

Natural under IMS/TM - Service Programs

This part of the Natural IMS Interface documentation describes the service programs of the Natural IMS/TM Interface. The following topics are covered:

- Introduction to the Natural IMS/TM Interface Service Programs
 - Description of the Natural IMS/TM Interface Service Programs
-

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:
Environments | Components | Configuration Macros | Service Modules | Special Functions | User Exits | Recovery Handling
-

Introduction to the Natural IMS/TM Interface Service Programs

Purpose of Natural IMS/TM Interface Service Programs

Service programs are Natural subprograms which provide Natural under IMS with additional functionality. You can call them from within a Natural program using a standard CALLNAT statement.

Location of Service Programs

The service programs are provided in the library SYSEXTP and you must copy them to the SYSTEM or steplib library.

Sample Natural programs to invoke the service programs are also provided in the library SYSEXTP.

Common Return Codes

The last parameter in each service program is the return code whose format is (I4). The following return code values are common for all service programs:

0 OK

-1 Non-supported function. This is an internal error, please contact Software AG support.

For specific return code values, refer to the individual service program descriptions below.

Error Handling

If an error occurs, either a Natural error message is issued or the session is terminated with a Natural IMS error message.

Description of the Natural IMS/TM Interface Service Programs

The following service programs are described below:

NIIBRCST | NIICMD | NIIDEFT | NIIDEFTX | NIIDIRT | NIIDIRTX | NIIEMOD | NIIGCMD | NIIGMSG | NIIGSEG | NIIGSPA | NIIIMSIN | NIIISRTF | NIIISRTM | NIIPCBAD | NIIPCOM | NIIPMSG | NIIPSBAD | NIIPSPA | NIIPURG | NIIRETRM | NIISASD | NIIU3962

NIIBRCST - Send Passed Message to Terminal

Sends the passed message to the specified terminal using the message output descriptor specified in the MOD_name parameter.

The following parameters are provided:

Terminal_name | Message | Message_length | MOD_name | Return_code

Terminal_name	(A8)
Message	(A1/1:V)
Message_length	(I4)
MOD_name	(A8)
Return_code	(I4)

Specific Return Code Values: None.

Sample Program: NIPGMSG

NIICMD - Pass IMS Command to IMS

Passes the IMS command specified to IMS. If there is a reply, it is moved into the reply area provided. If the reply does not fit into the reply area, it is truncated and the return code is set to 4.

The following parameters are provided:

Command | Command_length | Reply_area | Reply_area_length | Reply_length | Status_code | Return_code

Command	(A1/1:V)	Input
Command_length	(I4)	Input
Reply_area	(A1/1:V)	Input/Output
Reply_area_length	(I4)	Input
Reply_length	(I4)	Output
Status_code	(A2)	Output
Return_code	(I4)	Output

Specific Return Code Values: 4 (reply truncated)

Sample Program: NIPCMD

NIIDEFT - Prepare Deferred Switch to Natural Transaction Code

Prepares a deferred switch to the specified Natural transaction code. With the next terminal I/O, the output is sent to the terminal and the next input from this terminal is processed by the transaction code specified in the parameter Transaction_code.

The following parameters are provided:

Transaction_code | Return_code

Transaction_code	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPDEFT

NIIDEFTX - Prepare Deferred Switch to Non-Natural Transaction Code

Prepares a deferred switch to a non-Natural transaction code. With the next terminal I/O, the output is sent to the terminal using the given MOD_name and the next input from this terminal is processed by the transaction code specified in the parameter Transaction code.

If the suspend flag is set to "Y", the Natural session will be suspended and can be resumed later. If the Natural session is resumed, it will first issue the last Natural screen.

If the suspend flag is set to "Y" you may not switch from a conversational Natural session to a non-conversational transaction code. If you try to do so, a Natural error message is issued.

The following parameters are provided:

Transaction_code | Transaction_type | Suspend_flag | MOD_name | Message | Message_length | Return_code

Transaction_code	(A8)	Input
Transaction_type	(A4)	Input Possible values: CONV for conversational NONC for non conversational
Suspend_flag	(A1)	Input Possible values: Y the Natural session will be suspended else the Natural session will be terminated
MOD_name	(A8)	Input
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPDEFTX

NIIDIRT - Prepare Direct Switch to Natural Transaction Code

Prepares a direct switch to a specified Natural transaction code. On the next terminal write, the CHNG command to the specified transaction code is issued and the Natural screen is inserted using the alternate TP PCB.

If you switch from a conversational Natural session to a non-conversational one, the conversation is terminated and a dummy message using MOD_name NIIMODNC is inserted. This message unprotects the screen temporarily, and is thus overwritten by the first screen of the non-conversational Natural session.

Transaction_code | Return_code

Transaction_code	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPDIRT

NIIDIRTX - Prepare Direct Switch to Transaction Code

Prepares a direct switch to the specified transaction code. On the next terminal write, the CHNG call for the new transaction code is issued and the message and or the SPA are inserted using the alternate TP PCB. The transaction type defines the type of the new transaction code.

- If you switch from a conversational transaction code to a non-conversational one, the conversation is finished by issuing a dummy message using MOD_name NIIMODN, which unprotects the screen temporarily, thus it will be overwritten by the screen issued from the non conversational transaction code.
- If the suspend flag is set to "Y", the Natural session is suspended and may be resumed at a later time. When the Natural session is resumed, the last Natural screen is issued.
- If the suspend flag is set to "Y" you may not switch from a conversational Natural to a non conversational

transaction code. If you try to do so, a Natural error message will be issued.

- If message length is set to zero, no message at all is inserted. This however is only possible if you switch to a conversational transaction code.

The following parameters are provided:

Transaction_code | Transaction_type | Suspend_flag | Message | Message_length | Return_code

Transaction_code	(A8)	Input
Transaction_type	(A4)	Input Possible values: CONV for conversational transaction code NONC for non-conversational transaction code
Suspend_flag	(A1)	Input Possible values: Y the Natural session will be suspended else the Natural session will be terminated
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPDIRTX

NIIEMOD - Modify Setting of Module Output Descriptor

Modifies the current setting of the module output descriptor to be used in the insertion of the last message in a Natural session and sets it to the value specified in the parameter MOD_name.

The following parameters are provided:

MOD_name | Return_code

MOD_name	(A8)	Input
Return_code	(I4)	Output

Sample Program: NIPEMOD

NIIGCMD - Retrieve Next Reply Segment of Previous IMS/TM Command

Retrieves the next reply segment of a previously issued IMS/TM command. The length of the reply is return in the parameter reply length. If the reply does not fit into the reply area, the reply is truncated and return code 4 is issued.

The following parameters are provided:

Reply_area | Reply_area_length | Reply_length | Status_code | Return_code

Reply_area	(A1/1:V)	Input/Output
Reply_area_length	(I4)	Input
Reply_length	(I4)	Output
Status_code	(A2)	Output
Return_code	(I4)	Output

Specific Return Code Values: 4 (reply truncated)

Sample Program: NIPCMD

NIIGMSG - Retrieve First Segment Next Message

Retrieves the first segment of the next message from the message queue by issuing a GU. The message area will contain the retrieved message including the leading LLZZ bytes. If there are no messages in the message queue, LLZZ is set to zero.

The following parameters are provided:

Message_area | Message_area_length | Return_code

Message_area	(A1/1:V)	Output
Message_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Programs: NIPGMSG, NIPGSEG

NIIGSEG - Retrieve Next Segment of Input Message

Retrieves the next segment of the input message by issuing a GN call. The message area will contain the retrieved message including the leading LLZZ bytes. If there are no more message segments in the current message, LLZZ is set to zero.

The following parameters are provided:

Message_area | Message_area_length | Return_code

Message_area	(A1/1:V)	Output
Message_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPGSEG

NIIGSPA - Retrieve Data from SPA Beginning

Retrieves data from the SPA beginning at the specified offset in the specified length.

The following parameters are provided:

Offset | Length | Area | Return_code

Offset	(I4)	Input
Length	(I4)	Input
Area	(A1/1:V)	Input/Output
Return_code	(I4)	Output

Specific Return Code Values: 4

The retrieved data resides entirely or partially within the part of the SPA reserved for Natural.

Sample Program: NIPGSPA

NIIIMSIN - Retrieve IMS Environment Info

Retrieves the IMS environment information using the INQY ENVIRON call. If you specify a reply_area_length smaller than 102, the reply will be truncated and you will receive return code X'0100' with reason code X'000C'.

The following parameters are provided:

Reply_area | Reply_area_length | Return_code

Reply_area	(A1/1:v)	Output
Reply_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: nnxx

nn: The first two bytes contain the AIB return code. *xx*: The second two bytes contain the AIB reason code. AIB denotes "Application Interface Block" and is used when calling IMS through the AIBTDLI interface.

Sample Program: NIPIMSIM

NIIISRTF - Create Multi-Segment Messages

Creates multi-segment messages. NIIISRTF performs the CHNG call for the specified destination and inserts the first message segment without performing a PURG call. Further message segments may be inserted using NIIISRTM. The message has to be terminated using NIIPURG. The LLZZ bytes are created by the service module.

The following parameters are provided:

Destination | Message | Message_length | Return_code

Destination	(A8)	Input
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPISRTM

NIIISRTM - Insert Message Segment into Message Queue

Inserts the next message segment into the message queue without performing a CHNG or a PURG call. The LLZZ bytes are created by the service module.

The following parameters are provided:

Message | Message_length | Return_code

Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPISRTM

NIIPCBAD - Return PSB Name and PCB Address

Returns the currently scheduled PSB name and the address of the PCB identified by the logical name. If the logical PCB name is not defined in the transaction code table, a Natural error message is issued.

The following parameters are provided:

PSB_name | Logical_PCB_name | PCB_address | Return_code

PSB_name	(A8)	Output
Logical_PCB_name	(A8)	Input
PCB_address	(B4)	Output
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPPCBAD

NIIPCOM - Move Data to Reply Area

Moves the data provided in the data area into the reply area specified in the NIIBOOT5 call at the specified offset in the specified length. NIIPCOM may be called from the server environment only.

The following parameters are provided:

Offset | Data_area | Length | Return_code

Offset	(I4)	Input
Data_area	(A1/1:V)	Input
Length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values: 4 (calling environment, not server environment)

Sample Program: NIPPROM

NIIPMSG - Send Message

Sends a message using a given MOD_name to the destination which is represented by the I/O PCB. The message is taken from the message area in the specified message area length. The message area must not contain the leading LLZZ bytes. In this way you can send MFS-formatted output messages back to the originator of the input message.

The following parameters are provided:

Message | Message_length | MOD_name | Return_code

Message	(A1/1:V)	Input
Message_length	(I4)	Input
MOD_name	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPPMSG

NIIPSBAD - Return PSB Address

Returns the address of the PSB which is the address of the PCB address list.

The following parameters are provided:

PSB_address | Return_code

PSB_address	(B4)	Output
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPBOOTS

NIIPSPA - Replace Data in SPA

Replaces the data located in the SPA at the specified offset in the given length by the data provided in the data area.

The following parameters are provided:

Offset | Length | Data_area | Return_code

Offset	(I4)	Input
Length	(I4)	Input
Data_area	(A1/1:V)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

An attempt to override the header of the SPA (first 14 bytes) and/or data residing in the Natural-reserved area is refused and a Natural error message is issued.

Sample Program: NIPPSPA

NIIPURG - Issue PURG Call

Issues a PURG call.

The following parameter is provided:

Return_code

Return_code	(I4)	Output
-------------	------	--------

Specific Return Code Values: None.

Sample Program: NIPISRTM

NIIRETRM - Move Data into Message Area

Moves data from the input message beginning at the specified offset in the specified length into the provided message area.

The offset is calculated from the LLZZ bytes.

The following parameters are provided:

Offset | Length | Message_area | Return_code

Offset	(I4)	Input
Length	(I4)	Input
Message_area	(A1/1:V)	Input/Output
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPRETRM

NIISASD - Modify SENDER and OUTDEST Settings

Modifies the current setting of the Natural dynamic parameters SENDER and OUTDEST.

The following parameters are provided:

Sender | Outdest | Return_code

Sender	(A8)	Input
Outdest	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values: None.

Sample Program: NIPNTRD

NIIU3962 - Terminate Session

Terminates the session with user abend 3962 and produces a dump.

The following parameter is provided:

Return_code

Return_code	(I4)	Output
-------------	------	--------

Specific Return Code Values: None.

Sample Program: NIPU3962