DEFINE FUNCTION

```
DEFINE FUNCTION function-name
  [return-data-definition]
  [function-data-definition]
  statement...
END-FUNCTION
```

This chapter covers the following topics:

- Function
- Syntax Description
- Example

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

Function

The `DEFINE FUNCTION` statement may be used to create new user-defined functions which may be called instead of operands in the Natural statements. Functions can be defined inside the object type Function only.

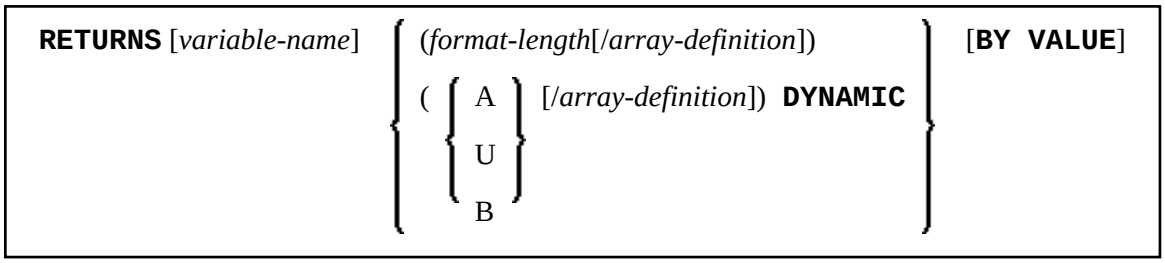
For further information, see the following sections in the *Programming Guide*:

- Natural object type Function
- *Function Call*
- *User-Defined Functions*

Syntax Description

Syntax Element	Description
function-name	**Name of Natural Function:** *function-name* is the symbolic name of the Natural function which is to be defined. The name must follow the same rules as used for user-defined variables, see *Naming Conventions for User-Defined Variables* in the *Using Natural Studio* documentation. This means that the name may have a maximum length of 32 characters and may start with a letter or some special characters such as a hash (#). You may not use the same function name twice in one library (including the libraries of the STEPLIB mechanism). Function overloading is not allowed. This means that all function definitions must have unique function names.
return-data-definition	**Return Data Definition Clause:** For details on this clause, see *Return Data Definition*.
function-data-definition	**Function Data Definition Clause:** For details on this clause, see *Function Data Definition*.
statement...	**Statement(s) to be Executed:** In place of *statement*, you must supply one or several suitable statements, depending on the situation. For an example of a statement, see *Example*.
END-FUNCTION	**End of DEFINE FUNCTION Statement:** The Natural reserved word END-FUNCTION must be used to terminate the DEFINE FUNCTION statement.

Return Data Definition

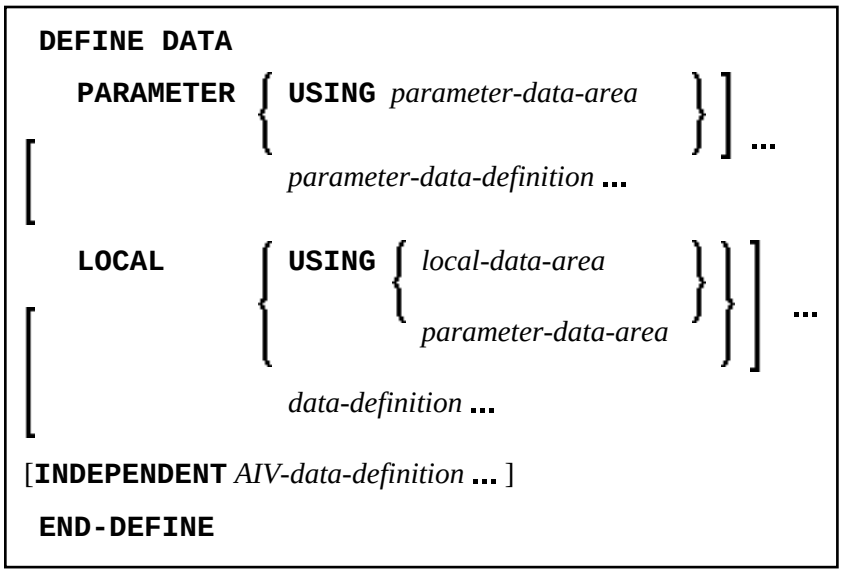


Syntax Element Description:

Syntax Element	Description
RETURNS	**RETURNS Clause:** Each function may contain only one definition of the return variable; that is, only one RETURNS clause is possible.
variable-name	**Return Value:** The return value may be assigned using the *variable-name*. If no explicit variable name is given in the definition, the name of the function is used as a return variable. The return value must not be an array.
BY VALUE	**Return Values by Value or by Reference:** Each parameter may be defined as BY VALUE RESULT or by reference, so that it is possible to return values to the caller using the parameters. Recursive function calls may be used inside a function definition. If you are using the BY VALUE keyword inside the RETURNS clause, the return value of the function will be converted into the return *format-length* which is set by the RETURNS clause.
format-length	**Format/Length Definition:** If the BY VALUE keyword is missing, the *format-length* of the RETURNS clause must match the *format-length* which is returned by the function evaluated at run time.
A, U or B	**Data Type:** Alphanumeric (A), Unicode (U) or binary (B).
array-definition	**Array Dimension Definition:** With *array-definition*, you define the lower and upper bounds of a dimension in an array-definition. For further information, see DEFINE DATA statement, *Array Dimension Definition*.
DYNAMIC	**Dynamic Variable:** A parameter may be defined as DYNAMIC. For more information on processing dynamic variables, see *Introduction to Dynamic Variables and Fields*.

Function Data Definition

Each Function object may contain only one function data definition.



When a function calls another Natural object which uses a global data area, it establishes its own global data area (GDA). Therefore, it is not possible to modify the current GDA data of the calling object. A GDA cannot be defined in the function.

Example

Function object containing function definition:

```
DEFINE FUNCTION GET-FIRST-BYTE
  RETURNS (A1)
  DEFINE DATA PARAMETER
    1 #PARA (A10)
  END-DEFINE
  GET-FIRST-BYTE := #PARA /* return value is assigned
END-FUNCTION
END
```