

LIST

This system command is used to display the source code of a single object or to list one or more objects which are contained in the current library. The options of the LIST command are explained below.

This chapter covers the following topics:

- Syntax Overview
- Listing the Contents of the Work Area
- Displaying an Individual Source Code
- Displaying Sources Sequentially
- Displaying a List of Objects
- Displaying a Presorted List of Preselected Objects
- Displaying Long Names of Cataloged Subroutines and Classes
- Displaying NOC Options of Cataloged Objects
- Displaying Compiler Options of Cataloged Objects
- Displaying Directory Information
- Displaying DDMs (Views)
- Options
- List of Objects
- List of Source
- Defining an Individual List Profile

See also separate documents describing LIST XREF, LISTDBRM, LIST COUNT and LISTSQL.

Application Programming Interfaces: USR1054N, USR1055N, USR1056N, USR2018N, USR4216N.
See SYSEXT - *Natural Application Programming Interfaces* in the *Utilities* documentation.

Syntax Overview

<u>L</u>IST	<p>[<i>object-type</i>] <i>object-name-range</i></p> <p>[<i>object-type</i>] <i>object-name</i> [<i>options</i>]</p> <p><i>object-name-range</i> [<i>range-clause</i>]</p> <p><u>SEQUENTIAL</u> [<i>object-type</i>] <i>object-name-range</i> [<i>options</i>]</p> <p><u>DIRECTORY</u> [{ <i>object-name</i> }] <i>object-name-range</i></p> <p><u>EXTENDED</u> [<i>extended-type</i>] <i>object-name-range</i></p> <p><u>NOCOPT</u> [<i>object-type</i>] <i>object-name-range</i></p> <p><u>OPTIONS</u> [<i>object-type</i>] <i>object-name-range</i></p> <p><u>DDM</u> [<i>ddm-name</i>]</p>
--------------------	--

Notes:

1. Instead of the keyword DDM, you can also use the keyword VIEW (or V for short).
2. Since LIST can display long lines containing up to 244 characters, set the line size as big as possible, using profile parameter LS. If possible, set LS=250.

object-type

In place of *object-type*, you may specify one of the object types shown below or an asterisk (*).

*
{ CLASS }
4
COPYCODE
DATA-AREAS
GLOBAL
LOCAL
PARAMETER
{ DIALOG }
3
{ FUNCTION }
7
{ ADAPTER }
8
{ RESOURCE }
9
MAP
{ PROCESSOR }
CP
5
PROGRAM
RECORDING
ROUTINES
HELPROUTINE
{ SUBPROGRAM }
N
SUBROUTINE
TEXT

object-name

In place of *object-name*, you may specify the name of an object (8 characters long at maximum; exception: 32 characters with LIST EXTENDED).

object-name-range

In place of *object-name-range*, you may specify asterisk (*) and wildcard (?) notations:

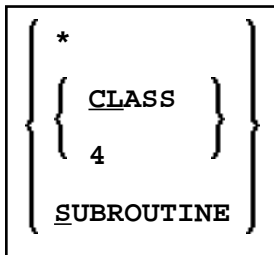
- To have all objects in the current library listed, you specify an asterisk (*) for the *object-name-range*, but no *object-type*.
- To have all objects of a certain type listed, you specify a certain *object-type* and an asterisk (*) for the *object-name-range*.
- If you wish a certain range of objects to be listed, you can use asterisk notation and wildcard notation for the *object-name-range*:
 - Asterisk notation is the option to specify an asterisk (*) in the *object-name-range*: the asterisk stands for any string of characters of any length.
 - Wildcard notation is the option to specify a question mark (?) in the *object-name-range*: the question mark stands for any single character.
- One or more asterisk and wildcard notations can be combined in an *object-name-range*.
- For a list of all objects from a specific start value or until a specific end value, you can use the notation > or < respectively.
- The notations < and > cannot be combined with each other or with asterisk or wildcard notation and can only be used for displaying a list of objects (see *List of Objects* below).

options

For a detailed description of the *options*, see *Options* .

extended-type

In place of *extended-type*, you may specify one of the object types shown below or an asterisk (*).



For a detailed description, see LIST EXTENDED below.

range-clause

[**TYPE**=*type-list*]

[**KIND**=*kind-range*]

[**MODE**=*mode-range*]

[**VERSION**=*version-range*]

[**USER**=*user-range*]

[**DATE**=*date-range*]

[**TIME**=*time-range*]

[**CP**=*code-page-range*]

[SORTED=	{	TYPE	}	[{	ASCENDING	}]																					
			MODE				VERSION			USERID	DATE	{	DT	}	TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}
			VERSION				USERID			DATE	{	DT	}	TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}	
			USERID				DATE			{	DT	}	TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}		
			DATE				{			DT	}	TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}			
			{				DT			}	TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}				
			DT				}			TIME	SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}					
			}				TIME			SIZE	LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}						
			TIME				SIZE			LINES	BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}							
			SIZE				LINES			BPSIZE	{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}								
			LINES				BPSIZE			{	DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}									
			BPSIZE				{			DSIZE	}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}										
			{				DSIZE			}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}											
			DSIZE				}			{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}												
}	{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}																					
{	DATSIZE	}	{	CP	}	{	CODE-PAGE	}																						
DATSIZE	}	{	CP	}	{	CODE-PAGE	}																							
}	{	CP	}	{	CODE-PAGE	}																								
{	CP	}	{	CODE-PAGE	}																									
CP	}	{	CODE-PAGE	}																										
}	{	CODE-PAGE	}																											
{	CODE-PAGE	}																												
CODE-PAGE	}																													
}																														

<i>type-list</i>	* (for all types) or a list of up to 11 valid 1-byte Natural object type characters (e.g. P for Program, M for Map).	
<i>kind-range</i>	*	List all objects.
	S	List only source objects.
	C	List only cataloged objects.
	S/C	List only objects which exist as source and cataloged.
	S/	List only objects which exist as source only.
	/C	List only objects which exist as cataloged only.
	W	List only stowed objects.

<i>mode-range</i>	*	List all objects.
	S	List only structured mode objects.
	R	List only report mode objects.
<i>version-range</i>	<p>The Natural version of the Natural objects.</p> <p>See also the definition of the term Version in the Glossray.</p> <p>Valid version format: <i>V.R.SM</i> where <i>V</i> is the 1-digit version, <i>R</i> the 1-digit release, and <i>SM</i> the 2-digit system maintenance level.</p> <p>You can also specify a range of versions: see range-notation.</p>	
<i>user-range</i>	<p>The ID of the user who saved or cataloged a Natural programming object.</p> <p>Specify a single user ID or a range of user IDs: see <i>range-notation</i> below.</p>	
<i>date-range</i>	<p>Selects all objects with a save or catalog date within the date range specified. Specify a single date or a date range.</p> <p>Valid date format: <i>YYYY-MM-DD</i></p> <p>Valid date ranges:</p> <ul style="list-style-type: none"> ● Leading characters (Example: 2002*) ● Start value (Example: 2002-05>) ● End value (Example: 2003-02<) <p>Special dates allowed are:</p>	
	<u>T</u> ODAY (+ / - <i>nnnn</i>)	<p>All items with the date of the current day.</p> <p>The day can be followed by <i>+nnnn</i> or <i>-nnnn</i> where <i>nnnn</i> has a maximum of 4 digits.</p> <p>The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days.</p> <p>Can be combined with the start value option (>) or the end value option (<), e.g. <i>TO-1></i> selects all objects that were saved or cataloged within the last 2 days.</p>
	<u>Y</u> ESTERDAY	All items with the date of the day before the current day.
	<u>M</u> ONTH	All items with the date range of the current month.
	<u>Y</u> EAR	All items with the date range of the current year.

<i>time-range</i>	<p>Selects all objects with a save or catalog date within the time range specified. Specify a single time or a time range.</p> <p>Valid time format: <i>HH:II:SS</i> (<i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).</p> <p>Valid time ranges:</p> <ul style="list-style-type: none"> ● Leading characters (Example: 10:*) ● Start value (Example: 10:30>) ● End value (Example: 11:20<)
<i>code-page-range</i>	Specify a single code page or a range of code pages: see <i>range-notation</i> below.

range-notation

- To have all objects in the current library listed, you use an asterisk (*).
- If you wish a certain range of objects to be listed, you can use asterisk notation and wildcard notation:
 - Asterisk notation is the option to specify an asterisk (*): the asterisk stands for any string of characters of any length.
 - Wildcard notation is the option to specify a question mark (?): the question mark stands for any single character.
- One or more asterisk and wildcard notations can be combined.
- For a list of all objects from a specific start value or until a specific end value, you can use the notation > or < respectively.
- The notations < and > cannot be combined with each other or with asterisk or wildcard notation.

Listing the Contents of the Work Area

LIST	If you enter only the LIST command itself, without any parameters, the contents of the work area will be listed.
-------------	--

Displaying an Individual Source Code

LIST <i>object-name</i> [<i>options</i>]	In both cases, the object's source code will be listed.
LIST <i>object-type object-name</i> [<i>options</i>]	If you enter a single object name with the LIST command, you need not specify the <i>object-type</i> . If you specify an <i>object-type</i> , you must also specify an <i>object-name</i> .

Displaying Sources Sequentially

LIST SEQUENTIAL <i>object-name-range</i> [<i>options</i>]	In both cases, you must use asterisk (*) and/or wildcard (?) notations for the <i>object-name-range</i> . Then the sources of all objects that meet the selection criteria will be displayed sequentially, i.e. one after the other.
LIST SEQUENTIAL <i>object-type object-name-range</i> [<i>options</i>]	

Displaying a List of Objects

LIST <i>object-name-range</i>	In both cases, you must use asterisk (*) and/or wildcard (?) notation for the <i>object-name-range</i> . You get a list of all objects that meet the specified selection criteria. On the list you can then select objects for display by marking them with the function code LI (see <i>Performing a Function on an Object</i>).
LIST <i>object-type object-name-range</i>	

Displaying a Presorted List of Preselected Objects

LIST <i>object-name-range</i>	In both cases, you must use asterisk (*) and/or wildcard (?) notation for the <i>object-name-range</i> . You get a list of all objects that meet the specified selection criteria. On the list you can then select objects for display by marking them with the function code LI (see <i>Performing a Function on an Object</i>). With the <i>range-clause</i> , you specify additional selection and sorting criteria. See also example below.
LIST <i>object-name-range range-clause</i>	

Displaying Long Names of Cataloged Subroutines and Classes

LIST EXTENDED <i>object-name-range</i>	Displays a list of the long names of cataloged subroutines and classes. For the name options, see <i>object-name-range</i> above.
LIST EXTENDED <i>extended-type object-name-range</i>	

Displaying NOC Options of Cataloged Objects

LIST NOCOPT [<i>object-type</i>] <i>object-name-range</i>	Displays a list of the cataloged objects that are compiled with Natural Optimizer Compiler (NOC), together with the initial NOC options used during CATALOG. For the name options, see <i>object-name-range</i> above.
--	--

Displaying Compiler Options of Cataloged Objects

<p>LIST OPTIONS [<i>object-type</i>] <i>object-name-range</i></p>	<p>Displays a list of the cataloged objects together with the compiler options used during CATALOG. For the name options, see <i>object-name-range</i> above.</p> <p>By default, the final compiler options (that is, the options setting active at the end of the CATALOG) are displayed. For objects cataloged with Natural Version 4.2.5 or above, also the initial compiler options (that is, the options setting active at the beginning of the CATALOG) or the changed compiler options (that is, the options setting that are changed within the source code) can be displayed. See the corresponding help map for the range fields on the online map.</p>
--	---

Displaying Directory Information

LIST DIRECTORY	<p>Displays the directory information on the object currently in the work area:</p> <ul style="list-style-type: none"> ● Source code: "Saved-on" date and time, library name, user ID, programming mode (reporting or structured), TP system, terminal ID, operating system, transaction, Natural version, code page information (if available), source size ● Object code: "Cataloged-on" date and time, library name, user ID, programming mode, TP monitor system, terminal I/O, transaction, Natural version, code page information (if available), operating system/version, GDA used, size of global data, size in DATSIZE, size in buffer pool, size of OPT-code (size of machine code generated by Natural Optimizer Compiler), initial OPT-string (OPT profile parameter value effective at STOW time), compiler options
LIST DIRECTORY <i>object-name</i>	Displays the directory information (as described for LIST DIRECTORY) on the specified object.
LIST DIRECTORY <i>object-name-range</i>	If asterisk (*) and/or wildcard (?) notation is used in place of <i>object-name-range</i> , the directory information of the corresponding objects is displayed sequentially.
LIST <i>object-name</i> WITH DIRECTORY	This command first displays the directory information (as described for LIST DIRECTORY) on the specified object and then lists the source code of the object.

Displaying DDMs (Views)

LIST DDM	Displays a list of all DDMs.
LIST DDM <i>ddm-name</i>	<p>If you specify a single DDM name, the specified DDM will be displayed.</p> <p>For the <i>ddm-name</i> you can use a single DDM name (up to 32 characters) or a range as for <i>object-name-range</i> to display a list of a certain range of DDMs.</p>

Note:

Instead of the keyword DDM, you can also use the keyword VIEW (or V for short).

Options

In place of *options*, you may specify one of the options shown below.

{ [[WITH] DIRECTORY] [NUMBERS OFF] [<i>expand-option</i>] } <i>formatted-option</i> CONVERTED

DIRECTORY	This option first displays the directory information (as described below, see <i>Displaying Directory Information</i>) on the specified object and then lists the source code of the object.
NUMBERS OFF	By default, the source code of an object will be listed with source-code line numbers. To list it without line numbers, specify the NUMBERS OFF option. (See also subcommands NUMBERS ON/NUMBERS OFF in the section <i>Subcommands for Listed Source</i> .)
CONVERTED	By default, the source is listed in the code page as stored on the system file. To list the source in the default code page (see system variable <code>*CODEPAGE</code>), specify this option.

expand-option

EXPAND [FORMATTED] [{ COMMENTS }] [<i>expand-type</i> ...10] [{ <i>object-name</i> }] [{ <i>object-name-range</i> }]
--

EXPAND <i>object-name</i>	<p>With the EXPAND option, you can have the sources of other objects referenced by the listed object - copycodes, data areas, maps, help routines, external subroutines, subprograms, programs called with a FETCH statement, error messages - listed <i>within</i> the source of the listed object. This option is particularly useful in batch mode.</p> <p>For example, if a listed source program contains an INCLUDE statement, you can have the source code of the included copycode listed within the listed source program immediately after the INCLUDE statement.</p>
EXPAND <i>object-name-range</i>	<p>Objects listed within a source will be referred to as "expand objects" in the explanations below.</p> <p>Subcommands in Expand Object</p> <p>Within a listed expand object, only the following subcommands are available:</p> <pre>PRINT + - - .</pre> <p>See <i>Examples of List of Objects Usage</i>.</p>

EXPAND FORMATTED	<p>The <code>EXPAND FORMATTED</code> option is only relevant for stowed data areas (where time stamp of source object and cataloged object are identical) and maps listed within a source.</p> <p>For data areas, the following applies:</p> <ul style="list-style-type: none"> ● If <code>FORMATTED</code> is not specified, the display of the data area will resemble that in the data area editor ● If <code>FORMATTED</code> is specified, the display of the data area will resemble a <code>DEFINE DATA</code> statement. This only applies to stowed data areas (i.e. the time stamp of source object and cataloged object are identical); see also subcommand <code>FORMAT</code>. <p>For maps, the following applies:</p> <ul style="list-style-type: none"> ● If <code>FORMATTED</code> is not specified, the map <i>source</i> will be listed. ● If <code>FORMATTED</code> is specified, the map <i>layout</i> will be displayed (that is, the map as it is displayed to the users at runtime).
EXPAND COMMENTS	<p>If you use the option <code>EXPAND COMMENTS</code>, only the initial comment lines of the expand object will be listed; that is, the expand object will be listed until (but not including) the first source-code line which is not a comment line.</p>
EXPAND <i>n</i>	<p>If you use the option <code>EXPAND <i>n</i></code>, only the first <i>n</i> lines of the expand object will be listed.</p> <p>If you use neither of these two options, the entire expand object will be listed.</p>

<i>expand-type</i>	As <i>expand-type</i> , you specify the object type(s) of the expand object(s). The following <i>expand-types</i> can be specified:	
	P	Programs
	N	Subprograms
	S	External subroutines
	H	Help routines
	G	Global data areas
	L	Local data areas
	A	Parameter data areas
	M	Maps
	C	Copycodes
	E	Error messages
	4	Classes
	*	All object types
<i>object-name</i>	As <i>object-name</i> or <i>object-name-range</i> , you specify the name(s) of the expand object(s) to be listed within the main listed source.	
<i>object-name-range</i>	The same notations are possible as for <i>object-name</i> or <i>object-name-range</i> , except < and >.	

formatted-option

FORMATTED ['c'] ['c'] [SETTINGS] <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>{</td> <td>FIELDS</td> <td>}</td> </tr> <tr> <td>{</td> <td>EXTFIELDS</td> <td>}</td> </tr> </table> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>{</td> <td>RULES</td> <td>}</td> </tr> <tr> <td>{</td> <td>INLINERULES</td> <td>}</td> </tr> <tr> <td>{</td> <td>FREERULES</td> <td>}</td> </tr> <tr> <td>{</td> <td>AUTORULES</td> <td>}</td> </tr> </table>	{	FIELDS	}	{	EXTFIELDS	}	{	RULES	}	{	INLINERULES	}	{	FREERULES	}	{	AUTORULES	}
{	FIELDS	}																
{	EXTFIELDS	}																
{	RULES	}																
{	INLINERULES	}																
{	FREERULES	}																
{	AUTORULES	}																

FORMATTED Option

The FORMATTED option applies to stowed data areas (where time stamp of source object and cataloged object are identical) and maps:

FORMATTED	<p>Stowed Data Area:</p> <p>If you specify the <code>FORMATTED</code> option for a data area, the data area will be displayed formatted; that is, the display resembles a <code>DEFINE DATA</code> statement; see also subcommand <code>FORMAT</code>.</p> <p>This only applies to stowed data areas (i.e. the time stamp of source object and cataloged object are identical). By default, data areas are displayed unformatted; that is, the display resembles that in the data area editor.</p> <p>The default setting can be changed with in the List Profile. (Refer to <i>Defining an Individual List Profile</i> below and see also subcommand <code>FORMAT</code>).</p> <p>Map:</p> <p>If you specify the <code>FORMATTED</code> option for a map, the map <i>layout</i> will be displayed; that is, the map as it is displayed to the users at runtime.</p>
------------------	--

FORMATTED Options for Listing Maps

When you are listing maps, you may specify options in addition to the keyword `FORMATTED`:

<code>['c'] ['c']</code>	<p>Using Filler Characters:</p> <p>You may specify filler characters <i>c</i> for input fields (<code>AD=A</code> and <code>AD=M</code>) and output fields (<code>AD=O</code>) to make these fields visible. You may specify any character as filler character.</p> <p>The following example shows all input fields with an underscore (<code>_</code>) and all output fields with a hash (<code>#</code>).</p> <pre>LIST MAP map-name FORMATTED ' _ ' '#'</pre>
SETTINGS	<p>Map Settings:</p> <p>Displays the map settings of the specified map.</p> <pre>LIST MAP map-name FORMATTED SETTINGS</pre>
FIELDS	<p>Field Summary:</p> <p>Displays the field summary; that is, the list of fields in the specified map.</p> <pre>LIST MAP map-name FORMATTED FIELDS</pre>
EXTFIELDS	<p>Extended Field Editing Information:</p> <p>Causes the extended field editing information for all map fields to be displayed.</p> <pre>LIST MAP map-name FORMATTED EXTFIELDS</pre>

Displaying Processing Rules for a Map

The following options cause the processing rules used by the map to be displayed. The rules are displayed in order of fields to which they are assigned, and per field in order of rank.

RULES	<p>All Rules:</p> <p>LIST MAP <i>map-name</i> FORMATTED RULES</p> <p>Displays <i>all</i> the rules for the specified map.</p>
INLINERULES	<p>Inline Rules Only:</p> <p>LIST MAP <i>map-name</i> FORMATTED INLINERULES</p> <p>Displays only the inline rules for the specified map.</p>
FREERULES	<p>Free Rules Only:</p> <p>LIST MAP <i>map-name</i> FORMATTED FREERULES</p> <p>Displays only the free rules for the specified map.</p>
AUTORULES	<p>Automatic Rules Only:</p> <p>LIST MAP <i>map-name</i> FORMATTED AUTORULES</p> <p>Displays only the automatic rules for the specified map.</p>

See also the subcommands LAYOUT and FORMAT in the section *List of Source*.

List of Objects

When you use asterisk or wildcard notation for the object name, you get a list of all objects that meet the specified selection criteria. On this list, you can then select objects for display, print, etc. by marking them with a function code, or you can enter a Natural system command or a LIST subcommand in the command line.

This section describes the functions, subcommands and function codes that are available in the list of objects which is displayed, for example, after you have issued a LIST * command. The following topics are covered:

- Explanation of the Column Headers
- Scrolling the Selection List of Objects
- New Criteria for the Selection List
- Information Displayed on the Selection List
- Items Intensified on the Selection List
- Subcommands for the Selection List
- Performing a Function on an Object
- Sorting the List of Objects
- Examples of List of Objects Usage

Explanation of the Column Headers

The list of objects contains the following columns:

Column	Explanation
Cmd	In this column, you can enter a code to perform a function on an object in the selection list. See <i>Performing a Function on an Object</i> .
Name	Name of object.
Type	Type of object.
S/C	Indicates whether the object exists as source (S) and/or cataloged object (C).
SM	The Natural programming mode that was used when the object was created. S = structured mode, R = reporting mode.
Version	Product version of Natural that was used to create or catalog the object.
User ID	User ID of the user who created or cataloged the object.
Date, Time	Date and time when the object was created or cataloged.

Scrolling the Selection List of Objects

Once a list of objects is displayed, you can scroll it as follows:

- To scroll the list one page forward or backward, press PF8 or PF7 respectively.
- To scroll the list to its beginning or end, press PF6 or PF9 respectively.

New Criteria for the Selection List

When a list of objects is displayed, the fields immediately underneath the column headings show the selection criteria for the current list. You can change the selection criteria by overwriting the values of these fields. For information on the possible values for one of these fields, you enter a question mark (?) in the field.

Information Displayed on the Selection List

If there exists both a source and an object module for an object (as indicated in the column S/C), the information displayed refers to the source, not the object module.

Note:

When the sort function is active the source and the object module may be displayed separately, e.g. when the list is sorted by the object date and the source and the object module have different date values.

To display more information on source and cataloged objects

- Press PF11 to shift right.

Or:

Press PF10 to shift left.

Note:

By default the number of source lines of source objects is not calculated due to performance reasons. If you want the number of source lines of source objects being displayed, you can either enter the subcommand `COUNTSOURCE ON` or set in the `LIST` profile (see *Defining an Individual List Profile* below) the parameter `COUNT-SOURCE-LINES` to `Y`.

Items Intensified on the Selection List

If an item is displayed intensified on the left-most list page, this indicates that there is a discrepancy between the object's source and its object module. For information on the discrepancy, you may mark the object with the function code `LD` to list its directory information. To eliminate the discrepancy, it is usually sufficient to stow the object again (function code `ST`).

Subcommands for the Selection List

In a list of objects, you can enter a Natural system command or a `LIST` subcommand in the command line. Valid subcommands are:

Code	Function	
CODE-PAGE or CP	ON	Display the code page information for each object. This is the default value.
	OFF	Do not display the code page information.
SC	List only objects containing a scan value (can only be used with long list).	
SC OFF	Switch off scan mode.	
SHORT	Display a short list of objects, i.e., display only the object names (can only be used if scan mode is off).	
LONG	Switch to long list including all fields available.	
PRINT	Print the list of objects.	
<u>EXTENDED</u>	Display the list of long names of subroutines/classes; same as <code>LIST EXTENDED *</code> .	
ALL <i>fx</i>	Enter the function code <i>fx</i> (where <i>fx</i> is a valid function code for a listed object) for all displayed objects.	
SORT	Invokes the sort window (see <i>Sorting the List of Objects</i> below).	
<u>COUNTSOURCE</u>	ON	Display the number of source lines for source objects.
	OFF	Do not display the number of source lines for source objects.

Code	Function	
MARK-LONG-LINES	ON	Mark long lines in the list of a source object with an L in the first two positions. The default value can be specified in the LIST profile; see <i>Defining an Individual List Profile</i> .
	OFF	Do not mark long lines in the list of a source object.
DEFINE-DATA	ON	A listed data area source is listed in DEFINE DATA format by default (same as LIST <i>dataarea</i> FORMATTED). The default value can be specified in the LIST profile; see <i>Defining an Individual List Profile</i> .
	OFF	A listed data area source is listed unformatted.
<u>LISTPROFILE</u>	Display the current value of the parameters of the LIST profile (see <i>Defining an Individual List Profile</i> below).	
<u>NOCOPT</u>	Displays a list of the cataloged objects that are compiled with Natural Optimizer Compiler (NOC), together with the initial NOC options used during CATALOG; same as LIST NOCOPT *, see <i>Displaying NOC Options of Cataloged Objects</i> .	
<u>OPTIONS</u>	Displays a list of the cataloged objects together with the initial compiler options used during catalog; same as LIST OPTIONS *, see <i>Displaying Compiler Options of Cataloged Objects</i> .	
<u>REUSE</u>	ON	Switch on reuse mode. The last displayed list is reused after execution of commands entered in the Cmd column, except for the following commands: E ED (Edit) CA (Catalog) UC (Uncat) S ST (Stow) D DE (Delete) RE (Rename)
	OFF	Switch off reuse mode. The list is rebuilt after execution of commands entered in the Cmd column.
<u>REFRESH</u>	Rebuild the currently displayed list. This subcommand can be used especially when reuse mode is switched on.	
+	Scroll one page forward.	

Code	Function
-	Scroll one page backward.
++	Scroll to the end (bottom) of the object list.
--	Scroll to the beginning (top) of the object list.
?	Command line help.

Performing a Function on an Object

To perform a function on an object in the selection list, you simply mark the object with the appropriate function code in the left-hand column (titled **Cmd**).

You can mark several objects on the selection list with different function codes; the functions will then be performed one after the other.

The following function codes are available (possible abbreviations are underlined).

Code	Function
?	<p>A window will be displayed which shows all the functions available for the marked object. The window will only list those functions that are actually available for the selected object (for example, if the object is a subroutine, it cannot be run; if the object is only available in source form, it cannot be executed).</p> <p>From the window you can select the function to be performed on the marked object.</p>
CA	Compile the object and store it in object form (equivalent to the system command CATALOG).
<u>DE</u>	Delete the object (equivalent to the system command DELETE).
DL	Download object from mainframe to personal computer (only available if Natural Connection is installed).
<u>ED</u>	Edit the object's source code (equivalent to the system command EDIT).
EX	Execute the object (equivalent to the system command EXECUTE).
LC	List object's source code converted into the default code page *CODEPAGE, (equivalent to LIST <i>object-name</i> CONVERTED).
LD	List directory information (equivalent to LIST DIRECTORY <i>object-name</i>) on the object.
LE	List object's source code in expanded form (equivalent to LIST <i>object-name</i> EXPAND *).
LF	Display a data area or map formatted (equivalent to LIST <i>object-name</i> FORMATTED).
<u>LI</u>	List the object's source code.
LN	Display long name of subroutine or class (only possible if a cataloged object exists) or resource.
NO	Displays the Natural Optimizer Compiler (NOC) options used during CATALOG (only possible if a catalog object exists).
<u>QP</u>	<p>Displays the initial, final and changed Natural compiler options used during CATALOG (only possible if a cataloged object exists).</p> <p>The initial and changed compiler options can be displayed for objects cataloged with Natural Version 4.2.5 or above only.</p>
<u>PR</u>	Print the object's source code.
RE	Rename the object (equivalent to the system command RENAME).
<u>RU</u>	Run (that is, compile and execute) the object's source code (equivalent to the system command RUN).
<u>ST</u>	Stow the object in source and object form (equivalent to the system command STOW).
UC	Delete the object module (equivalent to the system command UNCATALOG).
.	Exit (from selection list window)

Sort Field	Keyword in Sort Syntax
Natural object type	TYPE
Programming mode (reporting or structured mode)	MODE
Version	VERSION
User ID	USER
Date	DATE
Date and time	DATETIME
Time	TIME
Source size	SIZE
Number of source lines	LINES
Buffer pool size	BPSIZE
DATSIZE (size of local data buffer)	DS/DATSIZE
Code page	CP/CODE-PAGE
Member names of subroutines or classes (available in extended selection list only)	MEMBER

Once the sort has been started, all changes in the Criteria for the Selection List create a sorted list.

To switch off the sort mode

- Enter the subcommand `SORT OFF`.

Or:

Deactivate the sort function in the **Sort Options** window invoked by pressing PF4.

The sorted list is built in a Natural text object in library `WORKPLAN`. The name of the text object is generated by the `LIST` command. If the `LIST` profile is activated (see *Defining an Individual List Profile* below) the name of the text object and the library can be specified in the `LIST` profile.

Examples of List of Objects Usage

LIST *	Lists all objects in the current library.
LIST S *	Lists all subroutines in the current library.
LIST SYS*	Lists all objects (of any type) whose names begin with SYS.
LIST M SYS*	Lists all maps whose names begin with SYS.
LIST C *CODE	Lists all copycodes whose names end with CODE.
LIST NAT*AL	Lists all objects whose names begin with NAT and end with AL no matter which and how many other characters are between NAT and AL (this would include the names NATURAL and NATIONAL as well as NATAL).
LIST DOO?	Lists all objects with 4-character names beginning with DOO (this would include the names DOOR and DOOM, but not DOO or DOODLE).
LIST M NAT?AL	Lists all maps whose names begin with NAT and end with AL with exactly one character are between NAT and AL (this would include the names NAT1AL and NAT2AL, but not NATAL or NATIONAL).
LIST M *1*	Lists all maps whose names contains a 1.
LIST M F>	Lists all maps, starting from the first one whose name begins with F.
LIST M MA<	Lists all maps, from the first one until the one named MA (if present).
LIST N?T*AL	Lists all objects such as NATAL, NATURAL, NATvrAL (where vr stands for the relevant version and release numbers).
LIST E* TYPE=PM KIND=S DATE=YEAR SORTED=DATE ASCENDING	Creates a list of all source objects of Programs and Maps whose names start with E and which were saved in the current year. The list is sorted by object date in ascending order.

List of Source

The following topics are covered below:

- Subcommands for Listed Source
- Subcommand FORMAT
- Cursor-Sensitive Object Selection

Subcommands for Listed Source

When you have the source code of an object listed, you can enter in the command line one of the following subcommands.

Subcommand	Function
+	Scrolls one page forward.
-	Scrolls one page backward.

Subcommand	Function
++	Scrolls to the end (bottom) of the source.
<u>B</u> OTTOM	
--	Scrolls to the beginning (top) of the source.
<u>T</u> OP	
+ <i>n</i>	Scrolls <i>n</i> lines forward.
- <i>n</i>	Scrolls <i>n</i> lines backward.
<i>nnnn</i>	Scrolls to line number <i>nnnn</i> .
<u>C</u> ONVERTED	See <u>C</u> ONVERTED in <i>Options</i> .
DBFNR ON	Displays the database id (DBID) and file number (FNR) of the source library in the header line of the source code.
DBFNR OFF	Displays the header line of the source code without the database id (DBID) and file number (FNR) of the source library. This is the default value.
<u>E</u> XPAND	See <i>expand-option</i> .
<u>F</u> IELDS	Applies to maps only: displays the field summary; that is, the list of fields in the map.
FIND	Displays only those source lines which contain the specified <i>value</i> .
FIND <i>value</i>	If you enter only the command FIND itself, a window will be displayed in which you can enter the <i>value</i> to be sought and specify whether the search is to be absolute or not.
FIND <u>A</u> BSOLUTE <i>value</i>	If you specify FIND without ABSOLUTE after the command, the <i>value</i> will only be found if it is an isolated word. This is the default. If you specify ABSOLUTE after the command, the <i>value</i> will also be found if it is part of a larger string of characters.
<u>F</u> ORMAT	Applies to data areas and maps only: displays <i>formatted</i> data area or map, and other items related to the map.
<u>L</u> AYOUT	Applies to maps only: displays the map layout; that is, the map will be displayed as it is displayed to the users at runtime.

Subcommand	Function
<u>N</u> UMBERS ON	Displays the source code with line numbers. This is the default value.
<u>N</u> UMBERS OFF	Displays the source code without line numbers.
<u>P</u> RINT	Prints the listed source.
REF	Displays the line numbers of the source-code lines which contain the specified <i>value</i> in a table.
REF <i>value</i>	If you enter only the command REF itself, a window will be displayed in which you can enter the <i>value</i> to be sought for and specify whether the search is to be absolute or not.
REF <u>A</u> BSOLUTE <i>value</i>	If you specify REF without ABSOLUTE after the command, the <i>value</i> will only be found if it is an isolated word. This is the default. If you specify ABSOLUTE after the command, the <i>value</i> will also be found if it is part of a larger string of characters.
<u>R</u> ULES	Applies to maps only: displays the processing rules used by the map (the rules are displayed in order of fields to which they are assigned, and per field in order of rank).
<u>S</u> CAN	Displays all lines intensified which contain the specified <i>value</i> . The source will be scrolled to the first line that contains the <i>value</i> .
<u>S</u> CAN <i>value</i>	If you enter only the command SCAN itself, a window will be displayed in which you can enter the <i>value</i> to be sought for and specify whether the search is to be absolute or not.
<u>S</u> CAN <u>A</u> BSOLUTE <i>value</i>	If you specify SCAN without ABSOLUTE after the command, the <i>value</i> will only be found if it is an isolated word. This is the default. If you specify ABSOLUTE after the command, the <i>value</i> will also be found if it is part of a larger string of characters.

Subcommand	Function
SCAN= or SC=	Scans for the next occurrence of the last SCAN <i>value</i> (or press PF5).
SETTINGS	Applies to maps only: displays the map settings of the map.
ZOOM [<i>expand-type</i> ...10] <i>object-name</i>	<p>Specifying a single <i>object-name</i> with the ZOOM command has the same effect as marking the name in the listed source with the cursor (see the section <i>Cursor-Sensitive Object Selection</i>): the selected object will be displayed in a window.</p> <p>If you use asterisk/wildcard notation for the <i>object-name</i> or the <i>object-name-range</i>, all selected objects will be displayed in a window in the sequence in which they are referenced in the listed source.</p>
ZOOM [<i>expand-type</i> ...10] <i>object-name-range</i>	<p>The specification of an <i>expand-type</i> is the same as for the <i>expand-option</i>.</p> <p>For an object displayed within a window invoked by ZOOM, the same subcommands (except PRINT, EXPAND and ZOOM) are available as for the normal listed source. Moreover, if you have used asterisk or wildcard notation and several objects are displayed, you can use the commands NEXT and PREV (or PF4 and PF5) to move from one object in the window to the next one or previous one respectively.</p>
.	Exit.

Note:

By default, the database id (DBID) and file number (FNR) of the source library are not displayed in the header line of the listed source. If you want the DBID and FNR of the source library to be displayed, you can either enter the subcommand DBFNR ON or, in the LIST profile, set the parameter SOURCE-LIST-WITH-DBID-FNR to "Y" (see *Defining an Individual List Profile* below).

Subcommand FORMAT

This subcommand only applies to stowed data areas (where time stamp of source object and cataloged object are identical) and maps.

For data areas, this subcommand corresponds to the option FORMATTED.

In the List Profile, you can specify how data areas are listed by default:

- formatted (that is, the display resembles a DEFINE DATA statement) or
- unformatted (that is, the display resembles that in the Natural data area editor).

In the List of Objects, you can use the subcommand DEFINE-DATA ON/OFF to set the default for the time the LIST command is being executed.

If data areas are listed formatted by default and if it is not possible to convert the data area source code into DEFINE DATA format, a corresponding message is displayed and the data area is listed unformatted.

When you enter the subcommand FORMAT for a map, a window will be displayed in which you can select one or more additional items related to the map to be displayed:

- Map settings (corresponds to subcommand SETTINGS).
- Map layout (corresponds to subcommand LAYOUT). When you select this item, you have the option to specify filler characters for input fields (AD=A and AD=M) and output fields (AD=O) to make these fields visible. You may specify any character as filler character.
- Field summary (corresponds to subcommand FIELDS).
- Processing rules (corresponds to subcommand RULES).

The items you select are displayed one after the other in the order in which they appear in the selection window.

In FORMAT mode, the same subcommands for scrolling - except B - and the subcommands FIELDS, LAYOUT, PRINT, RULES and SETTINGS are available as for a normal listed source (see above). Additional subcommands are available as described below for each item.

- Additional Subcommands for Map Layout
- Additional Subcommands for Field Summary List
- Additional Subcommands for Processing Rules

Additional Subcommands for Map Layout

$S > n$	Shift map layout n columns to the right.
$S < n$	Shift map layout n columns to the left.

Additional Subcommands for Field Summary List

<u>EXTEND</u>	Displays the extended field editing information for all map fields. To have the extended field editing information for an individual field displayed, mark the field name on the field summary list with the cursor and press ENTER.
<u>RULES</u> <i>nn</i>	Displays the processing rules attached to field <i>nn</i> (<i>nn</i> being the sequential field number (first column of the field summary list)). To have the processing rules of a field displayed, you can also enter an R in the command line and then mark the field name on the field summary list with the cursor and press ENTER.
<u>SCAN</u> [<u>ABSOLUTE</u>] <i>value</i>	Same as for <i>Subcommands for Listed Source</i> .
<u>SCAN</u> =	

Additional Subcommands for Processing Rules

<u>SCAN</u> [<u>ABSOLUTE</u>] <i>value</i>	Same as for <i>Subcommands for Listed Source</i> .
<u>SCAN</u> =	

Cursor-Sensitive Object Selection

Within a source that is being listed, you can mark with the cursor the *name* of an object referenced within that source, and the source of the selected object will be listed in a window.

For the source displayed within the window, the same subcommands - except PRINT, EXPAND and ZOOM - are available as for the "normal" listed source.

Defining an Individual List Profile

You can define an individual profile for the LIST command. For this purpose, Natural provides the text object LISTPROF in the library SYSLIB.

In LISTPROF, you can enter general or user-specific profiles with corresponding defaults, such as COUNT-SOURCE-LINES. These defaults are used when you start the LIST command.

To activate the values defined in LISTPROF

1. Copy the text object LISTPR-S from library SYSLIB to any library.
2. Add the changes.
3. Save the text object LISTPR-S under the name LISTPROF.

4. Copy the text object LISTPROF to library SYSLIB.
5. Invoke the LIST command.

For a detailed description, see text object LISTPR-S in library SYSLIB.