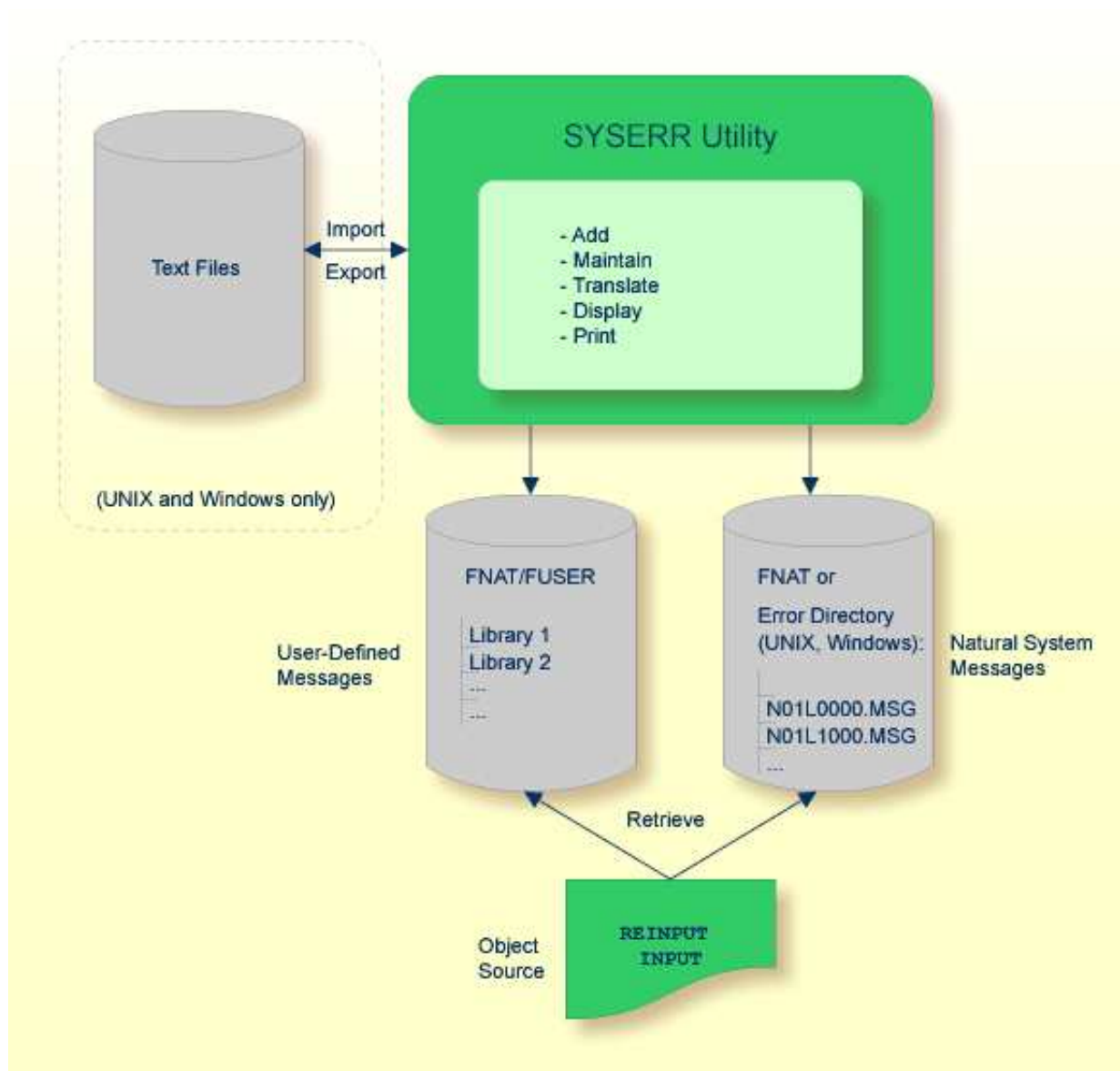


# General Information on Messages

This section contains information on the types of message and message languages that can be managed with the SYSERR utility and how messages are issued and retrieved in your Natural system environment.

The following graphic illustrates the features of the SYSERR utility and how messages are processed within Natural:



This section covers the following topics:

- Message Types
- Message Languages

- Issuing Messages
  - Retrieving Natural System Short Messages
  - Retrieving User-Defined Short Messages
  - Obtaining Message Information
- 

## Message Types

There are two types of message: Natural (system) messages and user-defined messages:

Natural system messages are issued by the Natural nucleus and Natural utilities. Natural system messages are delivered by Software AG and stored as message files in the Natural Err directory. Natural system messages begin with NAT, followed by a four-digit number, for example, NAT0230.

User-defined messages are issued by applications written by a user. User-defined messages are stored as message files in libraries (including SYS-libraries) in the system file FUSER or FNAT.

A message can be translated into different languages. Each language is stored in a separate message file. A maximum of 9999 messages can be stored per library and message file.

There are four types of message text:

- Natural system short message
- Natural system long message
- User-defined short message
- User-defined long message

A short message is the one-line message which is displayed in the message line when the corresponding error situation occurs.

A long message is a detailed explanation of the corresponding short message and includes instructions for solving problems.

### **Caution:**

Keep in mind that any modifications of Natural system messages can result in wrong messages or a loss of modifications when a new Natural version is released.

## Message Languages

Messages can be created in up to 60 languages as described for the system variable \*LANGUAGE in the *System Variables* documentation.

The following rules and restrictions apply:

- Natural system short messages must be entered in English first, and can then be translated into any other language.
- Natural system long messages can be entered in English, but cannot be translated into other languages.
- User-defined short messages can be entered in any language, and then be translated into any other language.
- User-defined long messages can be entered in any language, but only if the corresponding short message in the same language already exists.

## Issuing Messages

This section contains information on the Natural statements `INPUT` and `REINPUT` that are used to issue a Natural system short message or a user-defined short message in a Natural program.

### To issue a Natural system short message in a program

- Specify one of the following Natural statements:

```
INPUT WITH TEXT *-nnnn
```

or

```
REINPUT WITH TEXT *-nnnn
```

where *nnnn* is the number of the requested message (you can omit leading zeros).

### To issue a user-defined short message in a program

- Specify one of the following Natural statements:

```
INPUT WITH TEXT *nnnn
```

or

```
REINPUT WITH TEXT *nnnn
```

where *nnnn* is the number of the requested message (you can omit leading zeros).

## Dynamic Replacement of Message Text

A message text can contain variable parts that are identified by the notation `:n:`, where *n* represents occurrences 1 to 7. These variable parts are replaced by a value at runtime.

For details, see *operand3* in the section *INPUT Syntax 1 - Dynamic Screen Layout Specification* and *operand3* in the section *REINPUT* in the *Statements* documentation.

## Retrieving Natural System Short Messages

When a program references a Natural system short message, Natural looks for the requested message number in the Natural Err directory in the following order:

1. Under the current language code as determined by the system variable \*LANGUAGE,
2. Under language code 1 (English).

If neither of the above is found, a program references a message that does not exist and you only receive the message number prefixed with NAT, for example, NAT0230.

## Retrieving User-Defined Short Messages

When a program references a user-defined short message, Natural first looks for the requested message number *nnnn* under the current language code as determined by the system variable \*LANGUAGE (see the *System Variables* documentation). If that message does not exist, Natural looks for the requested message number *nnnn* under language code 1 (English). If that message does not exist either, Natural looks for message number *n000* (where *n* is the first digit of the requested message number) under language code 1.

These three search steps are first performed in the current library. If nothing is found there, further libraries are searched in the same way until a corresponding message is found.

The sequence of libraries for the search is as follows:

1. The current library as determined by the system variable \*LIBRARY-ID,
2. The steplib; if Natural Security is installed, the sequence in which the steplib are specified in the Natural Security profile of the current library,
3. The default steplib as determined by the system variable \*STEPLIB,
4. The library SYSTEM in the system file FUSER (\*),
5. The library SYSTEM in the system file FNAT (\*).

(\*) If the name of the current library begins with SYS, SYSTEM FNAT is searched before SYSTEM FUSER.

## Obtaining Message Information

When you receive a short message, you may be looking for additional information on the problem situation.

- With the system command HELP, you can display Natural system long messages or user-defined long messages.
- With the system command LASTMSG, you can list the short test of the message(s) that occurred last and additional information on the error situation. The information displayed includes associated error messages that possibly preceded the last message.

Both commands are described in the *System Commands* documentation.