

# ***NAT-PAD***

***NATURAL Change Management  
NATURAL Program Administration and Documentation  
NATURAL Application Control***

***Installation Guide and Administration Manual***

MVS and OS/390

---

**NAT-PAD Version 3.1.e**

**May 18, 1999**

© **Storr Consulting 1996 / 1997 / 1998 / 1999**

**Distribution, Change and Enhancements in Europe:**

**Storr Consulting, Fasanenstrasse 4, D-55270 Zornheim, Germany**

Voice: +49-6122-940611 - Fax: +49-6122-940612

**Distribution, Change and Enhancements in U.S.A.:**

**D.P. Solutions, Inc., 12444 Matteson Ave., Los Angeles, CA 90066**

Voice: 310-306-7917 and 310-390-6096 - Fax: 310-306-7917

Internet: <http://www.dpsi-ca.com>

E-mail: [dwstorr@aol.com](mailto:dwstorr@aol.com)

## Contents

<b>1</b>	<b>Functions.....</b>	<b>5</b>
<b>2</b>	<b>Contents of the Diskette.....</b>	<b>6</b>
2.1	Creation of file 1 .....	6
2.2	Creation of files 2 and 3.....	6
<b>3</b>	<b>Installation.....</b>	<b>7</b>
3.1	Installation with Rumba.....	7
3.2	Installation without Rumba.....	8
3.2.1	Load NPPDS.TXT (FILE 1).....	8
3.2.2	Load NPOBJ.BIN (file 2) and NPSRC.TXT (file 3).....	9
3.3	Adapt NATURAL Source and JCL .....	10
3.3.1	Parameter Source NPN10000 .....	10
3.3.2	Parameter Source NPN10001 .....	11
3.3.3	Parameter Source NPN10002 .....	11
3.3.4	Parameter Source NPN10003 .....	12
3.3.5	Parameter Source NPN10004 .....	12
3.3.6	Parameter Source NPN10005 .....	13
3.3.7	Parameter Source NPN10006 .....	13
3.3.8	Job control statements (JCL) NPN00210.....	14
3.3.9	Job control statements (JCL) NPN00220.....	14
3.3.10	Job control statements (JCL) NPP00620.....	15
3.3.11	Submit programs (ESS) NPNSUBNP .....	15
3.3.12	Define job card NPNSUBRJ .....	16
3.3.13	Define job card NPNSUBCJ .....	16
3.3.14	JCL text objects .....	17
3.3.15	JCL for ESS (NATURAL PROCESS) .....	29
3.3.16	Administration programs.....	29
<b>4</b>	<b>Create ADABAS Files and Copy DDMs.....</b>	<b>30</b>
4.1	ADABAS Files.....	30
4.1.1	Request File .....	30
4.1.2	Reference File .....	31
4.1.3	Archive File.....	32
4.1.4	FUSER File .....	32
4.1.5	Transition Library.....	34
4.2	Description of Data Definition Modules (DDM).....	34
4.2.1	Requests - NP-AUFT-E .....	34
4.2.2	Requests - NP-AUFT-M.....	35
4.2.3	Reference - NP-REF .....	35
4.2.4	Archive - NP-ARC.....	36
4.2.5	FUSER and FNAT .....	36

<b>5 NATURAL SECURITY .....</b>	<b>38</b>
5.1 Files .....	38
5.2 Automatic Logon .....	38
<b>6 Miscellaneous .....</b>	<b>39</b>
6.1 Delimiter .....	39
6.2 NATLOAD Condition Code 37 .....	39

## 1 Functions

Controlling the migration of NATURAL objects between environments presents numerous challenges: Tracking the current location of a program, avoiding program overwrites, verifying that program changes are made, changing all related objects, archiving and recovery, maintaining complete audit trails. Manual change management can be time-consuming which tends to lead to errors.

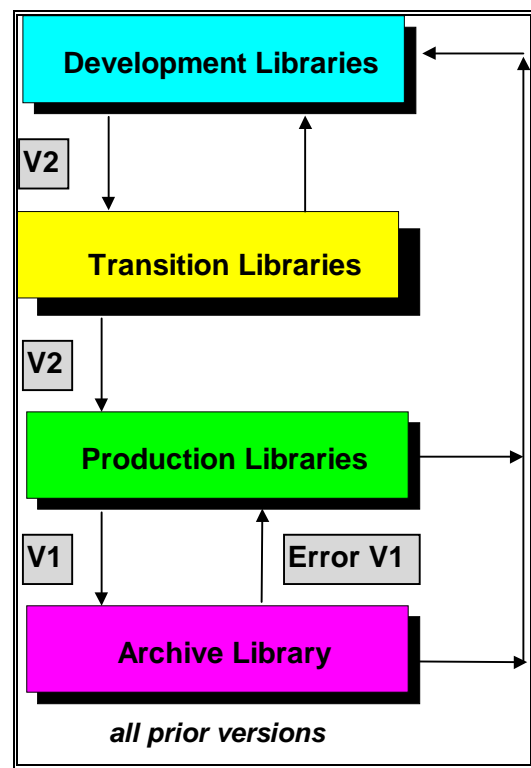
NAT-PAD was developed to provide cost-effective, automated change management for NATURAL objects, SYSERR messages, and PREDICT objects.

### NAT-PAD provides administration and documentation by

- Maintaining an audit trail
- Providing data security
- Controlling access

### NAT-PAD functionality

- Copies NATURAL source
  - from development into transition library
  - from transition into production library
  - catalogs the sources in production
- Archives the former version before moving new objects into production
- Browses in archive and copies source into development
- Browses in production and copies source into development
- Supports several search functions
- Supports several statistic reports
- Copies error messages from test into production and vice versa
- Copies PREDICT objects from test into production and vice versa
- Performs validation checks of all functions against NATURAL Security or equivalent facilities
- Connects to MVS job entry systems (JESx)
- Uses Entire System Server (if available)
- Uses NATURAL user exits eliminating problems with new release changes
- NAT-PAD is Year 2000 compliant



### Operating system and prerequisites

- MVS and OS/390
- ADABAS 5.2 / 5.3 / 6.2 and NATURAL 2.2 / 2.3
- NATURAL SECURITY and PREDICT optional

## 2 Contents of the Diskette

The following files (data sets) are on the diskette:

Disk #	Name	Description
1	NPPDS.TXT	Job control statements, DDMs, FDTs
2	NPOBJ.TXT	NAP-PAD NATURAL Objects
3	NPSRC.TXT	NAP-PAD NATURAL Source for changing the parameter values etc.

Figure 1: Description of the contents of the diskette

### 2.1 Creation of file 1

The members of the PDS data set are unloaded with the command:

#### IN\$FILE Format 80/800 CRF EBCDIC (PUT)

Control statements are provided for later load.

### 2.2 Creation of files 2 and 3

The 'cataloged objects' and the 'stowed objects' are unloaded with NATUNLD (see Figure 2). They are then loaded with IND\$FILE as file 2 and 3 onto the diskette.

```

15:57:33          ***** NATURAL OBJECT MAINTENANCE *****          97-06-09
User dstorr      - Unload Programming Objects -          Library NATPAD

                Code  Sub-function
                ----  -
                A   Unload All/Individual Objects
                C   Unload only Cataloged Object
                S   Unload only Saved Object
                W   Unload Stowed Object
                ?   Help
                .   Exit
                ----  -

Code ..... _
From Library ... FO_____ Set Number ..... _
To Library ..... _____ Xref Data ..... Y
Object Name .... _____ Object Type .... *_____
User ID ..... _____ Date/Time Fm ... _____
Date/Time To ... _____ Date/Time To ... _____
PC Download .... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
      Help      Exit      Views      File      Canc

```

Figure 2: NATUNLD

## 3 Installation

### 3.1 Installation with Rumba

With the RUMBA file transfer feature, files can be copied from a PC to a host or from a host to a PC. RUMBA software works transparently with standard host operating systems. This allows an application on one host system to access and use data from an application on another host system.

RUMBA software simplifies the file transfer setup process by using typical Windows menus and dialog boxes. It also reads and lists PC and host files, and permits file transfers in the background, allowing other host or PC tasks to be performed during file transfer. File transfer actions can be performed using the configurable tool bar.

#### To transfer files between your PC and the host

- Logon TSO session with RUMBA
- Allocate the sequential data sets xxxxxx.NPSRCE1 through xxxxxx.NPSRCE2 (options see Figure 3).

```
----- DATA SET INFORMATION -----
COMMAND ===>

DATA SET NAME: DIETER.NPSRC1

GENERAL DATA:                                CURRENT ALLOCATION:
Management class:      MCTSOPDS                Allocated cylinders:    6
Storage class          SCTSODA                 Allocated extents      2
Volume:                TSO001
Device type:           3390
Data class:            DCTSODA                CURRENT UTILIZATION:
Organization:          PS                     Used cylinders:         2
Record format:         FB                     Used extents:           2
Record length:         256
Block size:            27904
1st extent cylinders:  1
Secondary cylinders:   5
Data set name type:

Creation date:         1997/06/03
Expiration date:      ***NONE***
```

Figure 3: Data set Allocation

- End the ISPF session by typing in 'X'
- Select the 'Transfer' button from the top of the screen
- Select 'Send' from the pull-down menu
- Select the PC directory, fill in the file name (**from**), select the data set name, insert the host file name (**to**), and mark the CR/LF-box. This is important to convert the carriage return and the line feed characters from the PC to the equivalent characters used by the host environment (see Figure 4).

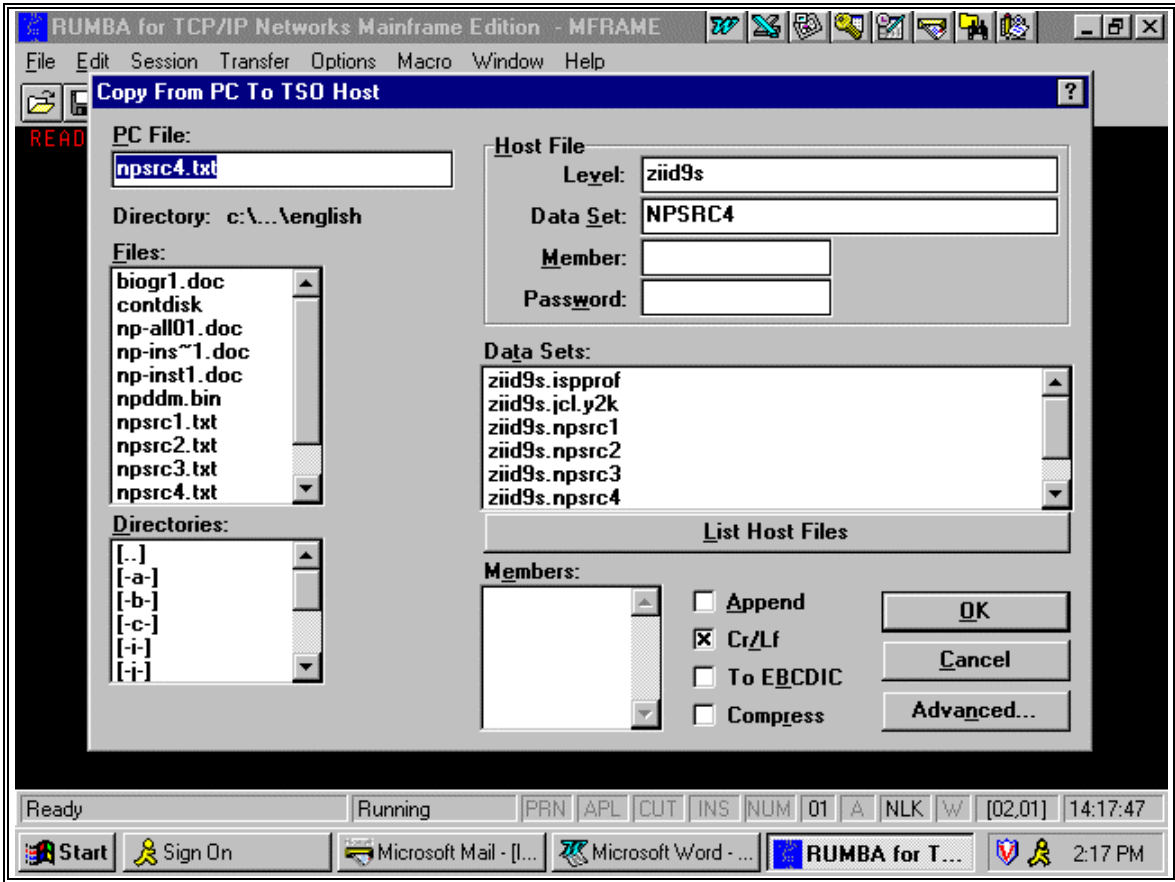


Figure 4: File transfer between PC and Host using Rumba

## 3.2 Installation without Rumba

### 3.2.1 Load NPPDS.TXT (FILE 1)

File 1 consists of several members that have to be copied from a PC or a 'single diskette unit' to a mainframe data set. Use the command:

```
IN$FILE Format 80 CRF EBCDIC
```

Load (copy) this data set with IBM utility IEBUPDTE into a PDS data set.

```
//STEP1      EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  DSN=input-dataset,DISP=SHR
//SYSUT2   DD  DSN=output-dataset,DISP=(NEW,CATLG,DELETE),
//           DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),
//           SPACE=(CYL,(1,1,5)),UNIT=SYSDA
```

Figure 5: IEBUPDT - load the PDS member



### 3.2.2 Load NPOBJ.BIN (file 2) and NPSRC.TXT (file 3)

Copy files 2 and 3 from a single diskette or a PC to a mainframe data set, by using the command:

IN\$FILE Format 256

Load these files with the NATURAL utility NATLOAD on-line or batch, into the NATURAL environment (see Figure 6). It is important to first create the target library NATPAD to load the objects. Do not forget the NATURAL SECURITY definitions.

Copy the delivered programs NPPBUFFE and NPPEROR into the production environment. NPPBUFFE is responsible to purge the buffer pool after copy the objects. NPPEROR is an error routine (see the jobs NPJ00700 and NPJ00780).

```
16:37:47          ***** NATURAL OBJECT MAINTENANCE *****          97-06-09
User dstorr          - NATURAL Load Utility -          Library NATPAD

          Code   Function
          ----   -----
          L     Load Objects in a Library
          V     Load Views
          S     Scan Load File
          ?     Help
          .     Exit
          ----   -----

Code ..... VL_          Load Except ... _
View Name .....
Library .....           S/C Type ..... A   XREF ... S
Object Name ...         Object Type ... *
User ID .....           Date/Time Fm ..
Replace ..... N         Date/Time To ..
                          PC Upload ..... N

Command ==> Enter-PF1--PF2--PF3-PF4--PF5--PF6--PF7--PF8--PF9-PF10-PF11-PF12--
              Help Menu Exit                                Canc
```

Figure 6: NATLOAD - online

```
000006 //NATBATCH EXEC NATBATCH,NATPARAM=' ,STACK=INPL,XREF=OFF '
000007 //CMPRINT DD SYSOUT=X
000008 //CMPRT01 DD SYSOUT=X
000009 //CMWKF01 DD DSN=DIETER.NPSRC2,DISP=SHR
000010 //          DD DSN=DIETER.NPSRC3,DISP=SHR
000011 //CMSYNIN DD *
000012 L
000013 FIN
```

Figure 7: NATLOAD - batch - INPL

## 3.3 Adapt NATURAL Source and JCL

Some NATURAL programs are delivered in source code (tables, parameter values, etc.) and have to be adapted and STOWed. Additionally, some job control statements (JCL) have to be changed for the target environment.

Examples are available in the relevant NATURAL objects.

### 3.3.1 Parameter Source NPN10000

NPN10000 contains descriptions of transition and production environment. Up to 10 entries for libraries are possible (see Figure 8).

Figure 9 shows the selection window from NAT-PAD map 'Register Request' to determine 'from library', 'development library', and 'production library'. These values become part of the new request.

```
0010 *
0020 *   NPN10000 Transition and target libraries
0030 *
0040 DEFINE DATA
0050 PARAMETER                USING NPA10000
0060 * #NPA10000              (10)
0070 * #TLIB                  (A8)
0080 * #TDBID                  (N3)
0090 * #TFNR                   (N3)
0100 * #PLIB                   (A8)
0110 * #PDBID                  (N3)
0120 * #PFNR                   (N3)
0130 END-DEFINE
0140 *
0150 RESET #NPA10000 (*)
0160 *
0170 #TLIB (01) := 'NP-TRANS'   /* Transition library 1
0180 #TDBID (01) := 196
0190 #TFNR (01) := 008
0200 #PLIB (01) := 'NP-PROD'   /* Production library
0210 #PDBID (01) := 196
0220 #PFNR (01) := 008
...
...
0370 *
0380 #TLIB (04) := 'ICSPST'    /* Transition: integration test
0390 #TDBID (04) := 196
0400 #TFNR (04) := 008
0410 #PLIB (04) := 'NP-PROD'   /* Production library
0420 #PDBID (04) := 196
0430 #PFNR (04) := 008
....
```

Figure 8: Subprogram NPN10000

```

19:44:51          *** N A T - P A D ***          97-05-22
STORR              - Register Request -          NP

Continue with Request:

>
S Request          NP_____ from Library
n 1                ___ Request Number
- 2
- 3
- 4                Developmt. Production
- 5                MIG          PROD
- 6                SYSNOMU     SYSNOMU
- 7                MIGRPC     PRODRPC
- 8                SYSNDM     SYSNDM
- 9                PRODCNT     PRODCNT
- 10

Request -----> <-- Job ----
Time   Status   Number Status
7 11:53 inprod    9496 OK
7 13:59 inproces 1269 OK
7 14:00 inproces 1622 OK
7 18:24 inprod    6834 OK
7 18:52 inprod    8908 OK
7 11:16 inprod    1944 OK
7 18:22 inprod    1605 OK
7 17:26 inprod    6337 OK
7 18:03 inprod    6471 OK
7 11:40 inprod     730 OK

Enter-PF1-- PF3-Endc PF8---PF9---PF10--PF11--PF12---
Arch Prod Devel
  
```

Figure 9: Pop-up window shows NATURAL table values to select

### 3.3.2 Parameter Source NPN10001

NPN10001 contains information for SYSMAIN for development, transition, production, and archive (see Figure 10).

```

0170 RESET #NPA10001
0180 *
0190 * Transition
0200 #TDBID := 196
0210 #TFNR := 008
0220 * Archive
0230 #ADBID := 196
0240 #AFNR := 122
0250 * Production
0260 #PDBID := 196
0270 #PFNR := 008
0280 * Development
0290 #EDBID := 196
0300 #EFNR := 008
0310 *
  
```

Figure 10: Subprogram NPN10001

### 3.3.3 Parameter Source NPN10002

NPN10002 contains default values and permissions.

```
0040 DEFINE DATA
0050 PARAMETER          USING NPA10002
0060 * 1 #NPA10002
0070 * 2 #PASSWORD      (A6)  password
0080 * 2 #BANNER        (L)   avoid first screen
0090 * 2 #SYSSEC        (L)   checks with NATURAL SECURITY
0100 * 2 #CHKOBJEKT     (L)   check-out byte one and two for object
0110 * 2 #COPYARCOBJ    (L)   permission to copy object from archive
0120 * 2 #UEBERGABE     (A1)  permission to copy/move/both in trans.lib
0130 * 2 #ZIEL          (A1)  permission to copy/move/both in target lib
0140 * 2 #SRCESUP       (L)   searching for source in FUSER (check-in)
0150 END-DEFINE
```

Figure 11: Subprogram NPN10002

### 3.3.4 Parameter Source NPN10003

NPN10003 contains descriptions of FUSER and FNAT for development (see Figure 12).

```
0160 *
0170 #FUSERFNR := 006
0180 #FUSERDBID := 196
0190 #FNATFNR := 008
0200 #FNATDBID := 196
0210 *
```

Figure 12: Subprogram NPN10003

### 3.3.5 Parameter Source NPN10004

NPN10004 contains parameters for batch jobs, in particular, the name of the library of the text objects for RJE (see 3.3.12).

```
0010 *
0020 * NPN10004 values to submit a job to production
0030 *
0040 DEFINE DATA
0050 PARAMETER USING NPA10004
0060 * 1 #NPA10004
0070 * 2 #BATCHVIA      (A3)   NAT Process / NAT RJE / COM-LETE RJE
0080 * 2 #BATCHSRC      (A44)  dsn / NATPAD
0090 * 2 #CATINPROD     (L)    CATALL in production
0100 * 2 #CATALL-NO-STOW (L)   CATALL w/o STOW (userid/date from SAVE)
0110 * 2 #XREFUNLOAD    (A1)   with XREF
0120 * 2 #LOGON         (A20)  how to logon
0130 * 2 #JOBS         (1:30) transition with batch jobs
0140 * 3 #JOBMEMBER     (A08)   - NATURAL source
0150 * 3 #JOBTEXT       (A40)   - library name
0160 * 4 #JOBTEXT-LIB   (A08)
0170 * 4 #JOBTEXT-FIL   (A32)
0180 * 2 #BACKADMIN     (L)    backout only for administrator
0190 * 2 #XREFARC       (A1)   XREF copy from and to archive
0200 END-DEFINE
0210 *
```

Figure 13: Subprogram NPN10004

### 3.3.6 Parameter Source NPN10005

NPN10005 contains parameters for NAT-PAD administrators. Up to 10 entries for userids are possible (see Figure 14).

```
0010 *
0020 * NPN10005 Who is NAT-PAD administrator?
0030 *
0040 DEFINE DATA
0050 PARAMETER USING NPA10005
0060 * 1 #NPA10005
0070 * 2 #ADMIN-USER (A8/10)
0080 END-DEFINE
.....
0140 RESET #ADMIN-USER (*)
0150 *
0160 * UserIDs of adminsitrators - max 10
0170 *
0180 #ADMIN-USER (01) := 'ZIxxx1'
0190 #ADMIN-USER (02) := 'ZIxxx2'
0200 #ADMIN-USER (03) := 'ZIxxx3'
.....
```

Figure 14: Subprogram NPN10005

### 3.3.7 Parameter Source NPN10006

NPN10006 contains parameters for NAT-PAD users and alternates.

```
0030 * NPN10006: Who are NAT-PAD users and alternates?
0040 *          TR / PR = test read    and production read    = TR
0050 *          TU / PR = test update and production read    = TU
0060 *          TU / PU = test update and production uptate = PU
<snip>
1680 * Programmer
1690 * -
1700 #N6-USERID (17,01) := 'ZEEJXE'      /* Jim Example
1710 #N6-USERID (17,02) := ' '          /* no alternate
1720 #N6-ACCESS (17)   := 'TU'          /* test update / production read
1730 * -
<snip>
```

Figure 15: Subprogram NPN10006

### 3.3.8 Job control statements (JCL) NPN00210

NPN00210 creates data set names for text module NPJ00700 for steps GNATD, GNATP, GPRDD, and GPRDP.

NPJ00700 contains jcl for copy/move NATURAL objects from transition into production.

```
0010 *
0020 * NPN00210 JCL to submit programs to production
0030 *
0040 DEFINE DATA
0050 PARAMETER
0060 1 #MSG                (A70)
0070 1 #USER              (A8)
0080 1 #EVENT             (N2)
0090 1 #TLIB             (A8)
0100 1 #JOB              (A8)
0110 1 #UEBERKZ         (A1)
....
0270 *
0280 COMPRESS *USER '.TEST' '.R' #EVENT '.NATUNLD' INTO #NATUNLD LEAVING NO
0290 COMPRESS *USER '.TEST' '.R' #EVENT '.PRDUNLD' INTO #PRDUNLD LEAVING NO
0300 COMPRESS *USER '.TEST' '.R' #EVENT '.NATUNLP' INTO #NATUNLP LEAVING NO
0310 COMPRESS *USER '.TEST' '.R' #EVENT '.PRDUNLP' INTO #PRDUNLP LEAVING NO
0320 *
0330 SUBMIT-PARAMETER.MEMBER := #JOB
```

Figure 16: Subprogram NPN00210

### 3.3.9 Job control statements (JCL) NPN00220

NPN00220 creates data set names for text module NPJ00780 for steps GNATD, GNATP, GPRDD, and GPRDP.

NPJ00780 contains jcl to backout a transition.

```
0010 *
0020 * NPN00220 Submit Backout of transition
0030 *
0040 DEFINE DATA
0050 PARAMETER
0060 1 #MSG                (A70)
0070 1 #USER              (A8)
0080 1 #EVENT             (N2)
0090 1 #CM               (A1)
0100 *
....
0250 *
0260 COMPRESS *USER '.TEST' '.E' #EVENT '.NATUNLP' INTO #NATUNLP LEAVING NO
0270 COMPRESS *USER '.TEST' '.E' #EVENT '.NATUNLD' INTO #NATUNLD LEAVING NO
0280 *
0290 SUBMIT-PARAMETER.MEMBER := 'NPJ00780'
....
```

Figure 17: Subprogram NPN00220

### 3.3.10 Job control statements (JCL) NPP00620

NPN00620 contains job control statements to submit PREDICT.

```
0010 *
0020 * NPP00620 Copy Predict objects
0030 *
0040 DEFINE DATA
.....
0540 RELEASE STACK
0550 IF #VON = 'D' /* D = Development '//NATBA EXEC NATBATD'
0560 STACK DATA FORMATTED '//NATBA EXEC NATBATD,NATPARM='',AUTO=ON''
0570 ELSE /* P = Production '//NATBA EXEC NATBATP'
0580 STACK DATA FORMATTED '//NATBA EXEC NATBATP,NATPARM='',AUTO=ON''
0590 END-IF
.....
1010 IF #NACH = 'P'
1020 STACK DATA FORMATTED
1030 '//NATBB EXEC NATBATB,NATPARM='',AUTO=ON'',COND=(0,NE)'
1040 ELSE
1050 STACK DATA FORMATTED
1060 '//NATBB EXEC NATBATB,NATPARM='',AUTO=ON'',COND=(0,NE)'
1070 END-IF
.....
```

Figure 18: Subprogram NPP00620

Subprogram NPP00620 enhances text object NPJ00620.

### 3.3.11 Submit programs (ESS) NPNSUBNP

NPNSUBNP contains information to adapt and catalog for relevant NATURAL PROCESS (ENTIRE SYSTEM SERVER, ESS) DDMs. If ESS is not available use NATRJE (see 3.3.13). Describe parameter #BATCHVIA in source NPN10004.

```
0010 * Send job to JES via INTERNAL READER - only for NATURAL PROCESS
0020 *
0030 *
0040 DEFINE DATA
0050 PARAMETER USING NPASUBMT
0060 *
0070 LOCAL
0080 1 NATPROC-LOGON VIEW OF NATPROC-LOGON
0090 2 ERROR-CODE
0100 2 ERROR-TEXT
0110 2 NODE
0120 2 LOGON-ID
.....
* change various statements
.....
```

Figure 19: Subprogram NPNSUBNP

### 3.3.12 Define job card NPNSUBRJ

NPNSUBRJ creates the job card and submits the job via NATRJE. Be sure that NATRJE is linked in NATURAL nucleus and the parameter RJESIZE is set to a value greater than 0. For more information please see NATURAL Utilities Manual. Describe parameter #BATCHVIA in source NPN10004.

```
.....
1310 IF RECJOBML1 = ' ' AND RECJOBML2 = ' '
1320 COMPRESS '// ' *USER 'N JOB (TECH,DBA),''
1330 #PNAME ''',MSGCLASS=X,CLASS=2,TYPRUN=SCAN'
1340 INTO #RECORD
1350 LEAVING NO SPACE
1360 ELSE
1370 COMPRESS '// ' *USER 'N JOB (TECH,DBA),''
1380 #PNAME ''',MSGCLASS=X,CLASS=2,TYPRUN=SCAN'
1390 INTO #RECORD
1400 LEAVING NO SPACE
1410 END-IF
.....
1450 COMPRESS '// REGION=4M,' INTO #RECORD LEAVING NO SPACE
.....
1960 CALL 'NATRJE' #JCL-CARD #RJE-COUNT #RJE-FLAG #RJE-RC
.....
```

Figure 20: NPNSUBRJ creates the job card and submits via NATRJE

### 3.3.13 Define job card NPNSUBCJ

NPNSUBCJ creates the job card and submits the job via COM-LETE's RJE. For more information please see 'COM-LETE Application Programmer's Manual', page 5-19. Describe parameter #BATCHVIA in source NPN10004.

```
.....
1310 IF RECJOBML1 = ' ' AND RECJOBML2 = ' '
1320 COMPRESS '// ' *USER 'N JOB (TECH,DBA),''
1330 #PNAME ''',MSGCLASS=X,CLASS=2,TYPRUN=SCAN'
1340 INTO #RECORD
1350 LEAVING NO SPACE
1360 ELSE
1370 COMPRESS '// ' *USER 'N JOB (TECH,DBA),''
1380 #PNAME ''',MSGCLASS=X,CLASS=2,TYPRUN=SCAN'
1390 INTO #RECORD
1400 LEAVING NO SPACE
1410 END-IF
.....
1450 COMPRESS '// REGION=4M,' INTO #RECORD LEAVING NO SPACE
.....
2100 CALL 'RJE' #RJE-RC #JCL-CARD #RJE-LEN #RJE-HOLD
.....
```

Figure 21: NPNSUBCJ creates the job card and submits via RJE



### 3.3.14 JCL text objects

Modify these programs only if NATRJE is used to submit a batch job. Otherwise, use JCL for ESS (see 3.3.11).

- NPJ00100
- NPJ00620
- NPJ00700
- NPJ00780
- NPJ00950

#### NPJ00100 Restart PREDICT

```
>
All      > + Text      NPJ00100      Lib NATPAD
.....1.....2.....3.....4.....5.....Mode Struc
0010 &JOB
0020 /* Restart PREDICT
0030 //RESTART EXEC NATBATP
0040 //CMSYNIN DD *
0050 LOGON SYSDICBE
0060 RESTART
0070 FIN
0080 /*
```

Figure 22: NPJ00100 NATURAL text module 'Restart PREDICT'

**NPJ00620 Copy PREDICT objects**

```

>
All .....1.....2.....3.....4.....5.....Mode Struc
L 0010 &JOB
  0020 /* copy / transmit PREDICT objects
  0030 /* * step 1 unload
L 0040 &CARD
L 0050 //CMWKF01 DD DSN=&&WKF01,UNIT=SYSDA,DISP=(,PASS),
L 0060 //          SPACE=(TRK,(15,15),RLSE),
L 0070 //          DCB=(SYS1.MODEL,RECFM=VB,BLKSIZE=4628,LRECL=4624)
L 0080 //CMWKF03 DD DSN=&&WKF03,UNIT=SYSDA,DISP=(,PASS),
L 0090 //          SPACE=(TRK,(15,15),RLSE),
L 0100 //          DCB=(SYS1.MODEL,RECFM=VB,BLKSIZE=4628,LRECL=4624)
L 0110 //CMSYNIN DD *
L 0120 LOGON SYSDICBE
L 0130 MENU
L 0140 &CARD
L 0150 &CARD
L 0160 &CARD
L 0170 &CARD
L 0180 &CARD
L 0190 &CARD
L 0200 &CARD
L 0210 &CARD
L 0220 &CARD
L 0230 &CARD
L 0240 &CARD
L 0250 &CARD
L 0260 &CARD
L 0270 &CARD
L 0280 /*
  0290 /* step 2 load
L 0300 &CARD
L 0310 //CMWKF01 DD DSN=&&WKF01,DISP=(OLD,DELETE)
L 0320 //CMWKF03 DD DSN=&&WKF03,DISP=(OLD,DELETE)
L 0330 //CMSYNIN DD *
L 0340 LOGON SYSDICBE
L 0350 MENU
L 0360 LOAD ALL;CODE=Y;REPLACE=Y
L 0370 FIN
L 0380 /*

```

Figure 23: NPJ00620 NATURAL text module 'Copy PREDICT objects'

NATURAL program NPP00620 creates NATBATCH procedure depends on development or production (see 3.3.10).

## NPJ00700 Copy to production

```

>
All      .....1.....2.....3.....4.....5.....Mode Struc
L 0010 &JOB
L 0020 /* Create control statements for unload and load takeover (transmit)
L 0030 //EVENT      EXEC NATBAT
L 0040 //SYSOUT     DD SYSOUT=Z
L 0050 //SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0060 //SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0070 //SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0080 //SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0090 //CMPRT01   DD SYSOUT=*
L 0100 //CMWKF01   DD DSN=&&WKFO1,DISP=(NEW,PASS,DELETE),
L 0110 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0120 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0130 //CMWKF02   DD DSN=&&WKFO2,DISP=(NEW,PASS,DELETE),
L 0140 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0150 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0160 //CMWKF03   DD DSN=&&WKFO3,DISP=(NEW,PASS,DELETE),
L 0170 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0180 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0190 //CMWKF04   DD DSN=&&WKFO4,DISP=(NEW,PASS,DELETE),
L 0200 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0210 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0220 //CMWKF05   DD DSN=&&WKFO5,DISP=(NEW,PASS,DELETE),
L 0230 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0240 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0250 //CMWKF06   DD DSN=&&WKFO6,DISP=(NEW,PASS,DELETE),
L 0260 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0270 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0280 //CMWKF07   DD DSN=&&WKFO7,DISP=(NEW,PASS,DELETE),
L 0290 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0300 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0310 //CMWKF08   DD DSN=&&WKFO8,DISP=(NEW,PASS,DELETE),
L 0320 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0330 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0340 //CMSYNIN DD *
L 0350 LOGON NATPAD
L 0360 &CARD
L 0370 FIN
L 0380 /*
L 0390 //DELFIL1   EXEC PGM=IDCAMS,COND=(0,NE)
L 0400 //SYSPRINT DD SYSOUT=Z
L 0410 //SYSIN DD *
L 0420           DELETE -
L 0430 &CARD
L 0440           NOERASE -
L 0450           PURGE
L 0460           IF LASTCC LE 8 THEN DO
L 0470             SET LASTCC = 0
L 0480             SET MAXCC = 0
L 0490           END
L 0500           DELETE -
L 0510 &CARD

```

Figure 24: NPJ00700 NATURAL text module 'Copy to production' - part 1 of 6

```

>
All      .....1.....2.....3.....4.....5.....Mode Struct
L 0520      NOERASE -
L 0530      PURGE
L 0540      IF LASTCC LE 8 THEN DO
L 0550          SET LASTCC = 0
L 0560          SET MAXCC = 0
L 0570      END
L 0580      DELETE -
L 0590 &CARD
L 0600      NOERASE -
L 0610      PURGE
L 0620      IF LASTCC LE 8 THEN DO
L 0630          SET LASTCC = 0
L 0640          SET MAXCC = 0
L 0650      END
L 0660      DELETE -
L 0670 &CARD
L 0680      NOERASE -
L 0690      PURGE
L 0700      IF LASTCC LE 8 THEN DO
L 0710          SET LASTCC = 0
L 0720          SET MAXCC = 0
L 0730      END
L 0740 /*
L 0750 //GNATE EXEC PGM=IEBGENER,COND=(0,NE)
L 0760 //SYSPRINT DD SYSOUT=Z
L 0770 //SYSUT1 DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 0780 //SYSUT2 DD DISP=(,CATLG),
L 0790 &CARD
L 0800 //          SPACE=(CYL,(25,5),RLSE),
L 0810 //          UNIT=SYSDA
L 0820 //SYSIN DD *
L 0830 /*
L 0840 //GPRDE EXEC PGM=IEBGENER,COND=(0,NE)
L 0850 //SYSPRINT DD SYSOUT=Z
L 0860 //SYSUT1 DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 0870 //SYSUT2 DD DISP=(,CATLG),
L 0880 &CARD
L 0890 //          SPACE=(CYL,(25,5),RLSE),
L 0900 //          UNIT=SYSDA
L 0910 //SYSIN DD *
L 0920 /*
L 0930 //GNATP EXEC PGM=IEBGENER,COND=(0,NE)
L 0940 //SYSPRINT DD SYSOUT=Z
L 0950 //SYSUT1 DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 0960 //SYSUT2 DD DISP=(,CATLG),
L 0970 &CARD
L 0980 //          SPACE=(CYL,(25,5),RLSE),
L 0990 //          UNIT=SYSDA
L 1000 //SYSIN DD *
L 1010 /*
L 1020 //GPRDP EXEC PGM=IEBGENER,COND=(0,NE)
L 1030 //SYSPRINT DD SYSOUT=Z
L 1040 //SYSUT1 DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 1050 //SYSUT2 DD DISP=(,CATLG),
L 1060 &CARD

```

Figure 25: NPJ00700 NATURAL text module 'Copy to production' - part 2 of 6

```
>
All      .....1.....2.....3.....4.....5.....Mode Struct
L 1070 //          SPACE=(CYL,(25,5),RLSE),
L 1080 //          UNIT=SYSDA
L 1090 //SYSIN    DD *
L 1100 /*
  1110 /** Unload NATURAL objects from development
L 1120 //NATUNLE  EXEC NATBAT,COND=(0,NE)
L 1130 &CARD
L 1140 //CMSYNIN DD DSN=&&WK01,DISP=(OLD,DELETE,DELETE)
L 1150 /*
  1160 /** Unload PREDICT objects from development
L 1170 //PRDUNLE  EXEC NATBAT,COND=(0,NE)
L 1180 &CARD
L 1190 //CMSYNIN DD DSN=&&WK02,DISP=(OLD,DELETE,DELETE)
L 1200 /*
  1210 /** Unload NATURAL objects from production
L 1220 //NATUNLP  EXEC NATBATP,COND=(0,NE)
L 1230 &CARD
L 1240 //CMSYNIN DD DSN=&&WK03,DISP=(OLD,DELETE,DELETE)
L 1250 /*
  1260 /** Unload PREDICT objects from production
L 1270 //PRDUNLP  EXEC NATBATP,COND=(0,NE)
L 1280 &CARD
L 1290 //CMSYNIN DD DSN=&&WK04,DISP=(OLD,DELETE,DELETE)
L 1300 /*
  1310 /** Archive
L 1320 //ARCHIV   EXEC NATBAT,COND=(0,NE)
L 1330 //CMPRT01 DD SYSOUT=*
L 1340 //CMWKF01 DD DSN=&&WK05,DISP=(OLD,PASS,DELETE)
L 1350 //CMSYNIN DD *
L 1360 LOGON NATPAD
L 1370 NPP00700
L 1380 FIN
L 1390 /*
  1400 /** Load new NATURAL objects into production
L 1410 //NATLOAD  EXEC NATBATP,COND=(0,NE)
L 1420 &CARD
L 1430 //CMSYNIN DD *
L 1440 LOGON NATPAD
L 1450 NATLOAD
L 1460 L;Y
L 1470 FIN
L 1480 /*
  1490 /** Load new PREDICT objects into production
L 1500 //PRDLOAD  EXEC NATBATP,COND=(0,NE)
L 1510 &CARD
L 1520 //CMSYNIN DD *
L 1530 LOGON SYSDICBE
L 1540 LOAD ALL CODE=Y REPLACE=Y
L 1550 FIN
L 1560 /*
```

Figure 26: NPJ00700 NATURAL text module 'Copy to production' - part 3 of 6

```

>
All      ....+....1....+....2....+....3....+....4....+....5....+....Mode Struc
1570 /* Catalog in target environment
1580 /* D O N ' T   remove this step
1590 /* also, when moving objects (#CATINPROD=FALSE)
1600 /*
L 1610 //CAT      EXEC NATBATP,COND=(0,NE),
1620 // PARM.NATBAT='ADASVC=$$$,BPID=$$$'      <----- change numbers
L 1630 //CMPRINT  DD SYSOUT=*
L 1640 //CMSYNIN  DD DSN=&&WKFO6,DISP=(OLD,DELETE,DELETE)
L 1650 /*
1660 /* Set status OK
L 1670 //STATOK   EXEC NATBAT,COND=(0,NE)
L 1680 //CMPRT01 DD SYSOUT=*
L 1690 //CMSYNIN DD *
L 1700 LOGON NATPAD
L 1710 &CARD
L 1720 FIN
L 1730 /*
1740 /* purge buffer
L 1750 //PBUFFER  EXEC NATBATP,COND=(0,NE),
L 1760 // PARM.NATBAT='ADASVC=$$$,BPID=$$$'      <----- change numbers
L 1770 //CMPRINT  DD SYSOUT=*
L 1780 //CMWKF01 DD DSN=&&WKFO7,DISP=(OLD,DELETE,DELETE)
L 1790 //CMPRT01 DD SYSOUT=*
L 1800 //CMSYNIN DD *
L 1810 LOGON NATPAD
L 1820 NPPBUFFE
L 1830 FIN
L 1840 /*
1850 /* Delete objects in takeover library if move is wanted.
1860 /* Also,   d o n o t   delete step if copy is wanted.
L 1870 //DELE     EXEC NATBAT,COND=(0,NE,CAT.NATBAT)
L 1880 //CMPRT01 DD SYSOUT=*
L 1890 //CMWKF01 DD DSN=&&WKFO8,DISP=(OLD,DELETE,DELETE)
L 1900 //CMSYNIN DD *
L 1910 LOGON NATPAD
L 1920 NPP00762
L 1930 FIN
L 1940 /*
1950 /* Error during delete: set status OK-ND
L 1960 //STATND   EXEC NATBAT,COND=((0,EQ,DELE.NATBAT),(0,NE,CAT.NATBAT))
L 1970 //CMPRT01 DD SYSOUT=*
L 1980 //CMSYNIN DD *
L 1990 LOGON NATPAD
L 2000 &CARD
L 2010 FIN
L 2020 /*
2030 /* Error during delete archive settings
L 2040 //ARCDEL   EXEC NATBAT,COND=(0,EQ,CAT.NATBAT)
L 2050 //CMPRT01 DD SYSOUT=*
L 2060 //CMWKF01 DD DSN=&&WKFO5,DISP=(OLD,DELETE,DELETE)
L 2070 //CMSYNIN DD *
L 2080 LOGON NATPAD
L 2090 NPP00740
L 2100 /*

```

Figure 27: NPJ00700 NATURAL text module 'Copy to production' - part 4 of 6

```
>
All      .....1.....2.....3.....4.....5.....Mode Struc
2110 /* Error: set status error
L 2120 //STATER EXEC NATBAT,COND=(0,EQ,CAT.NATBAT)
L 2130 //CMPRT01 DD SYSOUT=*
L 2140 //CMSYNIN DD *
L 2150 LOGON NATPAD
L 2160 &CARD
L 2170 FIN
L 2180 /*
2190 /* Error: Recover old NATURAL objects in production
L 2200 //NATRL0D EXEC NATBATP,COND=(0,EQ,CAT.NATBAT)
L 2210 &CARD
L 2220 //CMSYNIN DD *
L 2230 LOGON NATPAD
L 2240 NATLOAD
L 2250 L;Y
L 2260 FIN
L 2270 /*
2280 /* Error: Purge buffer
L 2290 //PBUFFRE EXEC NATBATP,
L 2300 // COND=(0,EQ,CAT.NATBAT),
2310 // PARM.NATBAT='ADASVC=$$$,BPID=$$$' <----- change numbers
L 2320 //CMPRINT DD SYSOUT=*
L 2330 //CMWKF01 DD DSN=&&WKF07,DISP=(OLD,DELETE,DELETE)
L 2340 //CMPRT01 DD SYSOUT=*
L 2350 //CMSYNIN DD *
L 2360 LOGON NATPAD
L 2370 NPPBUFFE
L 2380 FIN
L 2390 /*
2400 /* Error: Recover old PREDICT objects in production
L 2410 //PRDRLOD EXEC NATBATP,COND=(0,EQ,CAT.NATBAT)
L 2420 &CARD
L 2430 //CMSYNIN DD *
L 2440 LOGON SYSDICBE
L 2450 LOAD ALL CODE=Y REPLACE=Y
L 2460 FIN
L 2470 /*
L 2480 //DELFIL2 EXEC PGM=IDCAMS,COND=(0,NE,CAT.NATBAT)
L 2490 //SYSPRINT DD SYSOUT=Z
L 2500 //SYSIN DD *
L 2510 DELETE -
L 2520 &CARD
L 2530 NOERASE -
L 2540 PURGE
L 2550 IF LASTCC LE 8 THEN DO
L 2560 SET LASTCC = 0
L 2570 SET MAXCC = 0
L 2580 END
L 2590 DELETE -
L 2600 &CARD
L 2610 NOERASE -
L 2620 PURGE
```

Figure 28: NPJ00700 NATURAL text module 'Copy to production' - part 5 of 6

```

>
All      .....1.....2.....3.....4.....5.....Mode Struc
L 2630      IF LASTCC LE 8 THEN DO
L 2640          SET LASTCC = 0
L 2650          SET MAXCC = 0
L 2660      END
L 2670      DELETE -
L 2680 &CARD
L 2690      NOERASE -
L 2700      PURGE
L 2710      IF LASTCC LE 8 THEN DO
L 2720          SET LASTCC = 0
L 2730          SET MAXCC = 0
L 2740      END
L 2750      DELETE -
L 2760 &CARD
L 2770      NOERASE -
L 2780      PURGE
L 2790      IF LASTCC LE 8 THEN DO
L 2800          SET LASTCC = 0
L 2800          SET LASTCC = 0
L 2810          SET MAXCC = 0
L 2820      END
L 2830 /*
  
```

Figure 29: NPJ00700 NATURAL text module 'Copy to production' - part 6 of 6



## NPJ00780 Backout Transmit

```

>                                     > + Text      NPJ00780      Lib   NATPAD
Top   ....+....1....+....2....+....3....+....4....+....5....+....Mode Struct
L 0010 &JOB
    0020 /* Create unload and laod staetements for backout
L 0030 //EVENT      EXEC NATBAT
L 0040 //SYSOUT     DD SYSOUT=Z
L 0050 //SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0060 //SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0070 //SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0080 //SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,2)
L 0090 //CMPRT01   DD SYSOUT=*
L 0100 //CMWKF01   DD DSN=&&WKFO1,DISP=(NEW,PASS,DELETE),
L 0110 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0120 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0130 //CMWKF02   DD DSN=&&WKFO2,DISP=(NEW,PASS,DELETE),
L 0140 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0150 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0160 //CMWKF03   DD DSN=&&WKFO3,DISP=(NEW,PASS,DELETE),
L 0170 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0180 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0190 //CMWKF04   DD DSN=&&WKFO4,DISP=(NEW,PASS,DELETE),
L 0200 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0210 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0220 //CMWKF05   DD DSN=&&WKFO5,DISP=(NEW,PASS,DELETE),
L 0230 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0240 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0250 //CMWKF06   DD DSN=&&WKFO6,DISP=(NEW,PASS,DELETE),
L 0260 //           UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
L 0270 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
L 0280 //CMSYNIN DD *
L 0290 LOGON NATPAD
L 0300 &CARD
L 0310 FIN
L 0320 /*
L 0330 //DELFIL1   EXEC PGM=IDCAMS
L 0340 //SYSPRINT DD SYSOUT=Z
L 0350 //SYSIN DD *
L 0360           DELETE -
L 0370 &CARD
L 0380           NOERASE -
L 0390           PURGE
L 0400           IF LASTCC LE 8 THEN DO
L 0410             SET LASTCC = 0
L 0420             SET MAXCC = 0
L 0430           END
L 0440           DELETE -
L 0450 &CARD
L 0460           NOERASE -
L 0470           PURGE
L 0480           IF LASTCC LE 8 THEN DO
L 0490             SET LASTCC = 0
L 0500             SET MAXCC = 0
L 0510           END

```

Figure 30: NPJ00780 NATURAL text module 'Backout Transmit' - part 1 of 4

```

>
> + Text      NPJ00780      Lib  NATPAD
Top  ....+....1....+....2....+....3....+....4....+....5....+....Mode Struct
L 0520 /*
L 0530 //GNATP      EXEC PGM=IEBGENER
L 0540 //SYSPRINT DD SYSOUT=Z
L 0550 //SYSUT1    DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 0560 //SYSUT2    DD DISP=(,CATLG),
L 0570 &CARD
L 0580 //          SPACE=(CYL,(25,5),RLSE),
L 0590 //          UNIT=SYSDA
L 0600 //SYSIN     DD *
L 0610 /*
L 0620 //GNATE      EXEC PGM=IEBGENER
L 0630 //SYSPRINT DD SYSOUT=Z
L 0640 //SYSUT1    DD DUMMY,DCB=(RECFM=VB,LRECL=4624,BLKSIZE=4628)
L 0650 //SYSUT2    DD DISP=(,CATLG),
L 0660 &CARD
L 0670 //          SPACE=(CYL,(25,5),RLSE),
L 0680 //          UNIT=SYSDA
L 0690 //SYSIN     DD *
L 0700 /*
0710 /* Unload NATURAL objects in production
L 0720 //NATUNLP    EXEC NATBATP,COND=(0,NE)
L 0730 &CARD
L 0740 //CMSYNIN DD DSN=&&WK01,DISP=(OLD,DELETE,DELETE)
L 0750 /*
0760 /* Unload NATURAL objects in production for transmit library (MOVE)
0770 /* D o n o t delete for COPY!
L 0780 //NATUNLE    EXEC NATBATP,COND=(0,NE)
L 0790 &CARD
L 0800 //CMSYNIN DD DSN=&&WK06,DISP=(OLD,DELETE,DELETE)
L 0810 /*
0820 /* Archive production objects
L 0830 //ARCHIV     EXEC NATBAT,COND=(0,NE)
L 0840 //CMPRT01 DD SYSOUT=*
L 0850 //CMWKF01 DD DSN=&&WK03,DISP=(OLD,PASS,DELETE)
L 0860 //CMSYNIN DD *
L 0870 LOGON NATPAD
L 0880 NPP00700
L 0890 FIN
L 0900 /*
0910 /* Delete production objects
L 0920 //DELPROD    EXEC NATBAT,COND=(0,NE)
L 0930 //CMPRT01 DD SYSOUT=*
L 0940 //CMWKF01 DD DSN=&&WK03,DISP=(OLD,PASS,DELETE)
L 0950 //CMSYNIN DD *
L 0960 LOGON NATPAD
L 0970 NPP00770
L 0980 FIN
L 0990 /*

```

Figure 31: NPJ00780 NATURAL text module 'Backout Transmit' - part 2 of 4

```

>
> + Text      NPJ00780      Lib  NATPAD
Top  ....+....1....+....2....+....3....+....4....+....5....+....Mode Struct
1000 /* Ascertain old objects in archive
L 1010 //GETARC EXEC NATBAT,COND=(0,NE)
L 1020 //CMPRT01 DD SYSOUT=*
L 1030 //CMWKF01 DD DSN=&&WKFO2,DISP=(OLD,PASS,DELETE)
L 1040 //CMSYNIN DD *
L 1050 LOGON NATPAD
L 1060 NPP00790
L 1070 FIN
L 1080 /*
1090 /* Catalog old subroutines
L 1100 //CATSUB EXEC NATBATP,COND=(0,NE),
1110 // PARM.NATBAT='ADASVC=$$$,BPID=$$$' <----- Change numbers
L 1120 //CMPRINT DD SYSOUT=*
L 1130 //CMSYNIN DD DSN=&&WKFO4,DISP=(OLD,DELETE,DELETE)
L 1140 /*
1150 /* After move recover production objects to transmit.
1160 /* D o n o t delete this step
L 1170 //LADENTW EXEC NATBAT,COND=(0,NE)
L 1180 //CMPRINT DD SYSOUT=*
L 1190 &CARD
L 1200 //CMSYNIN DD *
L 1210 LOGON NATPAD
L 1220 NATLOAD
L 1230 L;Y
L 1240 FIN
L 1250 /*
1260 /* Set status 'OK'
L 1270 //STATOK EXEC NATBAT,COND=(0,NE)
L 1280 //CMPRT01 DD SYSOUT=*
L 1290 //CMSYNIN DD *
L 1300 LOGON NATPAD
L 1310 &CARD
L 1320 FIN
L 1330 /*
1340 /* Purge buffer
L 1350 //PBUFFER EXEC NATBATP,COND=(0,NE),
1360 // PARM.NATBAT='ADASVC=$$$,BPID=$$$' <----- change numbers
L 1370 //CMPRINT DD SYSOUT=*
L 1380 //CMWKF01 DD DSN=&&WKFO5,DISP=(OLD,DELETE,DELETE)
L 1390 //CMPRT01 DD SYSOUT=*
L 1400 //CMSYNIN DD *
L 1410 LOGON NATPAD
L 1420 NPPBUFFE
L 1430 FIN
L 1440 /*
1450 /* Error: Delete archive entries
L 1460 //ARCDEL EXEC NATBAT,COND=(0,EQ,CATSUB.NATBAT)
L 1470 //CMPRT01 DD SYSOUT=*
L 1480 //CMWKF01 DD DSN=&&WKFO2,DISP=(OLD,DELETE,DELETE)
L 1490 //CMSYNIN DD *
L 1500 LOGON NATPAD
L 1510 NPP00740
L 1520 /*

```

Figure 32: NPJ00780 NATURAL text module 'Backout Transmit' - part 3 of 4

```

>
> + Text      NPJ00780      Lib  NATPAD
Top  ....+....1....+....2....+....3....+....4....+....5....+....Mode Struct
1530 /* Error: Set status 'error'
L 1540 //STATER EXEC NATBAT,COND=(0,EQ,CATSUB.NATBAT)
L 1550 //CMPRT01 DD SYSOUT=*
L 1560 //CMSYNIN DD *
L 1570 LOGON NATPAD
L 1580 &CARD
L 1590 FIN
L 1600 /*
1610 /* Error: Recover objects
L 1620 //NATRL0D EXEC NATBATP,
L 1630 // COND=(0,EQ,CATSUB.NATBAT)
L 1640 &CARD
L 1650 //CMSYNIN DD *
L 1660 LOGON NATPAD
L 1670 NATLOAD
L 1680 L;Y
L 1690 FIN
L 1700 /*
1710 /* Error: Purge buffer
L 1720 //PBUFFRE EXEC NATBATP,
L 1730 // COND=(0,EQ,CATSUB.NATBAT),
1740 // PARM.NATBAT='ADASVC=$$$,BPID=$$$' <----- change numbers
L 1750 //CMPRINT DD SYSOUT=*
L 1760 //CMWKF01 DD DSN=&&WKF05,DISP=(OLD,DELETE,DELETE)
L 1770 //CMPRT01 DD SYSOUT=*
L 1780 //CMSYNIN DD *
L 1790 LOGON NATPAD
L 1800 NPPBUFFE
L 1810 FIN
L 1820 /*
L 1830 //DELFIL2 EXEC PGM=IDCAMS,COND=(0,NE)
L 1840 //SYSPRINT DD SYSOUT=Z
L 1850 //SYSIN DD *
L 1860 DELETE -
L 1870 &CARD
L 1880 NOERASE -
L 1890 PURGE
L 1900 IF LASTCC LE 8 THEN DO
L 1910 SET LASTCC = 0
L 1920 SET MAXCC = 0
L 1930 END
L 1940 DELETE -
L 1950 &CARD
L 1960 NOERASE -
L 1970 PURGE
L 1980 IF LASTCC LE 8 THEN DO
L 1990 SET LASTCC = 0
L 2000 SET MAXCC = 0
L 2010 END
L 2020 /*

```

Figure 33: NPJ00780 NATURAL text module 'Backout Transmit' - part 4 of 4

## NPJ00950 Archive Administration

```
>                                     > +           Text           NPJ00950       Lib       NATPAD
All      ....+....1.....2.....3.....4.....5.....Mode Struct
L 0010 &JOB
L 0020 //ARCH           EXEC NATBAT
L 0030 //CMPRT01       DD SYSOUT=*
L 0040 //CMSYNIN       DD *
L 0050 LOGON NATPAD
L 0060 &CARD
L 0070 FIN
L 0080 /*
```

Figure 34: NPJ00950 NATURAL archive administration

Enter the name of the library of these text objects into the program NPN10004 (see 3.3.1).

### 3.3.15 JCL for ESS (NATURAL PROCESS)

Adapt the following 4 JCL PDS member for the target environment:

- NPJ00100 Restart PREDICT
- NPJ00620 Copy PREDICT objects
- NPJ00700 Copy to production
- NPJ00780 Backout takeover

Enter the name of the PO data set of these members into the program NPN10004 (see 3.3.1).

### 3.3.16 Administration programs

Parts of archive and reference files can be unloaded if they become too large.

ARCHSAV contains a job to copy NATURAL objects in NATUNLD format to a sequential data set, e.g. tape. Additionally, all activities are recorded in another data set.

Parameters provide control to unload:

- Objects older than a given date
- Number of archived objects are greater than a given date

ARCHGMBR contains a job to find unloaded objects.

A report displays the time of archive and unload, of a relevant object. With this report it is possible to find the back-up tape. Then the object will be loaded with NATLOAD and the relevant REFName (reference name) into the NATURAL library 'ARCHIV'.

## 4 Create ADABAS Files and Copy DDMs

### 4.1 ADABAS Files

ADAFDTS contains all ADABAS FDTs necessary to create

FDT Name	DDM Name	File Number	DB ID
Requests	NP-AUFT-E NP-AUFT-M		
Reference	NP-REF		
Archive	NP-ARC		

Note: Transition is not an ADABAS file. It is a NATURAL library.

#### 4.1.1 Request File

Create an empty ADABAS file to store and maintain requests. Use ADACMP control statements and change the file number if necessary (see Figure 35).

Figure 36 shows the field description table (FDT) after ADALOD.

```

000001 // * 1. NP-AUFT      AUFTRÄGE - REQUEST
000002 ADACMP COMPRESS
000003 ADACMP FILE=20
000004 ADACMP MINISN=1
000005 ADACMP DEVICE=3380
000006 ADACMP FNDEF='01,AA,8,A,DE'
000007 ADACMP FNDEF='01,AB,1,A,NU'
000008 ADACMP FNDEF='01,AC,6,U'
000009 ADACMP FNDEF='01,AD,2,A,NU'
000010 ADACMP FNDEF='01,AE,8,A,NU'
000011 ADACMP FNDEF='01,AF,3,A,NU'
000012 ADACMP FNDEF='01,AG,8,A,NU'
000013 ADACMP FNDEF='01,AH,8,A,NU'
000014 ADACMP FNDEF='01,AI,8,A,NU'
000015 ADACMP FNDEF='01,AJ,8,U,NU'
000016 ADACMP FNDEF='01,AK,5,A,NU'
000017 ADACMP FNDEF='01,AL,7,U,NU'
000018 ADACMP FNDEF='01,AM,8,A,NU'
000019 ADACMP FNDEF='01,AN,5,U,NU'
000020 ADACMP FNDEF='01,AO,6,A,NU'
000021 ADACMP FNDEF='01,AP,8,U,NU'
000022 ADACMP FNDEF='01,AQ,7,U,NU'
000023 ADACMP FNDEF='01,AT,20,A,NU'
000024 ADACMP FNDEF='01,AV,50,A,MU,NU'
000025 ADACMP SUPDE='AR=AA(1,8),AC(1,6),AG(1,8)'
000026 ADACMP SUPDE='AS=AA(1,8),AC(1,6),AE(1,8)'
000027 ADACMP SUPDE='AU=AA(1,8),AC(1,6)'
000028 ADACMP SUPDE='AW=AC(1,6),AA(1,8),AG(1,8)'
000029 / *

```

Figure 35: ADACMP control statements to load REQUEST file (part or ADAFDTS)

Top	Ty	L	Name	F	Length	D	U	DB	S
			-----						
		1	USERID	A	8,0	D		AA	
		1	TEAM	A	1,0			AB	N
		1	EVENT	N	6,0			AC	
		1	LIB	A	2,0			AD	N
		1	MODULE	A	8,0			AE	N
		1	MODULE-TYP	A	3,0			AF	N
		1	LIB-FROM	A	8,0			AG	N
		1	LIB-TO	A	8,0			AH	N
		1	LIB-TEST	A	8,0			AI	N
		1	DATE-EVENT	N	8,0			AJ	N
		1	DATE-J	A	5,0			AK	N
		1	TIME-EVENT	N	7,0			AL	N
		1	STATUS	A	8,0			AM	N
		1	JOBNR	N	5,0			AN	N
		1	JOB-STATUS	A	6,0			AO	N
		1	DATE-OK	N	8,0			AP	N
		1	TIME-OK	N	7,0			AQ	N
		1	EVENTTXT	A	20,0			AT	N
	MU	1	INFOTXT	A	50,0			AV	N
	SP	1	USER-EVENT-LIB-FROM	A	22,0	D		AR	N
	SP	1	USER-EVENT-MODULE	A	22,0	D		AS	N
	SP	1	USERID-EVENT	A	14,0	D		AU	N
	SP	1	EVENT-USER-LIB-FROM	A	22,0	D.		AW	N

Figure 36: DDM request file with FDT

## 4.1.2 Reference File

Create an empty ADABAS file. Use ADACMP control statements and change the file number if necessary (see Figure 37). This file is used to reference member names in archive with real object names

```

000027 /** 2. NP-PROT   PROTOKOLL - Reference
000028 ADACMP COMPRESS
000029 ADACMP FILE=21
000030 ADACMP MINISN=1
000031 ADACMP DEVICE=3390
000032 ADACMP FNDEF='01,AA,8,A'
000033 ADACMP FNDEF='01,AB,8,U'
000034 ADACMP FNDEF='01,AC,7,U'
000035 ADACMP FNDEF='01,AD,8,U'
000036 ADACMP FNDEF='01,AE,7,U'
000037 ADACMP FNDEF='01,AF,3,A'
000038 ADACMP FNDEF='01,AG,8,A'
000039 ADACMP FNDEF='01,AH,8,A'
000040 ADACMP FNDEF='01,AI,8,A,NU'
000041 ADACMP SUPDE='AJ=AA(1,8),AD(1,8),AE(1,7)'
000042 /*

```

Figure 37: ADACMP control statements to load REFERENCE file (part or ADAFDTS)

### 4.1.3 Archive File

Create empty ADABAS file with FUSER FDT to archive NATURAL programs. Figure 38 shows the field description table (FDT). Archive file is used to store all prior versions of NATURAL objects. Superdescriptor LX is not necessary.

Lev	I	Name	I	Leng	I	Form	I	Options
1	I	AF	I	064	I	A	I	NU DE
1	I	BX	I	035	I	A	I	MU NU
1	I	LA	I	075	I	A	I	NU DE
1	I	LB	I	002	I	B	I	NU DE
1	I	LC	I	250	I	A	I	MU NU
1	I	LE	I	004	I	U	I	NU DE
1	I	LF	I		I		I	PE
2	I	LG	I	079	I	A	I	NU
1	I	LJ	I	018	I	A	I	NU DE
1	I	LK	I	094	I	A	I	MU NU
1	I	LL	I	018	I	A	I	NU DE
1	I	LM	I	250	I	A	I	MU NU
1	I	LO	I	075	I	A	I	NU DE
1	I	LP	I	075	I	A	I	NU DE
1	I	LQ	I	008	I	A	I	NU
1	I	LR	I	008	I	A	I	NU
1	I	LS	I	008	I	A	I	NU
1	I	LT	I	008	I	A	I	NU
1	I	LU	I	008	I	A	I	NU
1	I	LV	I	110	I	A	I	NU
1	I	LW	I	036	I	A	I	MU NU DE
1	I	LI	I	032	I	A	I	NU
1	I	L0	I	014	I	A	I	NU DE
1	I	L1	I	010	I	A	I	MU NU DE
1	I	L2	I	072	I	A	I	MU NU DE
1	I	L3	I	014	I	A	I	MU NU
1	I	L4	I	006	I	U	I	NU
1	I	L5	I	006	I	U	I	NU
1	I	L6	I	006	I	U	I	NU
1	I	L7	I	006	I	U	I	NU
1	I	L8	I	006	I	U	I	NU
1	I	NC	I	004	I	B	I	NU
1	I	ND	I	009	I	A	I	NU DE UQ
1	I	NP	I	009	I	A	I	NU DE UQ
1	I	NS	I	003	I	B	I	NU DE UQ
1	I	NW	I	020	I	A	I	MU NU
1	I	NZ	I	020	I	A	I	MU NU

Figure 38: Archive file - same FDT as FUSER (NAT232 example)

### 4.1.4 FUSER File

Check-in function requires superdescriptor ZJ. Copy functions ARCH (PF9), PROD (PF10), and DEVEL (PF11) from menu 'Register Requests' check out existing NATURAL source code in entire target environment FUSER and produces pop-up window with error message.





Also it is necessary to change parameter value **#SRCESUP** in source **NPN10002**.

## 4.1.5 Transition Library

Transition library is used to store NATURAL programs temporarily before moving into production. This library is mandatory. An existing library can be used. Noone should have access to update modules in this library.

All tests can be run using this library.

The name of the library must be defined in **NPN10001**.

## 4.2 Description of Data Definition Modules (DDM)

### 4.2.1 Requests - NP-AUFT-E

DB	11	File	137	-	NP-AUFT-E	Default	Sequence
T L	DB	Name		F	Leng	S	D Remarks
-----							
*		Generation started at	97-03-26	1	5		2 1
*		by user	DX70				
*							
1	AA	USERID		A	8		
1	AB	TEAM		A	1		N
1	AC	EVENT		N	6.0		
1	AD	LIB		A	2		N
1	AG	LIB-FROM		A	8		N
1	AH	LIB-TO		A	8		N
1	AJ	DATE-EVENT		N	8.0		N
1	AK	DATE-J		A	5		N
1	AL	TIME-EVENT		N	7.0		N
1	AM	STATUS		A	8		N
1	AP	DATE-OK		N	8.0		N
1	AQ	TIME-OK		N	7.0		N
1	AI	LIB-TEST		A	8		N
1	AN	JOBNR		N	5.0		N
1	AO	JOB-STATUS		A	6		N
1	AT	EVENTTXT		A	20		N
1	AV	INFOTXT		A	50		N MU(1:10)
1	AR	USER-EVENT-LIB-FROM		A	22		N S
*		----- SOURCE FIELD(S) -----					
*		USERID(1-8)					
*		EVENT(1-6)					
		LIB-FROM(1-8)					
1	AW	EVENT-USER-LIB-FROM		A	22		N S
*		----- SOURCE FIELD(S) -----					
*		EVENT(1-6)					
*		USERID(1-8)					
*		LIB-FROM(1-8)					

Figure 41: DDM NP-AUFT-E

## 4.2.2 Requests - NP-AUFT-M

DB	11	File	137	-	NP-AUFT-M	Default Sequence	
T L	DB	Name	F	Leng	S	D	Remarks
*		Generation started at 97-02-20	1	4	5	0	
*		by user DX70					
*		modified by D.Storr at 98-07-22					
*							
1	AA	USERID	A	8		D	
1	AC	EVENT	N	6.0			
1	AE	MODULE	A	8		N	
1	AF	MODULE-TYP	A	3		N	
1	AS	USER-EVENT-MODULE	A	22		N S	
*		----- SOURCE FIELD(S) -----					
*		USERID(1-8)					
*		EVENT(1-6)					
*		MODULE(1-8)					
1	AU	USERID-EVENT	A	14		N S	
*		USERID AND EVENT TO DETERMINE					
*		NUMBER OF MODULES PER REQUEST					
*		----- SOURCE FIELD(S) -----					
*		USERID(1-8)					
*		EVENT(1-6)					

Figure 42: DDM NP-AUFT-M

## 4.2.3 Reference - NP-REF

DB	11	File	136	-	NP-REF	Default Sequence	
T L	DB	Name	F	Leng	S	D	Remarks
*		Generation started at 97-02-06	1	5	2	1	
*		by user DX70					
*							
1	AA	OBJEKT	A	8			
1	AB	DATUM	N	8.0			
1	AC	UHRZEIT	N	7.0			
1	AD	DATUM-K	N	8.0			
1	AE	UHRZEIT-K	N	7.0			
1	AF	OBJEKTYP	A	3			
1	AG	REFNAME	A	8			
1	AH	USER	A	8			
1	AI	AUS-LIB	A	8		N	
1	AJ	SUP-OBJ-DAT-UHR-K	A	23		S	
*		----- SOURCE FIELD(S) -----					
*		OBJEKT(1-8)					
*		DATUM-K(1-8)					
*		UHRZEIT-K(1-7)					

Figure 43: DDM NP-REF

Reference file NP-REF must be defined with USERISN=YES

## 4.2.4 Archive - NP-ARC

T	L	DB	Name	F	Leng	S	D	Remarks
		DB 11	File 135 - NP-ARC					Default Sequence
*			Generation started at 97-05-31 1 7					1 3
*			by user DX70					
1	LJ		SRCID			A	18	N D
M 1	LK		SRCTX			A	90	N
1	LL		OBJID			A	18	N D
M 1	LM		OBJCHUNK			A	250	N
P 1	LF		ERROR					
			HD=ADACOM/NATURAL/MESSAGE					
2	LG		ERRTXT			A	65	N
			HD=ADACOM/NATURAL/ERROR-MESSAGE					
1	LO		INTERNA01			A	32	N D
1	LE		ERRNR			N	4.0	N D
			HD=ADACOM/NATURAL/ERROR-NUM					

Figure 44: DDM NP-ARC

## 4.2.5 FUSER and FNAT

The following DDMs are used to read NATURAL system and user files:

**NP-FUSER-DEVL** is used to read user file in development for transition.

**NP-FUSER-PROD** is used to read user file in production for transition.

**NP-FNAT-DEVL** is used to read system file in development for transition from syslibs.

**NP-FNAT-PROD** is used to read system file in production for transition from syslibs.

**Remark:** NATURAL utility **SYSMAIN** is used to store NATURAL source.

T	L	DB	Name	F	Leng	S	D	Remarks
-	-	-	-----	-	----	-	-	-----
*			Generation started at 97-05-31 1 7			4	8	
*			by user DX70					
*								
1	LJ		SRCID	A	18	N	D	
M 1	LK		SRCTX	A	90	N		
1	LL		OBJID	A	18	N	D	
M 1	LM		OBJCHUNK	A	250	N		
P 1	LF		ERROR					
			HD=ADACOM/NATURAL/MESSAGE					
2	LG		ERRTXT	A	65	N		
			HD=ADACOM/NATURAL/ERROR-MESSAGE					
1	LO		INTERNA01	A	32	N	D	
1	LE		ERRNR	N	4.0	N	D	
			HD=ADACOM/NATURAL/ERROR-NUM					

Figure 45: Same DDMs for FUSER-ENTW, FUSER-PROD, FNAT-ENTW, FNAT-PROD

## 5 NATURAL SECURITY

### 5.1 Files

All files must be defined in NATURAL Security:

```

19:18:49          *** NATURAL SECURITY ***          97-08-12
                  - File Retrieval -

Co File ID                Status Message
-----
___ NP-ARC                 PUBL
___ NP-AUFT-E              PUBL
___ NP-AUFT-M              PUBL
___ NP-FNAT-DEVL           PUBL
___ NP-FNAT-PROD           PUBL
___ NP-FUSER-DEVL          PUBL
___ NP-FUSER-PROD          PUBL
___ NP-REF                 PUBL
  
```

Figure 46: Define file entries in NATURAL SECURITY

### 5.2 Automatic Logon

All batch jobs will be submitted with *LOGON SYSDICBE* or *LOGON NATPAD* and an automatic logon in batch mode. Therefore, it is necessary to use NATURAL profile parameter *AUTO=ON*. The batch job name will be taken as user ID. This batch job name must be defined as a *USER* to NATURAL SECURITY. A logon with a user ID other than the batch job name will not be possible.

```

10:46:34          *** NATURAL SECURITY ***          97-08-11
                  - Display Group -

                                     Modified .. 97-08-10 by USER01
User ID .....      NP-BATCH
User Name ....     NAT-PAD Administration
User Type ....     G      (G=Group)

Members            Libraries
-----
USER01@           Default ..      NATPAD
USER02@
USER03@
USER04@           Batch User ID .....
USER05@

No. members      6

Additional Options ...  N
  
```

Figure 47: NATURAL SECURITY user group definition

NAT-PAD creates the jobname in this example (see Figure 47 and Figure 48) by using user-id and an add-sign (@) . See also: Define job card NPNSUBRJ - 3.3.12. Be sure

that all members have default library 'NATPAD'. Default description at group level does not suffice.

```
10:53:04          *** NATURAL SECURITY ***          97-08-11
                  - Display User -

No User ID          User Name          Default  Last  Ty No.  M G P O
Library            Library  pe Gr/M B E L W
-----
01 USER01@  user name 1 (QA)          NATPAD  NATPAD  M 0002  N
02 USER02@  user name 2 (QA)          NATPAD                M 0002  N 1
03 USER03@  user name 3 (QA)          NATPAD                M 0002  N 1
04 USER04@  user name 4 (QA)          NATPAD  NATPAD  M 0002  N 1
05 USER05@  user name 5 (QA)          NATPAD  NATPAD  M 0002  N 1
06 USER06@  user name 6 (QA)          NATPAD                M 0002  N 1

-----
Reposition to:          Exit:          _
```

Figure 48: NATURAL SECURITY member definition

If one library is defined with restrictions (Restrictions Y) make sure that your group is linked to the library via a 'special link' with 'no restrictions'. Otherwise, you will receive after CATAL the message: 'NAT0972 User is not authorized to use this command'.

Also, be careful to define these batch user ID's with owner ID's.

## 6 Miscellaneous

### 6.1 Delimiter

NATUNLD and NATLOAD utilities executing in batch mode using parameters, individual parameter values are separated by the delimiter ",". The input-mode parameter (IM) is recommended to be set to delimiter mode (IM=D).

### 6.2 NATLOAD Condition Code 37

NATLOAD processing in batch mode terminates with condition code 37, if objects requested for loading could not be found on work file (Workfile 1 contains only an header record); Workfile was empty; or no objects were loaded for the specified NATLOAD request

In the case of any condition code except 0, the user exit LOADEX01 will be invoked, if available. If the RET-CODE is set to 0 in LOADEX01, NATLOAD will end with condition code 0. For more information please see 'Utilities Manual for Mainframes', SYSUNLD Utilities, Condition Codes and User Exit in Batch Mode or LOGON to SYSUNLD, source code L-S-EX01

Table of Figures

Figure 1: Description of the contents of the diskette..... 6

Figure 2: NATUNLD..... 6

Figure 3: Data set Allocation ..... 7

Figure 4: File transfer between PC and Host using Rumba..... 8

Figure 5: IEBUPDT - load the PDS member ..... 8

Figure 6: NATLOAD - online..... 9

Figure 7: NATLOAD - batch - INPL..... 9

Figure 8: Subprogram NPN1000 ..... 10

Figure 9: Pop-up window shows NATURAL table values to select ..... 11

Figure 10: Subprogram NPN10001 ..... 11

Figure 11: Subprogram NPN10002 ..... 12

Figure 12: Subprogram NPN10003 ..... 12

Figure 13: Subprogram NPN10004 ..... 12

Figure 14: Subprogram NPN10005 ..... 13

Figure 15: Subprogram NPN10006 ..... 13

Figure 16: Subprogram NPN00210 ..... 14

Figure 17: Subprogram NPN00220 ..... 14

Figure 18: Subprogram NPP00620 ..... 15

Figure 19: Subprogram NPNSUBNP..... 15

Figure 20: NPNSUBRJ creates the job card and submits via NATRJE..... 16

Figure 21: NPNSUBCJ creates the job card and submits via RJE..... 16

Figure 22: NPJ00100 NATURAL text module 'Restart PREDICT' ..... 17

Figure 23: NPJ00620 NATURAL text module 'Copy PREDICT objects' ..... 18

Figure 24: NPJ00700 NATURAL text module 'Copy to production' - part 1 of 6 ..... 19

Figure 25: NPJ00700 NATURAL text module 'Copy to production' - part 2 of 6 ..... 20

Figure 26: NPJ00700 NATURAL text module 'Copy to production' - part 3 of 6 ..... 21

Figure 27: NPJ00700 NATURAL text module 'Copy to production' - part 4 of 6 ..... 22

Figure 28: NPJ00700 NATURAL text module 'Copy to production' - part 5 of 6 ..... 23

Figure 29: NPJ00700 NATURAL text module 'Copy to production' - part 6 of 6 ..... 24

Figure 30: NPJ00780 NATURAL text module 'Backout Transmit' - part 1 of 4 ..... 25

Figure 31: NPJ00780 NATURAL text module 'Backout Transmit' - part 2 of 4 ..... 26

Figure 32: NPJ00780 NATURAL text module 'Backout Transmit' - part 3 of 4 ..... 27

Figure 33: NPJ00780 NATURAL text module 'Backout Transmit' - part 4 of 4 ..... 28

Figure 34: NPJ00950 NATURAL archive administration ..... 29

Figure 35: ADACMP control statements to load REQUEST file (part or ADAFDTS)..... 30

Figure 36: DDM request file with FDT ..... 31

Figure 37: ADACMP control statements to load REFERENCE file (part or ADAFDTS)..... 31

Figure 38: Archive file - same FDT as FUSER (NAT232 example) ..... 32

Figure 39: Control cards for ADAINV INVERT superdescriptor ZJ for check-in function ..... 33

Figure 40: FDT of FUSER – enhanced with one superdescriptor (NAT232 example)..... 33

Figure 41: DDM NP-AUFT-E ..... 34

Figure 42: DDM NP-AUFT-M ..... 35

Figure 43: DDM NP-REF ..... 35

Figure 44: DDM NP-ARC ..... 36

Figure 45: Same DDMs for FUSER-ENTW, FUSER-PROD, FNAT-ENTW, FNAT-PROD ..... 37

Figure 46: Define file entries in NATURAL SECURITY ..... 38

Figure 47: NATURAL SECURITY user group definition ..... 38

Figure 48: NATURAL SECURITY member definition ..... 39