

Natural under IMS/TM - Environments

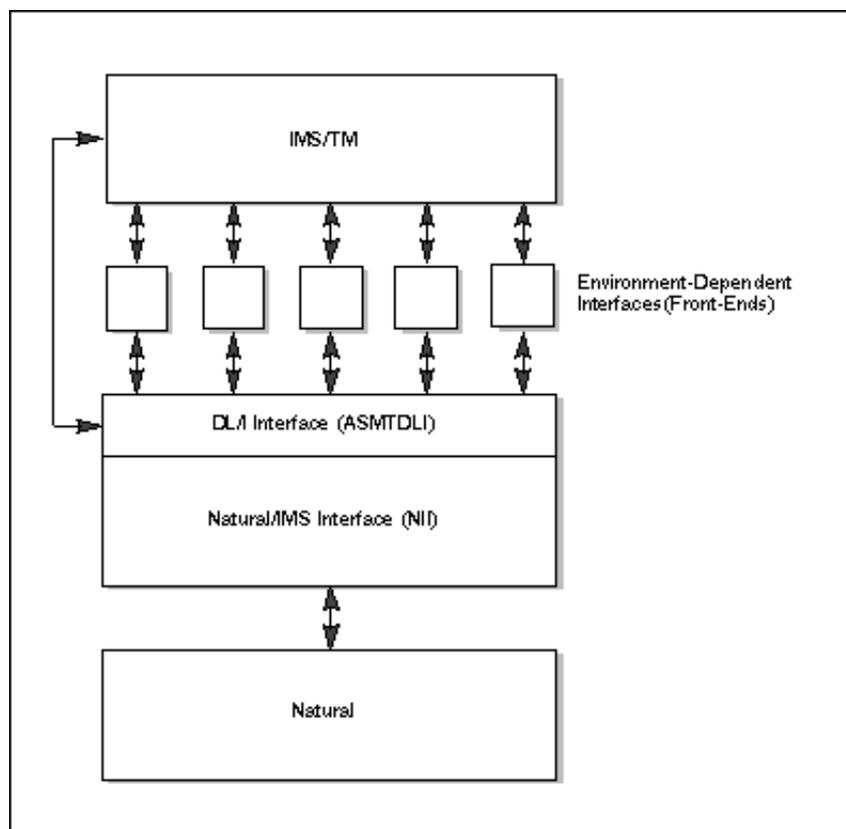
This document describes how Natural runs under various IMS/TM environments. The following topics are covered:

- IMS/TM Interface Overview
- IMS/TM Environments
- Dialog-Oriented Environments
- Message-Oriented Environment
- Batch Message Processing Environment
- Support of the Natural WRITE (*n*) Statement

Additional Information

- **Natural IMS/TM Error Codes** - for a list of the error codes and messages that may be issued by the Natural IMS Interface (NII) refer to Natural IMS/TM Error Codes (in the Natural Messages and Codes documentation).
 - **Installation** - refer to Installing the Natural IMS Interface (in the Natural Installation Guide for Mainframes).
 - **Further information** - refer to the following topics:
Components | Configuration Macros | Service Programs | Service Modules | Special Functions | User Exits | Recovery Handling
-

IMS/TM Interface Overview



IMS/TM Environments

IMS/TM provides three different types of environments:

- message processing regions,
- batch message processing regions,
- off-line DL/I batch region.

To be able to use Natural in each of these environments, different environment-specific interfaces are provided for the Natural IMS interface. The task of such an interface is to receive input (usually a terminal input message) from the environment, to pass the input to Natural for processing and to direct the resulting output back to the correct destination (usually a terminal output message). This way, it is possible to use the functionality of Natural in all available IMS environments.

In addition to different available environments, within each environment, there are different ways of operating.

Natural in a Message Processing Region (MPP Environment)

In a message processing region, Natural online transactions can be one of the following:

- dialog-oriented,
- message-oriented.

Dialog-Oriented Natural

A dialog-oriented Natural session establishes an ongoing interaction with an IMS screen. Input and output messages to and from Natural are logically related and, across dialog steps, Natural saves information so as to be able to correctly process the next input message. In a dialog-oriented way, Natural can be run as either a conversational or a non-conversational transaction.

In a dialog-oriented environment, Natural can be executed in multiple-message processing regions, as Wait-for-Input (WFI) transaction and with the parallel-scheduling option.

To run Natural in dialog-oriented environments, you either have to use the roll server or roll files (see The Roll File and Roll Server).

If the Natural IMS interface detects an error situation, a record containing information about this error situation is written to the IMS log file (see Recovery Handling). Thus, all terminals on which Natural is to be executed and all Natural transaction codes have to be authorized to issue the /LOG command using the automated operator interface (AOI).

Message-Oriented Natural

A message-oriented Natural session processes non-3270-formatted messages from the IMS message queue. The input messages are considered to be unrelated to each other and are not part of a dialog. In a message-oriented way, Natural must be run as a non-conversational transaction.

Natural in a Batch Message Processing Region (BMP Environment)

In a batch message processing region, Natural can have access to the IMS message queue by using an input transaction code. With batch-oriented BMP regions, Natural supports symbolic checkpoint and extended restart. The input messages are non-3270-formatted messages.

Natural in an Off-line DL/I Batch Region

The BMP Natural can also be executed as an off-line DL/I batch job.

If no IOPCB is available, all END TRANSACTION and BACKOUT TRANSACTION statements are ignored (as under NII 2.2).

For diagnostic purposes, the following feature is available: If Natural has been started with dynamic profile parameter TPF=(1), an informal WTO message is issued, indicating the above fact.

Dialog-Oriented Environments

This section discusses special points valid for the dialog-oriented conversational environment only.

- Special Considerations for a Conversational Environment
- Special Considerations for a Non-Conversational Environment
- Special Considerations for an MSC Environment

Special Considerations for a Conversational Environment

The dialog-oriented conversational environment is implemented by the Conversational MPP Interface which is linked with the Natural parameter module to the Conversational MPP Front-End. This front-end is the IMS/TM application program and is scheduled by IMS/TM if an input message for the assigned transaction code is available in the IMS/TM message queue.

The dialog-oriented conversational environment requires a scratch pad area (SPA) of at least 157 bytes plus the NRASTART value.

Special Considerations for a Non-Conversational Environment

The dialog-oriented non-conversational environment is implemented by the Non-Conversational MPP Interface which is linked with the Natural parameter module to the Non-Conversational MPP Front-End. This front-end is the IMS/TM application program and is scheduled by IMS/TM if an input message for the assigned transaction code is available in the IMS/TM message queue.

When a dialog-oriented non-conversational environment is used, the Authorized Services Manager with its SIP function enabled and the Physical Input Edit Routine are prerequisites.

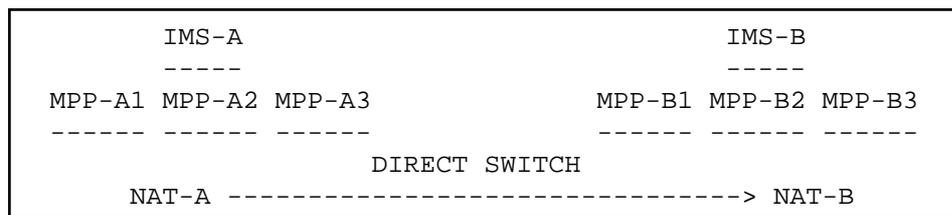
- The Authorized Services Manager is used to simulate the IMS/TM SPA.
- The Physical Input Edit Routine is used to insert the transaction code in front of the input message.

You must specify the same Natural subsystem ID in the

- SPATID of the NIMPARM macro,
- SPATID of the NIMPIXT macro,
- startup parameters of the Authorized Services Manager.

Special Considerations for an MSC Environment

Assuming the following environment, the Natural IMS interface prepares the message X'000500006D' for NAT-B, which means that the terminal user has pressed CLEAR.



Two entries must be created in the transaction code table: the first entry is for NAT-A, the second for NAT-B.

These two entries must specify different offsets for the Natural Reserved Area (NRA) and must ensure that these areas do not overlap.

NAT-B detects that a Natural session is to be started in IMS-B in the usual way and therefore gives control to its session-start exit routine. The session-start exit routine checks the input message for the string X'000500006D' and sets to "0" the length of the input message as seen by Natural.

If no additional logic is provided in either the exit NIIXSTAR or the exit NIIXSSTA, Natural starts a new user session in IMS-B.

It is assumed that IMS-A and IMS-B have different dedicated roll files allocated for Natural.

Both (or more) Natural sessions can communicate with each other by transferring data in the SPA when performing direct program-to-program switching.

For the time being, when two or more Natural sessions exist in such an environment, only the "active" session is terminated correctly.

Message-Oriented Environment

This section describes the message-oriented interface for use with Natural for IMS/TM.

- Introduction to the Message-Oriented Environment
- Operation of the Message-Oriented Environment
- Bootstrap Module NIIBOOT

Introduction to the Message-Oriented Environment

This interface is designed to process nett-data input messages, which means that the messages do *not* represent a 3270 data stream. The message-oriented interface is driven by a user-written Natural program which instructs the interface to access the IMS message queue for the purpose of retrieving input messages.

The message-oriented interface has been created to support non-conversational, non-terminal driven transactions which must be executed as non-conversational MPP transactions.

Operation of the Message-Oriented Environment

The message-oriented interface incorporates functions from both the MPP and the BMP interfaces. The BMP interface is used as a basis, since much of the processing required emulates BMP-type transactions.

Since the message-oriented interface is not terminal-oriented, no messages or screen images are automatically generated to be sent to a terminal. The Natural nucleus is informed that it is running in a batch environment; therefore output is interpreted to be printer output and input is expected from a CMSYNIN file. All output which is normally written to CMPRINT is sent to the IMS/TM destination (normally an LTERM) specified with the Natural profile parameter SENDER. If Natural attempts to retrieve input data and no input data has been supplied by the application through the STACK command, EOF indicates that no input exists and Natural is terminated.

You can set SENDER to a new value at runtime by using the service module CMSNFPRT.

Except for checkpoint processing, Natural for DL/I and Natural for DB2 process as if they were in BMP mode. This is necessary, since one physical scheduling can (and usually will) process several unrelated input messages. Under the conversational MPP interface, all transactions processed during one Natural session and all DL/I requests within this Natural session are considered to be related, requiring maintenance of database positioning and PCB usage. With the non-conversational interface, this Natural for DL/I logic is not applicable.

Since transactions which are processed during one scheduling (and one Natural session) are not related to each other, the retention of Natural session information in the roll file is not required. Thus, no roll dataset needs to be allocated for this interface. A roll slot area is still allocated via GETMAIN and used to store all Natural control blocks and work areas.

Since processing is performed on a message-by-message basis, there is no need for any relocation logic.

With the message-oriented interface, retrieval of all messages from the message queue is initiated by a front-end Natural program. This program must be user-written to meet your specific processing needs. However, it requires a specific structure, as shown in the following:

```

PROGRAM INITIALIZATION
REPEAT
CALL 'CMGETMSG' MESSAGE-AREA MESSAGE-LENGTH
IF MSG-LL = 0 /* QC on GU to message queue
TERMINATE
FETCH RETURN PGMA MESSAGE-AREA
REPEAT
CALL 'CMGETSEG' MESSAGE-AREA MESSAGE-LENGTH
IF MSG-LL = 0 /* QD on GN to message queue
ESCAPE
FETCH RETURN PGMB MESSAGE-AREA
END-REPEAT
END-REPEAT
END

```

The service module CMGETMSG reads the first message segment. The service module CMGETSEG reads all further message segments.

Since Natural cannot read input from CMSYNIN, it is required to use the Natural stack for input. This is done by using the Natural profile parameter STACK.

It is your responsibility to ensure that the IMS message queue is accessed by your application prior to the termination of Natural. If not, the Natural transaction abnormally ends with IMS abend code 462, indicating that a GU to the message queue has not been performed.

To obtain these Natural messages even in the case of an abnormal termination, you are recommended to define the first alternate PCB as an EXPRESS PCB.

The message-oriented environment is implemented by the NTRD Interface which is linked with the Natural parameter module to the NTRD Front-End. This front-end can either be called directly by the IMS/TM program controller or via a bootstrap module.

If it is called directly by IMS/TM, this front-end is the IMS/TM application program which is scheduled by IMS/TM if an input message for the assigned transaction code is available in the IMS/TM message queue. You are recommended to use a Natural profile which contains the required STACK parameter. Specify PROFILE=PROGRAM in your Natural parameter module and create a profile with a name equal to the transaction code with which the interface is invoked. This way, you have the flexibility to use a different profile with a different STACK for each transaction code used.

If it is called via a bootstrap module, this bootstrap module is the IMS/TM application program which is scheduled by IMS/TM if an input message for the assigned transaction code is available in the IMS/TM message queue. This bootstrap module provides a string of dynamic profile parameters, one of which is the STACK profile parameter, and calls the NTRD front-end whose name is specified during the generation of the bootstrap module.

Bootstrap Module NIIBOOT

The bootstrap module is called by the program controller DFSPCC20. It loads and calls the message-oriented front-end module passing the necessary parameters. The bootstrap module NIIBOOT is delivered as source module and has to be assembled and link-edited during the installation process. The source code is contained in the macro NIMBOOT.

NIMBOOT includes the following parameters:

DRIVERN | ENVTNAM | DYNPARN

Parameter	Possible Values	Default	Comment
DRIVERN	Any valid OS/390 module name	None	This parameter specifies the name of the front-end module.
ENVTNAM	Any valid OS/390 module name	None	This parameter specifies the name of the environment table. This parameter is optional. If it is not specified, the environment table is determined by the entry in the transaction code table which corresponds to the transaction code used.
DYNPARN	Any character string of up to 80 characters.	None	This parameter is used to define a valid string of up to 80 characters of Natural dynamic parameters.

Batch Message Processing Environment

The Batch Message Processing (BMP) environment is implemented by the BMP Interface which is linked with the Natural parameter module and the work file/print file access routine NATWKFO to the BMP Front-End. This front-end is the IMS/TM application program which is specified in the BMP JCL.

A standard batch Natural is executed in a Batch Message Processing region. In comparison with the standard batch Natural run, the optional input dataset CONTROL may be used.

The optional BMP CONTROL File contains a maximum of two input cards.

- The first input card must contain the following keyword:

Keyword	Meaning
ENV-TAB=	The name of the environment table to be used.

Example:

ENV-TAB=ENVBMP0

- The second input card of the CONTROL file contains the dynamic Natural parameters.

Using Both the CMPRMIN Dataset and the CONTROL File to Pass Dynamic Natural Parameters

If the CMPRMIN dataset is also used to pass dynamic Natural parameters, the input of CONTROL is appended to the input of CMPRMIN. This means the parameters specified in CONTROL overwrite the parameters specified in CMPRMIN.

Working without CONTROL File

If the CONTROL file is not used, the name of the environment table is determined by the entry in the transaction code table which corresponds to the transaction code used (transaction-oriented BMP) or to the PSB name used (batch-oriented BMP).

Support of the Natural WRITE (n) Statement

With the WRITE (n) statement, up to 31 different reports on different printers can be produced within the same Natural program. The reports are sent to the IMS terminals specified either in the Natural parameter module or by using the Natural DEFINE PRINTER (n) statement. You have to specify AM=IMS in the NTPRINT macro which controls the report.

To be able to use this statement, define as many additional alternate TP-PCBs in your PSB as the number of parallel reports you want to create within the same Natural program, and specify the number of additional alternate TP-PCBs in your transaction code table by using the parameter WRKPCBs.



Attention: Be aware that the first alternate TP-PCB is used by the Natural IMS interface.

When using the WRITE (n) statement in a dialog-oriented environment, the following restriction applies:

The generation of a report cannot span one or more screen I/Os. If you want to use the same printer after a screen I/O, you have to close it explicitly before the screen I/O using the CLOSE PRINTER(n) statement.

To create reports, the following parameters of the NTPRINT macro are relevant:

AM | DEST | BLKSIZE | DRIVER | NAME | FORMS | DISP | COPIES | CLASS | PRTY

Parameter	Meaning
AM	Must be set to "IMS".
DEST	Specifies the IMS/TM destination.
BLKSIZE	Specifies the size of the buffer which is sent to the destination. Report lines are buffered.
DRIVER	Specifies the driver to be used to create the report. For a list of possible values, see the PRTDRIV parameter of the Natural/IMS interface parameter module. The driver determines where you want to have the form feed (at the start of the report, the end, both the start and the end, or no form feed), where you want to start your page (on line 1 or on line 2 for NII 2.2 compatibility) and where you want to print your report (SCS or non-SCS printer). In addition, you can specify that you want to use the JES API.
NAME FORMS DISP COPIES CLASS PRTY	These parameters are only evaluated if you use the JES API.

Hints Concerning NTPRINT and CLOSE PRINTER

NTPRINT Settings

You are strongly recommended that you always use the defaults for the OPEN and CLOSE subparameters in the NTPRINT/PRINT definition for IMS/TM printers (i.e. for printers defined with AM=STD). This means either don't specify any values for OPEN and CLOSE or use the defaults OPEN=ACC and CLOSE=CMD.

This is especially important if you have statically defined a printer in the NATPARM for a different access method with other options for OPEN and CLOSE and if you dynamically overwrite the access method with AM=IMS. In this case, always specify AM=IMS,OPEN=ACC,CLOSE=CMD together.

Note:

The NTPRINT options are merged with the dynamically specified PRINT options, even though the access method has been overwritten.

Problems which may occur with non-default values:

1. With OPEN=OBJ you may print to a wrong destination or get a NAT8211 if the OUTPUT option has been specified in a DEFINE PRINTER statement. With OPEN=OBJ the printer is opened before the OUTPUT overwrite has been evaluated and the printer destination used is not the one which is specified in the OUTPUT option but the one specified with the PRINT parameter.
2. With CLOSE=FIN the printer is not closed at CLOSE PRINTER time but at FIN time. This means that the CLOSE may come after a GU has been issued to the message queue and the destination has been reset in the TP PCB. This will lead to NII error NII3641 for IMS/TM status code QF (MPP) or A3 (BMP and OBMP/NTRD). With CLOSE=CMD the printer is really closed with the CLOSE PRINTER.

Usage of CLOSE PRINTER or DEFINE PRINTER

A report written to an IMS/TM printer is implicitly closed by IMS/TM with the next GU call (i.e. either at terminal I/O or through CMGETMSG).

This means, IMS/TM will print the report regardless of a CLOSE PRINTER or DEFINE PRINTER statement in the program.

For Natural, the printer is still open, and the next WRITE statement with the same report number will continue the already printed report which will lead to a NAT1518 error.

Scenario:

```
DEFINE PRINTER (1)
  WRITE (1) 'line 1'
  INPUT 'Press ENTER' or CALL 'CMGETMSG' (both issue a GU)
  WRITE (2) 'line 2'
```

The INPUT/CMGETMSG will "physically" close the printer and IMS/TM will print a report containing the line 'line 1'.

As the printer is still "logically" open to Natural, the line 'line 2' will not start a new report and error NAT1518 will be caused as the destination is purged by the GU call.

You are therefore strongly recommended to observe the following rule:



A CLOSE PRINTER is required if, after the GU, the report with the same number is continued.

Please note that the DEFINE PRINTER statement does an implicit close in which case the CLOSE PRINTER is obsolete, for example:

Correct	Correct	Wrong (NAT1518)
<pre>REPEAT DEFINE PRINTER (1) WRITE (1) INPUT LOOP</pre>	<pre>DEFINE PRINTER (1) REPEAT WRITE (1) CLOSE PRINTER (1) INPUT LOOP</pre>	<pre>DEFINE PRINTER(1) REPEAT WRITE(1) INPUT LOOP</pre>