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 either in a local Windows environment or in a remote environment located on a Windows, a UNIX, an OpenVMS or a mainframe platform.

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:

- Prerequisite

- Basic Features of SYSEXT

- Invoking and Terminating SYSEXT

- SYSEXT Tree View Items

- Performing SYSEXT Utility Functions

- Using a Natural API

Related Topics:

- *Application Programming Interfaces - Natural Security* documentation

- *SYSAPI - APIs of Natural Add-on Products - Utilities* documentation
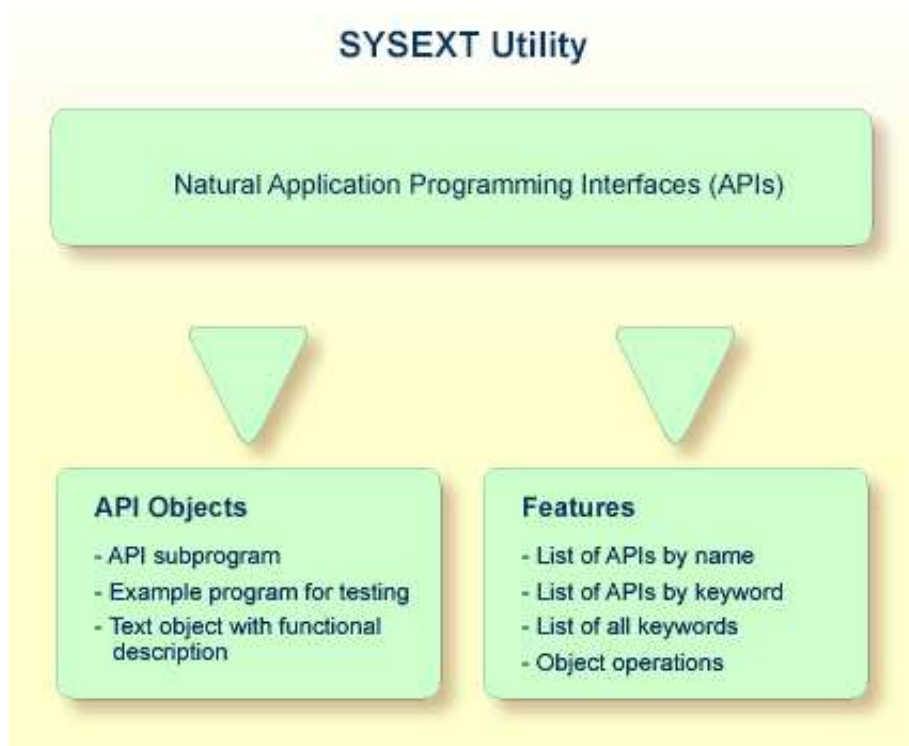

Prerequisite

- The **Enable Plug-ins** option must be selected. This option is selected by default. For details, see *Workspace Options* in the section *Setting the Options* in the *Using Natural Studio* 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*nnnn*N	The API subprogram (cataloged object) that performs the designated function.
USR*nnnn*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*nnnn*N.
USR*nnnn*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.

Caution:
Do *not* modify the source objects EXT-XML1 and EXT-XML2. They are required for configuring the SYSEXT utility and intended for Software AG internal use only.

Invoking and Terminating SYSEXT

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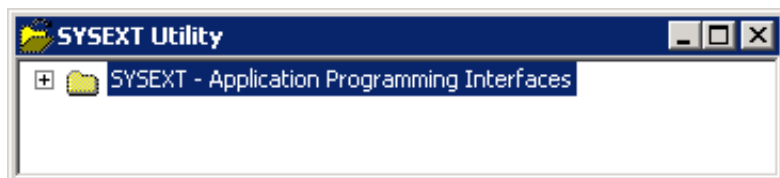
▶ To invoke SYSEXT

- Enter the following system command:

```
SYSEXT
```

Or:
From the **Tools** menu, select **Development Tools > Application Programming Interfaces**.

When invoking SYSEXT, the plug-in for the utility SYSEXT is activated and the SYSEXT utility window appears with the root node **SYSEXT - Application Programming Interfaces** as shown in the example below:

▶ To restart SYSEXT

- If you want to restart SYSEXT during the current Natural session, as an alternative to the start methods mentioned above, from the toolbar, choose the following icon:

This icon appears after initially invoking SYSEXT, when the plug-in for the utility SYSEXT is activated.
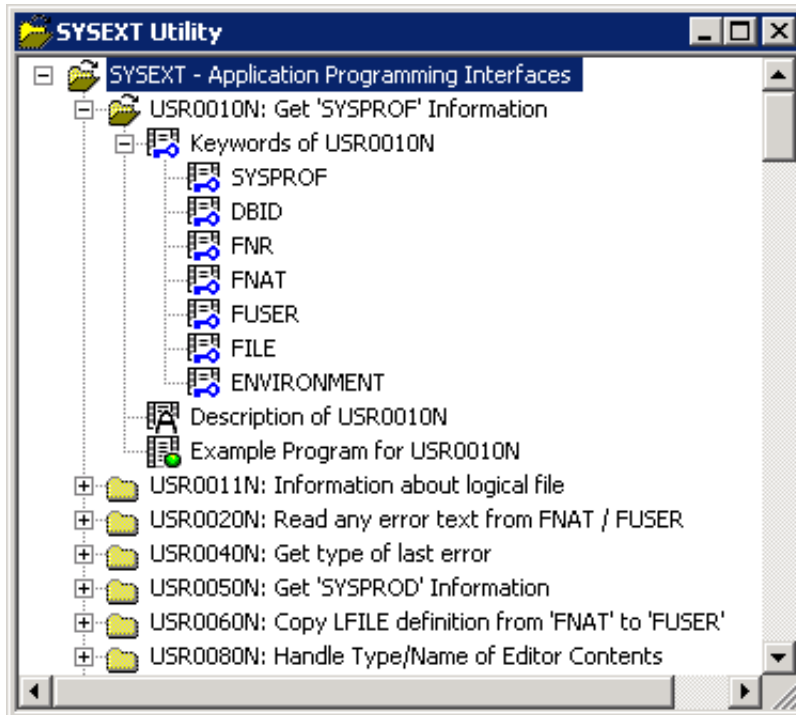
▶ To terminate SYSEXT

- Choose the standard Windows close function.

SYSEXT Tree View Items

This section describes the nodes and items contained in the tree view of the SYSEXT utility window.

If you expand all tree nodes, the tree looks similar to the example below:

Each Natural API is represented by an API node that contains the example program, description and keywords that relate to this API. The API node name consists of the name of the API subprogram (USR*nnnn*N) and a brief description of its functional purpose. The nodes are sorted by API names.

The following items are provided in an API node:

Item	Explanation
Keywords	All keywords relevant to the API. See also the functions Select Keyword and List All Keywords in *Performing SYSEXT Utility Functions*.
Description	A text object (USR*nnnn*T) that contains a description of the API. The description comprises purpose, function and calling conventions of the API and keywords relevant to the API.
Example Program	An example program (source object USR*nnnn*P) of how to invoke the API.

Performing SYSEXT Utility Functions

The SYSEXT utility functions can be used to perform operations on API-specific text objects (descriptions) and example programs or find APIs relevant to a current task by specifying a keyword.

Object operations include functions such as List, Open and Execute, which correspond to the standard functions available when maintaining or executing a Natural object of the type text or program. These functions can be performed by using the context menu associated with each object. For details of these functions, refer to the relevant sections in the *Using Natural Studio* documentation.

The section below covers the following topics:

- Select Keyword

- List All Keywords
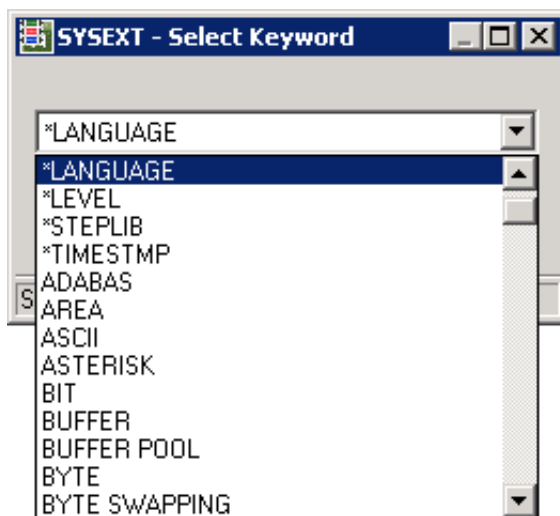
- Refresh

Select Keyword

This function is used to list APIs by keyword.

▶ To list APIs by keyword

1. Select the root node **SYSEXT - Application Programming Interfaces**, open the context menu and choose **Select Keyword** or press SHIFT+K.

   The **Select Keyword** window appears.

2. From the drop-down list box, select a keyword as shown in the example below:
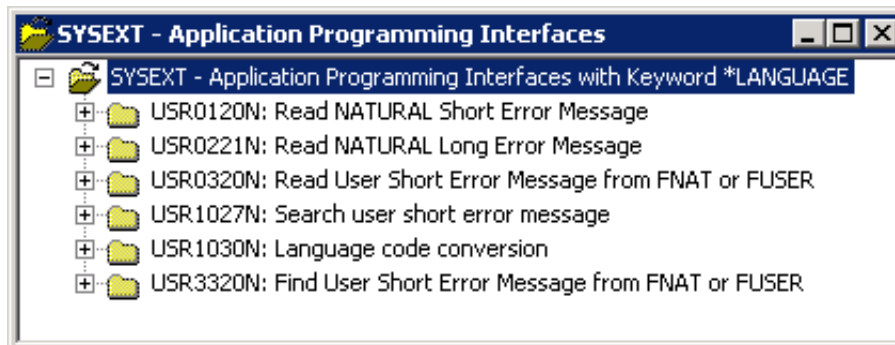

3. Choose the **OK** button.

   The root node **SYSEXT - Application Programming Interfaces with Keyword** appears for the selected keyword.

4. Expand the root node.

   The nodes of all APIs to which the specified keyword applies are displayed as shown in the example of keyword `*LANGUAGE` below:

5. If desired, to return to the display of all API nodes (default setting), in the **Select Keyword** window, select the asterisk (*).

List All Keywords

This function is used to list all keywords for the APIs available in the current system environment.
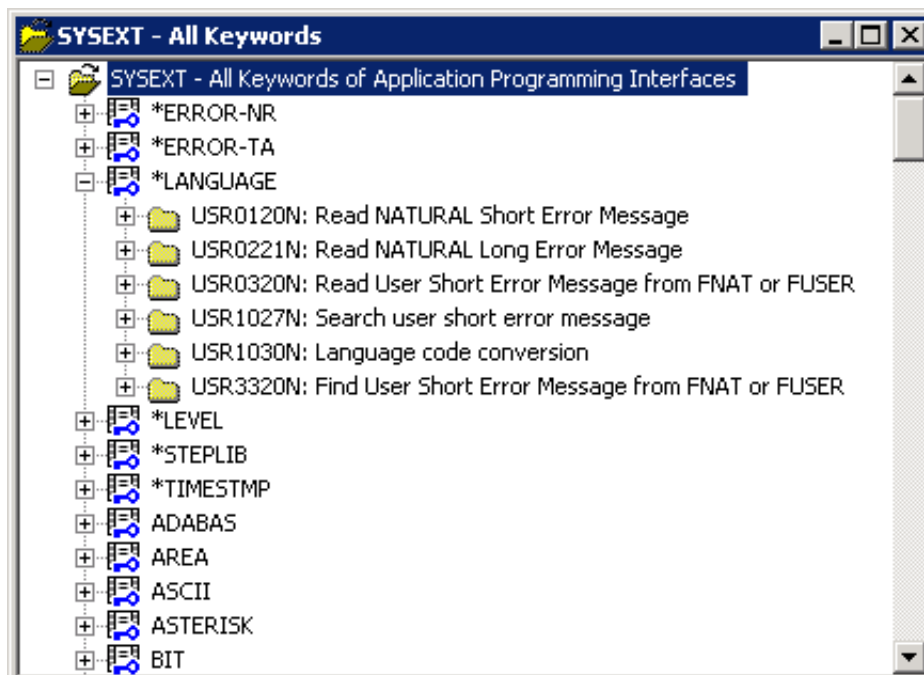
▶ **To list all API keywords**

1. Select the root node **SYSEXT - Application Programming Interfaces**, open the context menu and choose **List All Keywords** or press SHIFT+A.

   The **All Keyword** window appears as an additional window with the root node **SYSEXT - All Keywords of Application Programming Interfaces**.

2. Expand the root node.

   A list of all keywords is displayed as shown in the example below:

Refresh

This function updates API information in the tree view using the data from the objects contained in the current library SYSEXT. The refresh is only required if an API description or a keyword was modified or if a text object was removed.

Note:
Do *not* modify the source objects EXT-XML1 and EXT-XML2. They are required for configuring the SYSEXT utility and intended for Software AG internal use only.

▶ **To refresh API information**

1. Select the root node **SYSEXT - Application Programming Interfaces**, open the context menu and choose **Refresh** or press SHIFT+R.

   A **Refresh** window appears.

   If you access a remote environment on a mainframe, you must specify the parameter SORT in the map environment settings as indicated below:

   ```
   SORT=(WRKSIZE=50)
   ```

   For details on mapping a server, see *Accessing a Remote Development Environment* in the documentation *Remote Development Using SPoD*. For details on SORT, refer to the *Parameter Reference* of Natural for Mainframes.

2. Choose the **OK** button to confirm the refresh or choose **Cancel** to abort the operation.

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 steplibs) 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.