

Retrieving Information For Individual Members

This section covers the following topics:

- Two Ways to Retrieve Information on Individual Members
 - Information about Indirect Calls
 - Which Invocation Methods are Evaluated
 - Invoked Programs (Function I)
 - Error Numbers (Function E)
 - Data Areas and Variables (Function D)
 - Views and Their Fields (Function V)
 - Copy Code (Function C)
 - Printers (Function P)
 - Workfiles (Function W)
 - Retained Sets (Function S)
 - Processing Rules (Function R)
 - Natural Resources (Function F)
 - Report Programs with XRef Data (Function X)
-

Two Ways to Retrieve Information on Individual Members

Information on individual Natural members is either retrieved top-down or bottom-up. See also diagrams below. In both cases the retrieval starts with an individual member (program, data area, etc.):

- **Top-down Retrieval**

The functions look for objects that are used by a given member.

For example: The function Program using programs. A program is specified and all programs that are used by this program are shown.

- **Bottom-up Retrieval**

The functions look for objects that use a given member. The notation referenced in is always used in retrieval functions of this type.

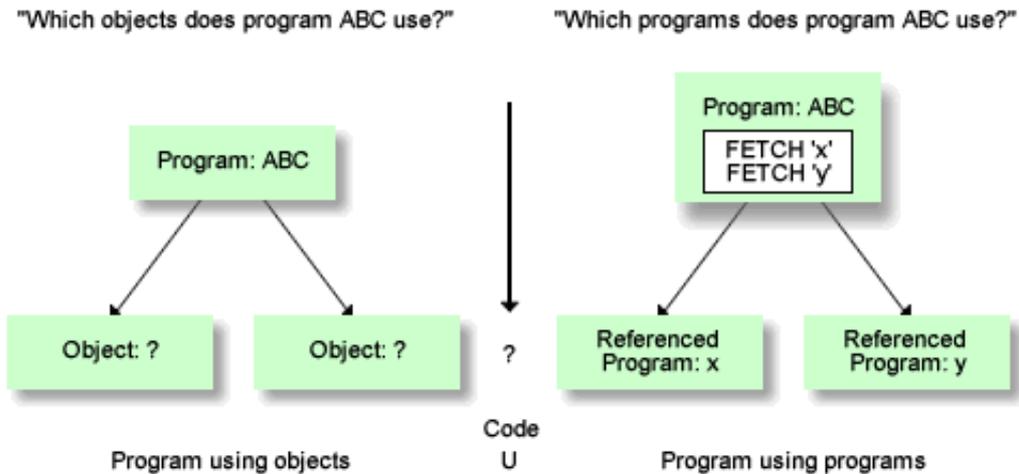
For example: The function Program referenced in programs. A program is specified and all programs that use this program are listed.

Note:

In the screens Data Areas and Variables and Copy Code only starting objects in the current library are taken into account.

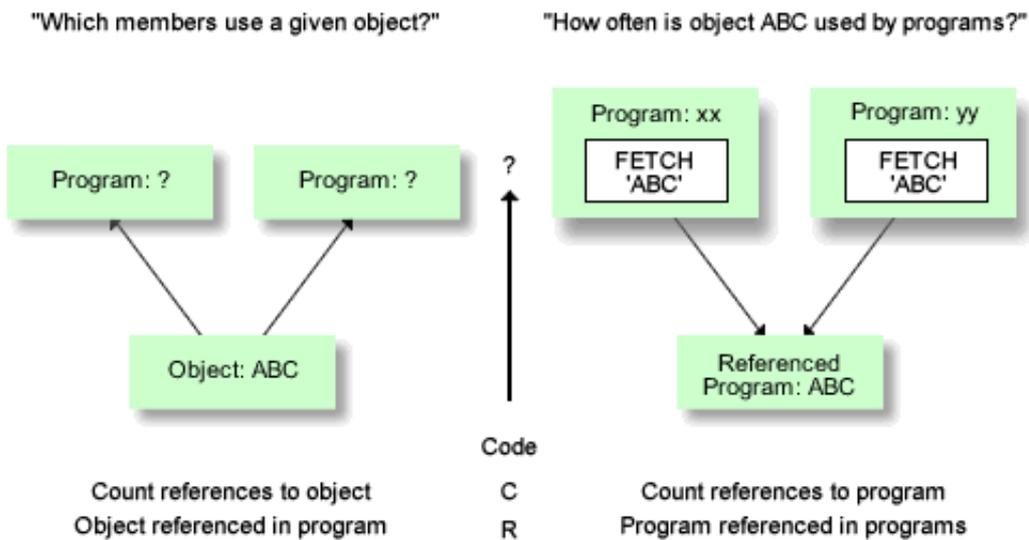
Top-Down: Finding Objects that a Given Member is Using

Top-down retrieval provides answers to question such as these:



Bottom-Up: Finding Members that Reference a Given Object

Bottom-up retrieval provides answers to question such as these:



Information about Indirect Calls

Indirect references can be evaluated, for example, "Which programs refer to programs which refer to the program ABC?"

Recursion can be extended up to seven levels to show the invocation structure of an application.

Which Invocation Methods are Evaluated

When Natural finds in a program a reference to another program, it stores not only the type and name of the program that is used but also how that program is invoked. The following invocation methods are reported by LIST XREF functions:

- CALL statements to 3GL programs;
- CALLNAT statements;
- FETCH statements;
- FETCH RETURN statements;
- Assignment of a program to a function key (using a SET KEY PFn= statement);
- INPUT USING MAP statements;
- CALLDBPROC statements to SQL stored procedures;
- Function call to Natural Functions;
- REINPUT USING HELP statements or an assignment via session parameter HE in an INPUT statement or within a map;
- PERFORM statements;
- RUN statements;
- STACK statements;
- CREATE OBJECT statements;
- SEND METHOD statements;
- Open dialog;
- SEND EVENT statements;
- Using property;
- Method implementation;
- Calling an entry in a DBRM (database request module) - only for Natural programs that use static SQL;
- Assignment of a program as an error transaction, a startup transaction or a restart transaction:
 - either by assignment in a Natural program (using a *ERROR-TA= or *STARTUP= statement)
 - or by definition in Natural Security for a Natural library or a special link to this library, provided that XREF for the application is set to either Y or F in Natural Security and that a user system file has been assigned to the library. These transactions are considered to be invoked by a dummy member *NSC (a program of type P) in the library.
- INVESTIGATE statements (invocation of a Natural Expert model).

Invoked Programs

```

13:30:57          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: PDCLASS          - Invoked Programs -          DBnr: 180 Fnr: 54

Code Function                                     Ss Pg PT RP PT Vi FE RD
-----
D Count references in program                    * 0 * 0 0 0 *
C Count references to program                    * 0 * 0 0 0 *
U Program using programs                          0 * 0 * 0 0 *
R Program referenced in programs                 0 * 0 * 0 0 *
E Program using programs recursively             * 0 * 0 0 0 0
I Program referenced in programs recursively     * 0 * 0 0 0 * 0
X XRef program using programs                   0 * 0 * 0 0 0
Y XRef program referenced in programs           0 * 0 * 0 0 0
-----
Code: ?          Save set: N (Y,N)
Using program:   Program type: (?)
Referenced program: Program type: (?) via:          (?)
Subroutine/Entry/Class/Method/Property/Event:
Rec. depth: 7 (1-7)
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit
    
```

The functions display active reference information about programs that are invoked from Natural programs. The input screen is displayed with code I in the XRef menu.

Code	Function	Answer to the question	Comments
D	Count references in program	How many programs does the program ABC refer to?	The total number of referenced programs is displayed and is listed for each invocation method
C	Count references to program	How many times is the program ABC referred to by programs	The total number of references is listed for each invocation method.
U	Program using programs	Which programs does the program ABC refer to	The invocation method is indicated. See also Effects of Steplib Support.
R	Program referenced in programs	Which programs refer to the program ABC	The invocation method is indicated. See also Effects of Steplib Support.
E	Program using programs recursively	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC,	The output is structured to show the direct references to programs on the next level. See also Termination of Listing.
I	Program referenced in programs recursively	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC,	The output is structured to show the direct references to programs on the next level. See also Termination of Listing.
X	XREF program using programs	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC,	The total number of programs at each level is reported as well as the type and ID of each program. This function summarizes the output of the function with code E but omits the structural information.
Y	XREF program referenced in programs	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC,	The total number of programs at each level is reported as well as the type and ID of each program. This function summarizes the output of the function with code I but omits the structural information.

Termination of Listing

In certain circumstances, the listing of a LIST XREF function is terminated and marked as follows:

- Recursion is marked with **rec** and the output is terminated.
- The output can be truncated from a level (up to 7) specified by the user. Truncated output is marked with **trunc**.
- The user can specify other parameters that limit the scope of the output, for example programs of a certain type or those with IDs that start with a particular character string. Output terminated in such a manner or due to a reference to a 3GL program is marked with **stop**.
- If a program that has already been displayed in the invocation hierarchy reappears at a lower recursion level, the hierarchy is not displayed again. Instead the program is marked with **suppr.** (for suppressed).
- If the called program is contained in another library, the output is stopped and the program is marked with **steplib**.

Save set	Any user defined in Predict can save the types and names of programs listed by functions U, R, X or Y in a set. See Using Sets. Note: Sets only contain the resulting objects of the called function.																						
Using program	The meaning of this parameter depends on the function: <ul style="list-style-type: none"> ● For the functions D, U, E and X, the ID of the program for which the function is to be executed. ● For the functions C, R, I and Y, the program or programs that use the program for which the function is to be executed. <p>Asterisk notation is possible.</p> <p>Note: Enter *NSC to list members that are assigned in Natural Security as startup, restart, end or error transactions.</p>																						
Program type	The type of the Natural program specified as the Using program. One of the following values can be specified: <table border="1"> <tr> <td>F</td> <td>function</td> </tr> <tr> <td>H</td> <td>help routine</td> </tr> <tr> <td>K</td> <td>ISPF Macro</td> </tr> <tr> <td>M</td> <td>map</td> </tr> <tr> <td>N</td> <td>subprogram</td> </tr> <tr> <td>P</td> <td>Main program</td> </tr> <tr> <td>S</td> <td>subroutine</td> </tr> <tr> <td>T</td> <td>dialog</td> </tr> <tr> <td>Y</td> <td>Natural Expert model</td> </tr> <tr> <td>O</td> <td>command processor</td> </tr> <tr> <td>4</td> <td>class</td> </tr> </table>	F	function	H	help routine	K	ISPF Macro	M	map	N	subprogram	P	Main program	S	subroutine	T	dialog	Y	Natural Expert model	O	command processor	4	class
F	function																						
H	help routine																						
K	ISPF Macro																						
M	map																						
N	subprogram																						
P	Main program																						
S	subroutine																						
T	dialog																						
Y	Natural Expert model																						
O	command processor																						
4	class																						

Referenced program	<p>The meaning of this parameter depends on the function:</p> <ul style="list-style-type: none"> ● For the functions U, E and X: the ID of the programs that are used by the program for which the function is executed. ● For the functions C, R, I and Y: the ID of the program for which the function is to be executed. <p>If asterisk notation is used, the function is applied to all programs which meet the given specification.</p> <p>Note: Enter *DYNAMIC to list all programs that invoke other programs via variables. An example: ASSIGN #A = 'SUB1' CALLNAT #A</p>																																												
Program type	<p>The type of the program specified as the Referenced program. See the list of program types above.</p>																																												
Via	<p>The method used to invoke programs. Valid values:</p> <table border="1" data-bbox="470 891 1441 2018"> <tr><td>C</td><td>CALL statement</td></tr> <tr><td>CN</td><td>CALLNAT statement</td></tr> <tr><td>CP</td><td>Command processor</td></tr> <tr><td>CR</td><td>CREATE OBJECT</td></tr> <tr><td>F</td><td>FETCH statement</td></tr> <tr><td>FR</td><td>FETCH RETURN statement</td></tr> <tr><td>H</td><td>REINPUT USING HELP statement or HE= assignment</td></tr> <tr><td>I</td><td>INVESTIGATE statement</td></tr> <tr><td>IR</td><td>Invoke return (stands for all methods that return control to the invoking program: CALL, CALLNAT, FETCH RETURN, Help, INVESTIGATE, Map, PERFORM and Static SQL).</td></tr> <tr><td>K</td><td>SET KEY statement</td></tr> <tr><td>M</td><td>INPUT USING MAP statement</td></tr> <tr><td>ME</td><td>Method implementation</td></tr> <tr><td>O</td><td>Open dialog</td></tr> <tr><td>P</td><td>PERFORM statement</td></tr> <tr><td>PR</td><td>Property</td></tr> <tr><td>R</td><td>RUN statement</td></tr> <tr><td>S</td><td>STACK statement</td></tr> <tr><td>SE</td><td>Send event</td></tr> <tr><td>SM</td><td>Send method</td></tr> <tr><td>SQ</td><td>Using static SQL</td></tr> <tr><td>T</td><td>Assignment to a Natural Security transaction</td></tr> <tr><td>U</td><td>Function</td></tr> </table>	C	CALL statement	CN	CALLNAT statement	CP	Command processor	CR	CREATE OBJECT	F	FETCH statement	FR	FETCH RETURN statement	H	REINPUT USING HELP statement or HE= assignment	I	INVESTIGATE statement	IR	Invoke return (stands for all methods that return control to the invoking program: CALL, CALLNAT, FETCH RETURN, Help, INVESTIGATE, Map, PERFORM and Static SQL).	K	SET KEY statement	M	INPUT USING MAP statement	ME	Method implementation	O	Open dialog	P	PERFORM statement	PR	Property	R	RUN statement	S	STACK statement	SE	Send event	SM	Send method	SQ	Using static SQL	T	Assignment to a Natural Security transaction	U	Function
C	CALL statement																																												
CN	CALLNAT statement																																												
CP	Command processor																																												
CR	CREATE OBJECT																																												
F	FETCH statement																																												
FR	FETCH RETURN statement																																												
H	REINPUT USING HELP statement or HE= assignment																																												
I	INVESTIGATE statement																																												
IR	Invoke return (stands for all methods that return control to the invoking program: CALL, CALLNAT, FETCH RETURN, Help, INVESTIGATE, Map, PERFORM and Static SQL).																																												
K	SET KEY statement																																												
M	INPUT USING MAP statement																																												
ME	Method implementation																																												
O	Open dialog																																												
P	PERFORM statement																																												
PR	Property																																												
R	RUN statement																																												
S	STACK statement																																												
SE	Send event																																												
SM	Send method																																												
SQ	Using static SQL																																												
T	Assignment to a Natural Security transaction																																												
U	Function																																												

Subroutine/Entry/Class/Method/Property/Event:	<p>One of the following:</p> <ul style="list-style-type: none"> ● the function of a Natural subroutine that is performed, or ● the point of entry into an external (non-Natural) program or DBRM that is called. ● the class, method or property name. <p>This parameter applies when a program is invoked without using its member name. To identify the (non-Natural) program to which an external entry belongs, Predict uses information from Adabas Native SQL or the Predict Preprocessor. To identify the DBRM to which an external entry belongs, Predict uses information created by the function CREATE DBRM of Natural DB2.</p>
Rec. depth	The recursion depth for the functions E, I, X and Y. A number from 1 to 7.

Effects of Steplib Support

Steplib support affects Invoked Programs functions as follows. A distinction is made between Top-down and Bottom-up functions:

Top-down

Example: Function Program using program

09:50:43			***** P R E D I C T 4.3.1 *****			2003-05-31		
Library: PDLX			- Invoked Programs -			DBnr: 180 Fnr: 54		
Command: PROG * (*) USING PROG * (*) WITH * VIA *						Page: 1		
T:Program			using			via		

1 P:ZPDFIELD			1 N:N-BUFEDT <<- nfn			Callnat		
2 P:ZPDP0			1 P:ZPDP1			Fetch		
			2 S:ZPDS1 (NEWDICLX,180,54)			Perform		
			Function: SUB-IN-ZPDS					

Comments:

Only programs in the current library that call other programs are displayed.

to 1	The note <<- nfn means that the called program was not found within the structure specified. "Not found" in this context means that no XRef data is present.
to 2	If the called program is not contained in the current library, Library, DBnr, Fnr are displayed in parentheses.

Bottom-up

Example: Function Programs referenced in programs

```

09:53:09          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: PDLX          - Invoked Programs -          DBnr: 180 Fnr: 54
Command: PROG * (*) REF PROG * (*) WITH * VIA *          Page: 1

      T:Program          referenced in          via
-----
1 ?:*DYNAMIC
          1 P:ZPDP3          Fetch

2 ? :N-BUFEDT
          1 P:ZPDFIELD          Callnat

3 M:ZPDM1
          1 P:ZPDP1          Map
          2 P:ZPDP2 (NEWDIC,180,54) Map

4 P:PGMCO002 (*SYSCOB*,255,255)
  Entry : PGMCO002
          1 P:ZPDP1          Call
          2 P:ZPDP2          Call
          3 S:ZPDS1          Call
    
```

Comments:

to 1	*DYNAMIC produces a list of all programs that call up other programs by means of variables: ASSIGN #A = 'SUB1' FETCH #A
to 2	The question mark means that the program N-BUFEDT was not found within the specified structure. "Not found" in this context means that no XRef data was found for the program object. The program is, however, referenced by program P:ZPDFIELD via CALLNAT.
to 3	Member ZPDM1 was found within the current library. If the called program is contained in the current library (here ZPDM1), programs not contained in the current library that call this program are also displayed (here ZPDP2 in Library NEWDIC). Library, DBnr, Fnr are displayed in parentheses.
to 4	The called program (PGMCO002) was found, but in another library within the structure (library *SYSCOB*). In this case only calling programs within the current library are displayed.

Example: Function Program referenced in programs recursively

```

13:44:35          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDICLX          - Invoked Programs -          DBnr: 180 Fnr: 54
Command: PROG XHMENU10 (*) REF REC * (*) WITH * VIA *          Page: 1
      DEPTH 7
      1 M:XHMENU10
1 -----2 -----3 -----4 -----5 -----6 -----7 -----
P:XPHELP   P:XPICIMPL P:XPVERI   M:XMCIMP00 P:XPICIMPL <--- rec
          M:XMREFE00 P:XPREFE   P:XPVERI   <--- rec
          P:XPICIMPL <--- rec
          P:XPICMDP   M:XMCIMP00 <--- suppr
          M:XMCONS00 P:XPVCONS P:XPVERI
          <--- rec
          M:XMCOPI00 P:XPICOPY P:XPMENU
          <--- steplib
    
```

Comments:

The note <--steplib means that the evaluation was stopped at this point because the called program is contained in another library.

Data Areas and Variables

The functions display active reference information about data areas and variables. The input screen is displayed with code D in the XRef menu.

```

13:45:48          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Data Area and Variables -          DBnr: 180 Fnr: 54

Code Function                                Ss Pg PT DA DT DB St Va Us
-----
C Count references to variable                * O * O          * * O
U Program using variables                     * O * O          * * O
R Variable referenced in programs            O * O * O          * * O
D Program using data areas                   O * O * O O
E Data area referenced in programs           O * O * O O

Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
Data area:        Data area type: (?)
Data block:
Structure:
Variable:
Usage:            (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit

```

Note:

All functions use objects contained in the current library as starting objects. In the reports, on the left side only objects in the current library are listed, regardless of whether or not a library structure was specified.

Code	Function	Answer to the question...
C	Count references to variable	How many times is the variable ABC (contained in the current library) referred to by programs (contained in any library)
U	Program using variables	Which variables (contained in any library) does the program ABC (contained in the current library) refer to
R	Variable referenced in programs	Which programs (contained in any library) refer to the variable ABC (contained in the current library)
D	Program using data areas	Which data areas (contained in any library) does the program ABC (contained in the current library) refer to
E	Data area referenced in programs	Which programs (contained in any library) refer to the data area ABC (contained in the current library)

Effects of Steplib Support

Steplib support affects the above Data Area and Variable functions as follows.

For programs cataloged using Natural 4.1.1 or above on a Windows or UNIX platform or Natural 3.1.2 or above on the mainframe, additional XRef information about each referenced data area is stored: the library name, database number and file number. This information is used to determine the referenced data areas/variables. In addition, the specified library structure is used to determine the data areas/variables that will be used when objects will be cataloged a new.

For programs cataloged using an older version of Natural the additional XRef information is not available, therefore the specified library structure is used to determine the referenced data areas/variables.

Parameters Limiting the Scope of the Functions																					
Save set	Any user defined in Predict can save the types and names of programs listed by functions R or E in one or more sets (one set per library). See Using Sets. Using function D, only types and names of the data areas will be included in the set(s).																				
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.																				
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified: <table border="1"> <tr> <td>F</td> <td>function</td> </tr> <tr> <td>H</td> <td>help routine</td> </tr> <tr> <td>K</td> <td>ISPF Macro</td> </tr> <tr> <td>M</td> <td>map</td> </tr> <tr> <td>N</td> <td>subprogram</td> </tr> <tr> <td>P</td> <td>main program</td> </tr> <tr> <td>S</td> <td>subroutine</td> </tr> <tr> <td>T</td> <td>dialog</td> </tr> <tr> <td>4</td> <td>class</td> </tr> <tr> <td>*, blank</td> <td>All</td> </tr> </table>	F	function	H	help routine	K	ISPF Macro	M	map	N	subprogram	P	main program	S	subroutine	T	dialog	4	class	*, blank	All
F	function																				
H	help routine																				
K	ISPF Macro																				
M	map																				
N	subprogram																				
P	main program																				
S	subroutine																				
T	dialog																				
4	class																				
*, blank	All																				
Data area	The data area or data areas that are used by the program selected above. Asterisk notation is possible. Note: Enter *DYNAMIC to list all area independent variables (AIV).																				
Data area type	Type of data area used by the program. One of the following three values can be specified: <table border="1"> <tr> <td>G</td> <td>Global data area</td> </tr> <tr> <td>L</td> <td>Local data area</td> </tr> <tr> <td>P</td> <td>Parameter data area</td> </tr> </table>	G	Global data area	L	Local data area	P	Parameter data area														
G	Global data area																				
L	Local data area																				
P	Parameter data area																				
Data block	The data block used by the program.																				

Parameters Limiting the Scope of the Functions	
Structure	<p>For fields contained in a group, the group name has to be specified in addition to the field name to ensure uniqueness. For example, a group is defined as follows:</p> <pre>1 FULL-NAME 2 FIRST-NAME 2 LAST-NAME</pre> <p>To identify the variable LAST-NAME, the field name LAST-NAME has to be specified in the field Variable and the group name FULL-NAME has to be specified in the field Structure.</p>
Variable	<p>The variable used by the program.</p> <p>Note: A counter field in a data area view has a name beginning with C*. Enter C** to specify all counter fields.</p>
Usage	Limits the function to variables that are used in a particular way:
	M Modified by the program
	U Used but not modified by the program
	S Used for dynamic source code creation (see description of the RUN statement in the Natural Statements documentation)
	If nothing is specified, then all variables are shown.

Views and Their Fields

The functions display active reference information about views and fields.
The input screen is displayed with code V in the XRef menu.

```

13:48:22          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Views and Fields -          DBnr: 180 Fnr: 54

Code Function                                     Ss Pg PT Vi DV VU Fi FU
-----
C Count references to view                          * 0 *          0
D Count references to field of view                  * 0 *          * 0
U Program using views                                * 0 *          0
R View referenced in programs                        0 * 0 *          0
F Program using fields of views                      * 0 *          * 0
G Field of view referenced in programs                0 * 0 *          * 0
A Program using data area views                      * 0 * * 0
B Data area view referenced in programs              0 * 0 * * 0
-----
Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
View:
Data area view:          View usage: (?)
Field:          Field usage: (?)
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit

```

Note:

In this section only, the term program applies to global and local data areas (types G and L) as well as to programs (types H, M, N, P and S).

Code	Function	Answer to the question...
C	Count references to view	How many times is the view ABC referred to by programs
D	Count references to field of view	How many times is the field ABC referred to by programs
U	Program using views	Which views does the program ABC refer to
R	View referenced in programs	Which programs refer to the view ABC
F	Program using fields of views	Which views does the program ABC refer to, and which fields of these views does it refer to
G	Field of view referenced in programs	Which programs refer to the field ABC
A	Program using data area views	Which data area views does program ABC refer to
B	Data area view referenced in programs	Which programs refer to data area view ABC

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by functions R, G or B in one or more sets (one set per library). See Using Sets.	
Program	The ID of the program (or data area) for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program (or data area) for which the function is to be executed. One of the following values can be specified:	
	F	function
	G	global data area
	H	help routine
	K	ISPF Macro
	L	local data area
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
*, blank	All	
View	The view (DDMs) that are used by the program (or data area). Asterisk notation is possible.	
Data area view	The data area view(s) used by the program or data area. Asterisk notation is possible. Note: Specify the string *DDM if direct use of a standard views (not via a data area view) in reporting mode is to be reported.	

Parameters Limiting the Scope of the Functions	
View usage	How the view is used by the program (or data area). One of the following values can be specified:
A	Defined in a data area
D	Deleted
M	Modified (either deleted, updated or stored)
N	Stored
O	Only read (neither deleted, updated nor stored)
P	Database addr.
R	Read, deleted or updated
T	Coupled search
U	Updated
*, blank	All
	Note: Usage A is valid only for data areas. All other values are valid only for programs.
Field	The field or fields used by the program (or data area). Asterisk notation is possible.
Field usage	How the field is used by the program (or data area):
A	Defined in a data area
C	Used as a counter field
H	Used in the WITH clause of a search criterion as hyperdescriptor search
L	Used in the WITH clause of a search criterion as non descriptor search
M	Stored or updated
N	Stored
O	Either used as a counter field or read (but not modified)
R	Updated, used as a counter field or read
S	Used in the WITH clause of a search criterion
T	Used in the COUPLED clause of a search criterion
U	Updated
*, blank	All
	Note: Usage A is valid only for data areas. All other values are valid only for programs.

Copy Code

The functions display active reference information about Natural copy code (source code). The input screen is displayed with C in the XRef menu.

```

13:50:21          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Copycode -          DBnr: 180 Fnr: 54

Code Function          Ss Pg PT Co Us
-----
C Count references to copycode          * O *
U Program using copycodes          O * O * O
R Copycode referenced in programs O * O * O
-----

Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
Copycode:          Usage: (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp GDAV Quit Sets Rule Copy XRef View OSet SPfk Main Exit
    
```

Note:

All functions use objects contained in the current library as starting objects. In the reports, on the left side only objects in the current library are listed, regardless of whether or not a library structure was specified.

Code	Function	Answer to the question...
C	Count references to copy code	How many times is the copy code ABC (contained in the current library) referred to by programs (contained in any library)
U	Program using copy codes	Which copy codes (contained in any library) does the program ABC (contained in the current library) refer to
R	Copy code referenced in programs	Which programs (contained in any library) refer to the copy code ABC (contained in the current library)

Effects of Steplib Support

Steplib support affects the above Copy Code functions as follows.

For programs cataloged using Natural 4.1.1 or above on a Windows or UNIX platform or Natural 4.1.1 or above on the mainframe, additional XRef information about each referenced copy code is stored: the library name, database number and file number. This information is used to determine the referenced copy codes. In addition, the specified library structure is used to determine the copy codes that will be used when objects will be cataloged a new.

For programs cataloged using an older version of Natural the additional XRef information is not available, therefore the specified library structure is used to determine the referenced copy codes.

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by functions U or R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Copy Code	Only copy code with IDs beginning with a particular sequence of characters is included in the report.	
Usage	Usage Only copy code that is used by programs in a particular way is included in the report. The following values can be specified:	
	D	Direct use of copy code
	I	Indirect use of copy code
		*, blank

Error Numbers

The functions display active reference information about Natural error numbers. The input screen is displayed with code E in the XRef menu.

```

13:52:04          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Error Numbers -          DBnr: 180 Fnr: 54

Code Function          Ss Pg PT EN ET
-----
C Count references to error number          * 0 0 0
U Program using error numbers          * 0 0 0
R Error number referenced in programs 0 * 0 0 0
-----

Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
Error number:    to: Error type: (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit
    
```

Code	Function	Answer to the question...
C	Count references to error number	How many times is the error number ABC referred to by programs
U	Program using error numbers	Which error numbers does the program ABC refer to
R	Error number referenced in programs	Which programs refer to the error number ABC

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Error number	<p>Only error messages in the range of error numbers are included in the report.</p> <p>Finding error messages that are used dynamically:</p> <p>Error number *DYN as input or output indicates an error number that is used dynamically, as in the following example:</p> <pre>VAR1(N4) = 1000 REINPUT WITH TEXT *VAR1</pre>	
Error type	<p>Only error messages of the given type are included in the report. The following values can be specified:</p>	
	S	Natural system messages
	U	Error messages defined in SYSDIC
	*, blank	Both types

Printers

The functions display active reference information about printers which are assigned in the OUTPUT statement of a DEFINE PRINTER statement. The input screen is displayed with code P in the XRef menu.

```

13:53:22          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Printers -          DBnr: 180 Fnr: 54

Code Function          Ss Pg PT Nr PN
-----
U Program using printers          * 0 0 *
R Printer referenced in programs 0 * 0 0 *
-----

Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
Printer number:  to: Printer name:

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit
    
```

Code	Function	Answer to the question...
U	Program using printers	Which printers does the program ABC refer to
R	Printer referenced in programs	Which programs refer to the printer ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
*, blank	All	
Printer name	Only printers that meet the given name are included in the report. Asterisk notation is possible.	
Printer number	Only printers with numbers in the specified range are included in the report.	

Workfiles

The functions display active reference information about Natural workfiles. The input screen is displayed with code W in the XRef menu.

```

13:58:17          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Work Files -          DBnr: 180 Fnr: 54

          Code Function          Ss Pg PT WN WU
-----
          U Program using work files          * 0 0 0
          R Work file referenced in programs 0 * 0 0 0
-----

          Code: ?          Save set: N (Y,N)
          Program:          Program type: (?)
Work file number: to:          Usage: (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Invp GDAV Quit Sets Rule Copy XRef View OSet SPfk Main Exit

```

Code	Function	Answer to the question...
U	Program using workfiles	Which workfiles does the program ABC refer to
R	Workfile referenced in programs	Which programs refer to the workfile ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
*, blank	All	
Workfile number	Only workfiles with numbers in the specified range are included in the report.	
Usage	Only workfiles used in a particular way are included in the report:	
	R	Read (statement READ WORKFILE)
	W	Write (statement WRITE WORKFILE)
	D	Define workfile (statement DEFINE WORKFILE), this usage is only applicable in a VAX/VMS environment
	P	Write PC command (statement WRITE PC FILE n COMMAND)
	C	Close workfile (statement CLOSE WORKFILE)

Retained Sets

The functions display active reference information about retained sets (as defined in the RETAIN clause of a Natural FIND statement). The input screen is displayed with code S in the XRef menu.

```

13:59:28          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Retained Sets -          DBnr: 180 Fnr: 54

Code Function          Ss Pg PT SN SU
-----
U Program using retained sets          * O * O
R Retained set referenced in programs O * O * O
-----

Code: ?          Save set: N (Y,N)
Program:          Program type: (?)
Retained set:
Usage: (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit
    
```

Code	Function	Answer to the question...
U	Program using retained sets	Which retained sets does the program ABC refer to
R	Retained set referenced in programs	Which programs refer to the retained set ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
*, blank	All	
Retained sets	<p>Only retained sets that meet the given name will be included in the report. Asterisk notation is possible.</p> <p>Note: Enter *DYNAMIC to list all programs that handle sets via variables. Examples:</p> <ul style="list-style-type: none"> ● FIND ... RETAIN AS #SET-NAME ● RELEASE SET #SET-NAME <p>Note: Enter *ALL to list all programs that handle all existing sets. Example:</p> <ul style="list-style-type: none"> ● RELEASE SET 	
Usage	Only retained sets used in a particular way are included in the report:	
	B	Sets built by FIND ... WITH ... RETAIN AS 'SET' statements.
	R	Sets released by RELEASE SET 'SET' or RELEASE SETS statements.
	U	Sets used by FIND ... WITH 'SET' statements.
	*, blank	All

Processing Rules

The functions display active reference information about processing rules.
The input screen is displayed with code R in the XRef menu.

```

13:00:59          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Processing Rules -          DBnr: 180 Fnr: 54

Code Function          Ss Ma Ru RS RU
-----
C Count references to rule          * * 0 0
U Map using rules          * * 0 0
R Rule referenced in maps          0 * * 0 0
-----

Code: ?   Save set: N (Y,N)
Map:
Rule:
Rule status:  (?)          Rule usage:  (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit

```

Code	Function	Answer to the question...
C	Count references to rule	How many times is the rule ABC referred to by cataloged maps
U	Map using rules	Which rule does the cataloged map ABC refer to
R	Rule referenced in maps	Which cataloged maps refer to the rule ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of maps listed by function R in one or more sets. See Using Sets.	
Map	The ID of the map for which the function is to be executed. Asterisk notation is possible.	
Rule	The ID of the rule for which the function is to be executed. Asterisk notation is possible.	
Rule status	Only rules of the given type are included in the report:	
	A	Automatic rules
	F	Free rules
	*, blank	All
Usage of rule	Only rules that are used in the given way are included in the report:	
	A	Used as automatic rule
	F	Used as free rule
	I	Used as inline rule
	*, blank	All

Natural Resources

The functions display active reference information about Natural resources.

The input screen is displayed with code F in the XRef menu.

```

10:02:03          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Natural Resources -          DBnr: 180 Fnr: 64

          Code Function          Ss Pg PT Nr Us
-----
C Count references to Natural resources          *  O  *  O
U Program using Natural resources          O  *  O  *  O
R Natural resources referenced in programs          O  *  O  *  O
-----

          Code: ?          Save set: N (Y,N)
          Program:          Program type: (?)
Nat.resource:

          Usage: (?)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit

```

Code	Function	Answer to the question...
C	Count references to Natural resources	How many times is the Natural resource ABC referred to by programs
U	Program using Natural resources	Which Natural resources does the program ABC refer to
R	Natural resources referenced in programs	Which programs refer to the Natural resource ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by functions U or R in one or more sets. See Using Sets.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
*, blank	All	
Natural resource	Only Natural resources with IDs beginning with a particular sequence of characters are included in the report.	
Usage	Only Natural resources which are used in a particular way are included in the report.	
	B	Sets built by FIND ... WITH ... RETAIN AS 'SET' statements.
	R	Sets released by RELEASE SET 'SET' or RELEASE SETS statements.
	U	Sets used by FIND ... WITH 'SET' statements.
	*, blank	All

Report Programs with XRef Data

The function Report programs with xref data displays summary active reference information about programs. The input screen is displayed with code X in the XRef menu.

```

13:02:53          ***** P R E D I C T 4.3.1 *****          2003-05-31
Library: NEWDIC          - Report Program -          DBNr: 180 Fnr: 54

      Program: _____ (?) Program type: _ (?)
      User Id: _____ Terminal Id: _____
      from date: _____ to date: 2003-05-31
      short list: N (Y,N)          save set: _____

Report              Nr          Report              Nr
-----
Statistical data    1          Predict description    2
Natural program list    _          Using/Referenced programs    4
Views, Da-views and fields    5          Da-areas and variables    6
Workfiles, printers, errors... 7          format extended description    N

expand copycodes / rules    N          suppress empty reports    Y
-----

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  XRef  View  OSet  SPfk  Main  Exit
    
```

Up to seven types of information can be retrieved:

Information Reported by Report Program with XRef Data	
Statistical data	The program ID, the program type, the date and time when it was cataloged, the ID of the user who cataloged it, the ID of the terminal where it was cataloged and the information that would be displayed for this program by the Natural command LIST DIR object-ID.
Predict description	The extended description and abstract that are stored in the Predict object for that program. If Format extended description is set to Y, Con-form, Software AG's text formatting facility, is used whenever a description is output. See Extended Description Skeleton in the section Defaults in the Predict Administration documentation .
Natural program list	A Source code listing of the Natural program itself. If Expand copycodes / rules is set to Y, all copy code used by an INCLUDE statement is expanded in the listing of the program and all processing rules (free and automatic) used in a map are expanded in the listing of the map.
Using/referenced programs	For each program that either invokes that program or is invoked by that program, the program type, the program ID, the method of invoking and - for maps - the processing rules used are included in the report. Note: For every method sent by a program, all classes which support this method will be listed.
Views, Da-views and fields	Each view, data area view and field of a view used by that program is included in the report.
Da-areas and variables	Each data area and each variable used by that program is included in the report.
Workfiles, printers, errors	Natural workfiles, retained sets, copy code, printers and error message numbers used by that program are included in the report.

Parameters		
All parameters except Save set limit the scope of the function.		
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	A	parameter data area
	G	global data area
	F	function
	H	help routine
	K	ISPF Macro
	L	local data area
	M	map
	N	subprogram
	O	command processor
	P	main program
	S	subroutine
	T	dialog
	Y	Natural Expert model
4	class	
*, blank	all	
User ID	Only those programs cataloged by the specified user will be processed. Asterisk notation is possible.	
Terminal ID	Only those programs cataloged from the specified terminal are processed. Asterisk notation is possible.	
From date ... to date	Only those programs cataloged within the range of dates specified are processed.	
Short list	Y	Displays the following information for each program selected in a single line: - Program ID - Program type - Information about the existence of source, cataloged object and XREF data - Program mode (report mode or structured mode) - ID of the user who last saved or cataloged the program - Version of Natural - Date and time when the program was last saved or cataloged.
	N	A more extensive report is created, containing up to seven categories of information. Enter digits in the Nr column to determine the sequence of the information displayed. Enter 0 to suppress categories that are not to appear.
Save set	Any user defined in Predict can save the types and names of listed programs in a set. This feature is normally used together with the Short list option. See Using Sets.	

Parameters		
Format extended description	Y	Any Con-form instructions contained in the extended description of any object are executed. See Extended Description Skeleton in the section Defaults in the Predict administration documentation .
Expand Copy Codes/Rules	Y	All copy code used by an INCLUDE statement is expanded in the listing of the program and all processing rules (free and automatic) used in a map are expanded in the listing of the map.
Suppress empty reports	Y	Suppress reports where no information is found in a particular category.

Example of NaturalX Support

The following sample output of function Report Programs with XRef data shows how features of NaturalX are documented in Predict.

```

*** Report for program references ***

4:CLASS1  class name NATURAL.BEAR
          interface SOUND
            method    1  SNIFF                      impl. SNIFF-N
            property  1  NATURALBEARPROPERTY
          interface IMOVE
            method    1  WALK                      impl. WALK-N

*** Report for data areas, blocks and variables ***

4:CLASS1  using data area  1  L:GUID-L
          variables      1  INTERFACESOUNDGUID      Used
                   2  NATURALBEARGUID             Used

          data area     2  L:BEAR-O
                   as object data area
          variables      1  BEAR-PROPERTY-VARIABLE  Used

          data area     3  P:SNIFF-A

          data area     4  P:WALK-A

```