Natural

Installation for BS2000

Version 8.2.7

October 2017

ADABAS & NATURAL

This document applies to Natural Version 8.2.7 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: NATMF-INSTALL-BS2-827-20180201

Table of Contents

Preface

This documentation describes the installation of base Natural and Natural add-on products on BS2000.

⚠ **Important:** Software AG products on BS2000 are affected by the BS2000 C/C++ time overflow problem which can occur after January 19, 2018. For instructions on how to proceed to avoid this problem, refer to *BS2000 C/C++ Time Overflow* in the current Natural *Release Notes*.

Basic Information:

Base Natural:

Base Natural - Optional Components:

TP Monitor Interfaces:

Other Natural Add-On Products:

1 Installation Process and Major Natural Features on BS2000

This document provides general information on the prerequisites and processes required to install base Natural and Natural add-on products. In addition, it describes installation tools and major Natural components required for installation.

Notation `vrs` or `vr`:

When used in this document, the notation `vrs` or `vr` represents the relevant product version (see also Version in the *Glossary*).

General Prerequisites and System Support

Before beginning the installation process, consider the following:

- Be sure to read the current Natural *Release Notes* for Mainframes for information on software and hardware requirements, known issues and changes to the documentation. These *Release Notes* apply to base Natural and Natural add-on products.
- A supported version of the operating system on which Natural is to run must be installed. For the supported operating systems and versions, refer the **Product Version Availability** section of Software AG's Empower web site at ***https://empower.softwareag.com/***.
- A supported version of the TP monitor/online interface used with Natural must be installed. For the supported versions, refer to *TP Monitors/Online Interfaces* in the current Natural *Release Notes* for Mainframes.
- A supported version of Adabas must be installed to store the Natural system files. See also ***Natural System Files***.

  For the supported versions, refer to *Database Management Systems* in the current Natural *Release Notes* for Mainframes.
- A supported version of each database management system used to store the user data processed with Natural must be installed.

  For the supported versions, refer to *Database Management Systems* in the current Natural *Release Notes* for Mainframes.

Installation Medium

The installation medium (for example, tape or CD-ROM) distributed for Software AG mainframe products contains all files required to install base Natural and the Natural add-on products.

The software required for the optional Natural components are contained in the files supplied for base Natural. The software required for the Natural add-on products are contained in separate product files which are listed in the product-specific sections of the *Installation for BS2000* documentation. In addition to the product files, the installation medium can contain the latest fix updates for the supplied products.

The names of the product files begin with a product code that identifies each product, as in the following table:

Product Code	Product Name
ETP	Entire Transaction Propagator
NAF	Natural Advanced Facilities
NAT	Natural
NOC	Natural Optimizer Compiler
NRT	Natural TIAM Interface
NSC	Natural Security
NTC	Natural Connection
NUT	Natural *open*UTM Interface

Software AG Product Delivery Report

Each installation medium is delivered with a Software AG Product Delivery Report providing the following information:

- A list of all files contained on the medium.
- The sequence in which the files are located on the medium.
- Attribute descriptions of each file.

Installation Method

The installation of Software AG products on BS2000 is performed by installation jobs that contain the JCL required to identify the job to the operating system and run the job.

There are two methods for creating and running the installation jobs:

- using the jobs generated by System Maintenance Aid (SMA), or
- using the jobs created from the sample installation jobs provided.

The *Installation for BS2000* documentation solely describes the installation procedure for the jobs generated by SMA. If you do not use SMA for installation, refer to the example installation jobs supplied on the installation medium.

SMA is supplied with base Natural.

System Maintenance Aid (SMA)

For each step of the installation procedure, System Maintenance Aid (SMA) generates an installation job according to your specifications in SMA. You then submit and run the generated job.

Before you can start generating the jobs, you have to load the `SMT111.TABS` file from the installation medium into the SMA system file. `SMT111.TABS` contains the tables SMA requires to build the jobs.

SMA is supplied with base Natural. For instructions on loading the file and using SMA, refer to the *System Maintenance Aid* documentation.

Readme File

For installation guidance and information on new or changed SMA parameters and Natural features, you can view the product-specific Readme files by using the appropriate SMA function.

To view a product-specific Readme

- From the product list on an SMA **Maintenance** screen, execute the `RM` (**Show Readme File**) command for the required product(s).

Sample Installation Jobs

The sample installation jobs that can be used as an alternative to SMA are provided in an LMS library contained on the file *product-code-vrs*.JOBS (for example, NAT824.JOBS) shipped on the installation medium. All sample installation jobs provided are listed and described in the README document that accompanies the shipment.

You need to adapt the sample installation jobs to your requirements.

Installation Job Identification

Each installation job indicates the **product code** and version (for example, NAT824) of the corresponding product (for example, Natural).

Each step of the installation procedure is identified by a job name (for example, I050) and one or more steps (for example, Steps 0100 and 0101 for Job I050) that indicate the tasks performed by the job. The job name can have a prefix such as a **product code** (for example, NATI050).

A sample installation job from the LMS library can also have a suffix letter which indicates a variant of the job. For example: Job I060L is a variant of Job I060 and used if support of the IBM Language Environment (LE) is required. In SMA, the same variant is executed with Job I060 and the appropriate SMA parameter setting.

Overall Installation Procedure

The installation process comprises the following:

1. Creating the Natural system files.
2. Creating the Natural parameter module.
3. Creating the Natural nucleus.
4. Loading the Natural objects.
5. Installing the optional Natural components.
6. Installing the Natural add-on products.

Naming Conventions for Installation Procedures

This section describes the naming conventions that apply to the installation procedures of the *Installation for BS2000* documentation:

- job-library
- Elements in job-library

job-library

job-library denotes one of the following:

- the sample job library (*product-code-vrs*.JOBS, for example, NAT824.JOBS) if you are *not* using **System Maintenance Aid (SMA)** or
- the SMA job library if you are using SMA. See the SMA parameter JOBLIB in the SMA Parameter Group BASIC.

Elements in job-library

The following naming conventions are used for the source elements contained in the *job-library* library:

```
Aproduct-codefunction
```

for assembler sources and

```
Lproduct-codefunction
```

for instructions for BINDER.

Examples:

ANATFRNT, ANATRENT, ANATSTUB and LNATFRNT

Installation Verification

Verify the successful completion of the installation by starting Natural and testing the system functions as described in the relevant sections of the *Installation for BS2000* documentation.

Note: If Natural Security is installed, certain Natural functions and libraries can be restricted to specific users.

INPL Utility

The installation instructions frequently refer to the Natural INPL utility which is used to load the files (for example, `NAT`*`vrs`*`.INPL`) contained on the Natural installation medium into the **Natural system files**. The INPL utility is invoked with the Natural system command `INPL`. For detailed information on the INPL utility, refer to the *Utilities* documentation.

Natural Nucleus Components

The Natural nucleus consists of two functional parts: the environment-independent nucleus and the environment-dependent nucleus.

Note: If you maintain different versions of Natural, you must use distinctive names for the nuclei to identify each version.

This section covers the following topics:

- Environment-Independent Nucleus
- Environment-Dependent Nucleus
- Modules for Static Linking
- Modules for Dynamic Loading
- Modules Called Dynamically

Environment-Independent Nucleus

The environment-independent nucleus contains components that are independent of the operating system or TP system (online interface) being used. The same instance of the environment-independent nucleus can be used in different online and batch environments in different tasks. The environment-independent nucleus is reentrant.

The environment-independent nucleus can reside in a common memory pool (CMP) or in Class 4 Memory where it can be shared between different tasks.

A module (such as the environment-independent nucleus) loaded into a CMP or Class 4 Memory is protected against modification. Therefore, tests for modifications of the environment-independent nucleus should be performed in a separate environment.

If the environment-independent nucleus resides in the CMP or Class 4 Memory, multiple batch jobs or TP tasks (for example, TIAM) share the same instance of the environment-independent nucleus. This results in a significant reduction of paging activities and virtual storage consumption.

Modules for Linking

The following modules must be linked to the environment-independent nucleus:

- Modules for base Natural
- Environment-independent modules of Natural add-on products
- Environment-independent user-supplied modules

When using System Maintenance Aid (SMA), the required modules are linked to the environment-independent nucleus during the appropriate installation job/step. Modules that can optionally be linked are mentioned in the *Installation Procedure*.

The installation of the environment-independent nucleus is described in *Link the Nucleus* in the *Installation Procedure*.

Specifying the Nucleus Name

The name of the environment-independent nucleus to be used is specified with the Natural profile parameter `NUCNAME` in the Natural parameter module during the installation of the environment-dependent nucleus. You can specify `NUCNAME` as a dynamic parameter in the primary parameter input, but you cannot specify `NUCNAME` in the input strings of the Natural profile parameter `PROFILE` or `SYS`.

The Natural parameter module is described in *Building a Natural Parameter Module* in the *Operations* documentation. `NUCNAME`, `PROFILE` and `SYS` are described in the *Parameter Reference* documentation.

If you maintain different versions of Natural, we recommend that you use distinctive names for the nucleus to clearly identify each version and environment, for example: `NAT824` for the environment-independent nucleus, `NAT824C` for the environment-dependent nucleus for a CICS interface, and `NAT824B` for the batch environment.

Rules for Using an Environment-Independent Nucleus

The rules that apply when using the environment-independent nucleus are described in the following section.

Concerning *open*UTM, see also *Several Applications with one Common Natural* in the *TP Monitor Interfaces* documentation.

1. The environment-independent nucleus is linked *without* the corresponding reentrant parts of the batch, TIAM and *open*UTM drivers (these modules must be linked to the **environment-dependent nucleus**).

   Example: The name of the environment-independent nucleus is `NATSHARE`.

```
/START-PROGRAM $BINDER
START-LLM-CREATION INTERNAL-NAME=NATSHARE,                                  -
SLCE-DEFINITION=*BY-ATTRIBUTES(RESIDENCY-MODE=*YES)
INCLUDE-MODULES NATINV,LIB=(NATvrs.MOD)
.
.
INCLUDE-MODULES NATURAL,LIB=(NATvrs.MOD)
.
.
INCLUDE-MODULES NATLAST,LIB=(NATvrs.MOD)
MODIFY-SYMBOL-ATTRIBUTES ADD-MODE=*31,RES-MODE=*ANY,READ-ONLY=*NO
MODIFY-SYMBOL-VISIBILITY                                                    -
          SYMBOL-NAME = *ALL,                                               -
          SYMBOL-TYPE = *DEFINITIONS,                                       -
          VISIBLE = *NO(KEEP-RESOLUTION=*YES)
SAVE-LLM LIB=NATURAL.USER.MOD
END
```

where *vrs* is the current product version of Natural.

2. Batch, TIAM and *open*UTM application-specific Natural parameter modules are also linked to the environment-dependent nucleus. In addition, the environment-independent nucleus can contain a common Natural parameter module, for example, for CSTATIC entries. The name chosen for the linked environment-independent nucleus is also identical with the name of the global common memory pool into which Natural is loaded. This name is to be used as operand for the following keyword parameters:

Name	Used in
NUCNAME	the macros NAMBS2, NAMTIAM and NATUTM
NAME	CMPSTART and ADDON (BS2STUB) described in the *Operations* documentation

Example:

```
NRTSTART NAMTIAM CODE=FRONT,
                    NUCNAME=NATSHARE
                    PARMODE=(31,ABOVE),
                    .
                    .
NUTFRONT NATUTM APPLNAM=NATUTM,
                    .
                    .
                    NUCNAME=NATSHARE
                    PARMODE=(31,ABOVE)
```

3. The environment-independent nucleus is started by the program CMPSTART.

Example:

```
/EXEC (CMPSTART,NATURAL.MOD)
NAME=NATSHARE,SIZE=2MB,POSI=ABOVE,ADDR=250,SCOP=GLOBAL
PFIX=YES,LIBR=NATURAL.USER.MOD
```

4. The link to the environment-independent nucleus is created in the batch, TIAM or *open*UTM applications through the generation of the macro `BS2STUB`; refer to *CMPSTART Program* described in the *Operations* documentation.

   Example:

```
NRTSTUB BS2STUB PARMOD=31,PROGMOD=ANY
             ADDON NAME=NATSHARE,
                   STAT=GLOBAL
NUTSTUB BS2STUB PARMOD=31,PROGMOD=ANY
             ADDON NAME=NATSHARE,
                   STAT=GLOBAL
```

5. The environment-dependent nucleus must contain the reentrant part of the corresponding driver (`NAMBS2 CODE=RENT`, `NAMTIAM CODE=RENT` or `NURENT`).

   Example for TIAM:

```
/START-PROGRAM $BINDER
START-LLM-CREATION INTERNAL-NAME=NRTFRONT,                           -
SLICE-DEFINITION=*BY-ATTRIBUTES(RESIDENCY-MODE=*YES)
INCLUDE-MODULES NRTFRNT,LIB=(user-lib)    /* Non-reentrant part of NAMTIAM   ↩

INCLUDE-MODULES NRTRENT,LIB=(user-lib)    /* Reentrant part of NAMTIAM
INCLUDE-MODULES NRTSTUB,LIB=(user-lib)    /* BS2STUB
INCLUDE-MODULES nat-parm-module,LIB=(user-lib)    /*
...
END
```

   Example for *open*UTM:

```
/START-PROGRAM $BINDER
START-LLM-CREATION INTERNAL-NAME=NUTvrs,                             -
SLICE-DEFINITION=*BY-ATTRIBUTES(RESIDENCY-MODE=*YES)
INCLUDE-MODULES KDCNUT,LIB=(user-lib)              /* openUTM KDCROOT
INCLUDE-MODULES NUSTART,LIB=(user-lib)             /* NATUTM
INCLUDE-MODULES NUTRENT,LIB=(user-lib)             /* NURENT
INCLUDE-MODULES NUTSTUB,LIB=(user-lib)             /* BS2STUB
INCLUDE-MODULES nat-parm-module,LIB=(user-lib)
INCLUDE-MODULES SWPPARM,LIB=(user-lib)    /* Swap pool parameter module
...
END
```

   where:

   `vrs` is the current product version and

user-lib is the name of the user library,
nat-parm-module is the Natural parameter module.

Environment-Dependent Nucleus

The environment-dependent nucleus contains components that depend on the operating or TP system being used.

In addition to the environment-independent nucleus, every single task in which Natural runs requires an environment-dependent nucleus containing modules that perform actions specific to the operating or TP system. The environment-dependent nucleus assumes control from the operating or TP system at the start of a Natural session, loads the environment-independent nucleus and passes control to it.

Modules for Linking

The following modules must be linked to the environment-dependent nucleus:

- Environment-specific Natural interface modules
- Environment-specific work file and print file modules
- Environment-specific Natural parameter module (see also *Building a Natural Parameter Module* in the *Operations* documentation)
- Environment-dependent modules of Natural add-on products
- Adabas link routine (`ADALNK` or `ADAUSER`)
- Environment-dependent user-supplied modules defined as `CSTATIC` in the Natural parameter module. The Natural profile parameter `CSTATIC` is described in the *Parameter Reference* documentation.

When using System Maintenance Aid (SMA), the required modules are linked to the environment-dependent nucleus during the appropriate installation job/step. Modules that can optionally be linked are mentioned in the *Installation Procedure*.

The installation of the environment-dependent nucleus is described in *Link the Nucleus* in the *Installation Procedure*.

Modules for Static Linking

Both the Natural configuration module `NATCONFG` (described in the *Operations* documentation) and the Natural parameter module contain the Natural-supplied list of additional modules to be statically linked to the nucleus.

The Natural parameter module also contains the user-supplied list of additional modules to be statically linked to the nucleus as specified with the Natural profile parameter `CSTATIC`.

Each entry of these lists consists of a program name and a V-type address constant which must be resolved by linking the corresponding module to the Natural parameter module.

The Natural-supplied list provided with `NATCONFG` is used if the Natural parameter module is not linked to the environment-independent nucleus. If modules are statically linked to the environment-independent nucleus, a Natural parameter module that defines all these modules must also be linked to the environment-independent nucleus.

Optionally, you can specify an alternative Natural parameter module by using the Natural profile parameter `PARM` (described in the *Parameter Reference* documentation). An alternative parameter module takes precedence over a parameter module that is linked to either the environment-independent or the environment-dependent nucleus.

Merging Module Lists

During initialization of a Natural session, up to three lists of statically-linked modules (specified with the Natural profile parameter `CSTATIC`) are merged:

- Base list for the merge is the list of the Natural parameter module specified with the Natural profile parameter `PARM`;
- V-type address constants not resolved in this list are resolved using the Natural parameter module linked to the environment-dependent nucleus;
- V-type address constants not yet resolved are resolved using the Natural parameter module linked to the environment-independent nucleus.

If a user-supplied module is to be statically linked to the environment-independent nucleus, it must be specified in the Natural parameter module linked to the environment-independent nucleus as well as in the Natural parameter module specified with the Natural profile parameter `PARM`.

Modules for Dynamic Loading

When initializing a Natural session, you can also dynamically load the modules (supplied by Software AG or user-defined) that have been defined for static linking. For information on whether the module of a Natural add-on product is suitable for dynamic loading, read the documentation for your specific Natural add-on product.

For information on defining external names for static non-Natural programs and dynamic linking and controlling these programs, see the Natural profile parameters `RCA` and `RCALIAS` described in the *Parameter Reference* documentation.

Modules Called Dynamically

If a module is not defined for static linking, Natural attempts to load and execute the module using environment-dependent functions when the corresponding Natural `CALL` statement is executed.

Natural System Files

The Natural system files are stored in an Adabas database.

The table below lists and describes the Natural system files that are usually available in a Natural environment. The availability of the system files and the data contained in the files depends on the Software AG products installed in addition to base Natural.

The settings for the system files are defined with Natural profile parameters of the same names (exception: scratch-pad file). You can follow the hyperlinks in the table below to read details about these parameters in the *Parameter Reference* documentation.

System File	Supplied with	File Contents
`FNAT`	Base Natural	All objects required for Natural system applications.
`FUSER`	Base Natural	User-specific objects required for user-defined applications.
`FPROF`	Base Natural	Parameter profiles specified by the profile parameter `PROFILE`, provided no database information is supplied as subparameter of `PROFILE`.
Scratch-pad file	Base Natural	Data that is not stored explicitly as a Natural object in another system file. See also *Natural Scratch-Pad File* in the *Operations* documentation.
`FDIC`	Base Natural	Natural Data Definition Modules (DDMs). If Predict is installed, `FDIC` also contains data for the Predict dictionary system. If the Natural Development Server is installed, `FDIC` also contains application data and holds object locking information.

System File	Supplied with	File Contents
FREG	Base Natural	Registry data that is not stored explicitly in another system file.
FSEC	Natural Security	Control information required for security definitions.
FSPOOL	Natural Advanced Facilities	Control and spooling information required to output a report on a screen or printer and obtain print statistics.

Defining a Scratch-Pad File

Like all other system files of Software AG products, the scratch-pad file is a logical file. The logical file number of the scratch-pad file is 212.

Since there is no mnemonic for the scratch-pad file such as `FNAT` and `FUSER` or `FDIC`, it has to be defined:

- either statically by using the macro `NTLFILE` in the Natural parameter module or
- dynamically by using the Natural profile parameter `LFILE`.

Examples of `NTLFILE` and `LFILE` Definitions:

`LFILE` Parameter:

```
LFILE=(212,physical-dbid,physical-fnr,password,cipher-key)
```

`NTLFILE` Macro:

```
NTLFILE 212,physical-dbid,physical-fnr,password,cipher-key
```

2 Installing Natural on BS2000

This document describes the steps for installing Natural (product code NAT) on BS2000.

Related Topic:

For information on how to run Natural in a BS2000 environment, see the relevant section in the *Operations* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support***.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
ICS*vrs*.MOD	Load modules and object modules for International Components for Unicode for Software AG (ICS)
MLC*vrs*.JOBS	Sample installation jobs for Software AG's mainframe license check software The *vrs* in the library name represents the version of the license check software, which is not necessarily the same as the version of Natural. For detailed information on the license check software, see *Software AG Mainframe Product Licensing*.
MLC*vrs*.MOD	Load modules and object modules for Software AG's mainframe license check software including the LICUTIL license utility The *vrs* in the library name represents the version of the license check software, which is not necessarily the same as the version of Natural. For detailed information on the license check software and the LICUTIL utility, see *Software AG Mainframe Product Licensing*.
NAT*vrs*.LICS	Product license file For information on the license file and product licensing, see *Software AG Mainframe Product Licensing*.
NAT*vrs*.SRC	Source modules

File Name	Contents
`NAT`*vrs*`.SYSF`	Natural system file definitions
`NAT`*vrs*`.MOD`	Load modules and object modules
`NAT`*vrs*`.MAC`	Macros
`NAT`*vrs*`.JOBS`	Sample installation jobs
`NAT`*vrs*`.INPL`	Natural objects
`NAT`*vrs*`.EXPL`	Natural example objects

Note: The file `NAT`*vrs*`.JOBS` will be converted automatically into an LMS library when it is copied from the installation medium to disk.

Copying Files to a BS2000 Disk

Copy the files (data sets) from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the files depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the files from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the files from tape, follow the instructions in this section.

This section explains how to copy all files from tape to disk.

- Step 1: Copy Library SRVvrs.LIB from Tape to Disk
- Step 2: Copy the Procedure COPY.PROC from Tape to Disk
- Step 3: Copy all Product Files from Tape to Disk

Step 1: Copy Library SRVvrs.LIB from Tape to Disk

This step is not necessary if you have already copied the library `SRV`*vrs*`.LIB` from another Software AG installation tape. For further information, refer to the element `#READ-ME` in this library. The library `SRV`*vrs*`.LIB` is stored on the tape as a sequential file named `SRV`*vrs*`.LIBS` containing LMS commands. The current version *vrs* can be obtained from the *Software AG Product Delivery Report*.

- Execute the following commands to convert `SRV`*vrs*`.LIBS` into an LMS library:

```
/IMPORT-FILE  SUPPORT=*TAPE(FILE-NAME=SRVvrs.LIBS,-
/   VOLUME=volser, DEV-TYPE=tape-device)
/ADD-FILE-LINK  LINK-NAME=EDTSAM, FILE-NAME=SRVvrs.LIBS,-
/   SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT,-
/   BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ  '/'
@SYSTEM 'REMOVE-FILE-LINK  EDTSAM'
@SYSTEM 'EXPORT-FILE  FILE-NAME=SRVvrs.LIBS'
@WRITE  'SRVvrs.LIBS'
@HALT
/ASS-SYSDTA  SRVvrs.LIBS
/MOD-JOB-SW  ON=1
/START-PROG  $LMS
/MOD-JOB-SW  OFF=1
/ASS-SYSDTA  *PRIMARY
```

where:

`tape-device` is the device type of the tape, for example, `TAPE-C4`, and
`volser` is the VOLSER of the tape (see the *Software AG Product Delivery Report*).

Step 2: Copy the Procedure COPY.PROC from Tape to Disk

- Call the procedure `P.COPYTAPE` in the library `SRVvrs.LIB` to copy the procedure `COPY.PROC` to disk:

```
/CALL-PROCEDURE  (SRVvrs.LIB,P.COPYTAPE), -
/   (VSNT=volser, DEVT=tape-device)
```

If you use a TAPE-C4 device, you can omit the parameter `DEVT`.

Step 3: Copy all Product Files from Tape to Disk

- Enter the procedure `COPY.PROC` to copy all product files to disk:

```
/ENTER-PROCEDURE  COPY.PROC, DEVT=tape-device
```

If you use a TAPE-C4 device, you can omit the parameter `DEVT`.

The result of this procedure is written to the file `L.REPORT.SRV`.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** and ***Naming Conventions for Installation Procedures*** before you start the installation procedure.

Step 1: Prepare, Convert, Assemble and Link the License File

(Job I010, Steps 0110, 0111, 0112)

You must install a valid Natural license file.

For detailed information on the license file and product licensing, see *Software AG Mainframe Product Licensing*.

1. Copy the license file from the supplied installation medium to disk or transfer it from the PC as described in *Transferring a License File from PC to a BS2000 Host Using FTP* in *Software AG Mainframe Product Licensing*.
2. Check, convert, assemble and link the license file:

Step 0110	Check license file NAT*vrs*.LICS. This job runs the CHECK function of the LICUTIL license utility (see below).
Step 0111	Convert license file into an assembler source. This job runs the MAKE function of the LICUTIL license utility (see below).
Step 0112	Assemble and link the assembler source to generate load module ANATLIC. This module is then linked to the nucleus in **Job I060**.

The functions and option settings provided by LICUTIL are described in *Using the License Utility: LICUTIL* in *Software AG Mainframe Product Licensing*.

Step 2: Load the FNAT System File Definition

(Job I050, Step 0100)

Skip this step if you want to use an existing Natural `FNAT` system file.

Load the new Natural `FNAT` system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new `FNAT` system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 9: Build the Natural Parameter Module***.

2. Load the `FNAT` system file definition contained in the `NAT`*vrs*`.SYSF` file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

   The following ADALOD utility parameter setting is recommended:

   ```
   USERISN=YES
   ```

   This avoids the Natural errors NAT9988 and NAT7397 after reorganizing the `FNAT` system file using the Adabas utility ADAULD or ADALOD.

Step 3: Load the FUSER System File Definition

(Job I050, Step 0101)

Skip this step if you want to use an existing Natural `FUSER` system file.

Load the new Natural `FUSER` system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new `FUSER` system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 9: Build the Natural Parameter Module***.

2. Load the `FUSER` system file definition contained in the `NAT`*vrs*`.SYSF` file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

```
ISNREUSE=YES
```

3. If you want to use existing Natural applications, copy all user-written objects to the empty `FUSER`.
4. If you want to use Natural Application Programming Interfaces (APIs), see *Using a Natural API* in the *SYSEXT Utility* documentation for further guidance.

Step 4: Load the Scratch-Pad File Definition

This step is only required if you want to use read-only system files. See also *Natural Scratch-Pad File* in the *Operations* documentation.

You can skip this step if you want to use an existing Natural scratch-pad file.

This step is not supported by System Maintenance Aid (SMA).

Load the new Natural scratch-pad system file definition:

1. Specify the database ID and file number of the Adabas file where to load the scratch-pad file by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 9: Build the Natural Parameter Module***.
2. Load the scratch-pad system file definition contained in the `NAT`*`vrs`*`.SYSF` file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

Step 5: Load the FREG System File Definition

(Job I050, Step 0104)

This step is only required if registry information must be available to control concurrent user sessions limited with the Natural profile parameter `UCONMAX` (see the *Parameter Reference* documentation).

You can skip this step if you want to use an existing Natural `FREG` system file.

This step is not supported by System Maintenance Aid (SMA).

Load the new Natural `FREG` system file definition:

1. Specify the database ID and file number of the Adabas file where to load the `FREG` system file by using the Adabas ADALOD utility.

In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 9: Build the Natural Parameter Module***.

2. Load the FREG system file definition contained in the NAT*vrs*.SYSF file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

Step 6: Load the FDIC System File Definition

(Job I050, Step 0103)

Skip this step:

- if you want to install Predict. In this case, use the corresponding installation step described in the Predict *Installation* documentation.
- if you want to use an existing Natural FDIC system file.

Load the new Natural FDIC system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new FDIC system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 9: Build the Natural Parameter Module***.
2. Load the FDIC system file definition contained in the NAT*vrs*.SYSF file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

Step 7: Load the FSEC System File Definition

(Job I050, Step 9900)

Skip this step, if you do not use Natural Security.

- If you use Natural Security, refer to ***Installing Natural Security***.

Step 8: Assemble the Natural BS2000 Stub Module and Batch Driver

(Job I055, Steps 0216 - 0218)

- Assemble the following source modules:

`ANATSTUB`	Natural BS2000 stub module
`ANATFRNT`	Natural BS2000 non-reentrant part of the Natural batch interface
`ANATRENT`	Natural BS2000 reentrant part of the Natural batch interface

These source modules are contained in the library *`job-library`*.

Step 9: Build the Natural Parameter Module

(Job I060, Step 0010)

Build the Natural parameter module for batch mode.

1. Modify the settings of the Natural profile parameters supplied with this job, if required. The parameters and corresponding macros (if applicable) are described in the *Parameter Reference* documentation. The most important parameter/macro settings are described below.

   - Adapt the following parameters:

     ```
     FNAT=(database-id,file-number)
     FUSER=(database-id,file-number)
     FDIC=(database-id,file-number)
     ```

     where *`database-id`* and *`file-number`* are either the database ID and file number you specified when loading the new `FNAT`, `FUSER` and `FSEC` system files (see ***Step 2***, ***Step 3*** and ***Step 6***, respectively), or the database ID and file number of your existing Natural system files.

     These parameters are supplied with the `NTPRM` macro described in the *Operations* documentation.

   - If you want to to limit the number of concurrent users with the Natural profile parameter `UCONMAX`, proceed as follows:

     Supply the following parameter with the `NTPRM` macro:

     ```
     FREG=(database-id,file-number)
     ```

     where *`database-id`* and *`file-number`* are either the database ID and file number you specified when loading the new Natural `FREG` system file (see ***Step 5***), or the database ID and file number of your existing Natural `FREG` system file.

   - If you want to use read-only system files, proceed as follows:

Supply the following parameter with the NTPRM macro:

```
ROSY=ON
```

Specify the NTLFILE macro (see the parameter LFILE):

```
NTLFILE 212,database-id,file-number
```

where *database-id* and *file-number* are the database ID and file number you specified when loading the new Natural scratch-pad file (see ***Step 4***), or the database ID and file number of your existing Natural scratch-pad file.

2. Assemble the Natural parameter module.

Step 10: Link and Start the Optimize Monitor Buffer Pool

(Job I060, Step 1230)

This step is only required if you want to use the Optimize Monitor Buffer Pool described in *Optimize Monitor Buffer Pool* in the *Operations* documentation.

1. Link the NAT04182 program for the Optimize Monitor Buffer Pool.
2. Create and start the job E.START.04IBP (supplied with Step 1230) before using Natural. See *Starting the Optimize Monitor Buffer Pool* in the *Operations* documentation.

Step 11: Link the Nucleus

(Job I060, Steps 3801, 3802)

1. Link the **environment-dependent nucleus** LNATFRNT (Step 3801) for batch Natural.

   The list of modules to be linked for the environment-dependent nucleus is supplied with Step 3801.

   Note:

   - If the parameter ILCS=CRTE is specified for the assembly of the reentrant part of the BS2000 batch driver, the BINDER statements for the masking of symbols have to be specified as follows:

```
INCLUDE-MODULES ELEMENT=(ITOSL#),LIB=$TSOS.SYSLNK.CRTE
RESOLVE-BY-AUTOLINK LIB=($TSOS.SYSLNK.CRTE)
MODIFY-SYMBOL-VISIBILITY                        -
          SYMBOL-NAME = *ALL,                   -
          SYMBOL-TYPE = *DEFINITIONS,           -
          VISIBLE = *YES
MODIFY-SYMBOL-VISIBILITY                        -
          SYMBOL-NAME = CMPRMTB,                -
          SYMBOL-TYPE = *DEFINITIONS,          -
          VISIBLE = *NO(KEEP-RES=*YES)
```

`VISIBLE = *YES`: Necessary to prevent duplicate loading of the CRTE routines.

`VISIBLE = *NO(KEEP-RES=*YES)`: The entry `CMPRMTB` in the Natural parameter module has to be masked if the Natural parameter module is bound to the environment-dependent nucleus.

2. Link the **environment-independent nucleus** `LNATSHAR` (Step 3802).

   The list of modules to be linked for the environment-independent nucleus is supplied with Step 3802.

   Ensure that the Natural profile parameter `NUCNAME` (see the *Parameter Reference* documentation) specified in the Natural parameter module contains the name of the module resulting from this link step.

Using a Sort Program

If you want to use a sort program (either Natural's internal one or an external one), include the module `NAT2SORT`.

It is also possible to place `NAT2SORT` in a load library from where it can be loaded dynamically at runtime; this requires that `NAT2SORT` is specified with the Natural profile parameter `RCA` (described in the *Parameter Reference* documentation).

Step 12: Start All Pools

(Job I061, Step 0001)

- Start the job `E.START.ALL` which is contained in the library *`job-library`*.

  To end all pools, start the job `E.END.ALL` which is contained in the library *`job-library`*.

Step 13: Load New Natural Objects and Natural Error Messages

(Job I061, Step 0100)

- Load the Natural objects and Natural error messages from the `NAT`*`vrs`*`.INPL` file into the Natural system files by using the Natural **INPL utility**.

  The Natural error messages comprise short and long message texts and the German (`ULANG=2`) short message texts. You can use the `ERRUPPER` program of the Natural SYSERR utility to convert the message texts to upper case.

  For details on the `ULANG` profile parameter and `ERRUPPER`, see the *Parameter Reference* and the *Utilities* documentation, respectively.

Step 14: Load the Natural Example Objects

(Job I061, Step 0103)

- Load the Natural example objects from the NAT*vrs*.EXPL file into the Natural system file by using the Natural INPL utility.

Step 15: Allocate and Format the Editor Work File

(Job I081, Step 1900)

1. Allocate the editor work file (see also *Editor Work File* in the *Operations* documentation) by issuing the command:

   ```
   /FILE file-name,LINK=CMEDIT,SPACE=...
   ```

2. Execute the formatting program (NATEDFM2, NAT*vrs*.MOD) by entering FORMAT to format and load the control record in the editor work file.

Installation Verification

This section provides instructions for verifying the successful installation of Natural.

- Test Batch Natural
- Start Natural
- Test Online Natural

Test Batch Natural

You can use the following procedure to invoke Natural in batch mode and check whether the Natural system files are available:

```
/   SET-PROCEDURE-OPTIONS         DATA-ESCAPE-CHAR = *STD
/        ADD-FILE-LINK  DDLIB,    ADALNK.LIB
/        ADD-FILE-LINK  DDLNKPAR, ADALNK.PARMS
/        ADD-FILE-LINK  CMEDIT,   EDTWORK
/   START-PROGRAM  NATBATCH
EDIT
WRITE 'TESTBAT'
END
.END
SAVE TESTBAT
RUN
FIN
/   EXIT-PROCEDURE
```

This procedure starts Natural (using the Adabas link routine ADALNK), creates the example program TESTBAT, and executes the program with the system command RUN.

Start Natural

Start a Natural session according to the procedure that applies to your TP monitor interface; see also *Installation Verification* described for the **Natural TIAM Interface** and the **Natural *open*UTM Interface**.

Test Online Natural

You can use the following verification procedure to test Natural system functions in online mode and check whether the Natural system files are available.

1. Log on to the Natural user library SYSTEM:

   ```
   LOGON SYSTEM
   ```

2. Enter the following Natural system command:

   ```
   MAINMENU
   ```

3. Select **Development Functions** and enter the following:

   C in the **Code** field,
   P in the **Type** field, and
   TEST in the **Name** field.
4. In the editing area of the program editor, type the following:

   ```
   WRITE 'HELLO'
   END
   ```

5. Save the source code and exit the program editor.
6. In the **Development Functions** menu, enter the following:

   L in the **Code** field and
   TES* in the **Name** field.
7. On the **LIST Objects in a Library** screen, enter the RU line command for the TEST program.
8. Enter the following Natural system command:

   ```
   SYSDDM
   ```

9. In the SYSDDM utility menu, enter the following:

   R in the **Code** field and
   EMPLOYEES in the **DDM Name** field.

10. After pressing ENTER, enter the following:

    C in the **Code** field,
    the appropriate Natural system file number in the **FNR** field,
    the appropriate database ID in the **DBID** field, and
    Y in the **Replace** field.

    The specified DDM has been adapted to your environment.

11. Repeat Steps 9 and 10 for the VEHICLES DDM.

12. You can check whether the DDMs EMPLOYEES and VEHICLES are now available in your environment by logging on to the Natural system library SYSEXSYN and executing the example programs (for example, AEDEX1R) with the system command RUN.

3 Installing International Components for Unicode for Software AG on BS2000

This document describes the steps for installing International Components for Unicode for Software AG (ICS) on BS2000 which allows Natural to convert code pages and support Unicode.

International Components for Unicode for Software AG (ICS) requires the use of an ICS module and an ICU data library. In addition, you can use ICU data items to load ICU components that are not contained in the ICU data library. The ICS module does not have to be linked to the Natural nucleus if neither code page conversion nor Unicode support are required.

The use of ICU functionality increases the required Natural thread size.

Note: For increased flexibility, it is also possible to load the ICS module during initialization of the Natural session. A dynamically loaded ICS module overrides the statically linked ICS module. You can also load an ICU data library during initialization of the Natural session. A dynamically loaded ICU data library overrides any statically linked ICU data library.

The ICS module, the ICU data libraries and the ICU data items are explained in *Enabling Unicode and Code Page Support* in the *Unicode and Code Page Support* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The ICS module, the ICU data libraries and the ICU data items are contained on the ICS*vrs*.MOD file supplied on the **installation medium** for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Link the ICS Module
- Step 2: Link an ICU Data Library
- Step 3: Load the ICS Module at Session Start
- Step 4: Load an ICU Data Library at Session Start

- Step 5: Load ICU Data Items on Request in a Session

Step 1: Link the ICS Module

1. Add the following `INCLUDE` statement to the link instructions for the **environment-independent nucleus**:

   ```
   INCLUDE-MODULES LIB=ICSvrs.MOD,ELEMENT=(SAGICU),TYPE=(R,L)
   ```

   Note: The ICU data library `ICSDT58J` is contained in the ICS module `SAGICU` and loaded by default.

2. Link the ICS module to the environment-independent nucleus as described in *Link the Nucleus* in *Installing Natural*.

Step 2: Link an ICU Data Library

This step is only required if you want to use another data library in addition to `ICSDT58J`.

1. Add one of the following `INCLUDE` statements to the `BINDER` instructions for the **environment-independent nucleus** depending on the ICU data library to be used:

   ```
   INCLUDE-MODULES LIB=ICSvrs.MOD,ELEMENT=(ICSDT58E),TYPE=(R,L)
   ```

2. Link the ICU data library to the **environment-independent nucleus** as described in *Link the Nucleus* in *Installing Natural*.

   Note: Since the size of the resulting module will exceed 16 MB, the environment-independent nucleus has to be linked with the linkage editor `BINDER` if ICU data libraries are to be included.

Step 3: Load the ICS Module at Session Start

1. Make sure that the ICS load library from the `ICSvrs.MOD` file is available to the execution JCL of your Natural or TP monitor interface. Depending on your environment, perform one of the following options:

   - Add `ICSvrs.MOD` with the link name `BLSLIBnn` as an alternative user library to your execution JCL.

2. At the start of a Natural session, set the Natural profile parameter `RCA` as follows:

```
RCA=SAGICU
```

RCA is described in the *Parameter Reference* documentation.

The ICS module SAGICU is described in the *Unicode and Code Page Support* documentation.

Step 4: Load an ICU Data Library at Session Start

1. Make sure that the ICS load library from the ICS*vrs*.MOD file is available to the execution JCL of your Natural or TP monitor interface. Depending on your environment, perform one of the following options:
   - Add ICS*vrs*.MOD with the link name BLSLIB*nn* as an alternative user library to your execution JCL.
2. At the start of a Natural session, set the Natural profile parameters RCA and CFICU for the ICU data library to be used:

   ```
   RCA=ICSDT58E  CFICU=(DATFILE=ICSDT586E)
   ```

   RCA and CFICU are described in the *Parameter Reference* documentation.

Step 5: Load ICU Data Items on Request in a Session

This step is optional and only required if you want to dynamically load an ICU data item on request during a Natural session instead of an entire ICU data library. For more information, see the *Unicode and Code Page Support* documentation.

- Make sure that the ICS load library from the ICS*vrs*.MOD file is available to the execution JCL of your Natural or TP monitor interface. Depending on your environment, perform one of the following options:
  - Add ICS*vrs*.MOD with the link name BLSLIB*nn* as an alternative user library to your execution JCL.

Installation Verification

After the last step of the installation procedure has been completed, proceed as follows:

1. Configure and activate your Unicode and code page environment by following the instructions in *Configuration and Administration of the Unicode/Code Page Environment* and *Profile Parameters and Macros* in the *Unicode and Code Page Support* documentation.

   For information on the code pages and ICU data files available in your current Natural environment, you can use the SYSCP utility (described in the *Utilities* documentation).

2. After successful activation, you can execute the example programs described in the *Unicode and Code Page Support* documentation.

4 Installation for REQUEST DOCUMENT and PARSE XML Statements on BS2000

This document describes the installation steps for enabling the use of the Natural statements `REQUEST DOCUMENT` and `PARSE XML` on BS2000.

Related Topics:

For information on the functions provided by `REQUEST DOCUMENT` and `PARSE XML`, see the following documents:

- *Statements for Internet and XML Access* in the *Programming Guide*
- `REQUEST DOCUMENT` in the *Statements* documentation
- `PARSE XML` in the *Statements* documentation

Prerequisites

The following requirements must be met to execute the `REQUEST DOCUMENT` and `PARSE XML` statements:

- The ICU library must be installed to convert data from one encoding to another (at least internally). For details, see the relevant section in the *Unicode and Code Page Support* documentation.
- A TCP/IP stack must be available and enabled for the execution environment.
- A DNS (Domain Name System) server or DNS services must be available in the execution environment to resolve internet addresses (`gethostbyname` function).

For Internet Protocol Version 6 (IPv6) support, the following additional prerequisites apply:

- An activated IPv6 stack must be available on the local host.
- The local network must support IPv6.
- An accessible and IPv6-capable DNS server must be available.
- For IPv6 internet communication, an IPv6 connection from the service provider must be available.
- If both IPv4 and IPv6 are used, a dual stack must be supported.
- IPv6 support must be configured with the appropriate keyword subparameters of the Natural profile parameter `XML` described in the *Parameter Reference* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Procedure

Be sure to read *Installation Process and Major Natural Features* before you start the installation procedure.

The installation procedure comprises the following:

- Step 1: Link the Module NATXML to the Nucleus
- Step 2: Enable the Environment-Dependent Nucleus for CRTE Execution
- Step 3: Link the Required Modules to the Nucleus

Step 1: Link the Module NATXML to the Nucleus

The module NATXML is required to execute the statements REQUEST DOCUMENT and PARSE XML.

1. The NATXML module is then linked to the **environment-independent nucleus** by using the following INCLUDE statement:

   ```
   INCLUDE NATLIB(NATXML)
   ```

Step 2: Enable the Environment-Dependent Nucleus for CRTE Execution

The REQUEST DOCUMENT and PARSE XML statements require CRTE (Common RunTime Environment) functions for execution.

1. Set the parameter ILCS=CRTE for the **environment-dependent nucleus**.
2. For the assembly step of the reentrant part of the **environment-dependent nucleus**, set the parameter ILCS=CRTE in the following macros:

   NAMBS2 for batch Natural,

   NAMTIAM for TIAM and

   NURENT for *open*UTM.

This applies to all batch and TP monitor system environments.

Step 3: Link the Required Modules to the Nucleus

Link the modules indicated in this section to the nucleus.

Batch, TIAM and openUTM

1. Add the statement RESOLVE-BY-AUTOLINK LIB=($TSOS.SYSLNK.CRTE) to the link step for the **environment-dependent nucleus**.
2. Link the following modules to the environment-dependent nucleus by using the appropriate INCLUDE statements:

INCLUDE NAT2LE	CRTE access module
INCLUDE NAT2TCP	TCP/IP access module Since NAT2TCP is delivered in LLM format, the front-end has to be linked with the linkage editor BINDER.
INCLUDE ITOSL#	CRTE initialization module

Natural Development Server/Natural Web I/O Interface

- Copy the LLM and the NCFIP4*vr* to the load libraries of the Natural Development Server and the Natural Web I/O Interface.

Installation Verification

After the last step of the installation procedure has been completed, proceed as follows:

1. Activate the statements in the runtime environment; see *Activation/Deactivation* in the section *Statements for Internet and XML Access* in the *Programming Guide*.

   For information on the profile settings that enable the support of the REQUEST DOCUMENT and/or PARSE XML statement, see the following documents:

   - *Profile Settings* in the section *Statements for Internet and XML Access* in the *Programming Guide*
   - Profile parameter XML in the *Parameter Reference* documentation
2. Try the example programs contained in the Natural system library SYSEXV.

5 Installing Natural Net Data Interface on BS2000

This document describes the steps for installing the Natural Net Data Interface NATNETTO on BS2000.

Related Topic:

For operational information, see *Natural Net Data Interface NATNETTO* in the *Operations* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Device Configuration in NATCONFG

Logical net data devices are configured with the IONET flag byte, all other flag and value settings must be made as in the sample definition given below. The module entry is VCNETTO, additional device entries must be specified with WXTRN=OFF. The settings of FLAG1, FLAG2 and RTAL must be according to the example below.

If the data delimited option (NEDLM) is set, the delimiter character which separates the fields in the value buffer can be set by using the BEL keyword subparameter of the NTDVCE macro.

IONET Settings

IONET	DS	XL1	NETDATA CONTROL FLAG
NECUFNR	EQU	X'01' 1	CURSOR POSITION = FIELDNR
NEMSG	EQU	X'02' .... ..1.	SEND MESSAGE LINE (if not set, message line will be skipped)
NEABO	EQU	X'04' .... .1..	ATTRIBUTE BUFFER OPTION

NEFBO	EQU	X'08' .... 1...	FORMAT BUFFER OPTION
NEFLG	EQU	X'10' ...1	FIELD LENGTH OPTION
NEDLM	EQU	X'20' ..1.	DATA DELIMITED OPTION
NEFIX	EQU	X'40' .1..	FIXED FORMAT OPTION
NEFBOPTE	EQU	X'80' 1...	EXTENDED FORMAT BUFF. OPT.

NATCONFG already has a device entry for NATNETTO.

The TTYPE is NETF. The protocol options are set as follows:

- The value buffer structure is fixed (without delimitation between the fields).
- Format buffer, extended format buffer and attribute buffer options are set.
- The cursor position is in the field number notation.

Message line and PF-key line are suppressed.

Example:

```
NTDVCE TYP=NETF,NAME=NETTF,ENTRY=VCNETTO,MSG=BOT,           HS06-
       FLAG1=CMNIXD,FLAG2=CMTNOPT,RTAL=255,                 HS07-
       FLAGS=(IONET,-,CO,IONET,+,NEFIX+NEFBO+NEABO+NECUFNR+NEFB-
       OPTE,WINDTITI,+,PFKNDISP)                            HS06
```

6 Installing Entire System Server Interface on BS2000

The Entire System Server Interface is required if the Entire System Server or Natural ISPF is to be used.

This document describes the steps for installing the Entire System Server Interface on BS2000. You can choose between default installation (recommended) and customized installation.

Related Topic:

For information on installing and using the Entire System Server or Natural ISPF, refer to the relevant product documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of either of the following products must be installed before you can install the Entire System Server Interface:

- Entire System Server or Natural ISPF, version as specified under *Software AG Product Versions Required with Natural* in the current Natural *Release Notes* for Mainframes.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Default or Customized Installation

If you want to use the default value settings in the modules ESYNODTB and NATPNIP (used by the Entire System Server and Natural ISPF), proceed with ***Link the Entire System Server Interface to the Nucleus***.

If you do *not* want to use the default value settings, edit the modules NATPNIP and ESYNODTB described in this section.

- NATPNIP

- ESYNODTB

NATPNIP

The NATPNIP module contains the following parameters and default values:

```
NAMVIEWP BUFLEN=12288,NUMREQ=5,MAXCBL=3000,MAXEDL=6000,EXTUSER=INIT-USER
```

The parameters are explained below:

BUFLEN	Length of all Adabas buffers in bytes
NUMREQ	Number of possible nested FIND loops in Natural calling the Entire System Server
MAXCBL	Complex FIND buffer length
MAXEDL	Editor session buffer length MAXEDL is used by the NSPF editor and incore database. The default value is 6000, which should be sufficient for an NSPF editor session and typical incore database applications. However, for large layouts within an incore database file that value might not be large enough and the following message is issued: NAT3077: Not enough space for extent. DB/FNR/Subcode :1:/:2:/:3:. (see the *Messages and Codes* documentation). In this case, the value of MAXEDL has to be increased.
EXTUSER	External user ID passed to the Entire System Server for security checks See also EXTUSER in the following section.

EXTUSER

The parameter EXTUSER describes how to inherit security definitions from an external security system such as RACF, ACF2 and TOP-SECRET.

The appropriate parameter setting depends on whether a multi-user address space or a single-user address space is used in your environment:

- A multi-user address space provides the option to maintain different user security definitions, for example, one for a CICS and one for a Com-plete user).
- A single-user address space supports a subsystem, for example, a Natural subtask for Entire Output Management or Entire Operations Managment, or a Natural RPC, Natural Web/IO Interface or batch server.

  Recommended values for EXTUSER are:

EXTUSER=INIT-USER	Recommended for a multi-user address space. The contents of the Natural system variable *INIT-USER must be identical to the user definiton in the external security system (for example, RACF). The Entire System Server transfers the value of *INIT-USER to the external security system, and all calls to security restricted resources are handled under this user ID. In this case, the security definition from a CICS or Com-plete user (for example) is inherited by Entire System Server, and a new logon is not required. If the value of *INIT-USER is not found in RACF (for example), an error occurs indicating that a logon is required. (*INIT-USER is described in the *System Variables* documentation.)
EXTUSER=USER	Recommended for *open*UTM. Recommended for a multi-user address space in a Natural Security environment. Processing is similar to EXTUSER=INIT-USER except that the Natural system variable *USER (described in the *System Variables* documentation) is used. (*USER is described in the *System Variables* documentation.)
EXTUSER=ADDRESS-SPACE	Recommended for a TIAM, batch or server environment. The security description of this address space is inherited for security evaluation.

ESYNODTB

The ESYNODTB module contains the following parameters and default values:

```
NAMXNOD ID=148,NAME=PRODUCTION-1
NAMXNOD ID=149,NAME=PRODUCTION-2
NAMXNOD ID=1490,NAME=DBID-ABOVE-255,LAST=Y
END
```

The parameters and default values are explained below:

ID	Entire System Server node number (also known as DBID)
NAME	Entire System Server node name
LAST	Indicator for last entry in table

Optional Node Name for Entire System Server Calls

Calls to the Entire System Server from Natural are usually handled with the NODE parameter which specifies the node number to be used for the call, for example:

```
FIND ACTIVE-JOBS WITH JOB-NAME = 'ADA*' AND NODE = 148
```

If the node number is defined in the ESYNODTB module, alternatively, you can specify the logical name of the required Entire System Server with the NODE-NAME parameter, for example:

```
FIND ACTIVE-JOBS WITH JOB-NAME = 'ADA*' AND NODE-NAME = 'PRODUCTION-2'
```

Assemble the Parameter Module for the Entire System Server Interface Component

Natural ISPF

If Natural ISPF is used as the INCORE database:

- (Job I055, Steps 1106, 1108)

  Link the parameter module NATPNIU (Step 1106 for *open*UTM) or NATPNIT (Step 1108 for TIAM and batch mode). In this case, the module ESYNODTB is not required.

Entire System Server

If the Entire System Server is used:

- **openUTM Environments:**

  (Job I055, Step 1108)

  Assemble and link the module ANATPNIU.

  ANATPNIU must be assembled with NAMVIEWP EXTUSER=USER in order to use the Natural Security user ID as user ID for Entire System Server calls (if no NATPROC-LOGON is issued).

- **TIAM and Batch Environments:**

  (Job I055, Step 1106)

  Assemble and link the module ANATPNIT.

  NATPNIT must be assembled with NAMVIEWP EXTUSER=ADDRESS-SPACE. The LOGON user ID will be used as Entire System Server user ID if no NATPROC-LOGON has been issued.

Link the Entire System Server Interface to the Nucleus

(Job I060, Step 3720)

- Link the following Entire System Server Interface modules to the **environment-dependent nucleus** by using the corresponding `INCLUDE` statements:

`INCLUDE NATPNIP, NATvrs.MOD`	Entire System Server Interface parameters
`INCLUDE ESXNUC, NATvrs.MOD`	Entire System Server Interface module
`INCLUDE NATPRBSU, NATvrs.MOD`	BS2000 service module
`INCLUDE ESYNODTB, NATvrs.MOD`	Optional, node table

Install the Entire System Server in Single-User Mode

(Optional installation for only batch mode or TIAM.)

This section describes the advantages of Entire System Server in single-user mode and the steps required for installation.

- Advantages of Single-User Mode
- Installation for Batch Mode and TIAM

Advantages of Single-User Mode

Running the Entire System Server in single-user mode is advantageous, for example, in the following cases:

- Executing long running batch jobs comprising a large number of calls to the Entire System Server.
- Performing test scenarios using a Natural session under TIAM performing many calls to the Entire System Server, without disturbing the production environment.
- Exploring new Entire System Server functionality or versions.

From a Natural point of view, the Entire System Server single-user mode is accessible as Entire System Server node 148, irrespective of whether such a node does already exist on your machine or network.

The following is an example of a Natural program that is running in a single-user environment:

```
FIND ACTIVE-JOBS WITH NODE = 148 AND JOB-NAME = 'XCOM*'
```

This statement calls a single-user Entire System Server that runs within the same address-space. Calling a different Entire System Server node that runs elsewhere in the network is possible by using a different node number, as shown in the following example:

```
FIND ACTIVE-JOBS WITH NODE = 53 AND JOB-NAME = 'NUC*'
```

This statement calls a multi-user Entire System Server with node number 53 out of the same Natural program which called the single-user Entire System Server session.

Installation for Batch Mode and TIAM

1. Link the optional `NATPSNGL` module to the **environment-dependent nucleus**:

   ```
   INCLUDE NATPSNGL, NATvrs.MOD
   ```

   This module enables use of single-user mode.

2. Add the mandatory `DDLIB2` card to assign the Entire System Server load library and the optional `PARMS` card to set the parameters for the Entire System Server:

   ```
   / ADD-FILE-LINK LINK-NAME=DDLIB2,FILE-NAME=$SAG.NPRvrs.MOD
   / ADD-FILE-LINK LINK-NAME=PARMS,FILE-NAME=parameter-file
   ```

   where *`parameter-file`* is the name of the Entire System Server parameter file.

   The Entire System Server parameter file is read during the execution of the first Entire System Server user request. You do not have to issue a `LOGON` or `LOGOFF` to the Entire System Server in single-user mode.

3. Modify your Entire System Server parameter file as required. The `NODE` parameter is ignored since all calls to the Natural node number 148 are routed through the Entire System Server node used for single-user mode.

7 Installing Software AG Editor on BS2000

The Software AG Editor is an optional Natural component that is required by several Natural utilities (for example, SYSRPC and SYSBPM), Natural add-on products (for example, Natural ISPF) and other Software AG products (for example, Predict).

When using SMA, the Software AG editor is automatically installed when installing Natural. If you are not using SMA, install the Software AG editor according to the procedure described in this document.

This document describes the steps for installing the Software AG Editor on BS2000.

Related Topics:

- For operational information, see *Operating the Software AG Editor* in the *Operations* documentation.
- For information on the features and functions of the *Software AG Editor*, see the relevant section in the *Editors* documentation.

Notation `vrs` or `vr`:

When used in this document, the notation `vrs` or `vr` represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Adapt the Natural Parameter Macro NTEDBP
- Step 2: Build the Natural Parameter Module
- Step 3: Allocate and Format the Editor Work File E.FRM.EDITWORK
- Step 4: Modify the Startup Procedures for the Nucleus
- Step 5: Start the Global Editor Buffer Pool

- Step 6: Link the Software AG Editor to the Nucleus

Step 1: Adapt the Natural Parameter Macro NTEDBP

The editor buffer pool parameters contained in Natural parameter module are stored in the editor buffer pool work-file control record during formatting (see ***Step 3: Allocate and Format the Editor Work File E.FRM.EDITWORK***). The Natural parameter module can be linked to the Software AG Editor work file formatting utility (described in *Editor Work File* in the *Operations* documentation).

Caution: Before adapting the editor parameter macro, first read the documentation of the editor buffer pool parameter `EDBP` and its corresponding macro `NTEDBP` in the *Parameter Reference* documentation.

- Assemble and link the Natural parameter module.

For the Initial Installation:

You can leave the defaults. In this case, it is not necessary to assemble and link the Natural parameter module.

Step 2: Build the Natural Parameter Module

(Jobs I060, I080)

1. Add the following parameter to your Natural parameter module to specify the size of the editor area:

   ```
   SSIZE=nn
   ```

   where *nn* must be set to at least 54.

2. Add the following macro to your Natural parameter module to specify the size of the editor buffer pool:

   ```
   NTBPI TYPE=EDIT
   ```

   The region size must be defined large enough to be able to allocate the buffer pool. For a detailed description of the `NTBPI` macro, refer to *Building a Natural Parameter Module* in the *Operations* documentation.

3. Assemble and link the Natural parameter module.

Step 3: Allocate and Format the Editor Work File E.FRM.EDITWORK

(Job I081, Step 1900)

1. Allocate the editor work file (see also *Editor Work File* in the *Operations* documentation) by issuing the command:

   ```
   /FILE file-name,LINK=CMEDIT,SPACE=...
   ```

2. Execute the Software AG editor work file formatting utility (`NATEDFM2`, `NATvrs.MOD`) by entering `FORMAT` to format and load the control record in the editor work file.

Step 4: Modify the Startup Procedures for the Nucleus

For TIAM and *open*UTM:

- Add a `FILE` command for the editor work file:

  ```
  /FILE file-name,LINK=CMEDIT
  ```

Step 5: Start the Global Editor Buffer Pool

(Job I061, Step 0001)

Once defined, a global editor buffer pool must be started.

- To start the global editor buffer pool, execute the module `CMPSTART`.

Step 6: Link the Software AG Editor to the Nucleus

(Job I080)

- Link the following Software AG Editor modules to either the **environment-independent** or the **environment-dependent nucleus** by using the corresponding `INCLUDE` statements:

`INCLUDE NATLIB(NATEDT)`	Software AG Editor module
`INCLUDE NATLIB(NATEDWA2)`	Software AG Editor work file access module

Installation Verification

You can verify the successful installation of the Software AG Editor by performing the following:

- Invoke Natural and enter the following system command:

  ```
  SYSEDT
  ```

  The SYSEDT Utility Main Menu appears, which can be used to display all buffer pool parameters and usage statistics; see *SYSEDT Utility - Editor Buffer Pool Administration* in the *Utilities* documentation.

You can only test the full operation of the Software AG Editor if another Software AG product (for example, Natural ISPF or Predict) that uses Software AG Editor functionality is installed.

8 Installing Natural TIAM Interface on BS2000

This document describes the steps for installing the Natural TIAM Interface (product code NRT) on BS2000.

Related Topics:

For information on the following topics, see *Natural under TIAM* in the *TP Monitor Interfaces* documentation:

- *Structure of the Natural TIAM Interface*
- *Parameters in Macro NAMTIAM*
- *Common Memory Pools under TIAM*

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
NRT*vrs*.MAC	Macros
NRT*vrs*.JOBS	Sample installation jobs

Copy the files into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** and ***Naming Conventions for Installation Procedures*** before you start the installation procedure.

Step 1: Build the Natural TIAM Interface Non-Reentrant Part

(Job I070, Step 0102)

Build the non-reentrant part of the Natural TIAM Interface.

1. Modify the parameters settings in the macro `NAMTIAM` if required. You can generally use the default values of the parameters. Modify only the values that do not suit your requirements.

   For descriptions of the individual parameters, see *Parameters in Macro NAMTIAM* in the Natural *TP Monitor Interfaces* documentation.
2. Assemble the source module `ANRTFRNT` in the library *`job-library`* to generate the macro `NAMTIAM` for the non-reentrant part.

Step 2: Build the Natural TIAM Interface Reentrant Part

(Job I070, Step 0103)

Build the reentrant part for the Natural TIAM Interface.

1. Modify the parameter settings in the macro `NAMTIAM` if required. You can generally use the default values of the parameters. Modify only the values that do not suit your requirements.

   For descriptions of the individual parameters, see *Parameters in Macro NAMTIAM* in the Natural *TP Monitor Interfaces* documentation.
2. Assemble the source module `ANRTRENT` in the library *`job-library`* to generate the macro `NAMTIAM`.

Step 3: Build the Natural Parameter Module

(Job I080, Step 0109)

Build the Natural parameter module for the Natural TIAM Interface.

1. Set the FNAT and FUSER profile parameters (see the *Parameter Reference* documentation) in the source module ANRTPARM of the Natural parameter module:

   ```
   FNAT=(database-id,file-number)
   FUSER=(database-id,file-number)
   ```

   where *database-id* and *file-number* are the values you specified when loading the system files during the ***Installation Procedure*** for base Natural.

2. Assemble the source module ANRTPARM in the library *job-library*.

Step 4: Link the Environment-Dependent Nucleus

(Job I080, Step 0110)

- Use the INCLUDE statements for BINDER contained in LNRTFRNT in the library *job-library*.

Notes:

1. The source module ANATSTUB has been assembled during the installation of base Natural.
2. If the parameter ILCS=CRTE is specified for the assembly of the reentrant part of the TIAM driver, the BINDER statements for the masking of symbols have to be specified as follows:

   ```
   INCLUDE-MODULES ELEMENT=(ITOSL#),LIB=$TSOS.SYSLNK.CRTE
   RESOLVE-BY-AUTOLINK LIB=($TSOS.SYSLNK.CRTE)
   MODIFY-SYMBOL-VISIBILITY                         -
             SYMBOL-NAME = *ALL,                    -
             SYMBOL-TYPE = *DEFINITIONS,            -
             VISIBLE = *YES
   MODIFY-SYMBOL-VISIBILITY                         -
             SYMBOL-NAME = CMPRMTB,                 -
             SYMBOL-TYPE = *DEFINITIONS,            -
             VISIBLE = *NO(KEEP-RES=*YES)
   ```

   VISIBLE = *YES: Necessary to prevent duplicate loading of the CRTE routines.

   VISIBLE = *NO(KEEP-RES=*YES): The entry CMPRMTB in the Natural parameter module has to be masked if the Natural parameter module is bound to the environment-dependent nucleus.

Installation Verification

You can verify the successful installation of the Natural TIAM Interface by performing the following steps:

1. Call the procedure `P.START.NRT` in the library *`job-library`* to start Natural under TIAM.
2. Proceed with the steps described in the section ***Test Online Natural***.

9 Installing Natural openUTM Interface on BS2000

This document describes the steps for installing the Natural *open*UTM Interface (product code NUT) on BS2000.

Related Topic:

For information on how to operate the Natural *open*UTM Interface, see *Natural under openUTM* in the *TP Monitor Interfaces* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install the Natural *open*UTM Interface:

- Adabas UTM Interface (product code AUT)

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
NUT*vrs*.SRC	Source modules
NUT*vrs*.MAC	Macros
NUT*vrs*.JOBS	Sample installation jobs

Copy the files into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** and ***Naming Conventions for Installation Procedures*** before you start the installation procedure.

Step 1: Generate the KDCROOT Source

(Job I070, Step 0110)

- Prepare the sample `KDCDEF` instruction source `NUT.KDCDEF` (LMS element type = D) in the library *`job-library`* so that it suits your environment.

  Therefore, refer to *Defining the openUTM Resources (KDCDEF)* in the *TP Monitor Interfaces* documentation.

Step 2: Assemble the BS2STUB Module for the Natural openUTM Interface

(Job I070, Step 0111)

- Assemble the source module `ANUTSTUB` in the library *`job-library`*.

  See also *Local Common Memory Pools* in the *Operations* documentation.

Step 3: Assemble the Non-Reentrant Part for the Natural openUTM Interface

(Job I070, Step 0112)

- Assemble the source module `ANUTFRNT` in the library *`job-library`*.

  See also *Keyword Parameters of Macro NATUTM* in the *TP Monitor Interfaces* documentation.

Step 4: Assemble the Reentrant Part for the Natural openUTM Interface

(Job I070, Step 0113)

- Assemble the source module `ANUTRENT` in the library *`job-library`*.

  See also *Keyword Parameters of Macro NURENT* in the *TP Monitor Interfaces* documentation.

Step 5: Generate the Swap Pool Parameter Module NTSWPRM

(Job I070, Step 0114)

- Assemble and link the module `ANUTSWP` in the library *`job-library`*.

  See also *Keyword Parameters of Macro NTSWPRM* in the section *Defining the Swap Pool* in the *Operations* documentation.

Step 6: Assemble KDCROOT

(Job I070, Step 0120)

- Assemble the sample source `ANUTKDCR` (generated in **Step 1**) in the library *`job-library`* for the `KDCROOT` module.

  See also *Generating KDCROOT* in the *TP Monitor Interfaces* documentation.

Step 7: Generate KDCFILE

(Job I075, Step 0100)

- Use the sample `KDCDEF` instruction source `NUT.KDCDEF` prepared in **Step 1**.

Step 8: Build the Natural Parameter Module

(Job I080, Step 0200)

Build the Natural parameter module for the Natural *open*UTM Interface.

1. Set the `FNAT` and `FUSER` profile parameters (see the *Parameter Reference* documentation) in the source module `ANUTPARM` of the Natural parameter module:

```
FNAT=(database-id,file-number)
FUSER=(database-id,file-number)
```

where *database-id* and *file-number* are the values you specified when loading the system files during the ***Installation Procedure*** for base Natural.

2. Assemble the source module ANUTPARM in the library *job-library*.

Step 9: Link the Environment-Dependent Nucleus

(Job I080, Step 0210)

- Use the INCLUDE statements for BINDER contained in LNUTFRNT in the library *job-library*.

  If the parameter ILCS=CRTE is specified for the assembly of NURENT, the BINDER statements for the masking of symbols have to be specified as follows:

```
INCLUDE-MODULES ELEMENT=(ITOSL#),LIB=$TSOS.SYSLNK.CRTE
RESOLVE-BY-AUTOLINK LIB=($TSOS.SYSLNK.CRTE)
MODIFY-SYMBOL-VISIBILITY                                    -
          SYMBOL-NAME = *ALL,                               -
          SYMBOL-TYPE = *DEFINITIONS,                       -
          VISIBLE = *YES
MODIFY-SYMBOL-VISIBILITY                                    -
          SYMBOL-NAME = (CMPRMTB,NU8*),                     -
          SYMBOL-TYPE = *DEFINITIONS,                       -
          VISIBLE = *NO(KEEP-RES=*YES)
```

  VISIBLE = *YES: Necessary to prevent duplicate loading of the CRTE routines.

  VISIBLE = *NO(KEEP-RES=*YES): The entry CMPRMTB in the Natural parameter module has to be masked if the Natural parameter module is bound to the environment-dependent nucleus.

Installation Verification

You can verify the successful installation of the Natural *open*UTM Interface by performing the following steps:

1. Submit the job E.START.SWP in the library *job-library* to start the Natural swap pool.
2. Submit the job E.START.NUT in the library *job-library*, which is a sample job for starting Natural under *open*UTM.
3. Enter the Natural transaction code.
4. Proceed with the steps described in the section ***Test Online Natural***.

10 Installing Natural Security on BS2000

This document describes the steps for installing Natural Security (product code NSC) on BS2000.

Related Topic:

For information on the features and functions provided by Natural Security, see the *Natural Security* documentation.

Notation `vrs` or `vr`:

When used in this document, the notation `vrs` or `vr` represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Additional Prerequisites for Natural Security in a Heterogeneous Environment

In addition to the prerequisites referred to above, the following software must be installed to use Natural Security in a heterogeneous environment:

- Entire Net-Work
- Natural Security for Mainframes

The following software must be installed as required:

- Natural Security for UNIX
- Natural Security for Windows
- Natural Security for OpenVMS

For further information, see *Using Natural Security on Multiple Platforms* in the *Natural Security* documentation.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
NSC*vrs*.SYSL	Natural Security log file
NSC*vrs*.INPL	Natural objects

Copy the files into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.JOBS file and are prefixed with the product code. The file is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Note for the Reinstallation:

- When you repeat a Natural Security installation, only the Natural objects are replaced; the Natural Security data defined for SYSSEC and DBA is *not* reset to the values defined after the initial installation. You can use the **Natural Security Recover** function of the Natural **INPL utility** to reset the data.

This section describes the actual installation steps:

- Step 1: Load the FSEC System File Definition
- Step 2: Load the Log File
- Step 3: Load the Logon and Error Log File
- Step 4: Build the Natural Parameter Module
- Step 5: Load New Natural Objects

Step 1: Load the FSEC System File Definition

(Job I050, Step 9900)

Skip this step if you want to use an existing Natural FSEC system file.

- Load the new Natural FSEC system file definition:

  1. Set the System Maintenance Aid (SMA) parameter NSC-FIRST-INSTALL to Y (Yes). This is the default setting.
  2. Specify the database ID and file number of the Adabas file where to load the FSEC system file by using the Adabas ADALOD utility.

In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 4: Build the Natural Parameter Module***.

3. Load the FSEC system file definition contained in the NAT*vrs*.SYSF file by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

Step 2: Load the Log File

(Job I050, Step 9901)

This step only applies if Adabas is installed and if the Natural Security function **Logging of maintenance functions** (see the *Natural Security* documentation) is to be used. It creates the log file to be used by the function.

- Load the log file by using the Adabas ADALOD utility. Input for ADALOD is the NSC*vrs*.SYSL file.

Step 3: Load the Logon and Error Log File

(Job I050, Step 9902)

This step only applies if Adabas is installed and if the Natural Security function **Store Logon and Error Data on Separate System Files** (see the *Natural Security* documentation) is to be used. It creates the logon and error log file to be used by the function.

- Load the logon and error log file by using the Adabas ADALOD utility. Input for ADALOD is the NAT*vrs*.SYSF file.

Step 4: Build the Natural Parameter Module

(Jobs I060, I080)

1. Specify the following profile parameter in your Natural parameter module:

   ```
   FSEC=(database-id,file-number)
   ```

   where *database-id* and *file-number* are the database ID and file number of either the new FSEC system file loaded in **Step 1** or your existing FSEC system file.

   The FSEC profile parameter is described in the *Parameter Reference* documentation.

   Repeat Job I080 for all your TP monitors.

2. Assemble and link your Natural parameter module.

Step 5: Load New Natural Objects

(Job I061, Step 0102 or Step 9900)

For the migration installation (Step 0102):

1. Set the System Maintenance Aid (SMA) parameter `NSC-FIRST-INSTALL` to `N` (No). The default setting is `Y` (Yes).
2. Load the Natural objects specific to Natural Security from the `NSC`*`vrs`*`.INPL` file into the appropriate Natural libraries in your `FNAT` system file by using the Natural **INPL utility**.

   Once this step has been performed, it is not possible to remove Natural Security from the Natural system file; to remove Natural Security from the system file, you would have to delete the entire contents of the system file and reinstall all Natural components again.

For the initial installation (Step 9900):

1. Set the System Maintenance Aid (SMA) parameter `NSC-FIRST-INSTALL` to `Y` (Yes).
2. Load the Natural objects specific to Natural Security from the `NSC`*`vrs`*`.INPL` file into the appropriate Natural libraries in your `FNAT` system file by using the Natural **INPL utility**.

   When you load the contents of the `NSC`*`vrs`*`.INPL` file for the first time, this step creates the following security profiles and relationships:

   - A library security profile with the library ID `SYSSEC`. The library is people-protected (**People-protected** set to `Y` and **Terminal-protected** set to `N`).
   - A user security profile with the user ID `DBA`, the user type Administrator, and the password set to `DBA`.

     The user `DBA` is linked to the library `SYSSEC` (ordinary link, no special link).

Installation Verification

Natural Security is operational after ***Step 5: Load New Natural Objects*** of the *Installation Procedure* has been completed successfully.

For the initial installation, proceed as described in the section *First Steps After the Installation* in the *Natural Security* documentation. If you upgrade from a previous Natural Security version, you can skip the steps described in this section.

11 Installing Natural Advanced Facilities on BS2000

This document describes the steps for installing Natural Advanced Facilities (product code NAF) on BS2000.

Related Topic:

For information on the features and functions provided by Natural Advanced Facilities, see the *Natural Advanced Facilities* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
NAF*vrs*.SYSF	Natural FSPOOL system file definition
NAF*vrs*.MOD	Load modules and object modules
NAF*vrs*.MAC	Macros
NAF*vrs*.JOBS	Sample installation jobs
NAF*vrs*.INPL	Natural objects
NAF*vrs*.ERRN	Natural error messages

Copy the files into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** and ***Naming Conventions for Installation Procedures*** before you start the installation procedure.

Step 1: Load the FSPOOL System File Definition

(Job I050, Step 0300)

Skip this step if you want to use an existing spool file of Natural Advanced Facilities.

Load the new Natural `FSPOOL` system file definition contained in the NAF*vrs*.SYSF file by using the Adabas ADALOD utility.

Step 2 : Build the NAFB2P Parameter Module

(Job I055, Step 0300)

1. If required, modify the settings of the parameters in the source module `ANAFB2P`, which is contained in the file NAF*vrs*.JOBS.

   The following examples illustrate how the parameters can be set:

   Example 1 - Installation with Natural Security and Two Spool Servers:

   ```
   DC01    = NAFDCAM1
   DC02    = NAFDCAM2
   DC03    = NO
   .
   .
   .
   .
   .
   DC30    = NO
   ```

```
NAFERK1 = NAFP1
NAFERK2 = NAFP2
PA01    = 'STACK=(LOGON SYSPRINT,userid1,password;SVPBS201)'
PA02    = 'STACK=(LOGON SYSPRINT,userid2,password;SVPBS201)'
PA03    = 'NO'
.
.
.
.
.
PA30    = 'NO'
```

Example 2 - Installation without Natural Security and One Spool Server:

```
DC01    = NAFDCAM1
DC02    = NO
.
.
.
.
.
.
DC30    = NO
NAFERK1 = NAFP1
NAFERK2 = NAFP2
PA01    = 'STACK=(LOGON SYSPRINT;SVPBS201)'
PA02    = 'NO'
.
.
.
.
.
PA30    = 'NO'
```

Note: If Natural Security is installed, link the library `SYSPRINT` to a user which is normally not active in the Natural Security environment. Moreover, link `SYSPRINT` to as many users as there are spool servers (1-9).

The parameters for DCAM connection (`DC01 - D30`) and for `P1-EVENTING` (`NAFERK1`, `NAFERK2`) must be different from those used for the same function in any other application for the same CPU.

Described below are the parameters that can be specified in the macro `NAFB2P`:

Parameter	Explanation
DC01 - DC30	7 bytes indicating the name of the corresponding DCAM application Unused entries should be set to NO.
NAFERK1	8 bytes indicating the event ID for the communication between the online Natural and the spool server(s) and defining the name of the used common memory pool
NAFERK2	8 bytes indicating the event ID for the automatic startup for spool server(s) using an *open*UTM TP monitor and the corresponding parameter SPOOL=(*name*, *number*)

PA01 - PA30	Stack data for the initialization of Natural by specifying the following:
	STACK=(LOGON *library*,*userid*;
	   program-name)
	If Natural Security is installed, specify the following:
	STACK=(LOGON *library*,*userid*,*password*;
	   program-name)
	Unused parameters must be set to NO.

MONEVT	Name of the event ID for the BS2000 monitor task (8 bytes maximum)
PAMO	LOGON command for the monitor task. The start program must always be set to SVPMON01. If Natural Security is installed, specify the following: STACK=(LOGON *library*,*userid*,*password*; *program-name*)
CMPSIZE	Size of the Natural Advanced Facilities common memory pool (in units of 1 KB) The value is rounded up to the next higher multiple of 64 KB (64 KB = segment size = allocation unit for the common memory pools).
RSOFORM	Name of a form to be used by the spool server when output is sent to an RSO printer The information in the report is ignored. If value ' ' is used, the information from the report is used.
BS2FORM	Name of a form to be used by the spool server when output is sent to a system printer The information in the report is ignored. If value ' ' is used, the information from the report is used.
PASEC	Determines whether at logon the spool server uses the FSEC value applicable to the invoking online application (YES/NO).
PANAT	Determines whether at logon the spool server uses the FNAT value applicable to the invoking online application (YES/NO).
ISO	Determines whether the spool server(s) use a DCAM ISO application in addition to the DCAM NEA application (YES/NO).

Parameter	Explanation
RSOPROT	Determines whether the RSO messages resulting from the PRINT invocation are to be written to a log if logical printers with RSO support are used. ON: log is written to SYSLST02. OFF: no log is written to SYSLST02.
MAXERR	Maximum number of acceptable Natural spool server abends (abnormal termination of Natural). 0: no limit. 1-255: limit. If the specified limit is exceeded, the spool server terminates itself.

2. Assemble the source module ANAFB2P to create the parameter module NAFB2P. If Natural Security is installed, check the LOGON command to application SYSPRINT.

Step 3: Assemble the Environment-Dependent Nucleus and the Natural Parameter Module

(Job I055, Steps 0301, 0302, 0303)

Assemble the environment-dependent nucleus for batch mode and the Natural parameter module for the Natural Advanced Facilities server task and the Natural monitor task.

- Assemble the following source modules:

ANAFFRNT	Natural BS2000 front-end for environment-dependent nucleus for batch
ANAFRENT	Natural BS2000 reentrant, environment-dependent nucleus for batch
ANAFPARM	Natural BS2000 parameter module for batch

Step 4: Build the Natural Parameter Module

(Job I060, Step 0010, and Job I080, Step 0109)

1. Modify the parameters FSPOOL, NTPRINT, NAFUPF and NAFSIZE in the Natural parameter module according to your site requirements. For more information on these parameters, see *Natural Profile Parameters for NATSPOOL* in the *Natural Advanced Facilities* documentation.
2. Assemble and link the Natural parameter module.

Step 5: Link the Environment-Independent Nucleus

(Job I060, Step 3802)

- Add the following INCLUDE statements to the sources LNATSHAR in the library NAF*vrs*.JOBS:

```
INCLUDE  NAFNUC   ,NAFvrs.MOD
INCLUDE  NAFREENT ,NAFvrs.MOD
INCLUDE  NAFB2RSO ,NAFvrs.MOD
INCLUDE  NAFSERVR ,NAFvrs.MOD
```

Step 6: Load the Natural Objects

(Job I061, Step 0300)

- Load the Natural objects specific to Natural Advanced Facilities from the NAF*vrs*.INPL file into the Natural system files SYSPOOL and SYSPRINT by using the Natural **INPL utility**.

  Ensure that the INPL load function finishes with the message:

  ```
  Natural Advanced Facilities initialized by INPL
  ```

  If this initialization fails, various problems will be encountered at execution time.

The maps contained on the data set are provided in source form so that you can modify them according to your requirements (for example, translate the maps from English into another language). If you modify these maps, ensure that all fields have the same format/length/relative position in the map. Failure to abide by this restriction will result in an invalid system.

Step 7: Load the Natural Error Messages

(Job I061, Step 0304)

- Load the Natural error messages specific to Natural Advanced Facilities from the NAF*vrs*.ERRN file into the Natural FDIC system file by using the ERRLODUS program of the Natural SYSERR utility (described in the *Utilities* documentation).

Step 8: Link the Spool Server

(Job I065, Steps 0300, 0301, 0302, 0303)

- Link the source members `LNAFSERV`, `LNAFSEND`, `LNAFMON` and `LNAFMEND` in the library `NAF`*`vrs`*`.JOBS`.

Source	Function
`LNAFSERV`	Links the program that starts the Natural Advanced Facilities server task
`LNAFSEND`	Links the program that terminates the Natural Advanced Facilities server task
`LNAFMON`	Links the program that starts the Natural Advanced Facilities monitor task See also *BS2000 Monitor* in the *Natural Advanced Facilities* documentation.
`LNAFMEND`	Links the program that terminates the Natural Advanced Facilities monitor task

Step 9: Link the Environment-Dependent Nucleus

(Job I080, Steps 0110, 0200)

- Add the following `INCLUDE` statements to the sources `LNUTFRNT`, `LNRTFRNT` and/or `LNATFRNT` (Job I060, Step 3801) in the library *`job-library`*:

```
INCLUDE  NAFB2P   ,job-library
INCLUDE  NAFFRONT ,NAFvrs.MOD
```

Step 10: Define Natural Advanced Facilities for Natural Security

This step must be performed only if Natural Advanced Facilities is being installed in a Natural Security environment.

- Define `SYSPOOL` to Natural Security with the startup program `MENU`.

Step 11: Start Natural

1. Start Natural and add the user profile, as defined in the `NAFUPF` parameter (see the *Parameter Reference* documentation) of the Natural parameter module, to the `SYSPOOL` file by using Function 31.1 (see the *Natural Advanced Facilities* documentation).

   Note: A NAT7201 message is issued at the start of the session indicating that the profile has not yet been added to the `SYSPOOL` file.

2. Configure the spool file by following the instructions in *NATSPOOL Initialization* in the *Natural Advanced Facilities* documentation.

Installation Verification

You can verify the successful installation of Natural Advanced Facilities by following the instructions provided in this section.

- System Testing
- NATSPOOL System Errors

System Testing

You can perform the following steps to determine whether Natural Advanced Facilities functions correctly.

To print a test report

1 In the appropriate logical printer, set the `Disposition` to `K` for testing.

2 Log on to the Natural system library `SYSPOOL` and execute the programs `NTEST` and `SPPTEST`. These programs contain `WRITE (1)` statements.

You can also create test reports with Function 42 (described in the *Natural Advanced Facilities* documentation.

To proceed if a test report cannot be printed

1 Check whether the report is contained in the spool file. If the report is not in the spool file, it is possible that a `FILE` command for a print file has been active and that the report has been written to the print file and not to the spool file. This would occur because the `FILE` command has a higher priority.

If the report is in the spool file but still is not printed, take the following steps.

2 Check whether the printer is in `FREE` status. If the printer is in status `BUSY`, `INOP`, `DEAC` or `HUNG`, reset it to `FREE`.

3 Check whether there is an error message on the console beginning with Natural Advanced Facilities. Normally, an error message will be sent to the console whenever a report is not printed. This allows the console operator to perform a more detailed analysis of the cause of the error.

4 Check whether the information in the printer definition (see Function 31.4 in the *Natural Advanced Facilities* documentation) and PDN are consistent with one another.

In addition, verify that the correct physical printer name and processor name have been specified.

5 Check whether the printer is reserved by another spool system and, although it is not currently in use, has been deactivated by the other spool system. In this case, it is best if the printer is reset via a direct operator command.

6 Check whether the printer is defined in an *open*UTM application (KDCDEF) and activated from that application. Deactivate the printer via an *open*UTM administration command.

If the printer prints only 80 characters per line and 24 lines per page, take the following step.

7 Check the PDN for the printer to see that the appropriate `LINELEN` and `MAXLINE` parameters have been included in the `XOPCH` macro.

For example, to cause the printer to print 132 characters per line and 80 lines per page, provide the following parameter values:

```
XOPCH LINELEN=132
MAXLINE=80
```

8 If a `WRITE (`*`rep`*`)` statement with attribute definition `AD=I` (intensified display) is used, the Natural profile parameter `INTENS` (see the *Parameter Reference* documentation) must be specified to enable intensified display.

The following must be true:

```
LS * INTENS ≤ printer buffer size
```

where `LS` is the line size as defined in the `FORMAT (`*`rep`*`)` statement, and the printer buffer size as specified in the printer definition (see Function 31.4 in the *Natural Advanced Facilities* documentation).

NATSPOOL System Errors

A system error may be issued upon initialization of *open*UTM if the `SPOOL` parameter has been included in the `NATUTM` macro and the spool server cannot be started during initialization.

Possible causes of this error are:

- The batch limit has been exceeded, and the spool server cannot be started.
- The entry job name specified in the `SPOOL` parameter either does not exist, or the entry job exists under a different name than the one specified in the `SPOOL` parameter.

12 Installing Natural Optimizer Compiler on BS2000

13 Installing Natural Optimizer Compiler Version 8.2.7 on BS2000

This document describes the steps for installing the Natural Optimizer Compiler Version 8.2.7 (product code NOC) on BS2000.

Related Topic:

For information on the features and functions provided by the Natural Optimizer Compiler, see the *Natural Optimizer Compiler* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following file required for product installation:

File Name	Contents
`NOCvrs.MOD`	Load modules and object modules

Copy the file into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the `NATvrs.JOBS` file and are prefixed with the product code. The file is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Build the Natural Parameter Module

Step 1: Build the Natural Parameter Module

(Jobs I060 and I080)

1. Activate the Natural Optimizer Compiler by adding the following macro to your Natural parameter module:

   ```
   NTOPT 'INDX,OVFLW,ZD=OFF'
   ```

   See also *Macro NTOPT* in the *Natural Optimizer Compiler* documentation.
2. Assemble and link the Natural parameter module.

Step 2: Relink the Nucleus

(Job I060)

Adapt the link steps for Natural:

1. Add the following `INCLUDE` statement to the link of the environment-independent nucleus to include the Natural Optimizer Compiler modules:

   ```
   INCLUDE NOCNUC,NOCvrs.MOD
   ```

2. Relink your nucleus as described in ***Link the Nucleus*** in *Installing Natural*.

Installation Verification

You can verify the successful installation of the Natural Optimizer Compiler by performing the following steps:

1. Recatalog an existing program or write a new program and then catalog it.
2. Check the directory information for the program you have just cataloged, by using the following `LIST` system command:

   ```
   LIST DIR object-name
   ```

   The directory information for the specified object will be displayed, showing the size of the machine code at the bottom of the screen.

14 Installing Natural Optimizer Compiler Version 8.3.5 on BS2000

This document describes the steps for installing the Natural Optimizer Compiler Version 8.3.5 (product code NOC) on BS2000.

Related Topic:

For information on the features and functions provided by the Natural Optimizer Compiler, see the *Natural Optimizer Compiler* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

The following software must be installed before you can install the Natural Optimizer Compiler Version 8.3.5:

- **Natural** Version 8.2.7 (or higher)

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following file required for product installation:

File Name	Contents
`NOC`*vrs*`.MOD`	Load modules and object modules

Copy the file into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the `NAT`*vrs*`.JOBS` file and are prefixed with the product code. The file is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Jobs I060 and I080)

1. Activate the Natural Optimizer Compiler by adding the following macro to your Natural parameter module:

   ```
   NTOPT 'INDX,OVFLW,ZD=OFF'
   ```

   See also *Macro NTOPT* in the *Natural Optimizer Compiler* documentation.
2. Assemble and link the Natural parameter module.

Step 2: Relink the Nucleus or Dynamically Load Modules

You can either statically link the Natural Optimizer Compiler modules to the nucleus or dynamically load them when initializing a Natural session.

Relink the Nucleus:

(Job I060)

Adapt the link steps for Natural:

1. Add the following `INCLUDE` statement to the link of the environment-independent nucleus to include the Natural Optimizer Compiler modules:

   ```
   INCLUDE NOCNUC,NOCvrs.MOD
   ```

2. Relink your nucleus as described in ***Link the Nucleus*** in *Installing Natural*.

Installation Verification

You can verify the successful installation of the Natural Optimizer Compiler by performing the following steps:

1. Recatalog an existing program or write a new program and then catalog it.
2. Check the directory information for the program you have just cataloged, by using the following `LIST` system command:

   ```
   LIST DIR object-name
   ```

   The directory information for the specified object will be displayed, showing the size of the machine code at the bottom of the screen.

15 Installing Natural Connection on BS2000

This document describes the steps for installing Natural Connection (product code NTC) on BS2000.

Related Topic:

For information on the features and functions provided by Natural Connection, see the *Natural Connection* documentation.

Notation ***vrs*** **or** ***vr*****:**

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install Natural Connection:

- Entire Connection

  See the *Installation* section in the *Entire Connection* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following file required for product installation:

File Name	Contents
NTC*vrs*.MOD	Load modules and object modules

Copy the file into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.JOBS file and are prefixed with the product code. The file is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Jobs I060, I080)

1. Adapt the Natural parameter module:

   - Specify the keyword subparameter AM=PC in the NTPRINT macro for all printer files and work files to be used for data transfer between the host and the PC. For example:

     ```
     NTPRINT (7),AM=PC
     NTWORK  (7),AM=PC
     ```

   - Specify the profile parameter PC=ON.

   The parameters and the macro are described in the *Parameter Reference* documentation.

2. Assemble and link the Natural parameter module.

Step 2: Adapt the Link Steps

(Job I080)

Adapt the link steps for online Natural:

1. Add the following INCLUDE statement to the element LNATSHAR in the library *job-library*:

   ```
   INCLUDE NTCPCAM3,NTCvrs.MOD
   ```

2. Relink your Natural shared nucleus as described in ***Link the Nucleus*** in ***Installing Natural***.

Installation Verification

You can verify the successful installation of Natural Connection by following the instructions below:

1. Invoke Entire Connection on the PC and start the terminal emulation.
2. Use the terminal emulation to invoke Natural on the mainframe.
3. Enter the terminal command `%+` to activate the PC connection.
4. Use the Natural Object Handler (described in the *Utilities* documentation) to download Natural objects to the PC. For information on transferring Natural objects, see also *Data Transfer* in the section *Terminal Emulation* in the *Entire Connection* documentation.
5. Verify that the downloaded objects are now on your PC.

16 Installing Entire Transaction Propagator on BS2000

This document describes the steps for installing the Entire Transaction Propagator (product code ETP) on BS2000.

Related Topic:

For information on the features and functions provided by the Entire Transaction Propagator, see the *Entire Transaction Propagator* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Entire Transaction Propagator Version 1.5.2 Compatibility with Earlier Releases

It is essential that the ETPNUC module and the Natural programs for Entire Transaction Propagator in the Natural system library SYSETP have the same version. When using a Version 1.5.2 ETPNUC module with Entire Transaction Propagator Version 1.4.1 Natural programs - or vice versa - can cause unpredictable results.

The Natural profile parameter ETPSIZE (see the *Parameter Reference* documentation) is still accepted by Natural, but it is no longer necessary to specify this parameter. The required storage (approx. 10 KB) will be automatically allocated when the first call to an Entire Transaction Propagator database is issued. Any value specified for the ETPSIZE parameter will be ignored. The size of the ETPSIZE parameter should be left at its default (ETPSIZE=0).

As the Natural profile parameter DATSIZE (see the *Parameter Reference* documentation) is automatically adjusted to the required length, it is no longer necessary to adjust the value of the Natural DATSIZE parameter for a Natural environment running a replication task. However, the DATSIZE in such an environment will be approximately 170 KB.

After Entire Transaction Propagator Version 1.5.2 is installed and when the MENU command is invoked for the first time in the Entire Transaction Propagator maintenance utility, the Entire Transaction Propagator administration file is automatically migrated to the Entire Transaction Propagator Version 1.5.2 format and an appropriate message is displayed.

Afterwards, any attempt to access the migrated administration file from an earlier version of the Entire Transaction Propagator maintenance utility is denied.

Before enabling the new Entire Transaction Propagator maintenance utility, all log files must be empty and all log and confirmation files must be accessible for the migration process. Migration of the administration file from the Entire Transaction Propagator Version 1.4 format to the Entire Transaction Propagator Version 1.5 format is performed automatically when the Entire Transaction Propagator Version 1.5 maintenance utility is invoked for the first time. If a previous Entire

Transaction Propagator version is installed, you should use the Natural SYSMAIN utility (see the *Utilities* documentation) to delete the contents of the library `SYSETP` before installing Entire Transaction Propagator Version 1.5.2. Following installation of Entire Transaction Propagator Version 1.5.2, the maintenance utility of any earlier Entire Transaction Propagator version will be denied access to the administration file.

Otherwise, Entire Transaction Propagator Version 1.5.2 is compatible with Entire Transaction Propagator Version 1.4.1. The FDT (Field Definition Table) of the administration, log and confirmation files remains unchanged.

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following files required for product installation:

File Name	Contents
`ETP`*`vrs`*`.SYSF`	System file definition for multiple use: administration, logging and confirmation
`ETP`*`vrs`*`.SYS1`	System file definition for administration
`ETP`*`vrs`*`.SYS2`	System file definition for logging
`ETP`*`vrs`*`.SYS3`	System file definition for confirmation
`ETP`*`vrs`*`.SRCE`	Example program for using Entire Transaction Propagator services from 3GL programs
`ETP`*`vrs`*`.PAMS`	Source modules
`ETP`*`vrs`*`.JOBS`	Sample installation jobs
`ETP`*`vrs`*`.INPL`	Natural objects
`ETP`*`vrs`*`.ERRN`	Natural error messages
`ETP`*`vrs`*`.FDTA`	System file for containing FDT definitions for all Entire Transaction Propagator files

Copy the files into your environment as described in ***Copying Files to a BS2000 Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Load the Entire Transaction Propagator System Files
- Step 2: Modify the Natural Parameter Module
- Step 3: Link the Assembler Modules
- Step 4: Specify Master File Databases
- Step 5: Install the Maintenance Utility
- Step 6: Load the Natural Objects and Natural Error Messages
- Step 7: Define Natural Security Options
- Step 8: Define the Administration File
- Step 9: Define a Master File and its Log File
- Step 10: Define and Initialize the Replicate and Confirmation Files
- Step 11: Set Related Parameters
- Step 12: Modify the WADUSER2 and/or WADUSER3 User Exits
- Step 13: Run the Entire Transaction Propagator in Batch Mode

Step 1: Load the Entire Transaction Propagator System Files

(Job I050, Step 5300)

The `ETP`*`vrs`*`.SYSF` file has a Field Definition Table (FDT) that is suitable for containing the administration, confirmation and log files within one physical Adabas file.

Load the `ETP`*`vrs`*`.SYSF` file from the installation medium by using Job I050.

The following files are suitable for installing individual Entire Transaction Propagator files:

- `ETP`*`vrs`*`.FDTA`

  This file contains the same FDTs as the `ETP`*`vrs`*`.SYSF` file, but in a format suitable for loading with the Adabas ADACMP utility.

- `ETP`*`vrs`*`.SYS`*`n`*

  This file contains separate sample Adabas files that can be loaded using the Adabas ADALOD utility. These files have FDTs suitable for defining individual administration, log and confirmation files.

The Adabas utility parameters `ISNREUSE=YES` (for mainframes) and/or `REUSE=ISN` (for UNIX, Windows or OpenVMS systems) can be set to reuse freed ISNs as they become available for Entire Transaction Propagator master, replicate, confirmation, administration and/or log files.

Step 2: Modify the Natural Parameter Module

Before linking the assembler modules as described in the next step, refer to ***Step 4: Specify Master File Databases*** and ***Step 8: Define the Administration File***.

Reassemble and relink the Natural parameter module after you have modified it.

> **Note:** You must relink the modified Natural parameter module to all Natural nuclei that update a master file or start a replication task.

Step 3: Link the Assembler Modules

You must link the `ETPNUC` module to all Natural nuclei that update a master file or start a replication task. We recommend that you relink Natural with the `ETPNUC` module.

To avoid the need to relink Natural with `ETPNUC`, specify the Natural profile parameter `RCA=ON` or `RCA=NATGWETP` in the Natural parameter module to allow dynamic loading of the Entire Transaction Propagator during Natural startup. In this case, perform the following steps:

1. Rename the module `ETPNUC` in the Entire Transaction Propagator module library to `NATGWETP`.
2. Place the `NATGWETP` module in a library from which your TP monitor or batch system can perform dynamic loads. In Com-plete systems, `NATGWETP` can be loaded as a resident program.

Step 4: Specify Master File Databases

Use the `NTDB` macro in the Natural parameter module to specify the databases containing the master files:

```
NTDB ADABAS,dbid,ETP
```

Or:

```
NTDB ADABAS,(dbid,dbid,...),ETP
```

where *`dbid`* specifies one or more databases separated by commas, each containing one or more master files.

You can define the same database as an `ENTIRE` database and also as an Entire Transaction Propagator database if you specify both options for the `NTDB` macro:

```
NTDB ADABAS,dbid,ETP,ENTIRE
```

It is also possible to specify the databases containing the master files dynamically at Natural startup using the DB parameter:

```
DB=(ADABAS,(dbid,dbid,...),ETP)
```

The first time that a database which is defined as an Entire Transaction Propagator database is accessed, Entire Transaction Propagator automatically obtains a buffer (ETPSIZE) of the required size (approx. 10 KB). Any value specified for the Natural profile parameter ETPSIZE is ignored. The size of the ETPSIZE parameter should be left at its default (ETPSIZE=0).

Step 5: Install the Maintenance Utility

The Entire Transaction Propagator maintenance utility is a menu-based control facility for defining and managing master and replicate files. Although not required, we recommend that you install the maintenance utility on every node containing a master file.

Step 6: Load the Natural Objects and Natural Error Messages

(Job I061, Steps 5300, 5301)

Load the Natural objects and Natural error messages specific to the Entire Transaction Propagator by performing the following steps:

1. Step 5300: Load the ETP*vrs*.INPL file (contains the Natural objects) into your Natural FNAT system file by using the Natural **INPL utility**.
2. Step 5301: Load the ETP*vrs*.ERRN file (contains the Natural error messages) into your Natural FNAT system file by using the ERRLODUS program of the Natural SYSERR utility (described in the *Utilities* documentation).

Caution: The SYSETP library should be protected against general access with Natural Security or an equivalent security facility to prevent uncoordinated changes to the administration file. Such changes can destroy the consistency and integrity of the master and replicate files.

Step 7: Define Natural Security Options

The Entire Transaction Propagator and the maintenance utility support the concept of functional security, meaning that selected functions can be allowed or disallowed under control of Natural Security. When a user is restricted by Natural Security from performing a specific Entire Transaction Propagator function, the function is not displayed on the user's corresponding menu.

Note: Disallowing the general dialog functions - for example, EXIT, CANCEL - can cause unpredictable results.

To install the Entire Transaction Propagator if Natural Security is already installed, perform the following steps:

1. Log on to the library `SYSSEC`.
2. Issue the command `ADD LIBRARY SYSETP`.
3. In the **Add Library** menu, enter the appropriate information and select **Additional Options**.
4. In the **ADDITIONAL OPTIONS** window, select **Functional Security**.
5. In the **FUNCTIONAL SECURITY** window:
   - Define the command processor `WADNCP1` for the library `SYSETP`.
   - Enable or disable keywords (for example, `DELETE`, `MASTER` ...) or functions (for example, `REPLICATE TRANSACTIONS`) as required for your site. Note that any restrictions you define here apply to *all* users.

   In a similar way, you can restrict the availability of keywords or functions for every user that has access to `SYSETP`.

Step 8: Define the Administration File

Each database containing an Entire Transaction Propagator master file should also contain an administration file to hold all master and replicate file definitions. The logical file ID of the administration file must be 200. To define the administration file for your Natural applications, specify the `NTLFILE` macro in the Natural parameter module:

```
NTLFILE 200,dbid,fnr
```

where *fnr* is the physical file number of the administration file and *dbid* is its physical database ID.

The administration file setting can also be changed dynamically at the start of the Natural session, using the Natural profile parameter `LFILE`:

```
LFILE=(200,dbid,fnr)
```

The administration file must always be defined before using the Entire Transaction Propagator; if the Entire Transaction Propagator maintenance utility is used, the utility prompts the user for an administration file if no valid `LFILE` definition is found.

The Entire Transaction Propagator also works correctly, even if the Natural macro `NTTF` or the profile parameter `TF` is used. However, the Entire Transaction Propagator may not work properly if you have an Adabas user exit installed which modifies the database ID or file number in the Adabas control block.

Caution: Do not change any of the information in the administration file while it is being used by a replication task.

To install an administration file in your database, use either Job I050 or the Adabas ADALOD utility to load `ETP`*`vrs`*`.SYS1`. You can keep the administration file quite small since it contains only a single record for every:

- master file definition
- replicate file definition
- log file
- replicate database
- user profile

Step 9: Define a Master File and its Log File

A master file is normally an existing Adabas file. If the master file is new, you must first create the file as a normal Adabas file. This description assumes that the file to be defined as a master already exists.

Before a master file can be defined, an administration file must first be defined. See ***Step 8: Define the Administration File***.

After defining the administration file, perform the following steps to define a master and log file:

1. Stop all updating on the file that is to be defined as a master file.
2. Copy the file to be defined as a master file. We recommend that you use the Adabas ADAULD utility for this purpose; the Adabas ADASAV utility may also be used, but only where the file will be reloaded on the same device type as before. Note the exact date and time of the copy.
3. Using the Entire Transaction Propagator maintenance utility's **Master File Definition Maintenance** menu (see *Master File Task Screens* in the *Entire Transaction Propagator* documentation), specify the master file and log file. If desired, all master files on a database can share the same log file.
4. Restart the database operation to make the master file available again.

From this time on, all changes applied to the master file will be recorded in the log file. Any replicate files of the master file can now be defined without interfering with the master file operation, providing the log file contains all changes applied to the master file since the copy in **Step 2** above was created. For more information, see *Updating the Administration File* in the *Entire Transaction Propagator* documentation.

Transaction logging will start as soon as a master file is defined and a new Natural session with the appropriate administration file is started or the master file is updated from within the Natural session which was used to define the master file. Therefore, the procedure described above might not be applicable, especially when a new master file is to be defined in a running environment. If it cannot be guaranteed that no updates are applied to the to-be-defined master file that is to be defined while the above steps are executed, perform the following steps (note that it is required that the log file specified for a master file is empty when the master file is defined):

1. Stop the transaction replication processes.
2. Define the new master file.
3. Shut down the database containing the new master and log files.
4. Copy the master file as described above.
5. Use the Adabas ADADBM utility with the `REFRESH` function to refresh the log file.
6. Restart the database to make the master file available again; transaction logging will start as soon as a user updates the master file;
7. Copy the master file's contents into the replicate file;
8. Restart transaction replication.

Install a log file in your database by using either Job I050 or the Adabas ADALOD utility to load `ETP`*`vrs`*`.SYS2`.

The number of records in the log file depends on the number of transactions that update master files between two successive invocations of the **Clean up log file** maintenance function (provided that all transactions are replicated), which is described in the *Entire Transaction Propagator* documentation. The approximate number of log file records equals:

```
(transaction count)*(updates per transaction, +1)
```

When loading the log file, the ADALOD utility parameter `PGMREFRESH=YES` is required if you want the **Clean up log file** function to refresh the log file for improved performance.

Note: It is impossible to refresh a log file that is also used as an administration or confirmation file.

Step 10: Define and Initialize the Replicate and Confirmation Files

After the master file is defined, the replicate files can be defined. A confirmation file must also be defined on each database where a replicate file is defined. If desired, multiple replicate files on a database can share the same confirmation file. To define a replicate file, perform the following steps:

1. Using the **Add replicate file definition** screen (described in the *Entire Transaction Propagator* documentation) of the online maintenance utility, specify the database IDs and file numbers for the replicate, master and confirmation files:
2. Load the unloaded copy of the master file into the replicate file using the Adabas ADALOD utility (if the file was unloaded with ADAULD) or ADASAV (if the file was unloaded with ADASAV). Specify the parameter `USERISN=YES` for the related Adabas mainframe utilities. For the replicate files on UNIX, Windows or OpenVMS systems, the option `USERISN` must be specified

when loading a non-empty file unless the distribution key is used as the replication criterion. If an empty replicate file is created, the option `USERISN` or parameter setting `USERISN=YES` is not required.

When specifying the `MAXISN` parameter while defining the replicate file you should remember that, when using the records' ISNs as the replication criterion, the Address Converter (AC) is not automatically extended. This can occur if the Entire Transaction Propagator issues Adabas N2 calls to add new replicate file records and specify an ISN value that exceeds the file's `MAXISN` value; in such a case, a response code 113 is returned. Specify a `MAXISN` value large enough for future extensions of the replicate file.

3. Using the **Display transactions** function of the maintenance utility (described in the *Entire Transaction Propagator* documentation), check for any log file entries that have been made since the master file was copied.
4. If a replicate file contains a subset of the master file records, you should either delete all unneeded records as defined by the specified distribution key ranges or copy only the selected subset of records from the master file. This can be done using one of the following methods:
   - Use a Natural program to delete the unneeded records from the replicate file.
   - Use a Natural program to copy only the selected records from the master file to an intermediate file which is then subsequently copied to the replicate file.
   - Use the `SELCRIT` and `SELVAL` parameters of the Adabas utility function `ADAULD UNLOAD` to select only the subset for unloading. This is the recommended method.

   Caution: After the replicate file has been initialized, it should not be manually changed. Otherwise, the consistency with the master file could be destroyed.

5. To install a confirmation file in your mainframe database, use either Job I050 or the Adabas ADALOD utility to load `ETP`*`vrs`*`.SYS3` from the installation medium. For confirmation files on UNIX, Windows or OpenVMS systems, use the confirmation file field definitions in file `ETP`*`vrs`*`.FDTA` to supply field definitions for the Adabas ADAFDU utility. For every replicate file that uses the confirmation file, the latter file contains a single record.
6. If the log file contains any entries for the master file, start the replication task for the replicate file using the **Replicate transactions** maintenance function described in the *Entire Transaction Propagator* documentation. The task checks the appropriate administration file for master or replicate files in the file range to be processed. The task then synchronizes the master and replicate file by applying all updates to the replicate file that are not already applied. The replicate file is now available for use.

To add other replicate files of an existing master file, perform the steps above after creating an up-to-date copy of the master file. Note that when creating a replicate file, no logged changes to the master file should be removed from the log file. If this rule is followed, a replicate file can be added without affecting the normal mode of operation.

Step 11: Set Related Parameters

- **Natural WH Parameter**

  When running multiple Entire Transaction Propagator replication tasks in parallel, specify WH=ON to avoid NAT3145 errors (record already in hold status for another user) when two tasks attempt to access the same record simultaneously.

- **Reusing ISNs in Entire Transaction Propagator Files**

  The Adabas utility parameters ISNREUSE=YES (for mainframes) and/or REUSE=ISN (for UNIX, Windows or OpenVMS systems) can be set to reuse freed ISNs as they become available for Entire Transaction Propagator master, replicate, confirmation, administration and/or log files.

- **Transactions with Many Updates**

  If transactions that include a lot of updates are to be logged, increase the value of the Adabas ADARUN parameter LDEUQP. The required size for transaction logging can be computed as:

  ```
  LDEUQP = (updates per transaction * 29)
  ```

Step 12: Modify the WADUSER2 and/or WADUSER3 User Exits

The two user exits WADUSER2 and WADUSER3 are delivered in source form in the library SYSETP.

- **Optional User Exit WADUSER2**

  The optional user exit WADUSER2 is a Natural subprogram for controlling file replication. WADUSER2 is called after the Entire Transaction Propagator decides whether the record in question is to be replicated or not.

  The WADUSER2 user exit, defined as User Exit 2, is called only if the user exit option is specified when a master file is defined (see *Master File Task Screens* in the *Entire Transaction Propagator* documentation). An example of WADUSER2 is included in the SYSETP library.

- **Message Handler WADUSER3**

  The subprogram WADUSER3 is used to display all messages issued by the replication task. WADUSER3 can be modified to filter the task messages and, if desired, send them directly to the operator console. The WADUSER3 subprogram receives the error number, severity level and the message text from the replication task. This allows the user to select the messages to be displayed. By means of the Natural CMWTO entry (for an example, see the program WTO in the library SYSEXTP), the messages can be sent to the operator console.

Caution: The user exits WADUSER2 and WADUSER3 should neither issue Adabas calls that update a database file nor should they issue any End Transaction (ET) or Back Out Transaction (BT) commands; otherwise, the results are unpredictable.

Step 13: Run the Entire Transaction Propagator in Batch Mode

In batch mode, command execution is possible only by means of direct commands. For a list of direct commands and their minimum abbreviations, see *Entering Direct Commands* in the *Entire Transaction Propagator* documentation.

The following example is for a batch file for starting a replication task:

```
LOGON SYSETP                                          (1)*
MENU                                                  (2)*
REPLICATE TRANSACTIONS                                (3)
1,1,65535,65535,1,65535,00:30:00,200,1000,N,EXIT      (4)
EXIT                                                  (5)
FIN                                                   (6)
```

* If Natural Security is installed, lines (1) and (2) may have to be changed (see *Starting the ETP Maintenance Utility* in the *Entire Transaction Propagator* documentation).

Line (4) contains the parameters for the corresponding **Replicate transactions** Entire Transaction Propagator maintenance utility screen. Parameters are entered from top to bottom, left to right. Line (5) exits from the Entire Transaction Propagator maintenance utility.

Caution: We recommend that you start replication tasks in batch mode.

The following example is for a batch file for deleting successfully replicated transactions:

```
LOGON SYSETP                                          (1)*
MENU                                                  (2)*
CLEANUP LOGFILE                                       (3)
1,1,65535,65535,,,N,00:30:00,200,1000,EXIT            (4)
EXIT                                                  (5)
FIN
```

* If Natural Security is installed, lines (1) and (2) may have to be changed (see *Starting the ETP Maintenance Utility* in the *Entire Transaction Propagator* documentation). Line (4) contains the parameters for the corresponding **Cleanup logfile** Entire Transaction Propagator maintenance utility screen. Parameters are entered from top to bottom, left to right. Line (5) exits from the Entire Transaction Propagator maintenance utility.

Caution: We recommend you to start tasks that successfully delete replicated transactions in batch mode.

If a window is displayed in batch mode, all fields are protected; the reason for this is that in most cases it is not possible to determine the number of selectable items. Therefore, the only meaningful command is `PROCESS`. The following is an example to reset the in-use flag for all replicate files:

```
MENU
RESET IN-USE * *
PROCESS
EXIT
EXIT
FIN
```

To run the above examples without problems, the following parameters in the Natural parameter module must be specified:

```
ID=','   (default setting)
IM=D
PC=OFF   (default setting)
```

Either the Natural statement `SET CONTROL '+'` or the terminal command `%=` cancel the effect of `PC=OFF`.