

Defining Application-Independent Variables

General syntax of `DEFINE DATA INDEPENDENT`:

```
DEFINE DATA
  INDEPENDENT AIV-data-definition...
END-DEFINE
```

This chapter covers the following topics:

- Function
- Syntax Description

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

Function

The `DEFINE DATA INDEPENDENT` statement is used to define application-independent variables (AIVs).

An application-independent variable is referenced by its name, and its content is shared by all programming objects executed within one application that refer to that name. The variable is allocated by the first executed programming object that references this variable and is deallocated by the `LOGON` command or a `RELEASE VARIABLES` statement.

The optional `INIT` clause is evaluated in each executed programming object that contains this clause (not only in the programming object that allocates the variable).

Note:

In an RPC server, application-independent variables (AIVs) are not deallocated implicitly, but stay active across RPC requests, because different clients may have access to the same variables on the RPC server. This means they must be deallocated explicitly using the `RELEASE VARIABLES` statement. See *Application-Independent Variables* in the *Natural Remote Procedure Call* documentation.

Syntax Description

Syntax Element	Description
INDEPENDENT <i>AIV-data-definition</i>	AIV Data Definition: The DEFINE DATA INDEPENDENT statement can be used to define a single or multiple application-independent variables (AIVs). For each AIV, the syntax shown in <i>AIV Data Definition</i> applies.
END-DEFINE	End of DEFINE DATA Statement: The Natural reserved word END-DEFINE must be used to end the DEFINE DATA statement.

AIV Data Definition

$level \left\{ \begin{array}{l} variable-definition \\ redefinition \\ handle-definition \end{array} \right\}$
--

Syntax Element Description:

Syntax Element	Description
<i>level</i>	<p>Level Number:</p> <p>An application-independent variable must be defined at Level 01. Other levels are only used in a redefinition.</p>
<i>variable-definition</i>	<p>Variable Definition</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>For further information, see <i>Variable Definition</i>.</p> <p>Note: The name of an application-independent variable must start with a plus (+) character.</p>
<i>redefinition</i>	<p>Redefinition:</p> <p>With a <i>redefinition</i>, you can partition an application-independent variable into one or more subfields.</p> <p>For further information, see <i>Redefinition</i>.</p> <p>The subfields resulting from the redefinition must not be application-independent variables; that is, their name must not start with a plus sign (+). These fields are treated as local variables.</p>
<i>handle-definition</i>	<p>Handle Definition:</p> <p>A handle identifies a dialog element in code and is stored in handle variables.</p> <p>For further information, see <i>Handle Definition</i>.</p>

Note:

The first character of the name must be a plus (+). Rules for Natural variable names apply, see *Naming Conventions for User-Defined Variables in Using Natural*.