View Definition
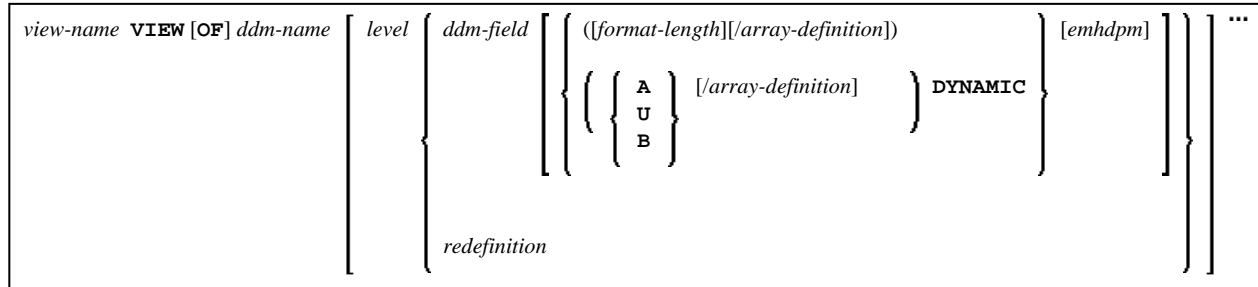
The *view-definition* option is used with DEFINE DATA LOCAL and DEFINE DATA OBJECT.

The *view-definition* option has the following syntax:

This chapter covers the following topics:

- Function

- Syntax Description

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Function

A *view-definition* is used to define a data view as derived from a data definition module (DDM).

Note:
In a parameter data area, *view-definition* is not permitted.

For further information, see *Accessing Data in an Adabas Database* in the *Programming Guide* and particularly the following topics:

- *Data Definition Modules - DDMs*

- *Database Arrays*

- *Datenbank-View definieren*

Syntax Description

Syntax Element	Description
view-name	**View Name:** The name to be assigned to the view. Rules for Natural variable names apply; see *Naming Conventions for User-Defined Variables* in *Using Natural Studio*.

Syntax Element	Description
VIEW [OF] *ddm-name*	**DDM Name:** The name of the data definition module (DDM) from which the view is to be taken.
level	**Level Number:** Level number is a 1- or 2-digit number in the range from 01 to 99 (the leading zero is optional) used in conjunction with field grouping. Fields assigned a level number of 02 or greater are considered to be a part of the immediately preceding group which has been assigned a lower level number. The definition of a group enables reference to a series of fields (may also be only one field) by using the group name. With certain statements (CALL, CALLNAT, RESET, WRITE, etc.), you may specify the group name as a shortcut to reference the fields contained in the group. A group may consist of other groups. When assigning the level numbers for a group, no level numbers may be skipped.
ddm-field	**DDM Field Name:** The name of a field to be taken from the DDM. When you define a view for a HISTOGRAM statement, the view must contain only the descriptor for which HISTOGRAM is to be executed.
redefinition	**Redefinition:** A *redefinition* may be used to redefine a group, a view, a DDM field or a single field/variable (that is a scalar or an array). For further information, see *Redefinition*.
format-length	**Format/Length Definition:** Format and length of the field. If omitted, these are taken from the DDM. In structured mode, the definition of format and length (if supplied) must be the same as those in the DDM. In reporting mode, the definition of format and length (if supplied) must be type-compatible with those in the DDM.

Syntax Element	Description
A, U or B	**Data Type:** Alphanumeric (A), Unicode (U) or binary (B) for dynamic variables. **Notes:** 1. For Adabas on mainframe computers, format U is available for LA fields (length <= 16381 bytes), but not for LB fields (length: <= 1 GB). 2. Format B is not available with Adabas.
array-definition	**Array Definition:** Depending on the programming mode used, arrays (periodic-group fields, multiple-value fields) may have to contain information about their occurrences. For further information, see *Array Definition in a View* below.
emhdpm	**EM, HD, PM Parameters for Field/Variable:** With this option, additional parameters to be in effect for a field/variable may be defined. See *EM, HD, PM Parameters for Field/Variable*.
DYNAMIC	**DYNAMIC Option:** Defines a view field as DYNAMIC. For further information on processing dynamic variables, see *Introduction to Dynamic Variables and Fields* in the *Programming Guide*.

Array Definition in a View

Depending on the programming mode used, arrays (periodic-group fields, multiple-value fields) may have to contain information about their occurrences.

- Structured Mode
- Reporting Mode

Structured Mode

If a field is used in a view that represents an array, the following applies:

- An index value must be specified for MU/PE fields

- When no format/length specification is supplied, the values are taken from the DDM.

- When a format/length specification is supplied, it must be the same as in the DDM.

Database-Specific Considerations in Structured Mode:

Adabas:	If MU/PE fields (defined in a DDM) are to be used inside a view, these fields must include an array index specification. For an MU field or ordinary PE field, you specify a one-dimensional index range, e.g. (1:10). For an MU field inside a PE group, you specify a two-dimensional index range, e.g. (1:10,1:5).		
Tamino:	**DDM definition**	**allowed**	**not allowed**
	A(*:X2)	A(*:Y2) Y2=<X2 A(Y1:Y2) Y2>Y1 Y2=<X2 A(Z:Z+Y) Y>=0	A(*:*) A(Y1:*)
	A(X1:*)	A(Y1:*) Y1>=X1 A(Y1:Y2) Y2>=X1, Y1>=X1 A(Z:Z+Y) Y>=0	A(*:*) A(*:Y2)
	A(X1:X2)	A(Y1:Y2) Y2<Y1 A(Z:Z+Y) 0=<Y>=X2-X1+1	A(*:*) A(Y1:*) A(*:Y2)

Examples for 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
```

Reporting Mode

In this mode, the same rules are valid as for structured mode, however, there are two exceptions:

- An index value needs not be supplied. In this case, the index range for the missing dimensions is set to (1:1).

- The format/length specification may differ from the specification in the DDM. Then the definition of format and length must be type-compatible with those in the DDM.

Examples:

```
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
```