

Natural in Batch Mode

This document contains general considerations that apply when running Natural in batch mode.

The following topics are covered:

- Datasets Required for Batch Mode Operation
- Natural Datasets
- Adabas Datasets
- Sort Datasets
- Subtasking Session Support for Batch Environments

For special considerations that refer to Natural in batch under the operating systems OS/390, VSE/ESA, CMS and BS2000/OSD, refer to the corresponding documents:

- Natural in Batch under OS/390
 - NATOS - Natural OS/390 Batch Interface
 - NTOS Macro - Natural for OS/390 Generation Parameters
 - Natural Datasets Used under an OS/390 Batch Session
 - Natural in Batch under VSE/ESA
 - NATVSE - Natural VSE/ESA Batch Interface
 - NTVSE - Generation Parameters for Natural under VSE/ESA
 - Natural Datasets Used under a VSE/ESA Batch Session
 - Debugging Facilities for Natural under VSE/ESA
 - Natural in Batch under CMS
 - Natural in Batch under BS2000/OSD
 - Natural Datasets Used under a BS2000/OSD Batch Session
 - KEYWORD Parameters
 - BS2000/OSD Job Variables
-

Datasets Required for Batch Mode Operation

The datasets required for executing Natural in batch mode depend upon which options have been selected, such as the Adabas mode of operation (single-user or multi-user), the installation's sort program, and whether or not Natural work files, additional reports, and input data for Natural INPUT statements are used.

These datasets are:

- Natural Datasets
- Adabas Datasets
- Sort Datasets

In addition, certain environment-specific Natural datasets are required for batch mode operation. For details, refer to the following sections:

- Natural Datasets Used under an OS/390 Batch Session,
- Natural Datasets Used under a VSE/ESA Batch Session or
- Natural Datasets Used under a BS2000/OSD Batch Session.

Natural Datasets

The following datasets are available: CMPRMIN | CMSYNIN | CMOBJIN | CMPRINT | CMPLOG | CMTRACE | CMHCOPIY | CMPRT

CMPRMIN - Dynamic Parameter Dataset

CMPRMIN can be used as dynamic parameter dataset to overcome the length restriction for the character string in the job control PARM keyword of the EXEC statement.

All input records from CMPRMIN are concatenated into one parameter string. Only the first 72 positions of each CMPRMIN record are significant. Trailing blanks at the end of each record are truncated; no commas are inserted.

Additional dynamic parameters from the job-control PARM keyword can be supplied. They are concatenated after the parameters from CMPRMIN, which means that the PARM character string can be used to overwrite dynamic parameters specified in the CMPRMIN dataset.

CMSYNIN - Primary Input

CMSYNIN is used for the primary input file that contains Natural commands, Natural source programs, and (optionally) data to be read by INPUT statements during the execution of Natural programs.

The number of characters actually processed per line is determined by the current setting of the profile parameter SL. This setting applies for both source statement and execution time input data. This enables identification or sequence numbers to be placed in the rightmost columns of every record if desired.

CMOBJIN - Input for Natural INPUT Statements

CMOBJIN is used for data intended to be read by Natural INPUT statements. This type of data can alternatively be placed in the CMSYNIN dataset immediately following the relevant source program or the relevant RUN or EXEC command.

When the setting for the profile parameter OBJIN is "N", Natural reads input from CMSYNIN. When OBJIN is set to "Y", Natural reads input from CMOBJIN.

When OBJIN is set to "R", Natural determines which option has been selected for a particular session depending upon the presence or absence of a CMOBJIN DD statement.

DCB considerations for CMOBJIN are the same as for CMSYNIN. The record read is interpreted up to the number of characters as specified with the parameter SL. If an error occurs, Natural reacts according to the setting of the parameter CC.

CMPRINT - Primary Report Output

CMPRINT is used for the primary output report resulting from DISPLAY, PRINT and WRITE statements in a Natural program.

CMPLOG - Optional Report Output for Dynamic Parameters

If profile parameter PLOG is set to ON, all dynamic profile parameters are written to this dataset during session initialization. If CMPLOG is not defined, CMPRINT is used instead.

CMTRACE - Optional Report Output for Natural Tracing

If profile parameter ETRACE is set to ON, all trace output is written to this dataset during the session.

CMHCOPY - Optional Report Output for Hardcopy

Hardcopy output destination. This dataset's name can be changed with the profile parameter HCDEST.

CMPRT - Additional Reports

CMPRT nn is used for each additional report referenced by any Natural program compiled or executed during the session. " nn " must be a two-digit decimal number in the range 01-31 corresponding to the report number used in a DISPLAY, PRINT or WRITE statement.

Instead of CMPRT nn , another file name may be used by setting the DEST subparameter of profile parameter PRINT to an appropriate value, for example:

```
PRINT=((nn),...,DEST=PRINTFIL).
```

CMWKF - Natural Work Files

CMWKF nn is used for each Natural work file referenced by any Natural program compiled or executed during the session. " nn " must be a two-digit decimal number in the range 01 - 32 corresponding to the number used in a READ WORK FILE or WRITE WORK FILE statement.

Instead of CMWKF nn , another file name may be used by setting the DEST subparameter of profile parameter WORK to an appropriate value.

Adabas Datasets

Adabas datasets must be specified only in single-user mode. They are identical to those required for the execution of any normal application program using Adabas. See the relevant Adabas documentation for detailed information on Adabas datasets.

Sort Datasets

Sort datasets must be specified if a Natural program containing a SORT statement is to be executed during the Natural session.

The requirements are identical to those for execution of a normal COBOL or PL/1 application program that invokes the operating system sort program and can vary according to the sort program in use.

Natural does not require the intermediate datasets SORTIN and SORTOUT, but communicates with the sort program via the E15 and E35 user-exit routine interfaces.

Subtasking Session Support for Batch Environments

- Purpose
- Prerequisites
- Functionality
- Starting A Natural Session
- Starting A Subtask
- Accessing the User Parameter Area

Purpose

With subtasking support, you can run multiple Natural batch sessions within one address space. This allows parallel processing within one address space, rather than executing subsequent job steps, and can increase throughput dramatically.

Typically, client-server applications and products would take advantage of this functionality, for example, the Natural remote procedure call. Multiple server subtasks can be started to communicate with remote clients.

Prerequisites

If you wish to restart the Natural nucleus, it must be linked as a reentrant module (linkage editor option RENT).

The Adabas link routine (ADALNK) must be generated with reentrancy support.

Functionality

You start a subtask by issuing a CALL statement from a Natural program. The new Natural session ("subtask") is started with an extended front-end parameter list. This list contains up to three parameter sets:

- dynamic Natural profile parameters,
- startup parameters,
- user parameters.

Variable names for standard I/O datasets (for example CMPRINT) and other parameters for the batch interface startup can be passed from the starting program in the startup parameter area. Standard I/O datasets can be undefined or dummy datasets; they can be owned by one session or shared by multiple sessions.

Furthermore, a CALL interface is provided for reading the user parameter area with a Natural program.

Starting a Natural Session

- Extended Parameter List
- Startup Parameter Area
- User Parameter Area

Extended Parameter List

The Natural batch interface without extended parameter list gets initial control from the operating system using standard linkage call. Register 1 points to an address with high-order bit on as the last address indicator. This address points to a halfword field containing the length of the following parameter area.

The extended parameter list contains up to three parameter addresses. This is indicated by the high-order bit in the last address which can be the first, second or third address. All parameter addresses point to a halfword field containing the parameter length of the following parameter area. Zero length indicates that there is no parameter area.

- The first parameter area contains the dynamic profile parameters for the Natural session.
- The second contains special startup parameters for the initialization of the batch interface.
- The third contains a user parameter area which can be accessed during the Natural session.

Startup Parameter Area

The startup parameter area is a name table with 16-byte fixed-length entries. One entry contains an 8-byte keyword followed by the 8-byte assigned value. Keywords and values must be padded with trailing blanks, if necessary. The following keywords are valid:

CMHCOPY	Permanent hardcopy destination
CMSYNIN	Command input dataset name
CMOBJIN	Object input dataset name
CMPRINT	Standard output dataset name
CMPRMIN	Dynamic parameter input dataset name
CMPLDG	Dynamic parameter output dataset name
CMTRACE	Trace output dataset name
INITID	Job step name (system variable *INIT-ID)
MSGCLASS	Spool class for dynamic allocation of CMPRINT and CMTRACE
NATRJE	Job submission dataset name
STEPLIB	Program load library name (see also profile parameter LIBNAM, Name of Load Library, OS/390 only)
SUBPOOL	OS/390 storage subpool (0 - 127, right justified)
USERID	Initial user identification (system variable *INIT-USER)

The usage of these entries is optional and no particular sequence is required. A blank value for a dataset means that this dataset is not available or is empty.

Platform:	Requirement:
VSE/ESA	By default, all print output (that is, the one resulting from CMPRINT, CMHCOPY, CMTRACE and CMPLDG) is routed to SYSLST. An overwrite specification for these files starting with "SYS" is considered a VSE/ESA system number overwrite. Possible format is <i>SYSnnn</i> where <i>nnn</i> is a three-digit number in the range from 000 to 099; if you specify an invalid number <i>nnn</i> , it is ignored.

User Parameter Area

The format of the user parameter area is free. It can be accessed from any Natural program by a special CALL interface see Accessing the User Parameter Area.

Starting A Subtask

The following call interface is supplied to be used by Natural programs to start a subtask in the same address space.

PGMNAME	Natural nucleus name getting control (mandatory). To restart with the same nucleus, an asterisk can be specified as the first character. The actual nucleus name is passed back in this field.
NATPARML	Natural dynamic parameter area
STRPARML	Startup parameter area
USRPARML	User parameter area

All parameter areas must start with the length of the following parameters. The following example illustrates the usage of CMTASK.

Example:

```

DEFINE DATA LOCAL
01 PGMNAME (A8) INIT <'*'>
01 PARM1
02 NATPARML (I2) INIT <30>
02 NATPARMS (A30) INIT <'INTENS=1,IM=D,STACK=MYPROG'>
01 PARM2
02 STRPARML (I2) INIT <32>
02 STRPARAM1 (A16) INIT <'CMPRINT SYSPRINT'>
02 STRPARAM2 (A16) INIT <'CMPRMIN MYPARMS'>
01 PARM3
02 USRPARML (I2) INIT <80>
02 USRPARMS (A80) INIT <'special user parameters'>
END-DEFINE
CALL 'CMTASK' PGMNAME NATPARML STRPARML USRPARML
END
    
```

A sample program, ASYNBAT, can be found in library SYSEXTP.

Accessing the User Parameter Area

The user parameter area passed during startup can be read from any Natural program with the following CALL statement:

```
CALL 'CMUPARM' USRPARML USRPARMS
```

USRPARML is the length (I2) of the USRPARMS area (before the call) and the length of the data returned (after the call). USRPARMS is the parameter data area.

If the length of the data to be returned is greater than the area length, the data is truncated to the area length. The following return codes are possible:

0	Data successfully moved
4	Data moved but truncated
8	No data available
12	Length value not positive
16	Insufficient number of parameters

A sample program, GETUPARM, can be found in library SYSEXTP.