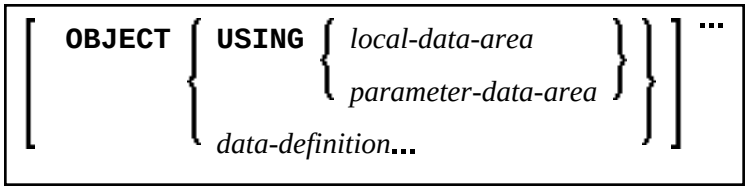
Defining NaturalX Objects

General syntax of `DEFINE DATA OBJECT`:



This chapter covers the following topics:

- Function
- Syntax Description

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

Function

The `DEFINE DATA OBJECT` statement is used in a subprogram or class in conjunction with NaturalX. For further information, refer to the section *NaturalX* in the *Programming Guide*.

Syntax Description

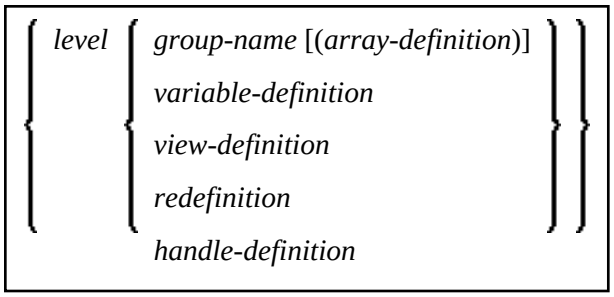
Syntax Element	Description

`USING` *`local-data-area`*	**LDA Name:**
	
	A local data area (LDA) contains data elements which are to be used in a single Natural module. You may reference more than one data area; in that case you have to repeat the reserved words `OBJECT` and `USING`, for example:
	
	`DEFINE DATA`
	`  OBJECT USING DATX_L`
	`  OBJECT USING DATX_P`
	
	`  ...`
	`END-DEFINE ;`
	
	For further information, see also *Defining Fields in a Separate Data Area* in the *Programming Guide*.

`USING` *`parameter-data-area`*	**PDA Name:** A data area defined with `DEFINE DATA OBJECT` may be a parameter data area (PDA). By using a PDA as an object data area you can avoid the extra effort of creating an object data area that has the same structure as the PDA.
`data-definition`	**Direct Data Definition:** Data can also be defined directly using the syntax shown in *Direct Data Definition*.
`END-DEFINE`	**End of DEFINE DATA Statement:** The Natural reserved word `END-DEFINE` must be used to end the `DEFINE DATA` statement.

Direct Data Definition

Data can also be defined directly using the following syntax:



For further information, see *Defining Fields within a DEFINE DATA Statement* in the *Programming Guide*.

Syntax Element	Description
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 1 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. A view-definition must always be defined at Level 1.
group-name	**Group Name:** The name of a group. The name must adhere to the rules for defining a Natural variable name. See also the following sections: ● *Naming Conventions for User-Defined Variables* in *Using Natural Studio*. ● *Qualifying Data Structures* in the *Programming Guide*.
array-definition	**Array Dimension Definition:** With an *array-definition*, you define the lower and upper bounds of dimensions in an array-definition. For further information, see *Array Dimension Definition*.
variable-definition	**Variable Definition:** A *variable-definition* is used to define a single field/variable that may be single-valued (scalar) or multi-valued (array). For further information, see *Variable Definition*.
view-definition	**View Definition:** A *view-definition* is used to define a view as derived from a data definition module (DDM). For further information, see *View Definition*.

Syntax Element	Description
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*.
handle-definition	**Handle Definition:** A handle identifies a dialog element in code and is stored in handle variables. For further information, see *Handle Definition*.