

EDIT

The EDIT command is used to invoke a Natural editor for the purpose of editing the source form of a Natural programming object. Three different forms of command syntax exist. These are documented in separate sections below.

- Syntax 1
- Syntax 2
- Syntax 3

Syntax 1

```
EDIT [object-type] [object-name [library-id]]
```

object-type

<pre> { CLASS } 4 COPYCODE DESCRIPTION GLOBAL HELPROUTINE LOCAL MAP PARAMETER PROGRAM { SUBPROGRAM } N SUBROUTINE TEXT </pre>

Which editor is invoked depends on the type of object to be edited:

- LOCAL data areas, GLOBAL data areas or PARAMETER data areas are edited with the **data area editor**.
- Maps are edited with the **map editor**.
- Classes are edited with the **Class Builder** (Windows) or with the program editor (Mainframe, UNIX).
- All other types of objects - PROGRAM, SUBPROGRAM, SUBROUTINE, HELPROUTINE, COPYCODE, TEXT, DESCRIPTION - are edited with the **program editor**. (A DESCRIPTION is a program description as stored and maintained in the Predict Data Dictionary; an object of this type can only be edited if Predict is installed.)

The object types are described in the Natural Programming Guide. The editors are described in the Natural Editors documentation.

If you specify the name of the object you wish to edit, you need not specify its object type.

object-name

With the EDIT command, you specify the name of the object you wish to edit. The maximum length of the object name is 8 characters. Natural will then load the object into the edit work area of the appropriate editor and set the object name for a subsequent SAVE, CATALOG, STOW command.

If you do not specify an *object-name* and there is no object in the edit work area, the empty program editor screen will be invoked where you can create a program.

For EDIT DESCRIPTION, the *object-name* must be the name as defined as a Natural member in the Predict program definition.

library-id

If the object you wish to edit is not contained in the library you are currently logged on to, you must specify the *library-id* of the library in which the object to be edited is contained.

The setting for *library-id* must not begin with "SYS" (except "SYSTEM").

If Natural Security is active, a *library-id* must not be specified, which means that you can only edit objects which are in your current library.

Syntax 2

<pre>EDIT [*] { * }</pre> <p style="text-align: center;"><i>object-type</i> <i>object-name</i></p>

If you do not remember the name of the object you wish to edit, you can use this form of the EDIT command to display a list of objects, and then select from the list the desired object.

EDIT * displays a list of all objects in your current library.

EDIT *object-type* * displays a list of all objects of that type in your current library.

To select an object from a certain range of objects, you can use asterisk notation and wildcard notation for the *object-name* in the same manner as described for the system command LIST.

Syntax 3

<pre>EDIT FUNCTION <i>subroutine-name</i></pre>

The EDIT FUNCTION command may be used to edit a subroutine using the subroutine name (not the object name) with maximally 32 characters.

Example:

```
DEFINE SUBROUTINE CHECK-PARAMETERS  
  . . .  
END-SUBROUTINE  
END
```

Assuming that the above subroutine has been saved under the object name CHCKSUB, you may edit subroutine CHECK-PARAMETERS either by issuing the command:

```
EDIT S CHKSUB
```

or by

```
EDIT F CHECK-PARAMETERS
```