

Using Predict in Batch Mode

Most Predict functions can be used either online or in batch mode. Batch mode is useful

- when processing a large number of objects
- when re-documenting entire libraries
- when reports are created regularly, for example with the same restrictions and output options.

This section covers the following topics:

- Where Predict Commands are Described
- Batch Mode Options
- Batch Mode Input
- Condition Codes
- Natural Workfiles and Additional Reports in Batch Mode
- Examples for Batch Mode Input

Where Predict Commands are Described

- Most commands are described in the section Predict Commands in the **Predict Reference documentation**.
- For generation, incorporation and comparison functions:
the valid keywords depend on the type of external object and are listed in the appropriate section of the respective sections of this documentation.
- For Coordinator functions:
in the section Data Transfer Examples in the **Predict Coordinator documentation**.
- For conversion functions:
in the section Converting Dictionary Data in Batch Mode of section **Conversion** in this documentation.

Keywords are also listed in the help text displayed by entering ? in the first input field of a particular function, for example Object ID for generation functions, or Retrieval type for retrieval functions. Command descriptions can easily be found using the index.

Batch Mode Options

This section contains:

- Overview of Possible Functions
- Using Batch Mode with Different Operating Systems

Overview of Possible Functions

The following functions can be performed in batch mode.

- **Predict Functions (Library SYSDIC)**
 - All generation functions. See the section Generation in the **External Objects in Predict documentation**.
 - Punch out previously generated code command: PUNCH or WRITE. See the section Predict Commands in the **Predict Reference documentation**.
 - All comparison functions. See the section Comparison in the **External Objects in Predict documentation**.
 - All incorporation functions except Incorporate DDM. See the section Incorporation in the **External Objects in Predict documentation**.
 - Produce retrieval reports. See the sections Retrieval, Active Retrieval and LIST XREF for Natural of the

Predict Reference documentation.

- Execute the preprocessor. See the section Preprocessor in the **External Objects in Predict documentation.**
- File Implementation. See the section File Implementation Plans in the **External Objects in Predict documentation.**
- The following special functions. See the section Special Functions in this documentation.
 - Delete old sets
 - Reposition implementation data
 - Recover
 - Consistency of Predict
- **Conversion Functions (Library SYSDICCO)**
Convert Predict data. See the section Conversion in this documentation.
- **Coordinator Functions (Library SYSDICBE)**
Unload/load Predict data. See the **Predict Coordinator documentation.**
- **Metadata Administration Functions (Library SYSDICMA)**
All metadata administration functions must be performed online.

Using Batch Mode with Different Operating Systems

Not all functions are available on all operating systems. This section lists the operating systems in which Predict can be run and lists any restrictions that apply to the particular operating system.

- **OS/390, VSE, BS2000/OSD, CMS**
All Predict commands available in batch mode can be used.

For some functions, work files or reports must be specified. The names of these files and the record format/length depend on the operating system and are given in the section Natural Workfiles and Additional Reports in Batch Mode.

- **UNIX**
See the Natural documentation on installation and operation.

Batch Mode Input

This section contains:

- Prerequisites
- Recommendation
- Starting Batch Input
- Terminating Batch Input
- Entering Predict Commands in Batch Mode

Prerequisites

To execute Predict functions in batch mode, a Natural batch session must be started. The following Natural parameters should always be set:

```
MAXCL=0
CC=OFF
INTENS=1
LANG=1
MADIO=0
```

Recommendation

System Variable *DEVICE

When entering commands in batch mode, do not specify any profile parameters which influence the system variable *DEVICE.

For example: If you specify PC=ON, the system variable *DEVICE=PC will be set.

Starting Batch Input

Predict functions that can be performed in batch mode are located in libraries SYSDIC, SYSDICBE or SYSDICCO. See Overview of Possible Functions.

Logon to the required library with the command:

LOGON<library>

or, if you are working with Natural Security,

LOGON<library><user-ID><password>

Then code the following command to put Predict into the required state for processing batch commands:

MENU

Terminating Batch Input

There are two commands to terminate your Predict/Natural session. With both commands, global variables are reset, but values in the Predict hot object table remain.

FIN

The FIN command terminates the Predict/Natural session.

END or .

The END or . command will perform a logoff in Natural Security if no command is on the stack, and the input mode is set to forms mode (IM=F). The other parameters are not altered.

Entering Predict Commands in Batch Mode

Any number of Predict commands may be coded between MENU and FIN or END. The batch mode commands are the same as the Predict commands in online mode. The following rules apply:

Format

In OS/390, VSE and BS2000/OSD environments, input is read in fixed-length format of 80-character records. The last 8 characters are ignored.

Input Mode

Input mode is always delimiter mode (Natural profile parameter IM=D).

Continuation Records

If the parameters will not fit in one record, the last parameter in the record must be followed by a percent character (%). Parameters in the next line are then concatenated.

Assign Character

If entering commands in keyword form (see below), an assign character is used to separate the keyword and the parameter value. There must be no blanks between keyword, assign character and parameter value. Throughout the documentations, an equals sign (=) is used as the assign character.

You can change the assign character with the Natural parameter IA.

Input Delimiter

Input parameters may be separated by blanks or by the input delimiter character. Throughout the documentations, a comma is used as the delimiter character.

You can change the input delimiter with the Natural parameter ID.

Comments

Comments in batch jobs must be preceded with an asterisk.

Entering Parameters in Line Following Command

With most commands the command parameters are entered in the same line as the command word. With the commands listed below, however, parameters are entered in the line(s) following the command.

- GENERATE ...
- COMPARE ... (with I option)
- INCORPORATE ...
- IMPLEMENT FILE
- SPECIAL ...

For example:

```
GENERATE COBOL  
FILE-ID=TEST-FILE,REPLACE=Y,PREFIX=NEW-
```

Note:

To generate an external object using the generation defaults, code the file ID in the same line as the command. No other parameters can be specified.

For example:

```
GENERATE COBOL TEST-FILE
```

Entering Command Parameters in Keyword or Positional Form

Most commands can be entered in keyword form, positional form or a mixture of both. Positional form is obtained by omitting the keywords and the Natural assign character.

Example: Keyword Form

```
COMMAND KEY1=value1,KEY2=value2,KEY3=value3
```

or, if parameter 2 is omitted:

```
COMMAND KEY1=value1,KEY3=value3,KEY4=value4
```

Example: Positional Form

```
COMMAND value1,value2,value3
```

or, if parameter 2 is omitted:

```
COMMAND value1,,value3,value4
```

Mixing Keyword and Positional Form

The keyword and positional form of specifying command parameters can be mixed.

```
COMMAND KEY5=value5, value6,, value8
```

Condition Codes

Some functions (for example generation, migration, incorporation, active references and preprocessor functions) may return condition codes. A non-zero condition code can terminate processing without executing any subsequent commands.

Code	Meaning
0	Function completed normally.
4	Natural detected an error condition. This condition code does not originate in Predict.
8	Natural could not be started. This condition code does not originate in Predict.
101	Generation warnings. An error condition was detected during the generation of record layouts and/or the format buffer which gave rise to a warning message.
103	Invalid statement found. An invalid Predict preprocessor statement was found. The preprocessor continues.
106	Functions in the implementation plan were executed with errors or terminated abnormally, or the plan contains at least one generation task that already failed or could not be executed (status impossible).
107	External object has not been generated or no code has been saved/generated (the requested code has not been included in the output file). For example, the generation function fails or the Copy Code must be regenerated before it can be included by the preprocessor.
109	Storage overflow in extended buffer. Increase the Natural parameter ESIZE.
110	The data dictionary administrator has specified that all programs must be documented in Predict. The preprocessor detected that a Predict program object documenting the program does not exist. The program is not processed.
111	The preprocessor could not be started. The specified input file could not be found.
112	Program name not found by preprocessor.
115	An error occurred. Detailed message is written to report 0.
117	Invalid command, or function not allowed in batch mode (FILOB, PUNCH).

Evaluating Predict Condition Codes under BS2000/OSD

Predict condition codes can be evaluated under BS2000/OSD.

The following is required:

- The BS2000/OSD facility "Job Variables" must be installed.
- The SET parameter "&JV" in the Natural BS2000/OSD batch driver NATBS2 must be set.
- The job variable used for the condition code must be declared in the Predict batch job with the link name *NATB2JV.

The necessary DECLARE statement looks as follows:

```
/DCLJV NATBJV , LINK= *NATB2JV
```

Note:

See the **Natural Operations documentation for Mainframes** for more information.

Natural Workfiles and Additional Reports in Batch Mode

For some batch commands, workfiles or additional reports must be defined.

This section contains:

- Workfile Required for Generation Commands
- Workfiles/Reports Required for Preprocessor Commands
- Workfiles/Reports Required for Incorporate COBOL Command
- Workfiles/Reports Required for Coordinator Commands

Workfile Required for Generation Commands

The following commands write output to Natural workfile 1:

- GENERATE (if Punch/output is set to Y), also if generation commands are called from an implementation plan.
- PUNCH / WRITE

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Generated code	CMWKF01	W01	F or FB	80
			V or VB	min. 80

Note:

When generating ADAFDU definitions with function Generate ADACMP/ADAWAN definitions in batch mode with parameter Punch / Output set to Y for one of the following ADABAS versions:

- any version starting with U,
- V3 and above or
- P2 and above,

we recommend setting the record format to V or VB with a record length of 250. Otherwise data may be truncated.

Workfiles/Reports Required for Preprocessor Commands

The PREPROCESS command requires the first three Natural workfiles:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset for Preprocessor	CMWKF01	W01	F or FB	80
Output dataset	CMWKF02	W02	F or FB	80
Temporary workfile	CMWKF03	W03	F or FB	91

The preprocessor produces three types of output on the first three Natural reports:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Report whether run was successful or not	CMPRT01	P01	F or FB	91
A list of preprocessor commands found	CMPRT02	P02	F or FB	133
A list of errors	CMPRT03	P03	F or FB	133

Workfiles/Reports Required for Incorporate COBOL Command

The INCORPORATE COBOL command requires the first two Natural workfiles:

Note:

If INPUT-TYPE=I, the copy code is read from the Natural command input stream and written to workfile 1.

If input type=W (default), the workfile must contain the copy code to be incorporated.

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset	CMWKF01	W01	F, FB, V or VB	80
Temporary dataset	CMWKF02	W02	V or VB	at least 44

The Incorporate COBOL command produces three types of output on the first three Natural reports:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset (COBOL Copy Code)	CMPRT01	P01	F or FB	91
COBOL Copy Code with error messages	CMPRT02	P02		
Predict file description	CMPRT03	P03		

Workfiles/Reports Required for Coordinator Commands

The Coordinator uses the Natural workfiles 1, 5, 6, and 7. See the **Predict Coordinator documentation**.

Transfer Medium	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Migrate format	CMWKF01, CMWKF05, CMWKF06 or CMWKF07	W01, W05, W06 or W07	V or VB	Rec length at least 1796, block size at least 1800
ALF format	CMWKF01, CMWKF05, CMWKF06 or CMWKF07	W01, W05, W06 or W07	V or VB	Rec length at least 110, block size at least 114

Examples for Batch Mode Input

The examples throughout the documentation assume that the input delimiter character is set to a comma with the Natural parameter

ID=,

and the input assign character is set to equals with the Natural parameter

IA==

This section contains:

- Retrieval
- Generation
- Preprocessor
- Punch Output
- File Implementation
- Redocument Natural Program
- COBOL Copy Code Incorporation
- Conversion
- Coordinator
- LIST XREF, LIST XREF for 3GL

Retrieval

- List all child and parent object of all types of files starting with 'TEST' and having keyword NEW. 60 lines per page are printed, the report will start with a cover page, and for each object up to 10 abstract lines are displayed. The options will stay in effect for all subsequent retrieval functions.

```
LOGON SYSDIC
MENU
SET RESTRICTION KEY=NEW
SET OUTPUT PS=60,COVER=Y,ABSTRACT-CUR=10
CHILD FILE ALL TEST*
PARENT FILE ALL TEST*
FIN
```

Note:

COMMENT-CUR and COMMENT-REL can be used as synonyms for ABSTRACT-CUR and ABSTRACT-REL respectively, so that batch jobs created for earlier versions are still valid.

- Perform retrieval model EX (standard Predict explode) with output mode Cross reference on all Files starting with 'TEST'

```
LOGON SYSDIC
MENU
EXECMODEL FILE TEST* MODEL=EX,MODE=X
FIN
```

Note:

In batch mode, you must specify output mode X (Cross reference) or T (Structured list).

Generation

Generate ADACMP cards for the file TEST-FILE in database TEST-DA, save them in member TEST in library WANLIB and punch them to workfile 1.

```
LOGON SYSDIC
MENU
GENERATE ,ADACMP
FILE-ID=TEST-FILE,DB=TEST-DA% <-- Continuation record follows
PUNCH=Y ,MEM=TEST ,LIB=WANLIB
```

Generate COBOL copy code for file TEST-FILE using default options.

```
GENERATE , COBOL , TEST-FILE
FIN
```

Preprocessor

Process the COBOL source program on workfile 1 with the preprocessor and write XRef data under member name COBTEST in library RBN.

```
LOGON SYSDIC
MENU
PREPROCESS , COBOL , COBTEST , RBN
FIN
```

Punch Output

Punch the previously generated copy code member TEST from library WANLIB to workfile 1.

```
LOGON SYSDIC
MENU
PUNCH TEST , WANLIB
FIN
```

File Implementation

Execute and then display the implementation plan IMP-TEST.

```
LOGON SYSDIC
MENU
IMPLEMENT FILE
FUNCTION=X , PLAN-ID=IMP-TEST
IMPLEMENT FILE
D , IMP-TEST
FIN
```

Redocument Natural Program

Create Predict program objects for all members of the Natural library NATTEST and replace existing program objects.

Construct the program IDs from the prefix NAT-, the library name and the member name.

Put all comment lines from the source into the extended description.

Assign to all redocumented program objects the owner NEWREDOC and the keyword NEW.

Link all redocumented programs to the system NATTEST-LIBRARY. Use the library structure TREE to determine the programs used.

```
LOGON SYSDIC
MENU
REDOCUMENT PROGRAM
PROCESS-OPT=R, LIBRARY=NATTEST, STRUCTURE=TREE, DESCRIPTION=A%
DEFAULT-OWNER=NEWREDOC, DEFAULT-KEY1=NEW%
SYSTEM=NATTEST-LIBRARY, PREFIX=NAT-
FIN
```

COBOL Copy Code Incorporation

Create a Predict sequential file object from the data definitions in the COBOL copy code member on workfile 1 with prefix PRD-, decimal character set to comma, literal delimiter set to single quote. An existing Predict file with the same name is replaced.

```
LOGON SYSDIC
MENU
INCORPORATE, COBOL
PREFIX=PRD-, DECIMAL-CHAR=C, DELIMITER=' , STORE=Y, %
REPLACE=Y, INPUT-TYPE=W
FIN
```

Create a Predict sequential file object from the data definitions in the COBOL Copy Code member with prefix PRD-, decimal character set to comma, literal delimiter set to single quote. An existing Predict file with the same name is replaced.

```
LOGON SYSDIC
MENU
INCORPORATE, COBOL
PREFIX=PRD-, DECIMAL-CHAR=C, DELIMITER=' , STORE=Y, %
REPLACE=Y, INPUT-TYPE=I
000100 01 FI-S-INCCOB-1
000110      02 PE-1 OCCURS 30 TIMES.
000120      03 PE-EL-1 PIC XX.
000130      03 PE-EL-2 PIC X(6).
000140 01 FI-S-INCCOB-2
000150      02 PERSON.
000160      03 CHR-NAME PIC X(32).
000170      03 LST-NAME PIC X(32).
000170      02 DUTY PIC X(25) OCCURS 5.
END
FIN
```

Conversion

Convert the data on the Predict system file from version 4.1 to version 4.2 format.

Convert the Predict objects of type report starting with "REP" and with keyword OLD to objects of type program.

Convert the Predict objects of type module with the prefix MOD* and with keyword OLD to objects of type program.

```
LOGON SYSDICCO
MENU
CONVERT VERSION42
CONVERT REPORT
REP* ,OLD , * ,N
CONVERT MODULE
MOD* , * ,OLD , * ,N
FIN
```

Coordinator

See the Predict Coordinator documentation for more information.

LIST XREF, LIST XREF for 3GL

```
LOGON SYSDIC
MENU
ACTIVE XREF *SYSPLI* PROGRAM USING VIEW
```