

EDIT

This 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 the following sections.

- Syntax 1
- Syntax 2
- Syntax 3

Related command: READ.

See also *Object Naming Conventions* in the *Using Natural Studio* documentation.

See also *Invoking Editors* in *Using Natural Studio*.

Syntax 1

<code>EDIT [<i>object-type</i>] [<i>object-name</i> [<i>library-id</i>]]</code>
--

object-type

The following object types can be edited:

{	<u>C</u> CLASS	}
	4	}
	<u>C</u> OPYCODE	
{	<u>D</u> IIALOG	}
	3	}
	<u>G</u> LOBAL	
	<u>H</u> ELPROUTINE	
	<u>L</u> OCAL	
{	<u>M</u> AP	}
	<u>P</u> ARAMETER	
	<u>P</u> ROGRAM	
{	<u>S</u> UBPROGRAM	}
	N	}
	<u>S</u> UBROUTINE	
	<u>T</u> EXT	
	<u>V</u> IEW	
{	7 (for Function)	}

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.
- Dialogs are edited with the dialog editor.
- Classes are edited with the Class Builder.
- EDIT VIEW only works in the current library and when an *object-name* is specified. If the object to be viewed is a DDM, the DDM editor is invoked.
- All other types of objects - program, subprogram, subroutine, 7 (for function), helproutine, copycode, text, description - are edited with the program editor.

Note:

The text object "description" is available on mainframes only. 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 *Programming Guide*. The editors are described in the *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.

Note:

For DDMs, the maximum length is 32 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 source work area, the empty program editor screen will be invoked where you can create a program. If the source work area is not empty, the object will be loaded in the appropriate editor.

Note:

If the source work area is not empty and contains an object that has been opened in an editor session, the corresponding editor window is displayed and gets the input focus. Any changes that are applied to the source work area in the meantime (for example, by running Natural programs) will not be displayed.

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.

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 [*] { * } [object-type] { object-name } </pre>

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 <i>object-type</i> *	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

```
EDIT FUNCTION subroutine-name
```

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

Note:

Please note that the keyword `FUNCTION` used in this syntax is not identical with the Natural object type 7 (for function) listed above. See the description of object type Function in the *Programming Guide*.

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
```