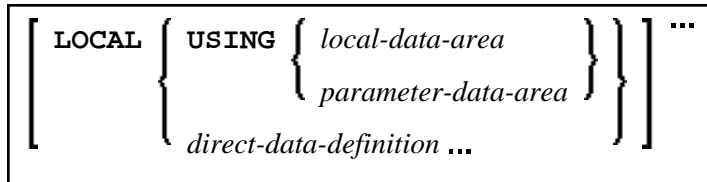


# Defining Local Data

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
<i>local-data-area</i>	<p><b>LDA Name:</b></p> <p>Specify the name of the local data area (LDA) to be referenced.</p> <p>An LDA is created using the <i>Data Area Editor</i>. It contains predefined data elements which can be included in the <code>DEFINE DATA LOCAL</code> statement.</p> <p>You may reference more than one data area; in that case you have to repeat the reserved words <code>LOCAL</code> and <code>USING</code>, for example:</p> <pre>DEFINE DATA LOCAL   LOCAL USING DATX_L   LOCAL USING DATX_P   . . . END-DEFINE ;</pre> <p>For further information, see also <i>Defining Fields in a Separate Data Area</i> and <i>Local Data Area, Example 2</i> in the <i>Programming Guide</i>.</p>
<i>parameter-data-area</i>	<p><b>PDA Name:</b></p> <p>Specify the name of a parameter data area (PDA).</p> <p><b>Note:</b> A data area referenced with <code>DEFINE DATA LOCAL</code> 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.</p> <p>A PDA is created using the <i>Data Area Editor</i>.</p> <p>For further information, see <i>Parameter Data Area</i> in the <i>Programming Guide</i>.</p>
<i>direct-data-definition</i>	<p><b>Direct Data Definition:</b></p> <p>For information on how to define elements or fields within the statement itself, that is, without using an LDA or PDA, see the section <i>Direct Data Definition</i> below.</p>
END-DEFINE	<p><b>End of DEFINE DATA Statement:</b></p> <p>The Natural reserved word <code>END-DEFINE</code> must be used to end the <code>DEFINE DATA</code> statement.</p>

## Direct Data Definition

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

<pre> { level { group-name [(array-definition)] } } {      { variable-definition          } } {      { view-definition             } } {      { redefinition                 } } {      { handle-definition           } } </pre>
--

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
<i>level</i>	<p><b>Level Number:</b></p> <p>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.</p> <p>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.</p> <p>A group may consist of other groups. When assigning the level numbers for a group, no level numbers may be skipped.</p> <p>A view-definition must always be defined at Level 1.</p>
<i>group-name</i>	<p><b>Group Name:</b></p> <p>The name of a group. The name must adhere to the rules for defining a Natural variable name.</p> <p>See also the following sections:</p> <ul style="list-style-type: none"> <li>● <i>Naming Conventions for User-Defined Variables</i> in <i>Using Natural</i>.</li> <li>● <i>Qualifying Data Structures</i> in the <i>Programming Guide</i>.</li> </ul>

Syntax Element	Description
<i>array-definition</i>	<p><b>Array Dimension Definition:</b></p> <p>With an <i>array-definition</i>, you define the lower and upper bounds of dimensions in an array-definition.</p> <p>See <i>Array Dimension Definition</i>.</p>
<i>variable-definition</i>	<p><b>Variable Definition:</b></p> <p>A <i>variable-definition</i> is used to define a single field/variable that may be single-valued (scalar) or multi-valued (array).</p> <p>See <i>Variable Definition</i>.</p>
<i>view-definition</i>	<p><b>View Definition:</b></p> <p>A <i>view-definition</i> is used to define a view as derived from a data definition module (DDM).</p> <p>See <i>View Definition</i>.</p>
<i>redefinition</i>	<p><b>Redefinition:</b></p> <p>A <i>redefinition</i> may be used to redefine a group, a view, a DDM field or a single field/variable (that is a scalar or an array).</p> <p>See <i>Redefinition</i>.</p>
<i>handle-definition</i>	<p><b>Handle Definition:</b></p> <p>A handle identifies a dialog element in code and is stored in handle variables. See <i>Handle Definition</i>.</p>