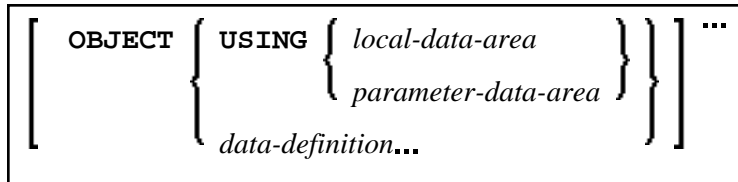


Defining NaturalX Objects

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

USING <i>local-data-area</i>	<p>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 <code>OBJECT</code> and <code>USING</code>, for example:</p> <pre style="margin: 0;">DEFINE DATA OBJECT USING DATX_L OBJECT USING DATX_P . . . END-DEFINE ;</pre> <p>For further information, see also <i>Defining Fields in a Separate Data Area</i> in the <i>Programming Guide</i>.</p>
USING <i>parameter-data-area</i>	<p>A data area defined with <code>DEFINE DATA OBJECT</code> 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.</p>
<i>data-definition</i>	<p>Data can also be defined directly using the syntax shown in the section <i>Direct Data Definition</i> below.</p>
END-DEFINE	<p>The Natural reserved word <code>END-DEFINE</code> must be used to end the <code>DEFINE DATA</code> statement.</p>

Direct Data Definition

Data can also be defined directly using the following syntax:

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

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

level	<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>
group-name	<p>The name of a group. The name must adhere to the rules for defining a Natural variable name. See also the following sections:</p> <ul style="list-style-type: none"> ● <i>Naming Conventions for User-Defined Variables</i> in the <i>Using Natural</i> documentation. ● <i>Qualifying Data Structures</i> in the <i>Programming Guide</i>.
array-definition	<p>With an <i>array-definition</i>, you define the lower and upper bounds of dimensions in an array-definition. See <i>Array Dimension Definition</i>.</p>
variable-definition	<p>A <i>variable-definition</i> is used to define a single field/variable that may be single-valued (scalar) or multi-valued (array). See <i>Variable Definition</i>.</p>
view-definition	<p>A <i>view-definition</i> is used to define a view as derived from a data definition module (DDM). See <i>View Definition</i>.</p>
redefinition	<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). See <i>Redefinition</i>.</p>
handle-definition	<p>A handle identifies a dialog element in code and is stored in handle variables. See <i>Handle Definition</i>.</p>