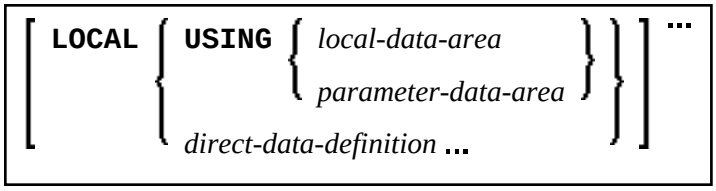
Defining Local Data

General syntax of `DEFINE DATA LOCAL`:



This chapter covers the following topics:

- Function
- Restriction
- Syntax Description

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

Function

The `DEFINE DATA LOCAL` statement is used to define the data elements that are to be used exclusively by a single Natural module in an application. These elements or fields can be defined in different ways:

- either within the `DEFINE DATA LOCAL` statement itself, using the *`direct-data-definition`* syntax (see *Direct Data Definition*)
- or outside the program in a separate LDA (*Local Data Area*) or PDA (*Parameter Data Area*), with the `DEFINE DATA LOCAL USING` statement referencing that data area.

Restriction

The LDA and the objects which reference it must be contained in the same library (or in a steplib).

Syntax Description

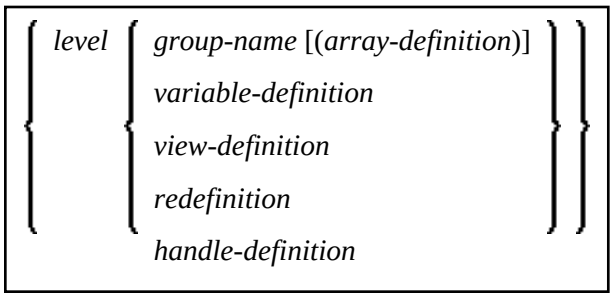
Syntax Element	Description

local-data-area	**LDA Name:**
	
	Specify the name of the local data area (LDA) to be referenced.
	
	An LDA is created using the *Data Area Editor*. It contains predefined data elements which can be included in the `DEFINE DATA LOCAL` statement.
	
	You may reference more than one data area; in that case you have to repeat the reserved words `LOCAL` and `USING`, for example:
	
	`DEFINE DATA LOCAL`
	`  LOCAL USING DATX_L`
	`  LOCAL USING DATX_P`
	
	`  ...`
	`END-DEFINE ;`
	
	For further information, see also *Defining Fields in a Separate Data Area* and *Local Data Area, Example 2* in the *Programming Guide*.

parameter-data-area	**PDA Name:** Specify the name of a parameter data area (PDA). **Note:** A data area referenced with `DEFINE DATA LOCAL` may also be a parameter data area (PDA). By using a PDA as an LDA you can avoid the extra effort of creating an LDA that has the same structure as the PDA. A PDA is created using the *Data Area Editor*. For further information, see *Parameter Data Area* in the *Programming Guide*.
direct-data-definition	**Direct Data Definition:** For information on how to define elements or fields within the statement itself, that is, without using an LDA or PDA, see the section *Direct Data Definition* below.
`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

Local data can be defined directly within a program or routine. For direct data definition, the following syntax applies:



For further information, see

- *Example 1 - DEFINE DATA LOCAL* (Direct Data Definition)
- *Defining Fields within a DEFINE DATA Statement* in the *Programming Guide*
- *Local Data Area, Example 1* in the *Programming Guide*

Syntax Element Description for Direct Data Definition:

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*. - *Qualifying Data Structures* in the *Programming Guide*.

Syntax Element	Description
array-definition	**Array Dimension Definition:** With an *array-definition*, you define the lower and upper bounds of dimensions in an array-definition. 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). See *Variable Definition*.
view-definition	**View Definition:** A *view-definition* is used to define a view as derived from a data definition module (DDM). See *View Definition*.
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). See *Redefinition*.
handle-definition	**Handle Definition:** A handle identifies a dialog element in code and is stored in handle variables. See *Handle Definition*.