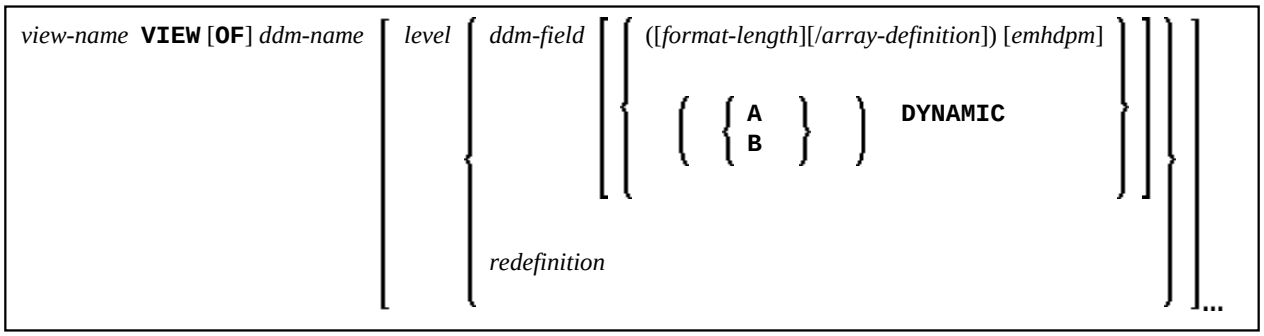
View-Definition

Die mit den Statements `DEFINE DATA LOCAL` und `DEFINE DATA OBJECT` benutzte Option *view-definition* hat die folgende Syntax:



Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung

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

Funktion

Eine *view-definition* stellt einen Ausschnitt eines Datendefinitionsmoduls (DDM) dar.

Anmerkung:
In einer Parameter Data Area ist eine *view-definition* nicht erlaubt.

Weitere Informationen siehe Abschnitt *Daten in einer Adabas-Datenbank aufrufen* im *Leitfaden zur Programmierung* und dort insbesondere die folgenden Themen:

- *Datendefinitionsmodule – DDMs*
- *Datenbank-Arrays*
- *DEFINE DATA-Views*

Syntax-Beschreibung

view-name	Der Name, den der View erhalten soll. Es gelten die Namenskonventionen für Natural-Variablen. Siehe *Namen von Benutzervariablen* in der Dokumentation *Natural benutzen*.
VIEW [OF] ***ddm-name***	Der Name des DDMs, aus dem der View gebildet wird.

level	Dies ist eine ein- oder zweistellige Zahl im Bereich von 01 bis 99 (die vorangestellte 0 ist nicht erforderlich), die in Verbindung mit der Gruppierung von Feldern verwendet wird. Felder mit einer Level-Nummer von 02 an aufwärts werden als Teil einer unmittelbar vorangehenden Gruppe mit einer jeweils nächstniedrigeren Level-Nummer betrachtet. Durch die Definition einer Gruppe (die auch nur aus einem Feld bestehen kann) ist es möglich, durch Angabe lediglich des Gruppennamens eine ganze Reihe von aufeinanderfolgenden Feldern gleichzeitig zu referenzieren. Bei manchen Statements (`CALL`, `CALLNAT`, `RESET`, `WRITE` usw.) können Sie den Gruppennamen als Kurznamen für die Referenzierung der in der Gruppe enthaltenen Felder angeben. Eine Gruppe kann ihrerseits Teil einer anderen Gruppe sein. Bei der Vergabe von Level-Nummern für eine Gruppe darf kein Level ausgelassen werden.
ddm-field	Der im verwendeten DDM definierte Name eines Feldes. Bei der Definition eines Views für ein `HISTOGRAM`-Statement darf dieser View lediglich den Deskriptor enthalten, den das `HISTOGRAM`-Statement benutzt.
redefinition	Eine *redefinition* kann zur Redefinition einer Gruppe, eines Views, eines DDM-Felds oder eines einzelnen Feldes oder einer einzelnen Variable benutzt werden (d.h. Skalar oder Array). Siehe *Redefinition*.
format-length	Format und Länge des definierten Feldes. Werden diese Angaben nicht gemacht, wird die Format-/Längendefinition aus dem DDM übernommen. Im Structured Mode muss die Definition von Format und Länge (wenn angegeben) dieselbe wie die vom DDM sein. Im Reporting Mode muss die Format-/Längendefinition kompatibel mit der im DDM sein.
A, U or B	Datentyp: Alphanumerisch (A), Unicode (U) oder binär (B) für dynamische Variablen. **Anmerkungen:** 1. Bei Adabas für Großrechner steht das Format U für LA-Felder (Länge: <= 16381 Bytes), aber nicht für LB-Felder (Länge: <= 1GB) zur Verfügung 2. Format B kann nicht bei Adabas verwendet werden.
array-definition	Abhängig vom benutzten Modus müssen Arrays (Periodengruppenfelder, multiple Felder) eventuell Informationen über ihre Ausprägungen aufnehmen. Siehe den Abschnitt *Array-Definition in einem View* weiter unten.
emhdpm	Mit dieser Option können zusätzliche Parameter definiert werden, die für ein Feld oder eine Variable gelten sollen. Siehe *Parameter EM, HD, PM für Feld/Variable*.

DYNAMIC	Definiert ein View-Feld als dynamisch. Weitere Informationen zur Verarbeitung von dynamischen Variablen siehe den Abschnitt *Dynamische und große Variablen benutzen*.

Array-Definition in einem View

Abhängig von dem benutzten Programmiermodus müssen Arrays, d.h. Periodengruppenfelder (PE), multiple Felder (MU), in Abhängkeit vom verwendeten Programmiermodus eventuell Informationen über ihre Ausprägungen aufnehmen.

- Array-Definition in einem View im Structured Mode
- Array-Definition in einem View im Reporting Mode

Array-Definition in einem View im Structured Mode

Wenn ein Feld in einem View benutzt wird, das einen Array darstellt, gilt Folgendes:

- Ein Indexwert muss für MU/PE-Felder angegeben werden.
- Wenn kein(e) Format/Länge angegeben ist, werden die Werte aus dem DDM genommen.
- Ist ein(e) Format/Länge angegeben, muss die Angabe mit der im DDM übereinstimmen.

Datenbank-spezifische Anmerkungen zum Structured Mode:

Adabas:	Wenn (in einem DDM definierte) MU/PE-Felder in einem View benutzt werden sollen, müssen diese Felder eine Array-Index-Angabe enthalten. Für ein MU-Feld oder ein normales PE-Feld geben Sie einen eindimensionalen Index-Bereich an, z.B. (1:10). Für ein MU-Feld in einer PE-Gruppe geben Sie einen zweidimensionalen Index-Bereich an, z.B. (1:10,1:5).

Beispiele für Structured Mode:

```
DEFINE DATA LOCAL
1 EMP1 VIEW OF EMPLOYEES
  2 NAME(A20)
  2 ADDRESS-LINE(A20 / 1:2)

1 EMP2 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE(1:2)

1 EMP3 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE(2)

1 #K (I4)
1 EMP4 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE(#K:#K+1)
END-DEFINE
END
```

Array-Definition in einem View im Reporting Mode

In diesem Modus gelten dieselben Regeln wie für Structured Mode. Es gibt aber zwei Ausnahmen:

- Ein Indexwert muss nicht angegeben werden. In diesem Fall wird der Index-Bereich für die fehlenden Dimensionen auf (1:1) gesetzt.
- Die Fomat/Längenangabe kann sich von der Angabe im DDM unterscheiden.

Beispiele:

```
DEFINE DATA LOCAL
1 EMP1 VIEW OF EMPLOYEES
  2 NAME(A30)
  2 ADDRESS-LINE(A35 / 5:10)

1 EMP2 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE(A40)        /* ADDRESS LINE (1:1) IS ASSUMED

1 EMP3 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE             /* ADDRESS LINE (1:1) IS ASSUMED

1 #K (I4)
1 EMP4 VIEW OF EMPLOYEES
  2 NAME
  2 ADDRESS-LINE(#K:#K+1)
END-DEFINE
END
```