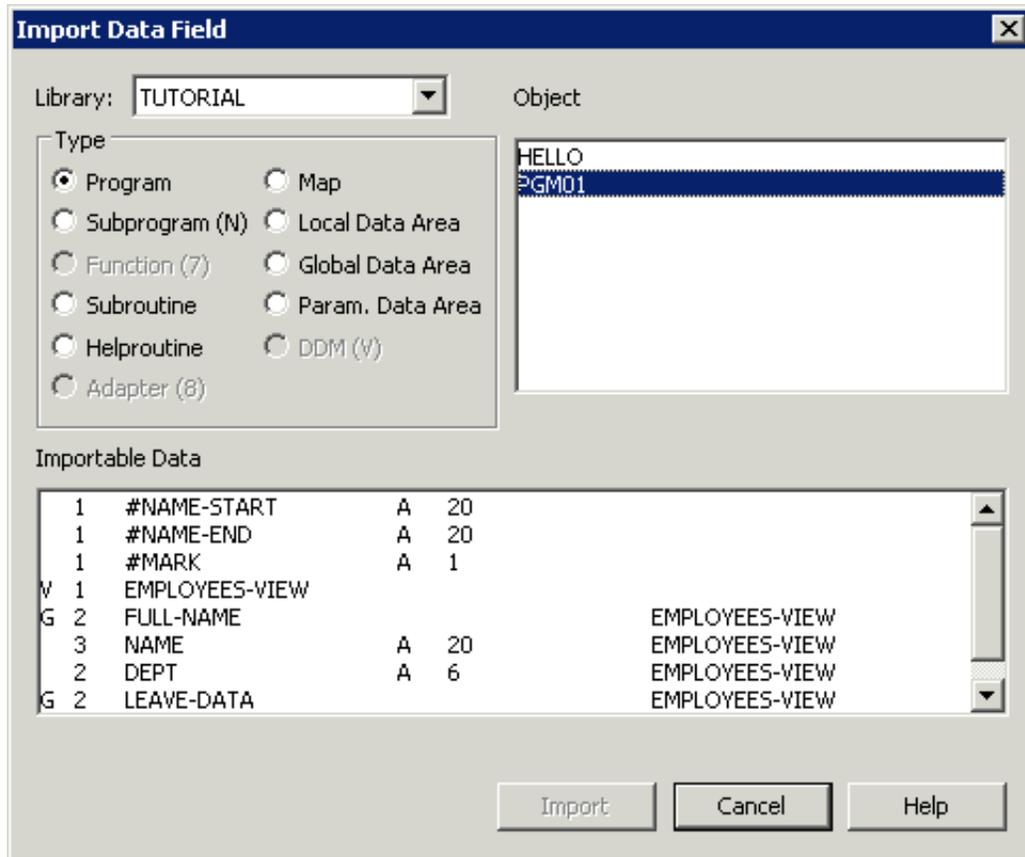
Das Dialogfeld **Import Data Field** erscheint.

3. Achten Sie darauf, dass **TUTORIAL** im Dropdown-Listefeld **Library** markiert ist.
4. Markieren Sie das Optionsfeld **Program**.

Alle zurzeit in Ihrer Library definierten Programme werden jetzt im Listefeld **Object** angezeigt.

5. Markieren Sie das Programm mit dem Namen PGM01.

Die importierbaren Felder werden jetzt unten im Dialogfeld angezeigt.



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

#NAME-END  
#MARK

7. Wählen Sie die Befehlsschaltfläche **Import**.

Die Befehlsschaltfläche **Cancel** im Dialogfeld **Import Data Field** trägt nun den Namen **Quit**.

8. Wählen Sie die Befehlsschaltfläche **Quit**, um das Dialogfeld **Import Data Field** zu schließen.

Die Felder #NAME-END und #MARK werden jetzt unter dem Feld #NAME-START im Editorfenster angezeigt.

## Datenfelder aus einem DDM importieren

Sie werden jetzt dieselben Datenfelder importieren, die Sie zuvor im DEFINE DATA-Statement des Programms definiert hatten. Die Felder werden direkt aus einem Natural-Daten-View in den Data-Area-Editor eingelesen. Ein Daten-View referenziert die Datenbankfelder, die in einem DDM (Data Definition Module) definiert sind.

Im Data-Area-Editor werden die importierten Datenfelder unter dem zurzeit markierten Datenfeld eingefügt.

 **Datenfelder aus einem DDM importieren**

1. Markieren Sie im Editor die Zeile, die #MARK enthält.
2. Wählen Sie aus dem Kontextmenü den Befehl **Import**.

Das Fenster **Import Data Field** erscheint.

3. Wählen Sie aus dem Dropdown-Listefeld **Library** den Eintrag **SYSEXDDM**.

Das Optionsfeld **DDM** ist markiert.

4. Markieren Sie im Listefeld **Object** das Beispiel-DDM mit dem Namen **EMPLOYEES**.

5. Drücken Sie STRG und markieren Sie die folgenden importierbaren Felder:

```
PERSONNEL-ID  
FULL-NAME  
NAME  
DEPT  
LEAVE-DATA  
LEAVE-DUE
```

**Anmerkung:**

Das Feld PERSONNEL-ID wird später benutzt, wenn Sie das Subprogramm erstellen.

6. Wählen Sie die Befehlsschaltfläche **Import**.

Das Dialogfeld **View Definition** erscheint.

7. Geben Sie denselben Namen an, den Sie vorher für den View definiert haben (d.h. EMPLOYEES-VIEW).

8. Wählen Sie die Befehlsschaltfläche **OK**.

9. Wählen Sie die Befehlsschaltfläche **Quit**, um das Dialogfeld **Import Data Field** zu schließen.

Die Local Data Area sollte nun folgendermaßen aussehen:

| Type | Level | Name           | Format | Length | Handle |
|------|-------|----------------|--------|--------|--------|
|      | 1     | #NAME-START    | A      | 20     |        |
|      | 1     | #NAME-END      | A      | 20     |        |
|      | 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    |        |

Die Spalte **Type** gibt den Variablentyp an. Der View ist mit "V" gekennzeichnet und jede Gruppe ist mit "G" gekennzeichnet.

**Anmerkung:**

Wenn das Kontrollkästchen **Expand/Collapse** in den Optionen für den Data-Area-Editor markiert wurde, erscheinen in der ersten Spalte Umschaltssymbole zum Ein- und Ausblenden der darunter liegenden Levels (für den View und jede Gruppe).

10. Speichern Sie die Local Data Area mit STOW.
11. Wenn Sie dazu aufgefordert werden, einen Namen für die Local Data Area einzugeben, geben Sie "LDA01" ein.

Im Library-Workspace erscheint ein neuer Knoten mit dem Namen **Local Data Areas** als Unterknoten von **TUTORIAL**. Dieser Unterknoten enthält die Local Data Area, die Sie eben gespeichert haben.

## Local Data Area aus dem Programm aufrufen

Sobald eine Local 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 Local Data Area benutzt wird.

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

### ▶ Local Data Area in Ihrem Programm benutzen

1. Kehren Sie zum Programmeditor zurück.
2. Löschen Sie im `DEFINE DATA`-Statement alle Variablen zwischen `LOCAL` und `END-DEFINE`.

- Referenzieren Sie die Local Data Area, indem Sie die LOCAL-Zeile folgendermaßen ändern:

```
LOCAL USING LDA01
```

Ihr Programm sollte nun folgendermaßen aussehen:

```
DEFINE DATA
  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
```

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

Sie können nun mit den nächsten Übungen fortfahren: *Global Data Areas*.