

DEFINE FUNCTION

```

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

Function

The DEFINE FUNCTION statement may be used in order to create new user defined functions, which may be called instead of operands in the Natural statements.

Functions can be defined inside the "Function" object type only. Each Function object may contain only one function definition. Each function may contain only one definition of the return variable, in other words, only one RETURNS clause is possible. But each parameter may be defined as BY VALUE RESULT or by reference, so that it is possible to return values back to the caller using the parameters. Recursive function calls may be used inside a function definition. If you're 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. 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.

function-name

The 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. This means that the name may consist of max. 32 characters and may start with a letter or some special characters like '#' etc.

You may not use the same function name twice in one library (including the libraries of the STEPLIB mechanism). Function overloading is not allowed, meaning that all function definitions must have unique function names.

return-data-definition

```

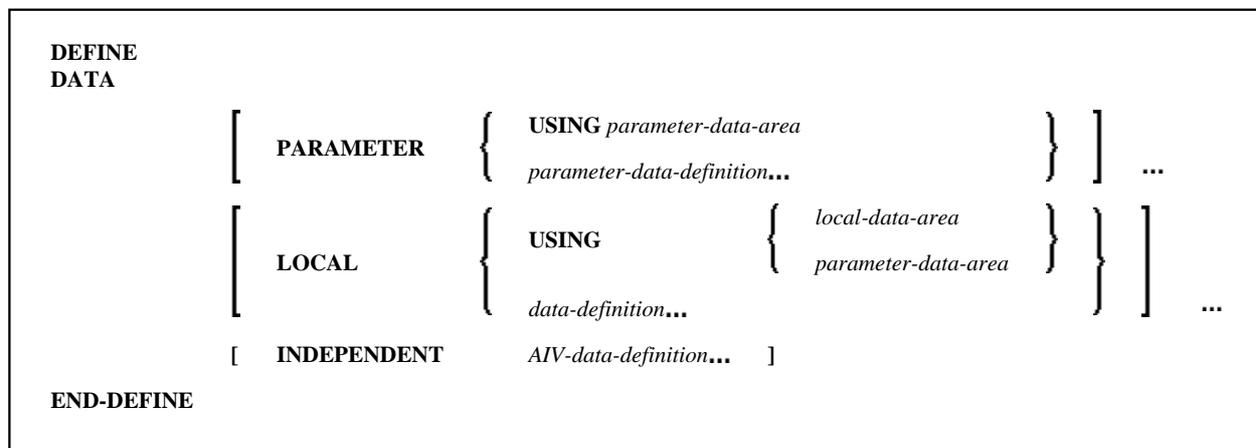
RETURNS [variable-name] { (format-length)
( { A } ) DYNAMIC } [BY VALUE]
    
```

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.

Note:

The return value must not be an array.

function-data-definition



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

Example

Function object containing function definition:

```

/* DFUNX1: Example for DEFINE FUNCTION
/*
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

```