

SYSEXT Utility - Natural Application Programming Interfaces

The utility SYSEXT is used to locate and test Natural Application Programming Interfaces (APIs) contained in the current system library SYSEXT.

A Natural API is a Natural subprogram (cataloged object) that is used for accessing and possibly modifying data or performing services that are specific to Natural, a subcomponent or a subproduct. The purpose of a Natural API is to retrieve or modify information or use services that are not accessible by Natural statements.

The *SYSEXT Utility - Natural Application Programming Interfaces* documentation covers the following topics:

- Basic Features of SYSEXT
- Invoking and Terminating SYSEXT
- SYSEXT Utility Screen
- Performing SYSEXT Utility Functions
- Using a Natural API
- List of Natural APIs

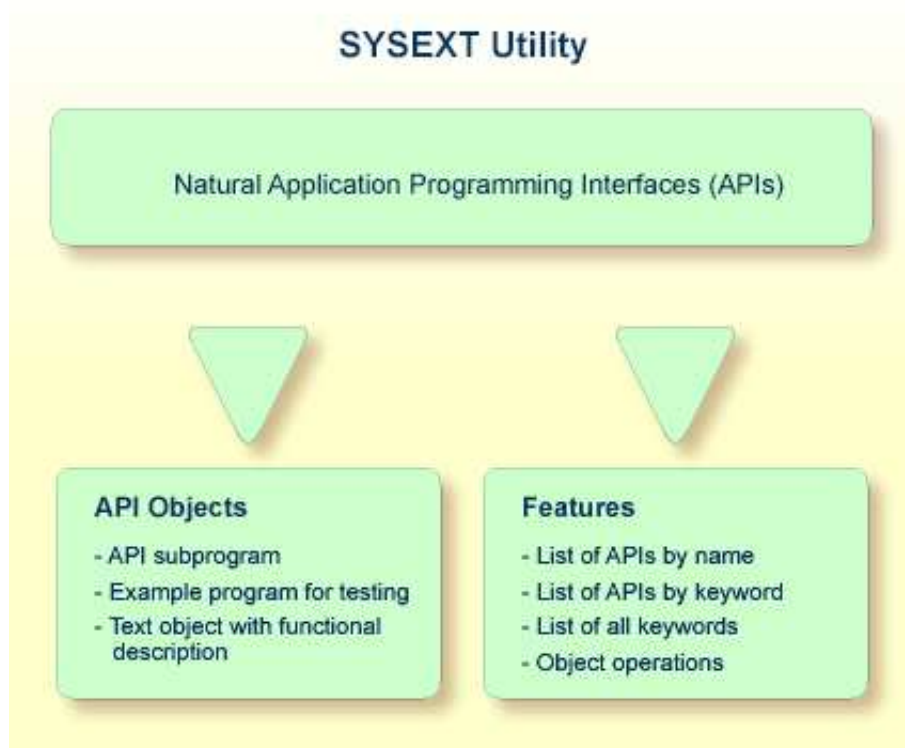
Related Topics:

- *Natural User Exits - Operations* documentation
 - *Application Programming Interfaces - Natural Security* documentation
 - *Application Programming Interfaces - Natural SAF Security* documentation
 - *SYSAPI - APIs of Natural Add-on Products - Utilities* documentation
-

Basic Features of SYSEXT

For each Natural API, the utility SYSEXT provides a functional description, one example program and API-specific keywords.

The following diagram is an overview of the Natural objects and major features SYSEXT provides to test and implement an API:



Objects Provided for Natural APIs

The types of Natural object typically provided for each Natural API are listed in the following section. The section does not list additional objects that may be required for particular APIs.

All API-related objects are contained in the library SYSEXT on the system file FNAT.

In the following table, *nnnn* denotes the 4-digit number assigned to the API. All API objects that relate to one another have identical numbers.

Object Name	Explanation
USR <i>nnnn</i> N	The API subprogram (cataloged object) that performs the designated function.
USR <i>nnnn</i> P	An example program (source object) that can be used to test the effect of the API. The example program invokes the corresponding subprogram USR <i>nnnn</i> N.
USR <i>nnnn</i> T	A text object that contains a description of the corresponding API. The description comprises purpose, function and calling conventions of the API and relevant keywords. You can display a text object by using the line command D as described in <i>Line Commands</i> in <i>Performing SYSEXT Utility Functions</i> . You can search for a text object by specifying a keyword as described in <i>Keyword Search</i> in <i>Performing SYSEXT Utility Functions</i> .

Invoking and Terminating SYSEXT

This section provides instructions for invoking and terminating the SYSEXT utility.

▶ To invoke SYSEXT

- Enter the following system command:

```
SYSEXT
```

A SYSEXT utility screen similar to the example below appears with a list of all available Natural APIs:

```
13:11:42          ***** NATURAL SYSEXT UTILITY *****          2005-08-01
User SAG              - Menu -                               Library SYSEXT

Cmd  Source      Interface  Comment                                     Prod
-   USR0010P  USR0010N  Get 'SYSPROF' Information                   NAT
-   USR0011P  USR0011N  Information about logical file              NAT
-   USR0020P  USR0020N  Read any error text from FNAT / FUSER      NAT
-   USR0040P  USR0040N  Get type of last error                     NAT
-   USR0050P  USR0050N  Get 'SYSPROD' Information                  NAT
-   USR0060P  USR0060N  Copy LFILE definition from 'FNAT' to 'FUSER' NAT
-   USR0070P  USR0070N  Default Editor Profile 'SYSTEM'           NAT
-   USR0080P  USR0080N  Handle Type/Name of Editor Contents       NAT
-   USR0100P  USR0100N  Control LRECL                              NVS
-   USR0120P  USR0120N  Read Natural Short Error Message           NAT
-   USR0210P  USR0210N  Save, cat or stow NATURAL object           NAT
-   USR0220P  USR0220N  Read Natural Long Error Message            NAT
-   USR0320P  USR0320N  Read User Short Error Message from FNAT or FUSER NAT
-   USR0330P  USR0330N  Read Natural Object Directory              NAT

Keyword .. _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit      --      +      Canc
```

The list is sorted by the names of the example programs (USRnnnnP). If required, press PF8 to scroll down one page in the list or PF7 to go to the beginning of the list.

▶ To terminate SYSEXT

- On the SYSEXT utility screen, press PF3 or PF12.

Or:

In the **Command** line, enter a period (.) or enter EXIT.

SYSEXT Utility Screen

The field and columns contained on the SYSEXT utility screen are explained in the following table:

Column	Explanation
Cmd	The input field for a line command to be executed on a text object or an example program: see Line Commands in <i>Performing SYSEXT Utility Functions</i> .
Source	The name of the example program (source object USRnnnnP) that can be used to invoke the API.
Interface	The name of the API subprogram USRnnnnN.
Comment	A brief description of the purpose of the API.
Prod	The product code of Natural (NAT) or a Natural add-on product affected by the API. For example: NAT = Natural, NDB = Natural for DB2, NVS = Natural for VSAM, PRD = Predict, RPC = Natural Remote Procedure Call.
Keyword	The input field for a keyword relevant to an API: see Keyword Search in <i>Performing SYSEXT Utility Functions</i> .

Performing SYSEXT Utility Functions

The SYSEXT utility functions can be used to perform operations on the text objects and example programs referenced in the API list on the SYSEXT utility screen. In addition, you can use the keyword search function to shorten the list of APIs or list all keywords available for the APIs.

The section below covers the following topics:

- Line Commands
- Keyword Search

Line Commands

Line commands are used to perform object operations. You enter a line command in the **Cmd** column next to the API required. For a list of valid line commands, enter a question mark (?) in this column.

The following line commands are available on the SYSEXT utility screen:

Line Command	Function
E	Edit example program USRnnnnnP.
L	List example program USRnnnnnP.
R	Run example program USRnnnnnP.
X	Execute example program USRnnnnnP.
D	List text object USRnnnnnT for a description of the corresponding API. The description comprises purpose, function and calling conventions of the API and keywords that are relevant to the API.
K	List keywords relevant to the specified API.
.	Terminate the SYSEXT utility.

Keyword Search

Keywords help you find the Natural APIs relevant to your current task. You can use the keyword search function to list APIs by keyword or list the keywords relevant to a specific API.

To list APIs by keyword

- Choose any of the following methods:

- Enter a single keyword:

On the SYSEXT utility screen, in the **Keyword** field, type in a keyword and press ENTER.

A list of all APIs to which the specified keyword applies appears.

- Select a single keyword from a list of valid keywords:

1. On the SYSEXT utility screen, in the **Keyword** field, enter an asterisk (*) and press ENTER.

The **List Keywords** window with a list of all keywords available for all APIs appears as shown in the example below.

If required, press PF8 to scroll down the list of keywords or press PF7 to go to the beginning of the list.

2. In the **M** column, next to the desired keyword, type in any character and press ENTER.

Or: In the **Direct** field, enter the desired keyword and press ENTER.

A list of all APIs to which the specified keyword applies appears.

- Select multiple keywords from a list of valid keywords:

1. On the SYSEXT utility screen, in the **Keyword** field, enter an asterisk (*) and press ENTER.

The **List Keywords** window with a list of all keywords available for all APIs appears as shown in the example below.

If required, press PF8 to scroll down the list of keywords or press PF7 to go to the beginning of the list.

2. In the **M** column, next to the desired keywords, type in an asterisk (*) and press ENTER.

The **Keyword** window appears with a list of all APIs to which the first keyword selected applies (here: *LANGUAGE) as shown in the example below:

```

11:08:58          ***** NATURAL SYSEXT UTILITY *****          2005-11-10
User SAG          - Menu -          Library SYSEXT

Cm +-----List Keywords-----+ +-----Keyword *LANGUAGE-----+  Prod
_ !                               ! !                               !   NAT
_ ! M Keyword                     ! ! Object                       !   NAT
_ ! _ *DATX                        ! ! USR0020P                     !   NAT
_ ! _ *ERROR-NR                    ! ! USR0120P                     !   NAT
_ ! _ *ERROR-TA                    ! ! USR0320P                     !   NAT
_ ! * *LANGUAGE                    ! ! USR1027P                     !   NAT
_ ! _ *LEVEL                       ! ! USR1030P                     !   NAT
_ ! _ *STEPLIB                      ! ! USR2034P                     !   NAT
_ ! * *TIMESTAMP                   ! ! USR3320P                     !   NVS
_ ! _ *TIMX                         ! !                               !   NAT
_ ! _ ABEND                         ! !                               !   NAT
_ !                               ! !                               !   NAT
_ ! Direct ... _____          ! !                               ! R NAT
_ !                               ! !                               !   NAT
+-----+-----+-----+-----+-----+-----+-----+-----+
Keyword .. * _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit      --      +      Canc

```

3. Press ENTER to open the **Keyword** window for the next keyword selected (here: *TIMESTAMP).
4. Press ENTER or PF3 to return to the **List Keywords** window.
5. Press PF3 to leave all windows and return to the SYSEXT utility screen.

To list keywords relevant to a specific API

- Next to the API required, enter the line command K. The **Keywords** window appears for the specified API with a list of all keywords that are relevant to this API as shown in the example below:

```

11:24:27          ***** NATURAL SYSEXT UTILITY *****          2005-11-10
User SAG              - Menu -              Library SYSEXT

Cmd  Source      Interface  Comment                                     Prod
K  USR0010P     USR0010  +--Keywords for USR0010P--+              NAT
_   USR0011P     USR0011  !                                     ! file              NAT
_   USR0020P     USR0020  ! DBID                                 ! FNAT / FUSER      NAT
_   USR0040P     USR0040  ! ENVIRONMENT                          !                   NAT
_   USR0050P     USR0050  ! FILE                                  !                   NAT
_   USR0060P     USR0060  ! FNAT                                  ! m 'FNAT' to 'FUSER' NAT
_   USR0070P     USR0070  ! FNR                                   ! YSTEM'            NAT
_   USR0080P     USR0080  ! FUSER                                 ! r Contents        NAT
_   USR0100P     USR0100  ! SYSPROF                              !                   NVS
_   USR0120P     USR0120  !                                     ! Message           NAT
_   USR0210P     USR0210  !                                     ! object            NAT
_   USR0220P     USR0220  !                                     ! essage           NAT
_   USR0320P     USR0320  +-----+ sage from FNAT or FUSER      NAT
_   USR0330P     USR0330N  Read Natural Object Directory          NAT

Keyword .. _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---;

```

If required, press PF8 to scroll down one page in the list. Press PF7 to go to the beginning of the list.

Using a Natural API

If you want to use a Natural API contained in the SYSEXT system library, perform one of the following steps:

- Define the SYSEXT system library in the FNAT system file as a steplib library for the user library which contains the Natural objects that use this API. Thus, no API-specific actions are required when upgrading your Natural version.
- Copy the required API to the SYSTEM system library in the FNAT system file. Thus, you only need to check a single library for APIs when upgrading your Natural version.
- Copy the required API to the SYSTEM library in the FUSER system file (not recommended).
- Copy the required API to the user library (or one of its steplib) in the FUSER system file which contains the Natural objects that use this API (not recommended).

An API can only be used in the Natural version with which it is delivered. It cannot be executed in any other Natural version. Therefore, it is strongly recommended that APIs are stored only in the FNAT system file.

List of Natural APIs

The following list of Natural Application Programming Interfaces (APIs) available with the SYSEXT utility is for general information only. For the current status of APIs, execute the system command SYSEXT.

The list columns correspond to the columns contained on the SYSEXT utility screen. For explanations on the columns, see the section *SYSEXT Utility Screen*.

For detailed information on the use of APIs that apply to Natural Remote Procedure Call (RPC), refer to the section *Application Programming Interfaces for Use with Natural RPC* in the *Natural Remote Procedure Call* documentation.

Interface	Comment	Product
USR0010N	Get SYSPROF information	NAT
USR0011N	Information about logical file	NAT
USR0020N	Read any error text from FNAT / FUSER	NAT
USR0040N	Get type of last error	NAT
USR0050N	Get SYSPROD information	NAT
USR0060N	Copy LFILE definition from FNAT to FUSER	NAT
USR0070P	Default Editor Profile SYSTEM	NAT
USR0080N	Handle type/name of editor contents	NAT
USR0100N	Control LRECL	NVS
USR0120N	Read Natural short error message	NAT
USR0210N	Save, catalog or stow Natural object	NAT
USR0220N	Read Natural long error message	NAT
USR0320N	Read user short error message from FNAT or FUSER	NAT
USR0330N	Read Natural object directory	NAT
USR0340N	Natural buffer pool interface	NAT
USR0341N	Natural buffer pool interface	NAT
USR0350N	Read current recording flags	NAT
USR0360N	Modify user short error message	NAT
USR0400N	Number of rows affected by searched UPDATE	NDB
USR0420N	Read user long error message from FUSER	NAT
USR0421N	Update user long error message on FUSER	NAT
USR0500N	Display a string in the title bar of a window	NAT
USR0600N	Display program level information	NAT
USR0610N	Display DB error information	NAT
USR0620N	Translate strings	NAT
USR0622N	Reset error counter	NAT
USR1002N	Save and restore Natural environment parameter	NAT
USR1005N	Information about some Natural system parameters	NAT

Interface	Comment	Product
USR1006N	Support skip-sequential processing	NVS
USR1007N	Display work file and printer file assignments	NAT
USR1009N	Convert *TIMESTAMP to numeric variable	NAT
USR1011N	Wildcard / asterisk check (short)	NAT
USR1012N	Read dynamic error part :1:	NAT
USR1013N	Display current character set	NAT
USR1014N	*** Line Calculator ***	NAT
USR1016N	Display error level for copycode	NAT
USR1017N	Add CATAL call to CATAL control list	NAT
USR1018N	Dynamic OPEN	NVS
USR1019N	Get SYSBUS information	NAT
USR1020N	Add user short error message to FUSER	NAT
USR1021N	Wildcard / asterisk check (long)	NAT
USR1022N	Type of data base	NAT
USR1023N	Date and time variables conversion	NAT
USR1024N	Read results of CATAL	NAT
USR1025N	Handle multiple steplibs	NAT
USR1026N	Display RETURN information	NAT
USR1027N	Search user short error message	NAT
USR1028N	Bit/byte conversion	NAT
USR1029N	Get type of Natural object	NAT
USR1030N	Language code conversion	NAT
USR1031N	Check object name	NAT
USR1032N	List cataloged Natural objects with type	NAT
USR1033N	Find DBID/FNR of a cataloged DDM	NAT
USR1034N	Display NTTF file table	NAT
USR1035N	Maintain objects via the Software AG editor engine	NAT
USR1036N	Maintain the user profile of the Software AG editor	NAT
USR1037N	Information about Natural ABEND data	NAT
USR1038N	Retrieve characteristics of the current platform	NAT
USR1040N	Get or set the UDB parameter	NAT
USR1041N	Sample error transaction program (*ERROR-TA)	NAT
USR1042N	Get or set the value of the UPDATE command	NAT

Interface	Comment	Product
USR1043N	Perform Adabas direct calls	NAT
USR1047N	Dynamic switch of file name	NVS
USR1048N	Modify PF-key labels	NAT
USR1050N	Get or set a work file name	NAT
USR1051N	Interface to various Predict data	PRD
USR1054N	List libraries	NAT
USR1055N	List objects in a library	NAT
USR1056N	List DDMs on the FDIC file or in a library	NAT
USR1057N	Read a Natural source into an array	NAT
USR1058N	Read a DDM source into an array	NAT
USR1066N	Display the Natural "Executing ..." message	NAT
USR1067N	Check library name	NAT
USR1068N	Get or reset the value of DBMS calls	NAT
USR1070N	Issue operator commands to Entire Net-Work	NAT
USR1071N	Set user ID and password for RPC	RPC
USR1072N	Get command ID of a retain set	NAT
USR2001N	Read information about last error	NAT
USR2002P	Default text strings for Help function	NAT
USR2003P	Default settings for MAINMENU	NAT
USR2004N	Information about logical file	NAT
USR2005N	Access to the internal file translation table	NAT
USR2006N	Get detailed message information	NAT
USR2007N	Set/get RPC default server information	RPC
USR2008N	Dynamic OPEN for VSAM/ISAM datasets	NVS
USR2009N	Read dynamic error part :1:	NAT
USR2010N	Display DB error information	NAT
USR2011N	Get or set a work file name	NAT
USR2012N	Get value of system variable *NET-USER	NAT
USR2013N	Get SYSPROF information	NAT
USR2014N	Maintain objects via the Software AG editor engine	NAT
USR2015N	EBCDIC or ASCII translation table for Natural RPC	NAT
USR2016N	Copy map profile from FNAT to FUSER	NAT
USR2017N	Activate map profile handling from FUSER	NAT

Interface	Comment	Product
USR2018N	Read Natural object directory	NAT
USR2019N	Read/save Natural source into/from the source area	NAT
USR2020N	Perform Adabas direct calls	NAT
USR2021N	Dynamic dataset allocation	NAT
USR2022N	Insert GUID into saved data area Note: See the section <i>Obsolete Application Programming Interface</i> in the current <i>Natural Release Notes</i> .	NAT
USR2023N	Type of data base (2 bytes)	NAT
USR2026N	Get TECH information	NAT
USR2027N	Define a wait interval for the session	NAT
USR2028N	Output the Natural version	NAT
USR2029N	Dynamic file allocation (BS2000/OSD)	NAT
USR2030N	Read dynamic error parts :1:,...	NAT
USR2031N	Get SYSPROD information	NAT
USR2032N	Support of commit for CLOSE CONVERSATION	RPC
USR2033N	Information about PRD List Xref sets	PRD
USR2034N	Read system or user error text from either FNAT or FUSER	NAT
USR2035N	Support of SSL	RPC
USR2036N	Convert *TIMESTAMP to numeric variable	NAT
USR2071N	Support of EntireX Security on client side	RPC
USR2072N	Support of EntireX Security on server side	RPC
USR2073N	Ping or terminate an RPC server	RPC
USR2075N	Terminate EntireX Broker Service	RPC
USR3001N	List Roll Server directory entries	NAT
USR3002N	Delete Roll Server directory entries	NAT
USR3005N	Process documentation objects	PRD
USR3013N	Get SYSPROF information	NAT
USR3025N	Handle multiple steplibs	NAT
USR3320N	Find user short error message from FNAT or FUSER	NAT
USR4001N	Set Natural profile parameter PROGRAM (corresponds to USR6002N but can be used on mainframes only)	NAT
USR4002N	Retrieve variables of the current system	NAT

Interface	Comment	Product
USR4003N	Retrieve Natural stack information (alphanumeric)	NAT
USR4004N	Retrieve dynamic Natural profile parameters	NAT
USR4005N	Read all current key settings	NAT
USR4006N	Select (and cancel) Natural sessions under CICS	NCI
USR4007N	Get/set current value of profile parameter SYNERR	NAT
USR4008N	Set library for RPC execution	RPC
USR4009N	Set parameters for EntireX	RPC
USR4010N	Retrieve runtime settings of server	RPC
USR4011N	Create A20 hash value for variable input	NAT
USR4201N	Process data area sources	NAT
USR4202N	Get code page table information	NAT
USR4203N	Check for lock, lock or unlock Natural object	NAT
USR4204N	Set or reset CICS Container Name	NCI
USR4206N	List objects in a library and return directory information	NAT
USR4207N	Return machine characteristics as provided by NATQVS	NAT
USR4208N	Read or write a Natural resource	NAT
USR4209N	Return short name of subroutine	NAT
USR4210N	Base64 conversion of alpha and binary bytes	NAT
USR4211N	Get DBCS characters	NAT
USR4212N	Analyze data area	NAT
USR4213N	String handling for DBCS Support	NAT
USR4214N	Enhanced program level information	NAT
USR4215N	Return a list of resources of a Natural library	NAT
USR4216N	Return a list of Natural objects of a Natural library	NAT
USR4340N	List buffer pool contents or buffer pool cache	NAT
USR4341N	Blacklist maintenance of Natural buffer pools	NAT
USR4371N	Set user ID and ETID for RPC	RPC
USR6002N	Get the current values of some internal counters	NAT
USR6204N	Set Natural profile parameter PROGRAM (corresponds to USR4001N but can be used on all platforms)	NAT
USR6303N	Retrieve Natural stack information (Unicode)	NAT
USR6304N	Set/get reliable state for RPC execution	RPC
USR6305N	Commit/rollback reliable RPC message(s)	RPC

Interface	Comment	Product
USR6306N	Status of UOWs of current EntireX Broker user	RPC