Con-nect

Con-nect Installation and Operations

Version 3.4.3

November 2016

Dieses Dokument gilt für Con-nect ab Version 3.4.3.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuausgaben bekanntgegeben werden.

Dokument-ID: CNT-INSTALL-343-20161125

Table of Contents

Con-nect Installation and Operations

This documentation provides information on installing and operating Con-nect in various operating system environments and under various TP monitors. It also describes how to install the Con-nect transport service used by Con-nect to send and receive mail items to/from other Con-nect nodes or mailing systems and how to install Con-nect Text Retrieval. Additionally, this manual describes how to install the Con-nect selectable units Con-nect Teleservices and Con-nect SNADS.

It is divided into the following parts:

Part 1. Installing Con-nect	Describes the procedures required to install Con-nect on the various operating systems.
Part 2. Installing the Transport Service	Describes the procedures required to install the Con-nect transport service on the various operating systems.
Part 3. Installing Con-nect Teleservices	Describes the procedures required to install Con-nect Teleservices.
Part 4. Installing Con-nect SNADS	Describes the procedures required to install Con-nect SNADS.

I Part 1. Installing Con-nect

This part contains all the information you need to install Con-nect. It is subdivided into the following topics:

1 General Information for Con-nect

Con-nect Operating Environment

Con-nect Version 3.4.3 can be used with any of the following operating systems:

- z/OS,
- z/VSE,
- BS2000.

Additionally, Con-nect Version 3.4.3 can be used with any of the following teleprocessing monitors:

- Com-plete,
- CICS,
- TSO,
- SHADOW,
- IMS/DC
- TIAM and UTM (BS2000).

Adabas Statistics Facility

If the Adabas Statistics Facility is installed and you want to send the output of critical reports to Con-nect, you must copy the following Con-nect API subprograms and the subprograms called by these API subprograms from SYSCNT2 to SYSASF:

Z-ADD11, Z-ERA11, and Z-MAILA.

For further information, see the *Con-nect Application Programming Interface* documentation.

Con-form and Language Modules

Con-form and the language modules *must* also be installed with Con-nect. Thus, when using SMA to install Con-nect (CNTxxx), Con-form (CMFxxx) and CSxnnn (language modules) must also be set to "to be installed".

For further information, see the Con-form Installation Manual.

Con-nect Text Retrieval

Con-nect Text Retrieval is based upon Adabas Text Retrieval (for further information see *Adabas TRS Call Interface: Programmer's Guide*).

When you plan to use Con-nect Text Retrieval, the following product must be installed:

- Adabas Text Retrieval System Version 2.1.4 or above.

The Con-nect Text Retrieval data sets are no longer delivered separately. They have been integrated into the Con-nect data sets. The installation of Con-nect Text Retrieval is incorporated in the Con-nect installation procedure.

When you plan to use Con-nect Text Retrieval and you are using SMA, you must set the parameter CNT-TEXT-RETR to **Y**.

Natural

Con-nect is a subsystem of Natural. Con-nect Version 3.4.3 is compatible with Natural Version 8.2.2 or above.

If OUTPUT 'CONNECT' is defined either in conjunction with a Con-nect system whose selected printer is CONNECT, or an application with the Natural statement DEFINE PRINTER (n) OUTPUT 'CONNECT', the Natural load library module NATPCNT must be linked to the Natural nucleus.

Printer files 1 through 8 should be defined in the NTPRINT macro. Otherwise, you have to define printer files 1 through 8 dynamically in the Natural parameter PRINT.

Natural Security

If Natural Security is installed and the language code for Natural Security is set to 0, it is important to note that the value of the Natural profile parameter ULANG is applied. This means that the value of ULANG and the Con-nect language code you are installing must be identical.

See the *Natural Security* documentation for information on user maintenance.

Predict

If Predict is installed, you must copy the following Con-nect API subprograms and the subprograms called by these API subprograms from SYSCNT2 to SYSDIC to enable the transfer of texts to and from Predict:

Z-ADD11A, Z-DIS11, Z-ERA11, Z-GET11B and Z-GET11C.

For further information, see the *Con-nect Application Programming Interface* documentation.

Transport Service

The installation of the Con-nect transport service method is incorporated in the Con-nect installation procedures. A description of the installation steps for the transport service is provided. See ***Part 2. The Transport Service***.

Before beginning the installation, it is recommended that you read the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *General Information*.

Installation Jobs

The installation of Software AG products is performed by installation jobs. These jobs are either created "manually" or generated by System Maintenance Aid (SMA).

For each step of the installation procedure, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA.

If you are not using SMA, an example installation job of the same number is provided in the job library on the Con-nect installation medium; you must adapt this example job to your requirements. Note that the job numbers on the medium may be preceded by a product code (for example, CNTI061).

Using System Maintenance Aid

If you plan to use SMA for the installation process, you must set the parameter CNT-FIRST-INSTALL to either **Y** if Con-nect is not installed, or **N** if you are upgrading from a previous Con-nect version.

When you plan to use Con-nect Text Retrieval, you must set the parameter CNT-TEXT-RETR to **Y**. If you do not want to install Con-nect Text Retrieval, set this parameter to **N**.

Additionally, if you want to install the transport service method, you must set the parameter CNT-TRANS-SERV (CNT-OPT-TRANS for BS2000) to **Y**. If you do not want to install the transport service, set this parameter to **N**.

All three parameters can be found in the SMA group Options.

For information on using SMA for the installation process, refer to the SMA documentation (included on the current edition of the Natural documentation CD).

Data Set Names

All references in this manual are made to data set names. The notation *nnn* in data set names represents the version number of the product, while the notation *x* represents the language code as shown below:

Character	Language	Code Number
E	English	1
D	German	2
F	French	3
S	Spanish	4
H	Dutch	6
K	Danish	8

2 Con-nect Installation Medium

z/OS Installation Medium

The following topics are covered below:

- Data Sets
- Copying Data Sets to a z/OS Disk

Data Sets

The z/OS installation medium contains the data sets listed below. The sequence of the data sets is shown in the Software AG Product Delivery Report which accompanies the installation medium.

Data Set Name	Contents
CNT*nnn*.JOBS	Example installation jobs.
CNT*nnn*.INPL	INPL data set containing language-independent components of the Con-nect processor and transport service.
CNT*nnn*.LOAD	Contains the CTMOD module which must be linked to the Natural nucleus. This data set is specifically for the transport service.
CNT*nnn*.SRCE	Source library for the transport service. This data set is specifically for the transport service.
CNT*nnn*.SYS2	An unloaded copy of an empty Con-nect help system file. This file has the internal identification LFILE number 222.
CNT*nnn*.SYS3	An unloaded copy of an empty Con-nect spool file. This file is used for the external mailing facility, e.g., Con-nect multi-node. It has the internal identification LFILE number 223.
CNT*nnn*.SYS4	An unloaded copy of an empty central directory file for the directory synchronization. It has the internal identification LFILE number 160.
CNT*nnn*.SYS5	An unloaded copy of an empty Con-nect system file in the format used by the Cabinet Transfer utilities. The internal identification LFILE number for this data set is 224.
CNT*nnn*.SYS6	An unloaded copy of an empty Con-nect Text Retrieval document file. The internal identification LFILE number for this data set is 233.
CNT*nnn*.SYS7	An unloaded copy of an empty Con-nect Text Retrieval vocabulary file. The internal identification LFILE number for this data set is 231.
CS*xnnn*.INPL	INPL data set containing components for the language-dependent parts of Con-nect and internal error text (language-dependent).
CS*xnnn*.HLPS	Con-nect help screens (language-dependent).

Copying Data Sets to a z/OS Disk

Copy the 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 data sets 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 data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy all data sets from tape to disk.

- Step 1: Copy Data Set COPY.JOB from Tape to Disk
- Step 2: Modify hilev.COPY.JOB on Your Disk
- Step 3: Submit COPY.JOB

Step 1: Copy Data Set COPY.JOB from Tape to Disk

- Modify the following sample job according to your requirements:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* --------------------------------
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=tape-volser),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=disk-volser,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

  where:

  `tape-volser` is the VOLSER of the tape, for example: `T12345`,
  `hilev` is a valid high-level qualifier, and
  `disk-volser` is the VOLSER of the disk.

- Execute the job to copy the data set `COPY.JOB` to your disk.

Step 2: Modify hilev.COPY.JOB on Your Disk

- Modify `hilev.COPY.JOB` according to your requirements:

  Set `EXPDT` to a valid expiration date, for example, `99365`.

  Set `HILEV` to a valid high-level qualifier, for example, `USERLIB`.

  Set `LOCATION` to a storage location, for example, `STORCLAS=ABC` or `UNIT=3390,VOL=SER=USR123`.

Step 3: Submit COPY.JOB

- Execute `hilev.COPY.JOB` to copy single, multiple, or all data sets to your disk.

z/VSE Installation Medium

The following topics are covered below:

- Data Sets
- Copying Data Sets to a z/VSE Disk

Data Sets

The installation medium contains the data sets listed below. The sequence of the data sets is shown in the Software AG Product Delivery Report which accompanies the installation medium.

Data Set Name	Contents
CNT*nnn*.INPL	INPL data set containing language-independent components of the Con-nect processor and transport service.
CNT*nnn*.LIBR	Contains the CTMOD module which must be linked to the Natural nucleus and the source library for the transport service. This data set is specifically for the transport service.
CNT*nnn*.LIBJ	Example installation jobs.
CNT*nnn*.SYS2	An unloaded copy of an empty Con-nect help system file. This file has the internal identification LFILE number 222.
CNT*nnn*.SYS3	An unloaded copy of an empty Con-nect spool file. This file is used for the external mailing facility, e.g., Con-nect multi-node. It has the internal identification LFILE number 223.
CNT*nnn*.SYS4	An unloaded copy of an empty central directory file for the directory synchronization. It has the internal identification LFILE number 160.
CNT*nnn*.SYS5	An unloaded copy of an empty Con-nect system file in the format used by the Cabinet Transfer utilities. The internal identification LFILE number for this data set is 224.
CNT*nnn*.SYS6	An unloaded copy of an empty Con-nect Text Retrieval document file. The internal identification LFILE number for this data set is 233.

Data Set Name	Contents
CNT*nnn*.SYS7	An unloaded copy of an empty Con-nect Text Retrieval vocabulary file. The internal identification LFILE number for this data set is 231.
CS*xnnn*.INPL	INPL data set containing components for the language-dependent parts of Con-nect and internal error text (language-dependent).
CS*xnnn*.HLPS	Con-nect help screens (language-dependent).

Copying Data Sets to a z/VSE Disk

Copy the 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 data sets 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 data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to disk. All other data sets can be installed directly from the tape.

- Step 1: Copy Data Set COPYTAPE.JOB to Disk
- Step 2: Modify COPYTAPE.JOB on Your Disk
- Step 3: Submit COPYTAPE.JOB

Step 1: Copy Data Set COPYTAPE.JOB to Disk

- Modify the following sample job according to your requirements:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                          +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****************************************
*     STORE COPYTAPE.JOB IN LIBRARY
* *****************************************
// ASSGN SYS004,nnn
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'
/*
```

```
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

where:

nnn is the tape address, and
lib.sublib is the library and sublibrary in which the data set `COPYTAPE.JOB` is to be stored.

- Execute the job to copy the data set `COPYTAPE.JOB` to disk.

  `COPYTAPE.JOB` contains the JCL required to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to disk.

Step 2: Modify COPYTAPE.JOB on Your Disk

- Modify `COPYTAPE.JOB` according to your requirements and set the disk space parameters as appropriate.

Step 3: Submit COPYTAPE.JOB

- Execute `COPYTAPE.JOB` to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` to your disk.

BS2000 Installation Medium

The following topics are covered below:

- Data Sets
- Copying Files to a BS2000 Disk

Data Sets

The installation medium contains the files listed below. The sequence of the files is shown in the Software AG Product Delivery Report which accompanies the installation medium.

Data Set Name	Content
CNT*nnn*.JOBS	Example installation jobs.
CNT*nnn*.INPL	INPL data set containing language-independent components of the Con-nect processor and transport service.
CNT*nnn*.MOD	Contains the CTMOD module which must be linked to the Natural nucleus. This data set is specifically for the transport service.

Data Set Name	Content
CNT*nnn*.SYS2	An unloaded copy of an empty Con-nect help system file. This file has the internal identification LFILE number 222.
CNT*nnn*.SYS3	An unloaded copy of an empty Con-nect spool file. This file is used for the external mailing facility, e.g., Con-nect multi-node. It has the internal identification LFILE number 223.
CNT*nnn*.SYS4	An unloaded copy of an empty central directory file for the directory synchronization. It has the internal identification LFILE number 160.
CNT*nnn*.SYS5	An unloaded copy of an empty Con-nect system file in the format used by the Cabinet Transfer utilities. The internal identification LFILE number for this data set is 224.
CNT*nnn*.SYS6	An unloaded copy of an empty Con-nect Text Retrieval document file. The internal identification LFILE number for this data set is 233.
CNT*nnn*.SYS7	An unloaded copy of an empty Con-nect Text Retrieval vocabulary file. The internal identification LFILE number for this data set is 231.
CS*xnnn*.INPL	INPL data set containing components for the language-dependent parts of Con-nect and internal error text (language-dependent).
CS*xnnn*.HLPS	Con-nect help screens (language-dependent).

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 `SRVvrs.LIB` from another Software AG installation tape. For further information, refer to the element `#READ-ME` in this library. The library `SRVvrs.LIB` is stored on the tape as a sequential file named `SRVvrs.LIBS` containing LMS commands. The current version `vrs` can be obtained from the *Software AG Product Delivery Report*.

- Execute the following commands to convert `SRVvrs.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`.

3 Installing Con-nect

This chapter describes step by step how to install Con-nect under the operating systems z/OS, z/VSE and BS2000.

Overview

When you install Con-nect, data sets required for the transport service are also installed. See *Part 2. The Transport Service* for an overview of the transport service installation procedures.

Determine if this is a first time installation or whether this version is to replace an existing Con-nect version.

If this is a first time installation and you are using SMA to install Con-nect, set parameter CNT-FIRST-INSTALL to **Y**. If you are migrating from a previous version of Con-nect, set parameter CNT-FIRST-INSTALL to **N**.

If you want to install the transport service method, you must set the parameter CNT-TRANS-SERV to **Y**. If you do not want to install the transport service, set this parameter to **N**.

Additionally, if you are migrating from a previous version of Con-nect, skip steps 1 through 6 and proceed with step 7.

Installation Procedures for Con-nect

1. Load Help System File (Job I050, Step 0203)

The help system file is used to store all help screens.

Load the empty Con-nect help system file (data set CNT*nnn*.SYS2) using the Adabas utility ADALOD.

The specifications of the NTLFILE macro call in the Natural parameter module *must* correspond to the file number and the ID of the database in which this file is contained. The internal file identification number 222 refers to the Con-nect help system file.

For a medium-sized installation the following ADALOD parameters can be used:

```
ADALOD LOAD FILE=nnn,MAXISN=20000,DSSIZE=20,ASSOPFAC=15,DATAPFAC=15,
ADALOD      TEMPSIZE=20,SORTSIZE=10,ISNREUSE=YES
```

2. Load Spool File (Job I050, Step 0205)

The spool file is used for intermediate storage of external mail information.

Load the empty Con-nect spool file (data set CNT*nnn*.SYS3) using the Adabas utility ADALOD.

The specifications of the NTLFILE macro call in the Natural parameter module *must* correspond to the file number and the ID of the database in which this file is contained. The internal file identification number 223 refers to the Con-nect spool file.

For a medium-sized installation the following ADALOD parameters can be used:

```
ADALOD LOAD FILE=nnn,MAXISN=20000,DSSIZE=20,ASSOPFAC=15,DATAPFAC=15,
ADALOD      TEMPSIZE=20,SORTSIZE=10,ISNREUSE=YES
```

Note: Whenever this file is loaded and reloaded, you must specify the USERISN option to ensure that the ISN assignments of the database records remain unchanged.

3. Load Central Directory File (Job I050, Step 0207)

The central directory file is used to store addresses uploaded to the central directory from Con-nect nodes participating in a Con-nect network.

Load the empty central directory file (data set CNT*nnn*.SYS4) using the Adabas utility ADALOD.

The specifications of the NTLFILE macro call in the Natural parameter module must correspond to the file number and the ID of the database in which this file is contained. The internal file identification number 160 refers to the central directory file.

For a medium-sized installation the following ADALOD parameters can be used:

```
ADALOD LOAD FILE=nnn,MAXISN=90000,DSSIZE=10,ASSOPFAC=15,DATAPFAC=15,
ADALOD      TEMPSIZE=20,SORTSIZE=10,ISNREUSE=YES
```

4. Load Cabinet Transfer File (Job I050, Step 0208)

If this is a first time installation, execute this step.

The Con-nect Cabinet Transfer functions and utilities CCTLOAD and CCTRLOAD are used to transfer one or more Con-nect cabinets from one Con-nect system to any other Con-nect system - regardless of the operating system.

Load the empty Cabinet Transfer file (data set CNT*nnn*.SYS5) using the Adabas utility ADALOD. Using the Adabas Online Services, you can obtain the size of the Con-form system file's MAXISN and DSIZE parameters.

If you specify the Cabinet Transfer file in the NTLFILE macro call in the Natural parameter module, it *must* correspond to the file number and the ID of the database in which this file is contained.

Otherwise, you can call it dynamically. For example:

```
*CON-NECT LFILE=(224,DBNR,FNR)
```

For a medium-sized installation the following ADALOD parameters can be used:

```
ADALOD LOAD FILE=nnn,MAXISN=20000,DSSIZE=20,ASSOPFAC=15,DATAPFAC=15,
ADALOD      TEMPSIZE=20,SORTSIZE=10,ISNREUSE=YES
```

The following Adabas parameters must be set to the highest values:

```
UNIT=3380NISNHQ=32767
NH=(4 x 32767)
LP=32767
WORK=400  Cyl. / 144.000 PAM Pages
```

5. Load Con-nect Text Retrieval Document File (DFNR) (Job I050, Step 4050)

If you do not plan to use Con-nect Text Retrieval, skip this step.

If Con-nect Text Retrieval is already installed, skip this step.

If Con-nect Text Retrieval has not yet been installed, load the empty Adabas file (data set CNT*nnn*.SYS6) using the ADALOD utility. The size of the file depends on the number of words in the data set. As a rule, 4 cylinders are required per 10000 documents.

ISNREUSE must equal yes (e.g. ISNREUSE=YES).

6. Load Con-nect Text Retrieval Vocabulary File (VFNR) (Job I050, Step 4051)

If you do not plan to use Con-nect Text Retrieval, skip this step.

If Con-nect Text Retrieval is already installed, skip this step.

If Con-nect Text Retrieval has not yet been installed, load the empty Adabas file (data set CNT*nnn*.SYS7) using the ADALOD utility. It is recommended that the vocabulary file (VFNR) is ten percent (10%) larger than the document file (DFNR). However, with full text retrieval the required size will increase.

ISNREUSE must equal yes (e.g. ISNREUSE=YES).

7. Delete Objects from Library SYSCNTx (Job I051, Step 0200)

Optionally: you can delete all objects from libraries SYSCNT2, SYSCNT2H, SYSCNT2B and SYSCNT2T, with the exception of the items below. It is recommended that you first save these items in a different library before you delete the other objects.

- User created programs for pre-formatting, post-formatting and procedures stored in library SYSCNT2.
- User forms (F*) stored in library SYSCNT2.
- Natural forms stored in library SYSCNT2.
- User exits (USR*) stored in library SYSCNT2B.

8. Modify Parameter Modules (Jobs I060 and I080)

Jobs I060 and I080 can be found in the SMA Joblib or in the example jobs data set.

Use the appropriate steps of this job (0010, 2210, 2300 or 2500) for your specific TP-monitor.

Note: If running in batch, use job I060, step 0010 (instead of job I080).

Generating an executable Natural/Con-nect nucleus is almost identical to generating a standard Natural nucleus. The changes that must be applied are given in the following sections.

1. The keyword parameter CSIZE=*nn* must be set to a value of 70 or more. This parameter can be specified either in the NTPRM macro call or dynamically when Con-nect is invoked. If this parameter is not specified correctly, Natural error 5300 or 5319 will occur whenever Con-form is called.
2. For Con-nect Text Retrieval, the keyword parameter TSIZE=*nn* must be set to a value of 32 or more. This parameter can be specified either in the NTPRM macro call, or dynamically when Con-nect is invoked.
3. The keyword parameters LS (line size) and PS (page size) must be set to at least 132 and 66 respectively.
4. The keyword parameter MAXCL (maximum number of program calls between successive terminal dialog steps) must be set to 0.
5. The keyword parameter MADIO (maximum number of DBMS calls between successive screen I/O operations) must be set to at least 1000.
6. Additional macro calls must be specified to define the Con-nect help system file to the nucleus. The macro's name is NTLFILE. The NTLFILE macro call must be placed after the NTPRM macro call.

NTLFILE is a keyword macro with the following parameters:

Parameter	Explanation
ID	Internal identifier. Must be set to 222 for the Con-nect help system file, 223 for the Con-nect spool file, 224 for the Cabinet Transfer File and 160 for the directory synchronization. This parameter is mandatory.
DBID	ID of the database where the Con-nect system files reside. This parameter may be specified if necessary.
FNR	File number of the Con-nect system files. This parameter is mandatory.
PASSW	Adabas password that may be required to gain access to the Con-nect system files. We advise against using password protection on the Con-nect spool file. This parameter may be specified if necessary.
CIPH	Adabas cipher key that may be required to encipher or decipher records in the Con-nect system files. We advise against using cipher protection on the Con-nect spool file. This parameter may be specified if necessary.

The NTLFILE macro is contained in the Natural source library.

As an alternative to using this macro, the dynamic parameter LFILE can be specified when Con-nect is invoked. The format is:

```
LFILE=(222,dbid,fnr<,passw><,ciph>)  /Con-nect Help
LFILE=(223,dbid,fnr<,passw><,ciph>)  /Con-nect Spool File
LFILE=(224,dbid,fnr<,passw><,ciph>)  /Con-nect Cabinet Transfer File
LFILE=(160,dbid,fnr<,passw><,ciph>)  /Con-nect Central Directory File
```

```
LFILE=(231,dbid,fnr<,passw><,ciph>)  /Con-nect Text Retrieval Vocabulary File
LFILE=(233,dbid,fnr<,passw><,ciph>)  /Con-nect Text Retrieval Document File
```

where the parameters shown in pointed brackets are optional.

Note: The specifications of the NTLFILE macro call in the Natural parameter module *must* correspond to the file number and the ID of the database in which this file is contained.

1. If Con-nect Teleservices is to be installed, an additional NTLFILE macro or LFILE keyword specifying the telex spool file must be entered. The internal identifier (ID parameter) is 223.
2. If the Con-nect directory synchronization is to be used, an additional NTLFILE macro or LFILE keyword specifying the central directory file must be entered. The internal identifier (ID parameter) is 160.
3. If the Con-nect cabinet transfer is to be used, an additional NTLFILE macro or LFILE keyword specifying the cabinet transfer file must be entered. The internal identifier (ID parameter) is 224.
4. If Con-nect Text Retrieval is to be installed, an additional NTLFILE macro or LFILE keyword specifying the vocabulary file must be entered. The internal identifier (ID parameter) is 231.
5. If Con-nect Text Retrieval is to be installed, an additional NTLFILE macro or LFILE keyword specifying the document file must be entered. The internal identifier (ID parameter) is 233.

9. Assemble and Link Parameter Module (Jobs I060 and I080)

Use the appropriate steps of this job (0015, 2220, 2310 or 2510) for your specific TP-monitor.

Note: If running in batch, use job I060, step 0015 (instead of job I080).

10. Link-edit Natural/Con-nect Nucleus (Jobs I060 and I080)

Con-nect provides full support of 31-bit addressing mode. This means that the Natural/Con-nect nucleus can be loaded either below or above 16 MB. For further information, see the *Natural Installation* documentation.

Use the standard link procedure as described in the *Natural Installation* documentation.

Use the appropriate steps of this job (0020, 2230, 2320 or 2570) for your specific TP-monitor.

Note: If running in batch, use job I060, step 0020 (instead of job I080).

For further information, see the *Con-form Installation* documentation.

Additionally, the CSTATIC list from the Natural parameter module must specify the module name CTMOD. See ***Installation Procedures for Com-plete***, ***Installation Procedures for Batch Mode*** or ***Installation Procedures for CICS***.

Note: To set-up the Adabas start-up procedures ensure the following are completed. The HEX12=TRSHEX12 parameter must be added to your Adabas nucleus start-up parameters. The database parameter NQCID must be equal to or greater than 200. Start-up the nucleus once Con-nect Text Retrieval is installed.

11. Load INPL (Job I061, Step 0200)

Note: Before you begin this step, ensure that sufficient disk space is available.

Use the Natural utility program INPL to load the Con-nect and transport service programs into the system file of your Natural system. This step is required.

The transport service programs Y8SEC and YCSETUP, which are distributed as source code, must be configured to your system's requirements.

Specify data set CNT*nnn*.INPL (Con-nect processor) as workfile 01.

```
INPL
B
FIN
```

Once the INPL performance program is completed, program INSTALL is called. See ***Run INSTALL (Job I100, Step 5020)***. If necessary, you can stop the INSTALL program and restart it later.

12. Load Language-dependent INPL (Job I061, Step 02x0)

Use the Natural utility program INPL to load the language-dependent Con-nect programs and error messages into the system file of your Natural system. This step is required.

Specify data set CS*xnnn*.INPL as workfile 01.

x in the step number refers to the language code.

Repeat this step for all languages which are to be installed.

```
INPL
B
FIN
```

13. Load Language-dependent Help Screens (Job I200, Step 02x0)

If Natural Security is installed, see ***General Note Concerning Natural Security*** before proceeding with this step.

Use the Con-nect utility HMENU to load the language-dependent help screens into the system file of your Natural system. This step is required.

Specify data set CS *xnnn*.HLPS as workfile 03.

x in the step number refers to the language code.

Ensure that the Con-nect Help File is defined to your batch Natural. See ***Modify Parameter Modules***.

Repeat this step for all languages which are to be installed.

If Natural Security is not installed use the following:

```
LOGON SYSCNT2H
HMENU
L,SYSCNT2,x
.
FIN
```

where x is the language code

If Natural Security is installed use the following:

```
SYSCNT2H, USER ID, PASSWORD
HMENU
L,SYSCNT2,x
.
FIN
```

x is the language code.

To load the help screens, you must always use function L (not N, as was required for the German language in previous Con-nect versions).

14. General Note Concerning Natural Security (Job I100, Step 2020)

The following description is applicable, if Natural Security is installed at your site.

Module SECCHCK1 must be present in the Natural steplib. This module can be obtained from the Natural library SYSLIB.

When limiting the commands via Natural Security for the Con-nect application, the following commands must not be disallowed, as they are needed by various Con-nect functions:

- CAT
- CLEAR
- GLOBALS
- SCRATCH
- SETUP
- UNCAT

Additionally, the Natural Security administrator must enter the following into the Natural Security system:

- SYSCNT2 (online application system)
- SYSCNT2H (online help screen maintenance system)
- SYSCNT2B (batch utilities).

See the *Natural Security* documentation for further information on command restrictions.

Note: If you plan to use the Central Directory File (CADMIN) to administrate the Con-nect addresses, you should use Natural Security to protect this file. This procedure is described in a technical information sheet, which is provided by Software AG. For further information, contact your Software AG support representative.

15. Run INSTALL (Job I100, Step 5020)

Note: Con-form must be installed before this step can be executed.

Using the newly installed online Con-nect system, the following step must be executed:

```
LOGON SYSCNT2
INSTALL
```

The INSTALL program first checks the version number of each established Software AG product (Natural, Con-form, Teleservices and SNADS) as well as the version number of each Con-nect language installed.

Note: Each time you install a new external mail driver (e.g. Con-nect Teleservices or Con-nect SNADS) you must run INSTALL.

If an upgrade is recommended the following screen is displayed. However, if an upgrade is not recommended or required this screen is not displayed.

```
                      Con-nect 3.4.n Product Validation
                      ---------------------------------

     Currently installed Software AG products:

     Natural                   8.n.n
     Con-form                  3.n.n  Upgrade recommended to 3.4.n
     English language          3.4.n
     German language           3.4.n
     French language           3.4.n
     Teleservices              2.6.n
     SNADS                     1.5.n















 Press ENTER to continue the installation process or press PF3 to Quit.
```

You can choose to stop the installation and complete the recommended upgrade(s), or continue with the installation procedure.

Important: If an upgrade is required, it is also displayed in the screen above. In this case, you are informed that the installation procedure has been stopped. You must perform the required upgrade(s) before you restart the INSTALL program.

Press ENTER to continue.

As a result, the following screen appears, listing all languages which are installed on your system:

```
                    Installation of Con-nect 3.4.n
                    ------------------------------

     Install : Mark with P for Primary Language and with any character for
               Secondary Languages

     PF-keys : Mark to reset existing PF-keys to default

                         Install   PF-keys

     1 English              _         _
     2 German               _         _
     3 French               _         _
     4
     5
     6
     7
     8
     9



 Press ENTER to continue the installation process or press PF3 to Quit.
```

Mark your primary language with P and mark all secondary languages to be used with any non-blank character, except P.

If a language code was previously used for another language, mark the PF-keys to reset them.

Press ENTER to display the "Installing System Profile" screen.

If a previous version of Con-nect has been installed, the values in the "Installing System Profile" screen are taken from cabinet SYSCNT. If you modify the values in this screen, cabinet SYSCNT is automatically updated.

```
                         Installing  System Profile

                     System Name:     * * *  C o n - n e c t  3  * * *

                    CPU Timezone: 0    (relative to Greenwich Mean Time)
 Con-nect Default Sliding Window: 50   (if blank, Natural YSLW is used, however,
                                        if Natural YSLW is 0, 50 is used)
             Days in Wastebasket: 60
           Days until Expiration: 365
           Archive on Expiration: _
             Trace Cabinet Usage: x

                       Day Start: 0     0-23
                      Week Start: 1     1-7

            Logical Printer Name: DAEPRTB2
             Printer Destination: DAEPRTB2

                      Initialize: _ System Command and Object Table

          Con-nect TRS installed: X     (Leave blank if not installed)


 Press ENTER to continue the installation process or press PF3 to Quit.
```

Note: You can use the Con-nect Administration System Maintenance function to modify the information specified in this screen at a later time. See the *Con-nect Administration* documentation for further information on this function.

Specify the following information:

System Name
The name which appears at the top of all screens (the default is: * * * C o n - n e c t 3 * * *).

CPU Timezone
The timezone in which the CPU is located, relative to Greenwich Mean Time.

Con-nect Default Sliding Window
The sliding window determines which two-digit year values belong to which century. The sliding window covers a period of 100 years. The start and end of the sliding window is determined by the value you enter in the "Con-nect Default Sliding window" field. As the range of the sliding window always begins in the past, the value you enter is automatically subtracted from the current year to determine the first year of the window range. To determine the last year of the sliding window, 99 is automatically added to the resulting year. For example, if the current year is 2004 and you enter 40 as the sliding window, the sliding window will cover the years 1964 to 2063. In this case, a two-digit year value (`nn`) from 64 to 99 is interpreted as 19`nn`, while a two-digit year value (`nn`) from 0 to 63 is interpreted as 20`nn`.

You can enter a value from 0 to 99. The value you specify here is supplied as the initial system value defined in the "Con-nect sliding window" field in the "Administration - System Defaults" screen, which is copied to all new cabinets. For further information, see the *Con-nect Administration* documentation, section *System Maintenance*, sub-section *System Defaults*.

If you leave this field blank, the value defined in the Natural YSLW parameter is used. However, if the Natural YSLW parameter is set to 0, the system value for all cabinets is automatically set to 50.

If you specify a value of 0, the sliding window mechanism is not used. This means that when a two-digit year is entered, Con-nect interprets it as in the current century.

Note: To insure that the sliding window covers adequate time in the past and in the present, it is recommended that you do not set the sliding window to either the low or top end of the range. For example, if you set the sliding window to 2, the range of the sliding window will be (assuming the current year is 2004) 2002 to 2101. This means that when a user enters 95 as a two-digit year, Con-nect will interpret it as 2095 and not 1995.

Days in Wastebasket
The number of days an item is kept in the Wastebasket before it can be deleted from the system with a batch utility. This is used as the default value for the utility B04130 Delete Folder Contents. See the *Con-nect Utilities* documentation for further information.

Days until Expiration
The number of days until the expiration date of an item is reached, at which time it can be deleted from the system.

Archive on Expiration
You can enter any non-blank character to specify that an item is to be archived onto a backup medium (tape, disc, etc.) when the expiration date is reached.

Trace Cabinet Usage
You can enter any non-blank character to specify that a record is to be kept of the last cabinet to logon to (use) another cabinet. See the *Con-nect Administration* documentation, section *Reports*, sub-section *Cabinet Usage* for details regarding the reports generated.

Day Start
The start of a 10-hour period within which you expect the greatest frequency of calendar entries. A day in the "Display Month" screen begins with this time. The values mean:

0 - day starts at midnight (0:00) and ends at 10:00

...

8 - day starts at 8:00 and ends at 18:00

...

Week Start

Indicates the first day of the week that is shown in the Con-nect menu and the calendar screens. The values mean:

1 - the first day of the week is Monday

2 - the first day of the week is Tuesday

...

7 - the first day of the week is Sunday.

Logical Printer Name

The name of the default printer in cabinet SYSCNT.

If you are installing Con-nect for the first time (or if that printer has not yet been defined), you should ADD that printer in cabinet SYSCNT after the installation process has been completed.

Printer Destination

The physical printer ID (destination) of the default printer in cabinet SYSCNT.

Initialize _ System Command and Object Table

When you mark this field, the following occurs:

- all user-defined system commands are deleted;
- all user-defined objects are deleted;
- all objects that are delivered with Con-nect are loaded;
- all commands (including aliases) that are delivered with Con- nect are loaded;
- the system cabinet defaults (defined in cabinet SYSCNT) are initialized.
- This function does not affect user aliases.

⚠ **Important:** If you are installing Con-nect 3 for the first time, this field must be marked.

Initialize: _ System Command and Object Table

If you are installing Con-nect Text Retrieval for the first time, mark this field with any character to install the new system object and command tables. In doing so, the new commands for objects CATEGORY, ROOT and SYNONYM are added.

Con-nect TRS installed

Mark with any character if TRS is installed on your system.

Specify all information in the "Installing System Profile" screen and press ENTER to start the installation process for the primary language.

When the installation of the primary language is completed, press ENTER to continue with the secondary languages.

In order to set or change the system defaults, or to perform system administration functions such as adding new users or changing the broadcast message, enter the following:

```
LOGON SYSCNT2
ADMIN <dba<dba>>
```

where the items in pointed brackets are optional. The second occurrence of the string "dba" represents the database administrator's password, which can be changed if required.

Once all information is specified, press ENTER.

4 Invoking Con-nect

Con-nect is contained in the Natural library SYSCNT2 as a Natural system application and the Natural library SYSCNT2B contains the Con-nect utilities.

Con-nect is invoked by executing the Natural transaction MENU. At the beginning of a Con-nect session, Con-nect checks the user's identification and determines the user's authorization, based on the stored Con-nect user definitions. Two parameters, user ID and password, may be provided when the MENU transaction is invoked. For example:

```
LOGON SYSCNT2 MENU
userID password
```

where `userid` and `password` must be substituted with the actual user ID and password.

If only a user ID is provided, Con-nect will prompt the user for the corresponding password.

If no parameters are specified, Con-nect will obtain the user ID from the Natural system variable *INIT-USER (if Natural Security is not in effect) or from the Natural Security user ID (if Natural Security is in effect).

See the *Con-nect User's Guide* for further information.

Possible Problems when Starting Con-nect

Besides the user's identification and user's authorization, Con-nect internally checks installation settings such as version, environment and modules. If something is missing, the user is informed. The following actions may be requested:

- the complete installation of another version of Con-nect,
- re-installation or validation of the copied Con-nect modules, or
- running the program INSTALL (see ***Run INSTALL (Job I100, Step 5020)***).

II Part 2. The Transport Service

This part contains all the information you need to install the Con-nect transport service. It is subdivided into the following topics:

⚠ **Important:** Con-nect must be installed with the required language modules and messages before you can being the installation procedures for the Con-nect transport service.

5 Installing the Transport Service for Execution under Com-plete

This chapter describes step by step how to install the Con-nect transport service for execution under Com-plete using the EntireX Broker Services LU6.2 API (referred to in this section as the LU6.2 API), EntireX Broker Services APPC Adapter (referred to in this section as the LU6.2 ACI) or Remote Database Access.

Note: As an alternative to the LU6.2 API that uses the Natural VIEW COMMUNICATE-LU62, you can now use the LU6.2 ACI. It uses the EntireX Broker ACI interface programs to map requests to LU6.2 verbs, which helps reduce the network traffic. For further information, contact your EntireX Broker Services support representative.

It is recommended that you review the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you begin the installation procedures.

Overview

Before you begin the installation process, you need the following information:

If your server programs use the LU6.2 ACI:

- The name under which the general startup routine will be cataloged in the Com-plete program library (see steps 2 and 6).
- The Broker ID defined for the EntireX Broker (see steps 8 and 9).
- The name of the server which identifies the directory information at the local node (see step 9). This name must correspond with the data set member specified in the EntireX Broker startup parameter APISERV.
- The name of the server which identifies the directory information at the adjacent node (see step 8). This name must correspond with the data set member specified in the EntireX Broker startup parameter APISERV.

If your server programs use the LU6.2 API:

- The LU names which the VTAM system programmer assigned to the local and adjacent systems (see steps 8 and 9).
- The name under which the general startup routine will be cataloged in the Com-plete program library (see steps2 and 6).
- The VTAM log mode name which is used for APPC sessions to the adjacent systems (see steps 8 and 9).
- The node number assigned to the respective LU6.2 API process (see steps 8 and 9).
- The TPNs which identify the corresponding receiving programs at the adjacent nodes (see step 8).

- The TPN which identifies the receiving program at the local node see step **9**).

If your server programs use remote database access:

- The database ID and file numbers of the adjacent systems' Con-nect spool files (see step**8**).
- The name under which the general startup routine will be cataloged in the Com-plete library (see steps **2** and **6**).

To configure and maintain the transport service, see the *Con-nect Administration* documentation, section *The Transport Service*.

Installation Procedures for Com-plete

1. Relinking your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs

This step is performed during the installation of Con-nect. See ***Installation Procedures for Con-nect***.

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) which will be used by the "end user" and "server programs" applications of the transport service. The requirements for the end user application are less stringent than those for the server programs application.

- The following requirements are for the transport service end user application, and will be performed during the installation of Con-nect:
  - The environment must include a fully operational Con-nect system version 3.
  - The environment must contain a valid reference to a Con-nect spool file. The spool file identifier number is 223. This reference can be specified in either the Natural parameter module with a NTLFILE macro call, or dynamically with the Natural keyword LFILE.

    Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The transport service system programs must be loaded into the appropriate Natural system file.
- The Natural keyword CSIZE which specifies the size of the Con-nect buffer must be assigned a value of at least 20.
- The CSTATIC list from the Natural parameter module must specify the module name CTMOD.
- CTMOD must be linked to the Natural nucleus.

The following requirements apply only if event-driven scheduling is used for the transport service inbound queue; that is, if the TS_ROUTER program is to be immediately invoked when a send request is submitted or when the TS_RECEIVE program receives a transport item from another node within the network.

- The Natural nucleus must run in a Com-plete environment.
- The Com-plete function, ATTACH, must be cataloged in the Com-plete library so that it can be accessed by the Natural nucleus.

The following requirements are for the transport service server programs application:

- All of the requirements for the end user application also apply to the server programs application.
- The statement NTDB PROCESS,148 must be added to the Natural parameter module. (Applicable only when the queues are serviced with the LU6.2 API.)
- The Natural PROCESS auxiliary buffer, ASIZE, must reflect the requirements of Software AGs XCOM Communication Module. The recommended value is 28. (Applicable only when the queues are serviced with the LU6.2 API.)
- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
RELO	OFF
MADIO	0
MAXCL	0
AUTO	OFF
ETID	Blank

Tip: The setting ETID=' ' (blank) is recommended but not required. However, it is important to note that if the Natural keyword ETID is set to blank and more than one transport service server program uses the same Natural user ID and they become active simultaneously Adabas will not issue an error message.

2. Modifying the queue server front-end parameter module CTPARMZ

The CTPARMZ module is an Assembler program that uses the keyword macro CTPAR. It is linked to the four transport service queue server front-end programs in order to provide the information required to handle the Natural front-end logic, e.g. the name of the Com-plete transaction code assigned to Natural, and to specify the dynamic parameters that must be passed to the Natural sessions.

Natural parameters that were not included as NATPARMs for the nucleus can be specified dynamically in the CTPARMZ parameter module.

In the CTPARMZ parameter module, you must specify:

- the name used for the general startup front-end program. The supplied source module of the general front-end program is CTCPST;
- the Natural transaction used for the server programs application; and
- appropriate values for the other CTPARMZ keyword parameters that are used by the CTPAR macro (see the table below).

Note: There may be conflicting requirements for the settings of several CTPARMZ keywords, e.g. in the case of simultaneous use of APPC and remote database facilities as transmission media, which result in conflicting requirements concerning the DYNDEMN and DYNSEND keywords. Such collisions can be resolved by creating multiple CTPARMZ-type modules and multiple sets of server front-end programs into which the respective CTPARMZ modules are link-edited. Tieing these procedures to the specific queues can be accomplished by using the concept of scheduling classes and appropriate entries in the "Server Program" field of the queue definitions.

CTPARMZ Keyword Parameters

The following table lists the required CTPARMZ keyword parameters and describes the value associated with each one (M=Mandatory, O=Optional):

Keyword	M	O	Description
TYPE	X		Environment in which the system will operate. Valid value: Com-plete
DYNROUT		X	Dynamic parameter string which leads to the execution of the transport service router program, YR, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNROUT: `STACK=(LOGON SYSCNT2;YR;F),PROGRAM=ctend`
DYNSEND		X	Dynamic parameter string which leads to the execution of the transport service send program, YOX0000, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current outbound queue identifier.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNSEND: `STACK=(LOGON SYSCNT2;YOX0000 %;F),PROGRAM=ctend`
DYNDEMN		X	Dynamic parameter string which leads to the execution of the transport service demon program, YIX7000 (in the case of remote database access), or YIX8DEM (in the case of the LU6.2 API and LU6.2 ACI) in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current receiving queue identifier.

Keyword	M	O	Description
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNDEMN: `STACK=(LOGON SYSCNT2;YIX8DEM %;F),PROGRAM=ctend`
DYNWTCH		X	Dynamic parameter string which leads to the execution of the transport service task re-initiation program, YFWTCH, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current scheduling class identifier.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNWTCH: `STACK=(LOGON SYSCNT2;YFWTCH %;F),PROGRAM=ctend`
DYNTERM		X	Dynamic parameter string which leads to the execution of the transport service termination program, YFSTOP, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. It specifies the transport service session termination program as the back-end program.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNTERM: `STACK=(LOGON SYSCNT2;YFSTOP;F),PROGRAM=ctend`

DYNAPPL		X	Dynamic parameter string which leads to the execution of the transport service target application program and specifies the transport service session termination program as the back-end program. The following substitution symbols which will be replaced by the respective specifications from the application queue definition at execution time can be used:
			`$ Natural Application Name`
			`& Natural Program Name (specify '&&'for one '&')`
			`/ transport service Node Name`
			`% transport service Application Name`

			Example: If "ctend" is the name of the transport service session termination program, DYNAPPL can contain the following parameters (among others): `STACK=(LOGON $;&& /,%;F),PROGRAM=ctend` **Note:** The above example is applicable to a Con-nect external mail application.
SPGRMID	X		Name of the general transport service startup program (source program is CTCPST).
SCHEDCL		X	A one-byte character that is used to classify the various queues which are maintained by the transport service. Specification of SCHEDCL restricts the transport service task re-initiation watchdog program to those queues whose scheduling class specifications match the value given in this parameter. If this parameter is omitted, the transport service task re-initiation program will handle all queues.

3. Specifying CTPARMZ keywords (if required) to perform periodic supervision (watchdog - task re-initiation) of the transport service tasks

The "watchdog" program is generated to periodically supervise the status of the system and attempt to recover it automatically when necessary. When a temporary system failure is detected, the watchdog program records the event in the transport service log records. Then, the watchdog program re-adjusts the status queues and re-starts the queue servers. However, the watchdog program cannot acquire inactive APPC connections.

The following table lists the applicable CTPARMZ keyword parameter and describes the value associated with it:

Keyword	Value
WATCHIN	The length of time (in minutes) between subsequent executions of the watchdog program. The default value is 15.

4. Assembling and linking (if required) the transport service Adabas interface module

Assign the transport service and Com-plete source data sets as macro libraries before assembling and linking the transport service Adabas interface module.

The transport service Adabas interface module, CTCPDB, is a subroutine which permits Adabas OPEN (OP) and CLOSE (CL) commands to be executed from within the front- and back-end programs. The interface module is a member of the transport service source data set and must be link-edited with each of the front-end programs and the session termination program.

It is not necessary to use the Adabas interface module unless your transport service front- and back-end programs are to issue OP and CL commands.

The following table lists the applicable CTPARMZ keyword parameters and describes the value associated with each one:

Keyword	M	O	Description
ADADBID		X	Specify the database number for which OP and CL commands are to be issued. If files in multiple databases are to be opened and closed (using the OP and CL commands), the database numbers must be specified in subparameter notation.
			Example: OP and CL commands are directed to database 2 with the following:

Keyword	M	O	Description

			`ADADBID=2`
			OP and CL commands are directed to databases 10 and 15 with the following:
			`ADADBID=(10,15)`

ADAUID		X	Specify whether the startup programs will generate an Adabas user ID when OP commands are issued. Adabas user IDs are not intended to be used when ETID=' ' (blank) and therefore, this keyword should not be specified. Default value: No
ADAOPRB		X	Specify the record buffer contents for the OP commands. If this parameter has not been specified, OP commands will not be issued. If OP commands are to be directed to multiple databases, multiple record buffer contents must be specified in subparameter notation. The number of all Natural and Con-nect system files which are used should be specified.

			Example: An OP command is directed to database number 2 with the record buffer contents 'UPD=4,8,12.' with the following:
			`ADADBID=2,ADAOPRB='UPD=4,8,12.'`
			However, an OP command is directed to databases 5 and 15 with the record buffer contents 'UPD=3.6.' and 'ACC.' with the following:
			`ADADBID=(5,10,15),ADAOPRB=('UPD=3,6.',,'ACC.')`
			OP commands are not directed to database 10.

ADACLSE		X	Specify whether the back-end (session termination) program CTCPTE will issue CL commands at the end of a session. Default value: No

5. Assembling the CTPARMZ parameter module using the transport service source data set as a macro library

6. Assembling and link-editing the queue server front- and back-end programs

The names of the supplied front- and back-end programs for the queue server functions are as follows:

Program	Queue Server Function
CTCPIN	Session re-initiation program
CTCPST	General start-up program
CTCPTE	Session termination program (back-end program)
CTCPTP	Termination program

Assign the appropriate Com-plete source data set and the transport service source data set as macro libraries, and assemble the queue server front- and back-end programs.

- Reference the Com-plete library, link-edit the queue server front- and back-end programs and CTCPDB, if required, with the CTPARMZ parameter module.

Note: If OP and CL commands are to be issued, the transport service Adabas interface module CTCPDB must be link-edited with each of the front- and back-end programs.

- Ensure that an execution will start with the first instruction of the queue server front- and back-end programs (by placing the queue server front- and back-end programs at the beginning of the load modules, or by specifying the appropriate ENTRY statements).
- Catalog the queue server front-end programs to the Com-plete library. No specifications are required for the region size and the programs do not need to be marked as privileged. However:
  - the name assigned to the CTCPST program must be identical to the "Server Program" specification in the respective transport service queue definitions and the value specified for the CTPARMZ keyword SPGRMID.
  - the name you specify for the CTCPTE program must be identical to the value specified for the PROGRAM parameter in the DYNROUT, DYNSEND, DYNDEMN, DYNWTCH, DYNTERM, and DYNAPPL CTPARMZ keywords.

7. Initializing the transport service

Initializing the transport service consists of two parts: defining the transport service node ID and the actual initialization.

How to initialize the transport service is described in the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Initializing the Transport Service*.

Note: If you have not already done so, it is recommended that you review the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you continue with the installation procedures. The next five steps define the path the transport items take bet

The user exit program YENTX is used to switch on the LU6.2 ACI. If this program has been renamed to YENTXACI, then the transport service will use the LU6.2 ACI to send/receive messages and not the LU6.2 API.

8. Creating one or multiple outbound queues for the storing of transport items intended for other nodes

At least one outbound queue must be created for each adjacent node in the network. When an outbound queue is created, the attributes of the LU6.2 API, LU6.2 ACI or remote database access, the link between the local and adjacent node, must be defined.

LU6.2 ACI Parameters

If you are using LU6.2 ACI, the server name of the outbound queue must correspond with the data set member specified in the EntireX Broker startup parameter APISERV. This member contains specific LU6.2 information for the adjacent node. The following table lists the parameters that are used:

Parameter	Description
DRIVER=COMMLU62	Specifies the LU6.2protocol.
TPNAME=	The TPN which identifies the corresponding receiving programs at the adjacent node. ■ If the adjacent node uses EntireX Broker LU6.2 ACI in Com-plete, this is the name of the general startup routine. ■ If the adjacent node uses EntireX Broker LU6.2 ACI in batch mode, this is the unique name used to identify the server. ■ If the adjacent node uses CICS LU6.2, this is the transaction code under which the receive program has been defined in the CICS PCT (source module name CTCIRC).
LUNAME=	The LU name which the VTAM system programmer assigned to the adjacent system.
MODENAME=	Specifies the VTAM LOGMODE name used for APPC sessions to establish a communication link to the adjacent system.
SYNCLEVEL=Y	Specifies confirmation level synchronization is used.
TRANSLATION=	Specifies the name of the exit routine used to translate data contained in the SEND/RECEIVE buffers.
LOCAL-LU=	Specifies the VTAM LOCAL-LU parameter.

To create one or more outbound queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Outbound Queues*.

9. Creating one or multiple receiving queues to control the demon processes

At least one receiving queue must be created if the local node will receive transport items by means of Software AG's LU6.2 API, LU6.2 ACI or remote database access. The receiving queue is used to control the demon processes which continuously listen for transaction requests from adjacent nodes or poll an intermediate queue in the case of remote database access.

To create a receiving queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Receiving Queues*.

LU6.2 ACI Parameters

If you are using LU6.2 ACI, the server name of the receiving queue must correspond with the data set member specified in the EntireX Broker startup parameter APISERV. This member contains specific LU6.2 information for the local node. The following table lists the parameters that are used:

Parameter	Description
DRIVER=COMMLU62	Specifies the LU6.2 protocol.
TPNAME=	The TPN which identifies the general startup routine at the local node.
LUNAME=	The LU name which the VTAM system programmer assigned to the local system. Omit this parameter if it is identical to either the LUNAME specified in the LOCAL-LU keyword or the EntireX Broker startup parameter DEFLUNAM.
MODENAME=	Specifies the VTAM LOGMODE name used by the adjacent system for APPC sessions to establish a communication link to the local system.
SYNCLEVEL=Y	Specifies confirmation level synchronization is used.
TRANSLATION=	Specifies the name of an exit routine used to translate data contained in the SEND/RECEIVE buffers.
LOCAL-LU=	Specifies the VTAM LOCAL-LU parameter.

10. Creating an application queue for incoming transport items that are to be passed to a local application

An application queue must be created for each application at the local node which will use the transport service. Transport items which are sent to the application are placed in these queues and remain there until they are processed by the local application.

When you create an application queue for Con-nect external mailing, specify the following on the "Application Queue" screen:

- In the `Node ID` field, enter the name you specified as your transport service ID. See ***Initializing the transport service***.
- In the `Application Name` field, enter A.
- In the `Application Library` field, enter SYSCNT2.

- In the `Application Program` field, enter YCINITO.

To create an application queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Application Queues*.

11. Setting the "Reset" and "Input" statuses for the inbound, outbound and application queues

When you initially test the transport service, the input status of the inbound, outbound and application queues must be set to active and the reset status to inactive. To set the input and reset statuses of the queues, perform the following steps; they must be repeated for each queue:

- Select the Queue Maintenance function from the "Transport Service Administration" screen to display the "Queue Maintenance" screen.
- Enter MO (modify) in the Cmd column for the appropriate queue and press ENTER. The corresponding screen is displayed. Modify the following fields:
  - In the `Reset Status` field, enter I (inactive).
  - In the `Input Status` field, enter A (active).
  - Press PF9.

For further information about the "Queue Maintenance" screen, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Queue Maintenance*.

12. Defining routing specifications for all other nodes with which you intend to communicate

For each node in your network with which you want to communicate, you must define a routing entry which determines how transport items are to be routed to that node.

To define a routing entry, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Routing Entries*.

13. Installing an external mail application which is based upon the transport service

Before you can test the installation of the transport service, you must install an external mail application which is based upon the transport service. See the appropriate installation manuals.

14. Testing the installation of the transport service

Once you have installed an external application which uses the transport service (e.g. multi-node Con-nect), you can test the installation of the transport service.

The following test uses Con-nect as an example.

When testing, use the following settings for **ALL** queues:

Input Status: A

Output Status: I

Reset Status: I

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you start a queue server when either the outbound or inbound queue is in the "hold" status, the queue server task will immediately exit without processing any information.

- Send a mail item from multi-node Con-nect or another external mail application based upon the transport service method and request a delivery notification. Then check the inbound queue to see if the transport item is there.
- Start the inbound queue from the "Queue Maintenance" screen. This activates the "router" which should route the transport item to the outbound queue. Since the outbound queue is inactive, the transport item should remain in the outbound queue. You can keep track of the ongoing sequence of events which are recorded in the `Output Status` field by pressing ENTER. Starting the queue causes the `Output Status` field to change from "I" to " " (scheduled), which means that your Natural session issued a Com-plete ATTACH command to start the general startup front-end program. This program accesses the specifications in CTPARMZ, such as those in the keywords NPGRMID and DYNROUT, in order to pass control to the Natural program, YR, which executes within a non-interactive Natural session. YR, at the beginning of its execution, changes the `Output Status` field to "A" and, at the end of its execution, back to "I" (if successfully terminated) or to "H" (if terminated due to unrecoverable errors). In both cases (successful and unsuccessful termination), you can check the log records in order to obtain additional information. If the status is "hung" in scheduled mode " ", this indicates that the startup of the asynchronous server program failed, typically due to an incorrect setup in CTPARMZ.
- Check to see if the transport item was routed to the outbound queue. (The same test procedure can be applied to the other queues as well.) Up to this point, LU6.2 or remote database access have not been involved. The following steps allow you to test sending a transport item to another node.
- Set the outbound queue with the routed transport item to the "inactive" status and start the outbound queue from the "Queue Maintenance" screen.

- Check the log records. The log records are accessed with the Log Information Maintenance function on the "Transport Service Administration" screen. If the transport item was successfully sent, records from the TS_SEND program YOX8000 (in the case of LU6.2 API), YOX8000B (in the case of LU6.2 ACI) or YOX7000 (in the case of remote database access) are displayed.
- Check for log records which were created when a status report was received from the target node. Records from the TS_RECEIVE program YIX8000 (in the case of LU6.2 API), YIX8000B (in the case of LU6.2 ACI) or YIX7000 (in the case of remote database access) should be displayed.
- Check to see whether the status report was delivered to the appropriate application queue.
- If you experience problems, the following commands and functions can be useful for debugging purposes: VTAM Operator Console Commands, EntireX Broker Services Operator Commands, and Transport Service Administration Functions.

  The following functions are offered on the "Transport Service Administration" screen:

  Queue Maintenance - Provides a transport item count, time stamps for recent activity, and the ability to manually start the queue server.

  Log Information Maintenance - Provides information that can be helpful to you or, in the case of severe malfunctions, to Software AG, in resolving any problems you experience with the transport service.

6

Installing the Transport Service for Execution in Batch Mode

This chapter describes step by step how to install the Con-nect transport service for execution in batch mode using the EntireX Broker Services LU6.2 API (referred to in this section as the LU6.2 API), EntireX Broker Services APPC Adapter (referred to in this section as the LU6.2 ACI) or Remote Database Access.

> **Note:** As an alternative to the LU6.2 API that uses the Natural VIEW COMMUNICATE-LU62, you can now use the LU6.2 ACI. It uses the EntireX Broker ACI interface programs to map requests to LU6.2 verbs, which helps reduce the network traffic. For further information, contact your EntireX Broker Services support representative.

It is recommended that you review the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you begin the installation procedures.

Overview

Before you begin the installation process, you need the following information:

If your server programs use the LU6.2 API:

- The LU names which the VTAM system programmer assigned to the local and adjacent systems (see steps 3 and 4).
- The VTAM log mode name which is used for APPC sessions to the adjacent systems (see steps 3 and 4).
- The node number assigned to the respective LU6.2 API process (see steps 3 and 4).
- The TPNs which identify the corresponding receiving programs at the adjacent nodes (see step 3).
- The TPN which identifies the receiving program at the local node (see step 4).

If your server programs uses the LU6.2 ACI:

- The broker ID for the EntireX Broker (see steps 3 and 4).
- The name of the server which identifies the directory information at the local node (see step 4). The name specified must correspond with the member that exists in the data set specified in the EntireX Broker startup parameter APISERV.
- The name of the server which identifies the directory information at the adjacent node (see step 3). This name must correspond with the data set member that specified in the EntireX Broker startup parameter APISERV.

If your server programs use remote database access:

- The database ID and file numbers of the adjacent systems' Con-nect spool files (See step 3).

To configure and maintain the transport service, see the *Con-nect Administration* documentation, section *The Transport Service*.

Installation Procedures for Batch Mode

1. Relinking your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs

This step is performed during the installation of Con-nect. See ***Installation Procedures for Con-nect***.

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) which will be used by the "end user" and "server programs" applications of the transport service. The requirements for the end user application are less stringent than the requirements for the server programs application.

- The following requirements are for the transport service end user application and will be performed during the installation of Con-nect:
  - The environment must include a fully operational Con-nect system version 3.
  - The environment must contain a valid reference to a Con-nect spool file. The spool file identifier number is 223. This reference can be specified in the Natural parameter module with a NTLFILE macro call, or dynamically with the Natural keyword LFILE.

  Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The transport service system programs must be loaded into the appropriate Natural system file.
- The Natural keyword CSIZE which specifies the size of the Con-nect buffer, must be assigned a value of at least 20.
- The CSTATIC list from the Natural parameter module must specify the module name CTMOD.
- CTMOD must be linked to the Natural nucleus.

The following requirements are for the transport service server programs application:

- All of the requirements for the end user application also apply to the server programs application.
- The statement NTDB PROCESS, 148 must be added to the Natural parameter module. (Applicable only when the queues are serviced with the LU6.2 API.)
- The Natural PROCESS auxiliary buffer, ASIZE, must reflect the requirements of Software AG's XCOM Communication Module. The recommended value is 28. (Applicable only when the queues are serviced with the LU6.2 API.)
- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
MADIO	0
MAXCL	0
AUTO	OFF
ETID	Blank

Tip: The setting ETID=' ' (blank) is recommended but not required. However, it is important to note that if the Natural keyword ETID is set to blank and more than one transport service server program uses the same Natural user ID and they become active simultaneously Adabas will not issue an error message.

2. Initializing the transport service

Notes:

1. If you have not already done so, it is recommended that you see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you continue with the installation procedures.
2. The user exit program YENTX is used to turn on the LU6.2 ACI. If the program has been renamed to YENTXACI, then the transport service uses the LU6.2 ACI to send/receive messages and not the LU6.2 API.

Initializing the transport service consists of two parts: defining the transport service node ID and the actual initialization. See the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Initializing the Transport Service*.

Note: The next steps define the path the transport items take from Con-nect to other nodes. If changes are made in your SNA network, you may need to re-execute the following steps.

3. Creating one or multiple outbound queues for the storing of transport items intended for other nodes

At least one outbound queue must be created for each adjacent node in the network. When an outbound queue is created, the attributes of the LU6.2 API, LU6.2 ACI or remote database access link between the local and the adjacent node must be defined.

LU6.2 ACI Parameters

If you are using LU6.2 ACI, the server name of the outbound queue must correspond with the member that exists in the data set specified in the EntireX Broker startup parameter APISERV. This member contains specific LU6.2 information for the adjacent node. The following table lists the parameters that are used:

Parameter	Description
DRIVER=COMMLU62	Specifies the LU6.2 protocol.
TPNAME=	The TPN which identifies the corresponding receiving programs at the adjacent node. ■ If the adjacent node uses EntireX Broker LU6.2 ACI in Com-plete, this is the name of the general startup routine. ■ If the adjacent node uses EntireX Broker LU6.2 ACI in batch mode, this is the unique name used to identify the server. ■ If the adjacent node uses CICS LU6.2, this is the transaction code under which the receive program has been defined in the CICS PCT (source module name CTCIRC).
LUNAME=	The LU name which the VTAM system programmer assigned to the adjacent system.
MODENAME=	Specifies the VTAM LOGMODE name used for APPC sessions to establish a communication link to the adjacent system.
SYNCLEVEL=Y	Specifies confirmation level synchronization is to be used.
TRANSLATION=	Specifies the name of the exit routine used to translate data contained in the SEND/RECEIVE buffers.
LOCAL-LU	Specifies the VTAM LOCAL-LU parameter.

To create one or more outbound queue, see the *Con-nect Administration* documentation, section *The Transport Service*, section *Outbound Queues*.

4. Creating one or multiple receiving queues to control the demon processes

At least one receiving queue must be created if the local node will receive transport items by means of the LU6.2 API, LU6.2 ACI or remote database access. The receiving queue is used to control the demon processes which continuously listen for transaction requests from adjacent nodes or poll an intermediate queue in the case of remote database access.

LU6.2 ACI Parameters

If you are using LU6.2 ACI, the server name of the receiving queue must correspond with the member that exists in the data set specified in the EntireX Broker startup parameter APISERV. This member contains specific LU6.2 information for the local node. The following table lists the parameters that are used:

Parameter	Description
DRIVER=COMMLU62	Specifies the LU6.2 protocol.
TPNAME=	The unique name used to identify the server at the local node. It must be the same as that specified in the Server Program field in the "EntireX LU6.2 ACI Receiving Queue" screen.
LUNAME=	The LU name which the VTAM system programmer assigned to the local system. Omit this parameter if it is the same LUNAME as specified with the LOCAL-LU keyword or the DEFLUNAM EntireX Broker startup parameter.
MODENAME=	Specifies the VTAM LOGMODE name used by the adjacent system for APPC sessions to establish a communication link to the local system.
SYNCLEVEL=Y	Specifies confirmation level synchronization is used.
TRANSLATION=	Specifies the name of an exit routine used to translate data contained in the SEND/RECEIVE buffers.
LOCAL-LU=	Specifies the VTAM LOCAL-LU parameter.

To create a receiving queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Receiving Queues*.

5. Creating an application queue for incoming transport items that are to be passed to a local application

An application queue must be created for each application at the local node which will use the transport service. Transport items which are sent to the application are placed in these queues and remain there until they are processed by the local application.

When you create an application queue for Con-nect external mailing, specify the following on the "Application Queue" screen:

- In the "Node ID" field, enter the name you specified as your transport service ID. See ***Initializing the transport service***.
- In the "Application Name" field, enter A.
- In the "Application Library" field, enter SYSCNT2.
- In the "Application Program" field, enter YCINITO.

To create an application queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Application Queues*.

6. Setting the "Reset", and "Input" statuses for the inbound, outbound and application queues

When you initially test the transport service, the input status of the inbound, outbound and application queues must be set to active, and the reset status to inactive. To set the input and reset statuses of the queues, perform the following steps; they must be repeated for each of the above mentioned queues:

- Select the Queue Maintenance function from the "Transport Service Administration" screen to display the "Queue Maintenance" screen.
- Enter MO (modify) in the Cmd column for the appropriate queue and press ENTER. The corresponding screen is displayed. Modify the following fields:
  - In the "Reset Status" field, enter I (inactive).
  - In the "Input Status" field, enter A (active).
  - Mark the "Reset" field with any character and press PF9.

For further information about the "Queue Maintenance" screen, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Queue Maintenance*.

7. Defining the JCL server task procedures

- The input handler is used as an interface to the LU6.2 API, LU6.2 ACI or to read mail items which have been copied from a remote database to an intermediate queue and initiates the demon processes. When the demon processes are initiated, they listen for transaction requests from remote partner nodes. When a request is caught, the demon processes execute the TS_RECEIVE program. The input handler procedure consists of the execution of the Natural nucleus in batch mode.

  The following is an example of a JCL input handler procedure in the case of the LU6.2 API or LU6.2 ACI:

  ```
  //CMSYNIN DD *
  LOGON SYSCNT2
  YIX8DEM xxxx
  FIN
  /*
  ```

  where *xxxx* is the name of the receiving queue. To stop the input handler procedure you must set the status of the dummy queue to H (hold).

  Note: If remote database access is used, YIX8DEM is replaced with YIX7DEM.

- The queue server processes the inbound, outbound, and application queues. A queue server can be specified to handle one or a multiple of queues.

The following is an example of a JCL queue server procedure:

```
//CMSYNIN DD *
LOGON SYSCNT2
YSCHED 2 x
FIN
/*
```

where 2 is the length of time (in minutes) between subsequent processing of the specified queues and *x* is the scheduling class.

The queue server processes will process all inbound, outbound and application queues whose scheduling match the value specified for *x* and whose output status is I (inactive). If a scheduling class has not been specified for YSCHED (i.e. *x* is empty), the queue server processes will process all queues.

To stop the queue server procedure you must set the statuses of all queues, specified within the procedure, to H (hold).

8. Defining routing specifications for all other nodes with which you intend to communicate

For each node in your network with which you want to communicate, you must define a routing entry which determines how transport items are to be routed to that node.

To define a routing entry, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Routing Entries*.

9. Installing an external mail application which is based upon the transport service

Before you can test the installation of the transport service, you must install an external mail application which is based upon the transport service. For further information, see the appropriate installation manuals.

10. Testing the installation of the transport service

Once an external application which uses the transport service has been installed, you can test the installation of the transport service.

The following test uses multi-node Con-nect as an example.

When testing, use the following settings for the inbound and receiving queue:

Input Status: A

Output Status: I

Reset Status: I

All other queues must be set to the following:

Input Status: A

Output Status: H

Reset Status: H

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you try to start a queue server when either the outbound or inbound queue is in the "hold" status, the queue server task will immediately exit without processing any information.

- Start the input handler procedure. This activates the demon processes so that they listen for transmission requests from remote Con-nect nodes.
- Send a mail item from Con-nect or another external mail application based upon the transport service method and request a delivery notification. Then check the inbound queue for the transport item.
- Start the JCL server procedure for the inbound queue YSCHED program with the scheduling class. This activates the "router", which should route the transport item to the outbound queue. Since the outbound queue is set in the hold status, the transport item should remain in the outbound queue. As soon as the appropriate Natural queue server program has been started by YSCHED (in the case of the inbound queue's router program, it is YR), that program will change the output status from I to A, and, at the end of its execution, back to I (if successfully terminated) or to H (if terminated due to errors). In both cases (successful and unsuccessful termination), you can check the log records in order to obtain additional information.
- Check to see if the transport item was routed to the outbound queue. (The same test procedure can be applied to the other queues as well.) Up to this point, LU6.2 or remote database access have not been involved. The following steps allow you to test sending a transport item to another node.
- Set the respective outbound queue status to the following: Input Status: A Output Status: I Reset Status: I and the scheduling class to the same value as the inbound queue, thus, causing the server to process the outbound queue.
- Check the log records. The log records are accessed from the Log Information Maintenance function on the "Transport Service Administration" screen. If the transport item was successfully sent, records from the TS_SEND program, YOX8000 (in the case of the LU6.2 API), YOX8000B (in the case of the LU6.2 ACI) or YOX7000 (in the case of remote data base access), are displayed.
- Check for log records which were created when a status report was received from the target node. Records from the TS_RECEIVE program YIX8000 (in the case the LU6.2 API), YIX8000B (in the case of the LU6.2 ACI) or YIX7000 (in the case of remote database access) should be displayed.
- Check to see whether the status report was delivered to the appropriate application queue.

- If you are still experiencing problems, the following commands and functions can be useful for debugging purposes: VTAM Operator Console Commands, EntireX Broker Services Operator Commands, and the Transport Service Administration Functions.

  The following functions can be performed from the Transport Service Administration Menu:

  Queue Maintenance - Provides a transport item count, time stamps for recent activity, and the ability to manually start the queue server.

  Log Information Maintenance - Provides information that can be helpful to you or, in the case of severe malfunctions, to Software AG, in resolving any problems you experience with the transport service.

7

Installing the Transport Service for Execution in CICS

This chapter describes step by step how to install the Con-nect transport service for execution in CICS using CICS APPC Facilities or Remote Database Access.

It is recommended that you review the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you begin the installation procedures.

Overview

Before you begin the installation process you need the following information, depending upon the type of transmission format your server programs use.

If your server programs use CICS LU6.2:

- The local system's TCT system entries which define the SNA LU6.2 communication link to the respective adjacent nodes (see step **10**).
- The transaction codes and program names under which the general server startup front-end program and the receive front-end program will be defined in the CICS PCT and PPT. (See steps **3**, **4** and **8**).
- The transaction code and program name assigned to Natural in the CICS PCT and PPT (see step **3**).
- The communication profile which is used while establishing a LU6.2 communication link to the adjacent node (see step **8**).
- The TPN which identifies the corresponding receiving program at the adjacent node (see step **10**).

If your server programs use a remote database access:

- The database ID of the adjacent nodes' system spool files (see step **10**).
- The file number of the adjacent nodes' system spool files (see step **10**).
- The transaction codes under which the general server startup front-end program will be defined in the CICS PCT. (see steps **3** and **8**).
- The transaction code and program name assigned to Natural in the CICS PCT and PPT (see **3**).

To configure and maintain the transport service, see the *Con-nect Administration* documentation, section *The Transport Service*.

Installation Procedures for CICS

1. Assembling and link-editing the CICS command-level program CXLU62 (the transport service interface to CICS services)

CXLU62 is the transport service communication module and is located on the CNT*nnn*.SRCE data set. The CXLU62 module is written in Assembler and uses CICS commands.

- Process the CXLU62 communication module using the CICS command language processor DFHEAP1$.
- Assemble the translated output using the appropriate CICS, Natural and Natural CICS source data sets as macro libraries.
- Link-edit the CXLU62 module. Since a BALR instruction is used to branch to the CXLU62 module, rather than a CICS LINK or XCTL command), the system administrator must use one of the following three options to link-edit the CXLU62 module:
    - Option one: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is part of a Natural CICS nucleus and CXLU62 is linked to the Natural CICS nucleus. In this case, the CXLU62 module uses the CICS exec interface stub included with the Natural CICS nucleus.

      With option one the CXLU62 module should not be linked with the CICS exec interface stub modules.
    - Option two: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is part of a Natural CICS nucleus and CXLU62 is not linked to the Natural CICS nucleus. In this case, the CXLU62 module is loaded by CICS each time a Natural session is initialized by using the Natural RCA (Resolve CSTATIC Addresses) feature. The setting RCA=ON is used to activate the RCA feature.

      With option two the CXLU62 module must be linked with the CICS exec interface stub modules. Since Natural will branch to the beginning of the CXLU62 load module, the CSECTs of the load module must be linked in the following sequence: CXLU62, DFHEAI, and then DFHEAI0. The name of the resulting load module must be CXLU62.

      Next, the CXLU62 module must be copied into the Relocatable Program Library (RPL) of the CICS nucleus and referenced by an appropriate PPT entry. The PPT entry must indicate that CXLU62 is resident. Additionally, if you are running CICS/ESA 3.3 or above and this module is intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.
    - Option three: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is not part of a Natural CICS nucleus but is, nevertheless, loaded at Natural session initialization by the Natural keyword PARM.

With option three, both the CXLU62 module and CICS exec interface stub modules must be linked with the Natural parameter module. Since Natural assumes the parameter module referenced in the PARM keyword is located at the beginning of the load module, the CSECTs that comprise the Natural parameter module must be located at the beginning of the resulting load module.

Additionally, the module must be copied into the Relocatable Program Library (RPL) of the CICS nucleus and referenced by an appropriate PPT entry. The PPT entry must indicate that the module is resident. If you are running CICS/ESA 3.3 or above and this module is intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.

- The linkage editor may flag a few unresolved external references when you link-edit a Natural parameter module that is designed to be loaded dynamically. This is unavoidable and will not cause any problems. If CICS refuses to load a module that has been marked as "not executable", specify either the linkage editor NCAL or the LET option in order to force the module to be marked as executable.

2. Relinking your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs, the interface module CXLU62

This step is performed during the installation of Con-nect. See *Link-edit Natural/Con-nect Nucleus*.

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) to be used by the end user and server programs applications of the transport service. The requirements for the end user application are less stringent than the requirements for the server programs application.

- The following requirements are for the transport service end user application:
  - The environment must include a fully operational Con-nect version 3 system.
  - The environment must contain a valid reference to a Con-nect spool file with the identifier number 223. This reference can be specified in the Natural parameter module with a NTLFILE macro call, or dynamically using the Natural keyword LFILE.

Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The transport service system programs must be loaded into the appropriate Natural system file.
- The Natural keyword CSIZE, which specifies the size of the Con-nect buffer, must be assigned a value of at least 20.
- The CSTATIC list from the Natural parameter module must specify the module name CTMOD.
- The CTMOD module must be linked to the Natural nucleus.

Important: The following requirements only apply if event-driven scheduling is used for the transport service inbound queue; that is, if the TS_ROUTER program is to be immediately invoked when a send request is submitted or when the *TS* RECEIVE program receives a transport item from another node within the network.

- The Natural nucleus must run in a CICS environment.
- The CSTATIC list from the Natural parameter module must specify the module name CXLU62.
- The CXLU62 module must be either linked to or accessible to the Natural nucleus.

The following requirements are for the transport service server programs application:

- All of the requirements for the end user application also apply to the server programs application.
- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
PSEUDO	OFF
RELO	OFF
MADIO	0
MAXCL	0
AUTO	OFF
ETID	Blank

Tip: The setting ETID=' ' (blank) is recommended but not required. However, it is important to note that if the Natural keyword ETID is set to blank and more than one transport service server program uses the same Natural user ID and they become active simultaneously Adabas will not issue an error message.

3. Modifying the queue server front-end parameter module CTPARM2

The CTPARM2 module is an Assembler program that uses the keyword macro CTPAR. It is linked to the four transport service queue server front-end programs in order to provide the information required to handle the Natural front-end logic, e.g. the name of the resident Natural CICS nucleus and to specify the dynamic parameters that must be passed to the Natural sessions.

Natural parameters that were not included as NATPARMs for the nucleus can be specified dynamically in the CTPARM2 parameter module.

In the CTPARM2 parameter module, you must specify:

- the name used for the general startup front-end program. The supplied source module of the general front-end program is CTCIST;
- the Natural transaction used by the transport service;
- the Natural transaction used for the server programs application; and

- appropriate values for the other CTPARM2 keyword parameters that are used by the CTPAR macro (see the table below).

Important: Conflicting requirements for the settings of certain CTPARM2 keywords may occur, e.g. simultaneous use of APPC and Remote Database facilities as transmission media will result in conflicting requirements for the DYNDEMN and DYNSEND keywords. Such collisions can be resolved by creating multiple CTPARM2-type modules and multiple sets of server front-end programs into which the respective CTPARM2 module is link-edited. Tieing these procedures to the specific queues can be accomplished by using the concept of scheduling classes and appropriate entries in the "Server Program" field of the queue definitions.

CTPARM2 Keyword Parameters

The following table lists the required CTPARM2 keyword parameters and describes the value associated with each one:

Keyword	M	O	Description
TYPE	X		Environment in which the system will operate.
			Valid value: CICS
DYNROUT		X	Dynamic parameter string which leads to the execution of the transport service router program, YR, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNROUT: `STACK=(LOGON SYSCNT2;YR;F),PROGRAM=ctend`
DYNSEND		X	Dynamic parameter string which leads to the execution of the transport service send program, YOX0000, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current outbound queue identifier.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNSEND: `STACK=(LOGON SYSCNT2;YOX0000 %;F),PROGRAM=ctend`
DYNRECV		X	Dynamic parameter string which leads to the execution of the transport service APPC receive program, YIX9000, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNRECV:

Keyword	M	O	Description
			`STACK=(LOGON SYSCTN2;YIX9000;F),PROGRAM=ctend`
DYNDEMN		X	Dynamic parameter string which leads to the execution of the transport service demon program, YIX7000 (in the case of remote database access) in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current receiving queue identifier.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNDEMN: `STACK=(LOGON SYSCNT2;YIX7000 %;F),PROGRAM=ctend` **Note:** DYNDEMN is to be specified only if remote database access is used.
DYNWTCH		X	Dynamic parameter string which leads to the execution of the transport service task re-initiation program, YFWTCH, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program. The percent sign (%) is substituted at execution time by the current scheduling class identifier.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNWTCH: `STACK=(LOGON SYSCNT2;YFWTCH %;F),PROGRAM=ctend`
DYNTERM		X	Dynamic parameter string which leads to the execution of the transport service termination program, YFSTOP, in the SYSCNT2 application and specifies the transport service session termination program as the back-end program.
			Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNTERM: `STACK=(LOGON SYSCNT2;YFSTOP;F),PROGRAM=ctend`

DYNAPPL		X	Dynamic parameter string which leads to the execution of certain transport service target application programs and specifies the transport service session termination program as the back-end program. The following substitution symbols, which will be replaced by the respective specifications from the application queue definition at execution time, can be used:
			`$ Natural Application Name`
			`& Natural Program Name (specify '&&' for one '&')`
			`/ transport service Node Name`
			`% transport service Application Name`

			Example: If "ctend" is the name of the transport service session termination program, DYNAPPL may contain the following parameters (among others):

Keyword	M	O	Description
			`STACK=(LOGON $;&& /,%;F),PROGRAM=ctend` **Note:** The above example is suitable for a Con-nect external mail application
NPGRMID	X		The name of the resident Natural CICS nucleus as defined in the CICS PPT. If the "OS Core" load option is used, this is not the same name as the Natural CICS nucleus load module.
NTRANID	X		The name of the CICS transaction code assigned to the Natural CICS nucleus in the CICS PCT.
SPGRMID	X		Name of the general transport service startup program (CTCIST) as defined in the CICS PPT.
SCHEDCL		X	A one-byte character that is used to classify the various queues which are maintained by the transport service. Specification of SCHEDCL restricts the transport service task re-initiation watchdog program to those queues whose scheduling class specifications match the value given in this parameter. If this parameter is omitted, the transport service task re-initiation program will handle all queues.

4. Specifying keyword parameters (if required) to perform periodical supervision (watchdog - task re-initiation) of the transport service tasks

The watchdog program is generated to periodically supervise the status of the system and attempt to recover it automatically when necessary. When a temporary system failure is detected, the watchdog program records the event in the transport service log records. Then, the watchdog program re-adjusts the status queues and re-starts the queue servers. However, the watchdog program cannot acquire inactive APPC connections.

The following table lists the applicable CTPARM2 keyword parameters and describes the value associated with each one:

Keyword	Description
WTRANID	The CICS transaction code which is assigned to the CTCIIN program.
WATCHIN	The length of time (in minutes) between subsequent executions of the watchdog program.
	Default value: 15
WATCHRQ	A character string used by the watchdog. The watchdog program occupies one CICS request ID, which is specified here. The request ID specified must not conflict with other CICS request IDs.
	Default value: WDOG

5. Assembling and linking (if required) the transport service Adabas interface module

Before assembling the transport service Adabas interface module, process CTCIDB using the CICS command language translator DFHEAP1$ with the default options PROLOG and EPILOG.

The transport service Adabas interface module is a subroutine which permits Adabas OPEN (OP) and CLOSE (CL) commands to be executed from within the front-end programs. The interface module CTCIDB is a member of the transport service source data set and must be link-edited with each of the front-end programs and the session termination program.

It is not necessary to use the Adabas interface module unless your transport service front-end programs are to issue OP and CL commands. The functionality can be useful if the value of the Natural keyword ETID is set to blank.

The following table lists the applicable CTPARM2 keywords and describes the value associated with each one:

Keyword	M	O	Description
ADADBID		X	Specify the database number for which OP and CL commands are to be issued to. If files in multiple databases are to be opened and closed, using the OP and CL commands, the database numbers must be specified in subparameter notation.

			Example: OP and CL commands will be directed to database 2 with the following:
			`ADADBID=2`
			OP and CL commands will be directed to databases 10 and 15 with the following:
			`ADADBID=(10,15)`
			In this example, OP and CL commands will be directed to databases 10 and 15:
			`ADADBID=(5,10,15),ADAOPRB=('UPD=3,6.',,'ACC.')`

ADAOPRB			Specify the record buffer contents for the OP command. If this parameter has not been specified, OP commands will not be issued.
			If OP commands are to be directed to multiple databases, multiple record buffer contents must be specified in subparameter notation. The number of all Natural and Con-nect system files which are used should be specified.
			Example: An OP command will be directed to database number 2 with the record buffer contents 'UPD=4,8,12.' with the following:

Keyword	M	O	Description

			`ADADBID=2,ADAOPRB='UPD=4,8,12.'`
			However, an OP command will be directed to databases 5 and 15 with the record buffer contents 'UPD=3.6.' and 'ACC.' with the following:
			`ADADBID=(5,10,15),ADAOPRB=('UPD=3,6.',,'ACC.')`
			OP commands will not be directed to database 10.

ADACLSE		X	Specify whether the back-end program (session termination) CTCITE will issue CL commands at the end of a session.
			Default value: NO
ADANAME			The name of the Adabas Link module in your CICS environment.
			Default value: Adabas
ADAUID		X	Specify whether the startup programs will generate an Adabas user ID when OP commands are issued. Adabas user IDs are not intended to be used when ETID=' ' (blank). Therefore this keyword should not be specified.
			Default value: NO

6. Assembling the CTPARM2 parameter module using the transport service source data set as a macro library

Con-nect provides full support of 31-bit addressing mode. This means that the Natural/Con-nect nucleus can be loaded either below or above 16 MB.

7. Assembling and link-editing the queue server front- and back-end programs

The names of the supplied front- and back-end programs for the queue server functions are as follows:

Program	Queue Server Function
CTCIIN	Session re-initiation program
CTCIRC	APPC receive program (not required in the case of remote database access)
CTCIST	General start-up program.
CTCITE	Session termination program (back-end program)
CTCITP	Termination program

- Process the front- and back-end programs using the CICS command language processor DFHEAP1$ with the default options PROLOG and EPILOG.
- Assign the appropriate CICS source data set and the transport service source data set as macro libraries, and assemble the preprocessed programs.

  - Referencing the CICS library, link-edit the CICS exec interface stub modules DFHEAI and DFHEAI0, the queue server front- and back-end programs, and CTCIDB (if required) with the CTPARM2 parameter module. The queue server front- and back-end programs must be placed in the CICS exec interface stub program DFHEAI first.
- Copy the queue server front-end programs into the appropriate Relocatable Program Library (RPL) of the CICS procedure.

Note: If OP and CL commands are to be issued, the transport service Adabas interface module CTCIDB must be link-edited with each of the front- and back-end programs.

You can start the front-end programs from your terminal for debugging purposes when you are using the CICS diagnostic aids (CECI or CEDF). However, the front-end programs are not designed to be invoked from a terminal since the programs do not perform terminal I/Os and your keyboard would remain input-inhibited until you pressed the RESET or CLEAR keys.

8. Updating the CICS PPT and PCT using CEDA (the CICS resource definition transaction)

- Define the queue server front-end programs in the CICS PPT. The name you specify for the queue server front-end program CTCITE in the CICS PPT must be identical to the values specified for the PROGRAM parameter within the DYNROUT, DYNSEND, DYNRECV, DYNDEMN, DYNWTCH, DYNTERM, and DYNAPPL keywords of the CTPARM2 parameter module. If CICS/ESA 3.3 or above is installed and these programs are intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.
- Assign a CICS transaction code to the front-end programs, but not for CTCITE, in the CICS PCT. The transaction code assigned to the CTCIRC front-end program must be identical to the value given in the TPN fields at the adjacent nodes' outbound queues. The transaction code assigned to the CTCIST general startup program must match the value specified in the "Server Program" field in the respective transport service queue definitions. The size of the Transaction Work Area (TWA) must be a minimum of 256 bytes. The actual TWA space requirement is dependent on the length of the dynamic parameter string specified in the CTPARM2 parameter module. In most cases, a TWA size of 512 bytes is adequate. If the TWA size is not adequate, the queue server front-end programs will abend with code CT00. If CICS/ESA 3.3 or above is installed and these programs are intended to be used in 31-bit addressing mode, TASKDATALOC=ANY should be specified in the PCT entry.

9. Initializing the transport service

Notes:

1. If you have not already done so, it is recommended that you see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Overview of the Transport Service Architecture* before you continue with the installation procedures.
2. The user exit program YENTX must be renamed to YENTXACI, if the adjacent node uses the LU6.2 ACI to send/receive messages. When done, the transport service cannot communicate with the LU6.2 API used at the adjacent node.

Initializing the transport service consists of two parts: defining the transport service node ID and the actual initialization. See the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Initializing the Transport Service*.

10. Creating one or multiple outbound queues for the storing of transport items intended for other nodes

At least one outbound must be created for each adjacent node in the network. When an outbound queue is created, the attributes of the LU6.2 or remote database access link between the local and the adjacent node must be defined.

To create one or more outbound queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Outbound Queues*.

11. Creating one or multiple receiving queues to control the demon processes

At least one receiving queue must be created when the local node uses remote database access as a transmission medium. The receiving queue is used to control the demon processes which continuously listen for transaction requests from adjacent nodes.

To create a receiving queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Receiving Queues*.

12. Creating an application queue for incoming transport item that are to be passed to a local application

An application queue must be created for each application at the local node which will use the transport service. Transport items which are sent to the application are placed in these queues and remain there until they are processed by the local application.

When you create an application queue for Con-nect external mailing, specify the following on the "Application Queue" screen:

- In the "Node ID" field, enter the name you specified as your transport service ID. See step 12.

- In the "Application Name" field, enter A.
- In the "Application Library" field, enter SYSCNT2.
- In the "Application Program" field, enter YCINITO.

To create an application queue, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Application Queues*.

13. Setting the "Reset" and "Input" statuses for the inbound, outbound and application queues

When you initially test the transport service, the input status of the inbound, outbound and application queues must be set to active and the reset status to inactive. To set the input and reset statuses of the queues, perform the following steps; they must be repeated for each queue:

- Select the Queue Maintenance function from the "Transport Service Administration" screen to display the "Queue Maintenance" screen.
- Enter MO (modify) in the Cmd column for the appropriate queue and press ENTER. As a result, the corresponding screen is displayed. Modify the following fields:
  - In the "Reset Status" field, enter I (inactive).
  - In the "Input Status" field, enter A (active).
  - Press PF9.

For further information about the "Queue Maintenance" screen see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Queue Maintenance*.

14. Defining routing specifications for all other nodes with which you intend to communicate

For each node in your network with which you want to communicate, you must define a routing entry which determines how transport items are to be routed to that node.

To define a routing entry, see the *Con-nect Administration* documentation, section *The Transport Service*, sub-section *Routing Entries*.

15. Installing an external mail application which is based upon the transport service

Before you can test the installation of the transport service, you must install an external mail application which is based upon the transport service. See the appropriate installation manuals.

16. Testing the installation of the transport service

Once an external application which uses the transport service has been installed, you can test the installation of the transport service. The following is an example with multi-node Con-nect.

When testing, use the following settings for ALL queues: Input Status: A Output Status: I Reset Status: I

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you try to start a queue server when either the outbound or inbound queue is in the hold status, the queue server task will immediately exit without processing any information.

- Send a mail item from a Con-nect node or, another external mail application based upon the transport service method, with the request for a delivery notification, and then check the inbound queue to see if the transport item is there.
- Start the inbound queue from the "Queue Maintenance" screen. This activates the router, which should route the transport item to the outbound queue. Since the outbound queue is inactive, the transport item should remain in the outbound queue.

You can keep track of the ongoing sequence of events which are recorded in the "Output Status" field by pressing ENTER. Starting the queue causes the "Output Status" field to change from "I" to " " (scheduled) which means that your Natural session issued a CICS START command to start the general startup front-end program. This program accesses the specifications in CTPARM2, such as those in the keywords NPGRMID and DYNROUT, in order to pass control to the Natural program YR which executes within a non-interactive Natural session. YR, at the beginning of its execution, changes the "Output Status" to "A" and, at the end of its execution, back to "I" (if successfully terminated) or to "H" (if terminated due to unrecoverable errors).

In both cases (successful and unsuccessful termination), you can check the log records in order to obtain additional information. If the status seems to hang in scheduled mode " ", this indicates that the startup of the asynchronous server program failed, typically due to an incorrect setup in CTPARM2.

- Check to see if the transport item was routed to the outbound queue. (The same test procedure can be applied to the other queues as well.) Up to this point, LU6.2 or remote database access have not been involved. The following steps allow you to test sending a transport item to another node.
- Set the outbound queue with the routed transport item to "inactive" status; then start the outbound queue from the "Queue Maintenance" screen.
- Check the log records. The log records are accessed from the Log Information Maintenance function on the "Transport Service Administration" screen. If the transport item was successfully sent, records from the TS_SEND program (YOX9000 in the case of CICS APPC facilities),

YOX9000B (in the case of CICS APPC facilities and if the adjacent node uses LU6.2 ACI) or YOX7000 (in the case of remote database access) are displayed.

- Check for log records which were created when a status report was received from the target node. Records from the TS_RECEIVE program YIX9000 (in the case of CICS APPC facilities), YIX9000B (in the case of CICS APPC facilities and if the adjacent node uses LU6.2 ACI) or YIX7000 (in the case of remote database access) should be displayed.
- Check to see whether the status report was delivered to the appropriate application queue.
- If you are experience any problems, the following commands and functions can be useful for debugging:
    - CICS Supplied Transactions CEMT, CEDC and CEDF

      For example, use the following command to verify whether or not an LU6.2 session has been acquired:

      ```
      CEMT I CONN (conn)
      ```

      where "conn" is the connection ID.

      Use the following command to examine the characteristics of an LU6.2 connection:

      ```
      CEDC VIEW CONNECTION(conn) GROUP(isc)
      ```

      where "isc" is the resource group ID.

      Use the following command to display the online trace utility:

      ```
      CEDF
      ```

      Note: Do not use the above transactions unless you are familiar with them.

    - Transport Service Administration Functions

      The following functions are among those offered on the "Transport Service Administration" screen:

      Queue Maintenance - Provides a transport item count, time stamps for recent activity, and the ability to manually start the queue server.

      Log Information Maintenance - Provides information that can be helpful to you or, in the case of severe malfunctions, to Software AG in resolving any problems you are experiencing with the transport service.

III Part 3. Con-nect Teleservices

This part describes how to install Con-nect Teleservices. It is subdivided into the following topics:

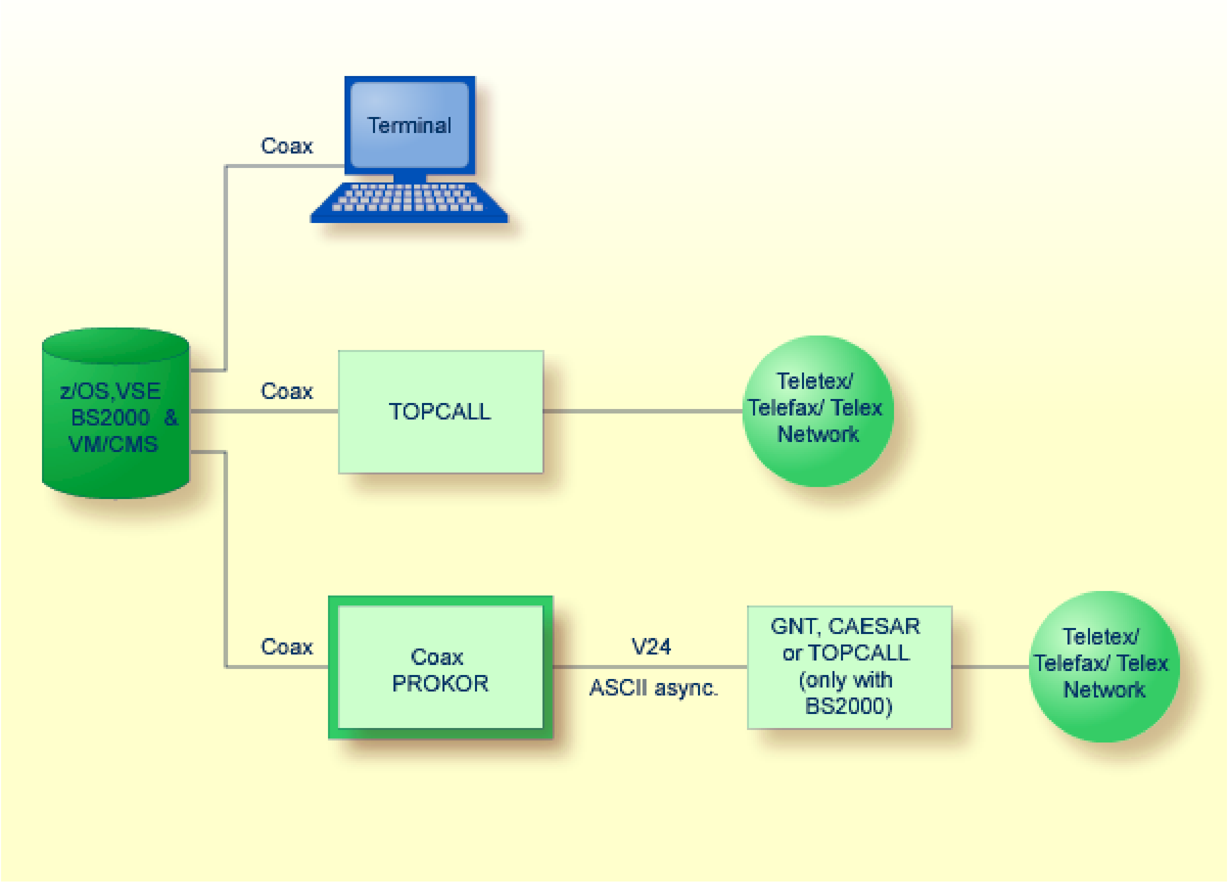
8 General Information for Con-nect Teleservices

This chapter contains general information for installing Con-nect Teleservices.

The Telex and Telefax System Connection

The currently supported telex and telefax systems are the TOPCALL system, GNT system, and CAESAR system. For the TOPCALL system, the protocol converter is not required; it is directly connected via a coaxial line.

For the GNT and CAESAR systems, the terminal controller is connected with the GNT or CAESAR system via the PROKOR protocol converter, which emulates a terminal and, thus, can be connected via a coaxial line. When the host transmits messages, the PROKOR receives them as if it were a terminal, and forwards them to the GNT or CAESAR system. Conversely, the PROKOR accepts messages from the GNT or CAESAR system and forwards them in accordance with the corresponding protocol, emulating the keyboard of a terminal, to the host computer.



The main prerequisite for this data exchange is a Natural session in the corresponding system environment that can be started on the coaxial line attached to the PROKOR. This Natural session must be able to be started automatically. The Telex Driver start program is, in turn, called from this session.

Prerequisites for the Installation of the Telex Driver

The following requirements must be met:

- Con-nect Version 3.4.3 must be installed.
- Special administration tasks for the definition of external mail nodes must be completed. These tasks are described in the *Con-nect Administration* documentation, section *System Maintenance*, sub-section *External Mail Nodes*.
- The required hardware configuration must be operable.

Note: In the Siemens environment, the German umlauts use different hex-codes than in the IBM environment. For this reason, the macro CDTLX must be modified. The source for assembling the macro CDTLX is contained in member CONFSRT of the source library CMF*nnn*.SRC.

Installation Jobs

The installation of Software AG products is performed by installation jobs. These jobs are either created "manually" or generated by System Maintenance Aid (SMA).

For each step of the installation procedure, the job number of a job performing the respective task is indicated. This job number refers to an installation job generated by SMA. Note that the job numbers on the installation medium may be preceded by a product code (for example, CMTI060).

Using System Maintenance Aid

For information on using Software AG's System Maintenance Aid (SMA) for the installation process, refer to the System Maintenance Aid documentation.

Running INSTALL

The following step must be executed after Con-nect Teleservices has been installed:

```
L
LOGON SYSCNT2
INSTALL
```

which displays the "Installing System - Profile" screen.

```
                        Installing  System Profile


                    System Name:   * * *  C O N - N E C T  3  * * *

                   CPU Timezone: 1     (relative to Greenwich Mean ,Time)
                 Sliding Window: 50

            Days in Wastebasket: 60
          Days until Expiration: 365
          Archive on Expiration: x
            Trace Cabinet Usage: x

                      Day Start: 0     0-23
                     Week Start: 1     1-7

           Logical Printer Name: DAEPRTB2
            Printer Destination: DAEPRTB2

                     Initialize: _ System Command and Object Table

         CON-NECT TRS installed: X     (Leave blank if not installed)
```

You are not required to modify the information on this screen, simply press ENTER. Each time you install Con-nect Teleservices you must run INSTALL even though it is run when Con-nect is installed.

9 Installing Con-nect Teleservices Under z/OS and z/VSE

This chapter describes step by step how to install Con-nect Teleservices under the operating systems z/OS and z/VSE.

The Installation Medium

The installation medium for z/OS contains the data set listed in the table below. The sequence of the data set is shown in the Software AG Product Delivery Report which accompanies the installation medium.

The installation medium for z/VSE is delivered along with the documentation for the package (Product Installation Package or Product Maintenance Package). In this documentation, the format of the installation medium (tape marks etc.) is described. The volume serial number (VOLSER) of the medium is listed in the Software AG Product Delivery Report.

The notation *nnn* in data set/file name represents the version number of the product.

Data Set Name	Contents
CMT*nnn*.INPL	INPL data set containing the Con-nect Teleservices programs.

Installation Steps

1. Load Data Set CMTnnn.INPL

Use the INPL utility (Job I061, Step 0295) to load the data set/file CMT*nnn*.INPL.

2. Copy Library to SYSCNT2 and SYSCNT2T

The data set CMT*nnn*.INPL contains the libraries X-TLXCAE (for the CAESAR system), X-TLXGNT (for the GNT system), and X-TLXTOP (for the TOPCALL system). Depending on the telex box which is being used, the corresponding library must be copied into the libraries SYSCNT2 and SYSCNT2T. Use the SYSMAIN utility to do this.

3. Define STEPLIB

Define SYSCNT2 as the STEPLIB for SYSCNT2T in the application definition for Natural Security. For further information, see ***Con-nect Teleservices and Natural Security***.

4. Run INSTALL

Once Con-nect Teleservices is installed, run the Con-nect INSTALL program. See ***Running INSTALL***.

Natural Parameter

The Natural parameter CSIZE must be set to 32KB. Specify this parameter in the Natural parameter module. See the *Natural Installation and Operations* documentation for additional information on profile parameters.

See also *Modify Parameter Modules* (Job I080).

Translating Special Characters in Telexes

It is possible to translate special national characters or any other special characters that might occur in a text sent as a telex into those characters required for a telex transmission. Con-nect Teleservices translates these characters according to a translation table prior to sending the telex to the telex box. The translation table, which is shown below with its default characters, is part of the module CONFSRT. This module is contained in the data set CMF*nnn*.SRC (see the *Con-form Installation* documentation).

```
CONFSRT
         ...
         ...
         CDTLX C'Ä',C'AE'
         CDTLX C'Ö',C'OE'
         CDTLX C'Ü',C'UE'
         CDTLX C'ä',C'AE'
         CDTLX C'ö',C'OE'
         CDTLX C'ü',C'UE'
         CDTLX C'ß',C'SS'
         CDTLX C'$',C'DLR'
         CDTLX C'"',C''''
         CDTLX C'^',C''''
         ...
         ...
```

If you want other characters to be translated, you can modify this table according to your requirements. After modification of the table, the module CONFSRT must be assembled and linked to the Natural nucleus.

Con-nect Teleservices under Net-pass and Com-plete

The following description pertains to an environment comprising the Software AG products Net-pass and Com-plete. To further control the application, you can use Natural Security.

General Description of the Process

A permanent terminal address for the special "Telex line" that runs from the host to the telex box, is defined in Net-pass tables. The Net-pass profile defines that, when this line is active, the user ID TELEX automatically logs on to Com-plete. From Com-plete, this user then logs on to Natural Security, which further specifies that the user directly logs on to the library containing the Telex Driver programs. The user then automatically calls the Telex Driver start program from this library. The following sections describe how this is accomplished for specific applications.

Con-nect Teleservices and Net-pass

In the case of normal telecommunications between a host and terminals, the user has to log on to the system before any other activities can be started. The addresses that are allocated can vary from logon to logon. Such logons are not possible from a Telex device. When generating the system, a permanent terminal address must be reserved for the coaxial port that is allocated to the Telex device. In this example, this is the terminal ID LOCS130. When this address (line) is active, a VTAM-type logon is made to Net-pass.

```
*/* THIS LIST SUPPORTS LOCAL TERMINALS ON A3274-1A         00060000
*/* CONTROLLER. THE TERMINAL TYPES ARE:                    00070000
*/*                                                        00080000
*/* 3278 MODELS 2                                          00090000
*/*                                                        00110000
...
...
...
LOCS129 LU LOCADDR=30                                      00300000
LOCS130 LU LOCADDR=31,LOGAPPL=ENETPASS                     00300000
LOCS131 LU LOCADDR=32                                      00300000
DAE82132 LU LOCADDR=33,DLOGMOD=DSC2K,                     X00300000
MODETAB=MODTBSAG
*                                                          01400000
```

When the LU is activated and connected to Net-pass, NPEXIT03 (Net-pass physical terminal logon exit) gains control and supplies logon data (user ID and password) for the telex "user". Therefore, NPEXIT03 has to be modified as follows.

Insert the following statements after label VTAMLGN:

```
         L      R1,EPLPTERM              GET LUNAME
         CLC    0(8,R1),=CL8'LOCS130'    TELEX BOX LU ???
         BE     VTAMLGN1                 YES - SUPPLY LOGON DATA
         B      NOTELEX                  NO - SOMETHING ELSE
*
VTAMLGN1 DS     0H                       TELEX DRIVER
         L      R4,EPLPTB                PHYSICAL TERMINAL C.B.
         CMUSE  DPTB,R4
         L      R5,PTBPTE                PHYSICAL TERMINAL ENTRY
         CMUSE  DPTE,R5
         MVC    PTEUSER,=CL8'TELEX'      SET USERID
         MVC    PTBPASSW,=CL8'TELEX'     & PASSWORD
         XC     PTBPASSW,NCBPCODE
         OI     PTBSLGND,PTBMLGND        NOTE LOGON DATA IS SET
         B      RET0                     AND RETURN

         CMDROP R4
         CMDROP R5

*
NOTELEX  DS     0H
```

Note: In the above examples, LOCS130 must be replaced with the LU name assigned to your Telex device. ENETPASS and ECOMPLET are example names which must be replaced by the names which are valid in your environment.

In the NETPASS profile for the Telex Driver, COMPLET is entered for APPL and YES is entered for START. This means that an automatic LOGON to ECOMPLET is made after the LOGON to Net-pass.

```
         ...
         ...
         CMUSER
                USERID=TELEX             USERID AS FROM NPEXIT03
                PROFILE=PROFTLX          ... USES PROFILE PROFTLX
         ...
         ...
         CMPROF=PROFTLX
                CONTROL=NO              AUTHORIZED USER
                BRODCST=NO              NO BROADCAST MESSAGE
                GROUP=TELEX             GROUP CAN BE USED FOR SEND
                RESCAN=NO               DO NOT SCAN FOR OTHER PROFILES
                SUSPKEY=NULL            NO SUSPEND KEY
                LGNLGF=YES              AFTER LOGOFF NET-PASS LOGO
                MENUFMT=L               N.A.T FORMAT
                AUTOLGF=NO              EXEMPT FROM AUTO-LOGOFF
                AUTODSC=NO              EXEMPT FROM AUTO-DISCONNECT
                MAXLVL=1                MAXIMUM OF ONE SESSIONS
```

```
               NOTIFY=NO              NOTIFY USER ON RECEIVE+LOSTSESS
               RESTART=YES            AUTO RESTART LOST SESSIONS
               COMP=0                 NO COMPRESSION
               WSLV=0                 NO INCORE SUSPEND BUFFERS
               READBUF=NO             DON'T DO READBUFFER
*  MAYBE USED  CLSPASS=YES            DO CLOSEDEST PASS
*  ...LATER    RELOGON=YES            BACK TO NET-PASS IF SESS DROP
               STRTLVL=1              AUTO START FOR LEVEL 1
               CMLEVEL=1
               ID=CONNECT
               APPL=ECOMPLET
               START=YES              AUTO-START APPLICATION
```

Con-nect Teleservices and Com-plete

The user TELEX is created in ULOGM (USER ID MAINTENANCE UTILITY, see the *Com-plete Utilities* documentation). In our example, CON2 (call Con-nect session) is entered as the user startup program in the associated user profile which then appears. This means that this startup transaction is automatically called each time this user logs on to Com-plete. Because Con-nect runs under Natural Security in our example, the user TELEX is passed to Natural Security, where the corresponding security definitions come into effect.

Con-nect Teleservices and Natural Security

Define SYSCNT2 as the STEPLIB for SYSCNT2T in the application definition in Natural Security.

The application SYSCNT2T is created under Library Maintenance (see the *Natural Security* documentation). SYSCNT2T is the library that contains the Telex Driver programs. The name of the Telex Driver start program must be entered in the input field Startup. In most cases, this will be the supplied program TLXDRV. This program will then be called after each successful logon to this application.

The user TELEX is subsequently defined under User Maintenance, and SYSCNT2T is entered in the input field Libraries Default (see the *Natural Security* documentation). This application is automatically called when the user TELEX logs on to Natural.

If Natural Security has not been installed, an Assembler program, which will be provided by Software AG on request, must be written to call the library and start the program.

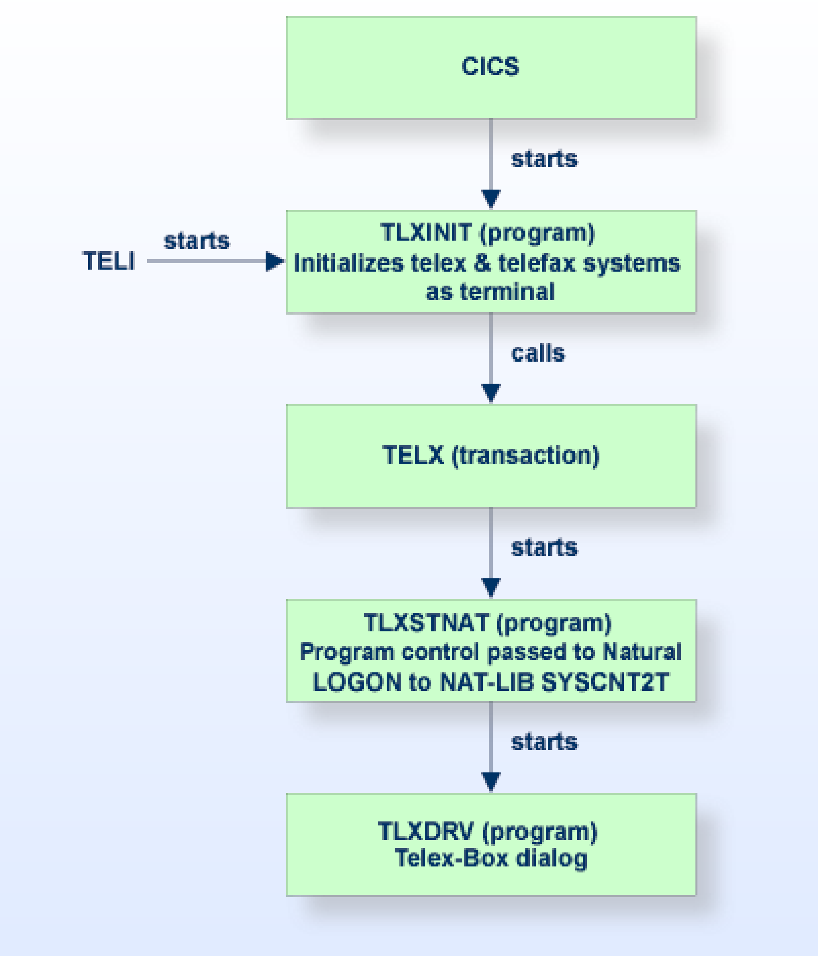
Con-nect Teleservices under CICS

The following description pertains to an environment with the transaction monitor CICS. To further control the application, you can use Natural Security.

General Description of the Process

A permanent terminal address for the special "Telex line" that runs from the host to the telex box is defined in the CICS Terminal Control Table (TCT) tables. The transaction TELI starts the Telex Driver program TLXINIT which initializes the telex and/or telefax system and then calls the transaction TELX. TELX starts the program TLXNAT which logs on to the Natural library SYSCNT2T and automatically calls the Telex Driver Start program from this library. The following sections describe how this is accomplished for specific applications.

The following is an example of Con-nect Teleservices under CICS.



The steps below are required in order to install Con-nect Teleservices under CICS:

1. Define the telex and/or telefax system in the TCT (Terminal Control Table)
2. Define the transaction TELI in the PCT (Program Control Table)
3. Define the transaction TELX in the PCT (Program Control Table)
4. Define the program TLXINIT in the PPT (Processing Program Table)
5. Define the program TLXSTNAT in the PPT (Processing Program Table)
6. Define the program TLXINIT in the PLT (Program Load Table)

An example for each of these steps, and programs TLXINIT and TLXSTNAT are given in the following sections.

Con-nect Teleservices and CICS

Define Telex and Telefax Systems in the TCT

The telex and telefax system works in the same way as a normal terminal. It must be defined in the TCT using the TRANSCEIVE option. The terminal ID (TID) must be defined as TERMID(' *xxxx* ') in the program TLXINIT (for example, TERMID('c1281')).

Example:

```
D081     DFHTCT TYPE=TERMINAL,
               ACCMETH=VTAM,
               FEATURE=(AUDALARM,UCTRAN),
               GMMSG=YES,
               NETNAME=c1281,
               PGESTAT=PAGE,
               RELREO=(,YES),
               TCTUAL=255,
               TIOAL=300,
               TRMIDNT=c1281,
               TRMMODL=2,
               TRMSTAT=TRANSCEIVE,
               TRMTYPE=L3277,
```

Define the Transaction TELI in the PCT

The transaction TELI starts the Telex Driver and is also necessary for testing or error handling. Set TRANSID=TELI, TWASIZE=150 and PROGRAM=TLXINIT in the PCT.

Example:

```
DFHPCT TYPE=ENTRY,TRANSID=TELI,
      DTB=YES,
      TWASIZE=150,
      CLASS=SHORT,PROGRAM=TLXINIT,
      RESTART=NO
```

Define the Transaction TELX in the PCT

The program TLXINIT calls the transaction TELX which starts the program TLXSTNAT. Set TRANSID=TELX, TWASIZE=150 and PROGRAM=TLXSTNAT in the PCT.

Example:

```
DFHPCT TYPE=ENTRY,TRANSID=TELX,
      DTB=YES,
      TWASIZE=150,
      CLASS=SHORT,PROGRAM=TLXSTNAT,
      RESTART=NO
```

Define the Program TLXINIT in the PPT

The program TLXINIT starts the transaction TELX with the terminal ID specified when the telex and telefax system was defined in the TCT (c1281 in the example).

Example:

```
DFHPPT TYPE=ENTRY,PROGRAM=TLXINIT,
      PGMLANG=ASSEMBLER,RES=YES
```

Define the Program TLXSTNAT in the PPT

The transaction TELX calls the program TLXSTNAT. This program logs on to the Natural library SYSCNT2T, and starts the program TLXDRV.

Example:

```
DFHPPT TYPE=ENTRY,PROGRAM=TLXSTNAT,
      PGMLANG=ASSEMBLER,RES=YES
```

Define the Program TLXINIT in the PLT

The program TLXINIT must be included in the PLT, so that the Telex Driver will be activated when CICS is started.

Example:

```
DFHPLT TYPE=ENTRY,PROGRAM=TLXINIT
```

Example: Program TLXINIT (Assembler)

```
*
*ASSEMBLER
*-----------------------------------------------------------------
* TLXINIT INITIALIZES THE PROKOR. THE TRANSACTION TELX IS CALLED
* AND THE PROGRAM TLXSTART IS SUBSEQUENTLY STARTED.
*-----------------------------------------------------------------
*
*
TLXINIT  TITLE 'TINIT'
         EJECT
         SPACE 1
*
*
TELE     START 0
*
*
*
         EXEC CICS START              ,                           +
              TRANSID('TELX')         ,                           +
              TERMID('c1281')         ,                           +
              INTERVAL(000100)        ,                           +
              NOCHECK
         EXEC CICS RETURN
*
*-----------------------------------------------------------------
*  ENTER THE TERMINAL ID WHICH WAS SPECIFIED IN THE TCT
*  UNDER TERMID('XXXX').
*-----------------------------------------------------------------
*
         END
```

Example: Program TLXSTNAT (Assembler)

```
*ASSEMBLER
*
*----------------------------------------------------------------
* TLXSTNAT IS CALLED FROM TLXINIT WHEN CICS IS STARTED VIA THE
* TRANSACTION TELX AND ACTIVATES THE CON-NECT TELESERVICES DRIVER.
*----------------------------------------------------------------
*
*!!!!!!!!! THIS PROGRAM APPLIES TO SYSTEMS WITHOUT NAT-SEC !!!!!!
*
*
TLXSTART  TITLE 'TSTART'
          EJECT
          SPACE 1
TLXSTR    START 0
*
*
          MVC   RSTRTRID,EIBTRNID
          MVC   NATRID,=CL4'NAT2'      Natural START
TRANSACTION
          MVC   REPGID,=XL8'00'
          LA    2,FILLER
          ST    2,DYPAAD
          LA    2,L'PADA
          STH   2,DYPALE
*
          EXEC  CICS XCTL              ,                          +
                PROGRAM('NCI21RE')     ,                          +
                COMMAREA(WSCOM)
*
*-----------DATA DEFINITIONS----------------------------------
FILLER    DS    CL12
PADA      DC    C'
AUTO=OFF,MENU=OFF,LC=ON,FS=OFF,ID='','',                          +
                STACK=(LOGON SYSCNT2T;TLXDRV)'
*
*-----------WS-COMMAREA----------------------------------------
DFHEISTG  DSECT
WSCOM     DS    0CL22                  WS-COMMAREA
RSTRTRID  DS    CL4                    RESTART-TRANS-ID
DYPAAD    DS    A                      DYN-PARMS-ADDRESS
DYPALE    DS    H                      DYN-PARMS-LENGTH
NATRID    DS    CL4                    NAT-TRAN-ID
REPGID    DS    CL8                    RET-PGM-ID
*
*
          END
```

Con-nect Teleservices and Natural Security

Define SYSCNT2 as the STEPLIB for SYSCNT2T in the application definition in Natural Security.

The application SYSCNT2T is created under Library Maintenance (see the *Natural Security* documentation). SYSCNT2T is the library that contains the Telex Driver programs. The name of the Telex Driver start program must be entered in the input field Startup. In most cases, this will be the supplied program TLXDRV. This program will then be called after each successful logon to this application.

The user TELEX is subsequently defined under User Maintenance, and SYSCNT2T is entered in the input field Libraries Default (see the *Natural Security* documentation). This application is automatically called when the user TELEX logs on to Natural.

Define Telex and Telefax Systems in the TCT

For further information, see ***Con-nect Teleservices and CICS***.

Define the Transaction TELI in the PCT

For further information, see ***Define the Transaction TELI in the PCT***.

Define the Transaction TELX in the PCT

For further information, see ***Define the Transaction TELX in the PCT***.

Define the Program TLXINIT in the PPT

For further information, see ***Define the Program TLXINIT in the PPT***.

Define the Program TLXSTNAT in the PPT

For further information, see ***Define the Program TLXSTNAT in the PPT***.

Define the Program TLXINIT in the PLT

For further information, see ***Define the Program TLXINIT in the PLT***.

Example: Program TLXSTNAT (Assembler)

Data definition in program TLXSTNAT must be specified as follows:

```
*ASSEMBLER
*
*----------------------------------------------------------------
* TLXSTNAT IS CALLED FROM TLXINIT WHEN CICS IS STARTED VIA THE
* TRANSACTION TELX AND ACTIVATES THE CON-NECT TELESERVICES DRIVER.
*----------------------------------------------------------------
*
.
.
.
*-----------DATA DEFINITIONS----------------------------------
FILLER    DS    CL12
PADA      DC    C'
AUTO=OFF,MENU=OFF,LC=ON,FS=OFF,ID='','',                           +
                STACK=(LOGON SYSCNT2T, USERID, PASSWD; TLXDRV)'
*
*-----------WS-COMMAREA----------------------------------------
.
.
.
```

10

Installing Con-nect Teleservices Under BS2000

This chapter describes step by step how to install Con-nect Teleservices under the operating system BS2000.

The Installation Medium

The installation medium contains the file listed in the table below. The notation *nnn* in file name represents the version number of the product.

Data Set Name	**Contents**
CMT *nnn*.INPL	INPL data set containing the Con-nect Teleservices programs.

Installation Steps

1. Load Data Set CMTnnn.INPL

Use the INPL utility to load the file CMT*nnn*.INPL.

2. Copy Library to SYSCNT2 and SYSCNT2T

The file CMT*nnn*.INPL contains the libraries X-TLXCAE (for the CAESAR box), X-TLXGNT (for the GNT system), and X-TLXTOP (for the TOPCALL system). Depending on the telex box which is being used, the corresponding library must be copied into the libraries SYSCNT2 and SYSCNT2T. Use the SYSMAIN utility to do this. Additionally, the INSTALL program must be invoked to complete the installation.

3. Define STEPLIB

Define SYSCNT2 as the STEPLIB for SYSCNT2T in the application definition for Natural Security.

4. Run INSTALL

Once Con-nect Teleservices is installed, run the Con-nect INSTALL program. See ***Running INSTALL***.

Natural Parameter

The Natural parameter CSIZE must be set to 32KB. Specify this parameter in the Natural parameter module. See the *Natural Installation and Operations* documentation for further information on profile parameters.

See also ***Modify Parameter Modules*** (Job I080).

Translating Special Characters in Telexes

It is possible to translate special national characters or any other special characters that might occur in a text sent as a telex into those characters required for a telex transmission. Con-nect Teleservices translates these characters according to a translation table prior to sending the telex to the telex box. The translation table, which is shown below with its default characters, is part of the module CONFSRT. This module is contained in the file CMF*nnn*.SRCE (see the *Con-form Installation* documentation).

```
CONFSRT
         ...
         ...
         CDTLX C'Ä',C'AE'
         CDTLX C'Ö',C'OE'
         CDTLX C'Ü',C'UE'
         CDTLX C'ä',C'AE'
         CDTLX C'ö',C'OE'
         CDTLX C'ü',C'UE'
         CDTLX C'ß',C'SS'
         CDTLX C'$',C'DLR'
         CDTLX C'"',C''''
         CDTLX C'^',C''''
         ...
         ...
```

If you want other characters to be translated, you can modify this table according to your requirements. After modification of the table, the module CONFSRT must be assembled and linked to the Natural nucleus.

Con-nect Teleservices under UTM/TIAM

The following description pertains to an environment with a UTM transaction monitor or under TIAM. To further control the application, you can use Natural Security.

General Description of the Process

A permanent terminal address for the special "Telex line" that runs from the host to the telex box is defined as a user who is operating a Natural application. The protocol converter (PROKOR) must be defined as a terminal in the PDN.

If Con-nect Teleservices is operating under UTM, the protocol convertor (PROKOR) must be programmed to open the connection to the UTM application with a UTM transaction code (TAC) to start Natural and the Telex Driver start program (TLXDRV) from library SYSCNT2T.

If Con-nect Teleservices is operating under TIAM, the protocol converter (PROKOR) must be programmed to open the connection with a BS2000 logon ID, start Natural and the Telex Driver start program (TLXDRV) from library SYSCNT2T.

Define the Protocol Converter in the PDN

The PDN generation for the protocol converter is dependent on the box used. An example for the appropriate PDN generation can be obtained from the hardware supplier. The following example applies to the CAESAR or TOPCALL system.

```
XLTNG UEWEG=FE-STAND,
      DUETYP=TYP05,
      LPUFTYP=96512,
      UEGSW=4800,
      UEPROZ=MSV1/GEMISCHT,
      MODTAKT=JA,
      EASS=0,
      LPUFADR=41
XSTAT EAADRES=4041,
      SAADRES=4040,
      STATTYP=TYP45,
      CONMODE=DYN,
      DEXP=STANDARD,
      PARTNAM=$DIALOG,
      PARTPRO=0/0,
      PREDIAL=JA,
      STAS=MULTIKOS,
      PUFLIM=3482,
      AKTIV=JA,
      STATNAM=L41c131,
      STATNUM=601
```

11 Special Considerations

This chapter contains special information for installing Con-nect Teleservices.

Assembler Programs Called from the Telex Driver Programs

The following Assembler programs are used by the Telex Driver programs when necessary:

```
CMWTO                      , for output to the console
CMROLL                     , for time-rollout
```

CMWTO and CMROLL are routines of the TP driver, e.g., NATCOMP if the TP monitor Complete is being used, or NATCICS if CICS is being used. In another TP environment where the above mentioned functions of these routines may not be supported, it may be necessary to adjust them accordingly.

Special Considerations for the CAESAR System

Certain system programming questions (see the *CAESAR System User's Manual*) must be answered as follows, in order to guarantee correct operation of the Telex Driver.

Question Number	Answer
11	1
20	02
55	affirmative
57	0
60	affirmative
61	affirmative

The new CAESAR system uses channel address 44 instead of the previous channel address 20. The channel address must be adjusted in the driver program as specified in your CAESAR manual.

Special Considerations for the Protocol Converter Under the CAESAR System

Certain system programming questions must be answered as follows in order to guarantee correct communication between Telex Driver, PROKOR and CAESAR system.

Question 06: 0200 Question 07: 0300

Special Considerations for the TOPCALL System

1. The TOPCALL channel(s) attached to Con-nect must be defined for QUERY. The following so-called "masks" must be available in the TOPCALL system: +A, +X, and +Y.
2. The following parameters may be modified in the program TLXDRV:

Parameter	Function
Logical-device-ID	Logical device ID of the driver which is used in the case of multiple installed modules serving the same spool file. Default=` ' (blank).
Sleeping-time	Number of minutes the driver remains inactive. Default=3.
Console-messages	Specifies whether console messages will be written. N=no messages written. Y=messages written. Default=N.
Fax-line-length	Maximum length of a text line until word wrapping occurs. Default=68. **Note:** When the full screen protocol is *not* used the value must be 68.
Tc-ddrive	Default disk drive used when storing outgoing messages. Default=B.
Tc-doc-prefix	Document prefix identifier for outgoing messages stored at the TOPCALL site. The name of a document consists of this prefix and a sequence number (e.g. TTC0000003). In the case of multiple simultaneously running driver programs, serving the same spool file, a different document prefix must be used with each driver, e.g. TTD. The last character of the document prefix is used to distinguish the different drivers. Default=TTC.
Tc-charge-pfx	Prefix which is used to store charge files in TOPCALL. Default=B:KK.
Tc-chargef-upload	Enables charge files to be uploaded. Y=yes; N=no. If yes, the charge file from the last month will be uploaded. Default=Y.
Tc-sys-cab	Target cabinet for uploaded charge files and journals. Default=TELEX.
Tc-default-cab	Default target cabinet for incoming messages. Default=TELEX.
Tc-max-in	Maximum number of incoming messages to be processed before switching to the outgoing mode. Default=10.
Tc-max-out	Maximum number of outgoing messages to be processed before switching to the incoming mode. Default=10.
Tc-max-hold	Maximum number of outgoing messages TOPCALL can hold before sending. Default=20.

Parameter	**Function**
Tc-full-screen	Uses the full screen protocol for the transmission of data between TOPCALL and the host. Y=yes; N=no. Default=N. For further information, see special considerations number [3].
Tc-main-link	Specifies whether it is the main link. The main link is the only link that sets the date and time on the TOPCALL system when it is started. The main link requests the charge file from the TOPCALL system at the beginning of each month. Y=yes; N=no. Default=Y.
Tc-delete-indoc	Delete incoming messages and journals at the TOPCALL site after reception (Y=Yes, N=No). Deletion of the documents can be performed either by the driver or the TOPCALL device. Default=Y.
Tc-data-out	Defines the data-out character as it is used in the TOPCALL protocol. The data-out character should be either "D" or "#". The character # is used in the case of a BS2000 system using a DAF@R protocol converter. No other values are supported. Default=D.
Tc-trace-enabled	Enables an extended trace facility which writes log records to the spool file. The log records can be viewed with the X-BLG-M program (TRUE=enabled, FALSE=disabled). Default=FALSE.
	Enabling and running the trace facility for a long period of time requires a large amount of storage in the spool file, therefore, it is recommended to enable this facility only in the case of problems. For further information, see the *Con-nect Administration* documentation for a description of the D-TELEX program.
Tc-cmquery-ext	An extension to the TOPCALL query command. The query extension is concatenated to the query command '..2Q'. For example, the extension can be N=AS:.
Tc-stop-enabled	Enables or disables utility program X-BSTOP to stop the driver between the driver's "sleeping time".
	Y=Enable utility program X-BSTOP to stop driver. N=Disable utility program X-BSTOP to stop driver.

3. If the TOPCALL full screen protocol is used (see Tc-full-screen), the Natural telex device driver NATTLX must also be used. How to assemble, link edit and include it in the executable Natural module is described in the *Natural Operations* documentation , section Operating Natural - General Information.
4. When multiple driver programs, serving the same spool file, are used to send telex/telefax messages to the TOPCALL system, the Natural global session parameter WH must be set to ON.

12 User Modifications

Transferring Telexes/Teletexes Directly to the Addressee

Incoming telexes or teletexes are normally sent to the Inbasket of the cabinet TELEX. Authorized users can then send the telex or teletex (via Con-nect) to the addressee.

With Software AG's telex driver, a user-written exit (X-BI-U) may be called for each incoming telex or teletex in order to guarantee direct delivery to the intended addressee.

Note: This feature does not apply to telefaxes.

The module X-BI-U must exist in the SYSCNT2T library.

The following topics are covered below:

- Description of X-BI-U
- Parameters
- Return Codes

Description of X-BI-U

The API may scan the telex or teletex for a string (name of an addressee) and route the telex or teletex directly to the receiver via Con-nect.

The API may return to the telex driver (in the "addressees" parameter) up to 20 addressees (receivers) to whom this telex or teletex is to be sent. If the 20 lines of text ("lines" parameter) currently being processed by the API contain further addressees, then the API must:

a) delete the 20 addressees already found in the parameter "lines", and

b) terminate with the "return-code" set to 2.

The telex driver will then distribute the telex or teletex to the 20 addressees found, and invoke the API again to proceed with further processing of the same "lines".

If less than 20 "addressees" have been found in the first 20 "lines", further "lines" may be requested from the telex driver by terminating the API with the "return-code" set to 1.

A call from the telex driver with the return code already set to 77 indicates to the API that all lines of the telex or teletex have already been processed.

Parameters

Parameter	Format	User- Supplied	Returned	Remarks
Return Code	(N2)		X	
Addressees	(A8/1:20)		X	
Lines	(A70/1:20)	X	X	for GNT system
	(A250/1:20)	X	X	for Caesar system
	(A250/1:20)	X	X	for TOPCALL system

Parameters are *passed* from the telex driver, and *returned* by the API to the telex driver.

Return Codes

01	More lines required
02	Not all lines processed, but 20 addressees found
77	End of text
others	No further call required

Adding Text to an Outgoing Telex/Teletex/Telefax

Outgoing telexes, teletexes and telefaxes are normally sent with sender/addressee information such as To, From, and Subject with the document following.

With Software AG's telex driver, a user-exit (X-BT-U) may be called for each outgoing telex, teletex or telefax to suppress the sender/addressee information and/or insert additional text lines.

The module X-BT-U must exist in the library SYSCNT2T.

The following topics are covered below:

- Description of X-BT-U
- Parameters

- Return Codes

Description of X-BT-U

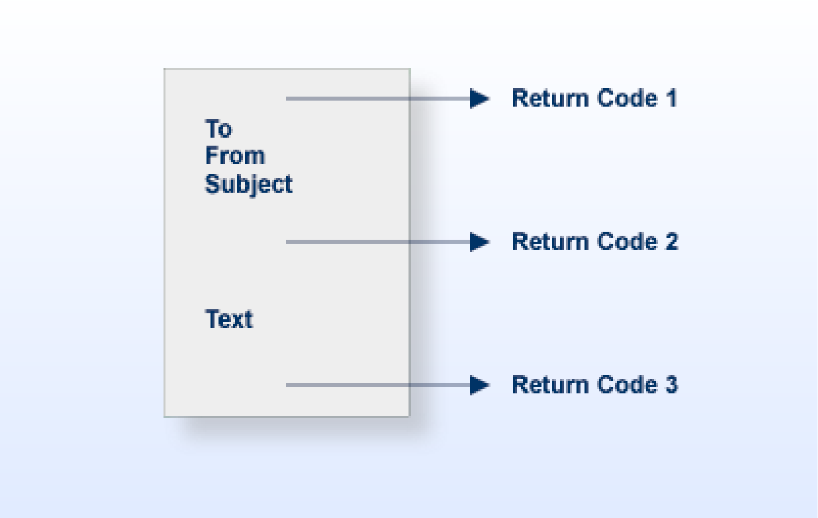
Additional text lines may be inserted before the sender/addressee information, after the sender/addressee information or after the text of the document. (A maximum of 20 lines may be inserted with each subsequent call.)

When the telex driver calls the API, the driver passes a return code to the API which specifies where the driver is within the document:

- Return Code 1: The driver is located above the sender/addressee information. When the return code 7 is passed from the API to the telex driver, the sender/addressee information is suppressed.

Note: Suppressing the sender/addressee information can be performed only when the API is invoked at the beginning of the telex, teletex or telefax.

- Return Code 2: The telex driver is located below the sender/addressee information.
- Return Code 3: The telex driver is located at the end of the document.



If no additional text is to be inserted at the position, the parameter number-lines must be set to zero.

If in the above three cases, more than 20 lines of text is required, the API will pass the first 20 lines to the telex driver and the return code 6 (subsequent calls requested). The telex driver in turn, invokes the API again to proceed with further processing.

If in the above three cases, the API returns a 0 to the telex driver, this indicates to the driver that after inserting the additional text, the driver should continue processing the document.

Parameters

Parameter	Format	User- Supplied	Returned	Remarks
Return Code	(N2)	X	X	
User-ID	(A8)	X	X	Originator's ID
Number-lines	(N2)	X	X	Number of lines to be inserted.
Lines	(A100/1:20)	X	X	
Context	(A250)	X	X	Stores data between subsequent calls and is not modified by the driver program. This field also contains the telex/teletex/telefax number and the answer back (telex/teletex only) on the first call ofX-BT-U.

Return Codes

00	Subsequent calls not requested
06	Subsequent calls requested
07	No addressee information sent when current position is at top of document (only valid when the API has been called with return code 1)

Creating and Sending a Delivery or Non-Delivery Notification

Outgoing telexes, teletexes and telefaxes are normally delivered without notifying the sender whether the item was delivered successfully or unsuccessfully. With Software AG's telex driver, a user-exit (X-BO-U) can be called for each outgoing telex, teletex, or telefax to create and send a delivery or non-delivery notification to the sender's Inbasket.

The module X-BO-U must exist in library SYSCNT2T.

The following topics are covered below:

- Description of X-BO-U

- Parameters

Description of X-BO-U

The API can be used to generate and send a delivery or non-delivery notifcation. In addition, the API can be called to process the data which is submitted to the API and include it in the delivery or non-delivery notification. This data can include, for example, addressee information, delivery date and time, and the transmission fee.

The following is an example of a delivery report. This report can be implemented with the example program, X-BO-U-I, which is delivered with Con-nect Teleservices. To activate the example program, copy X-BO-U-I and rename it to X-BO-U.

```
************** YOUR TELEX/FAX WAS DELIVERED! *************
*                                                       *
*         Tlx/Fax-Number:    +496151922610-             *
*         Subject:           Subject Information        *
*         Delivery Time:     92-06-05-10:32             *
*         Transmission Fee:  $ 0.23                     *
********************************************************* ↩
```

Note: Delivery and non-delivery reports are also collected in the TOPCALL system on a daily basis in journals, and on a monthly basis in charge files, which can be uploaded to a Con-nect cabinet as an incoming telex. For further information, see *Special Considerations for the TOPCALL System*.

Parameters

Parameters	Format	Tlx Driver Supplied	Out	Remarks
Return-code	N2			Not used
Record	N10	X		ISN from current spool record
Sender-id	A16	X		Sender ID
Teleservice-number	A16	X		Number
Teleservice-status	A70	X		GNT system
	A70	X		Caesar system
	A70	X		TOPCALL system
Teleservice-status redefined for the GNT system				
Refno	A4	X		
Go-01	A13	X		
Go-cnt	A2	X		
Go-reject	A1	X		
Go-status	A4	X		

Parameters	Format	Tlx Driver Supplied	Out	Remarks
Go-02	A7	X		
Go-asterisks	A4	X		
Teleservice-status redefined for the Caesar system				
Status	A70	X		The status notification can be either: "Rejected:" and the Caesar system driver variable postcode, or "Transmitted:" and the Caesar system driver variable price.
Teleservice-status redefined for the TOPCALL system				
Status	A1	X		The status notification can be either: R (rejected), or T (transmitted).
Filler	A14	X		Not used
Delivery-time	A14	X		Delivery date and time
Error	A2	X		Error code in case of non-delivery
Fee	A5	X		Transmission fee
Break	A1	X		TOPCALL break value (0=normal return)
Delivery-info	A30	X		Teleservice number and answer back message

Filling Telex/Telefax Addressee Information with a User Application Program

Outgoing telex, teletexes and telefaxes are normally sent with send/addressee information such as To, From and Subject preceding the document. In most cases, this information is provided by an address which is stored in the personnel address book of the user or a system cabinet, or provided manually by the user. With Software AG's telex driver, a user-exit (API X-BS-U) may be called for each outgoing telex, teletex or telefax to fill in this address information with a user application program.

In conjunction with user exit X-BT-U, user exit X-BS-U can be used to replace the default "Send Telex/Teletex/Telefax to `Node-name`" screen with a customer defined screen.

The module X-BS-U must exist in library SYSCNT2T.

The following topics are covered below:

- Description of X-BS-U
- Parameters

- Return Codes for Parameter Code

Description of X-BS-U

Address information such as To: From: Subject:, or the telex/telefax number may be filled after pressing PF5 (the confirm key) in the "Send Memo" screen.

When the Con-nect send program calls the API, the program passes the name of the external mail node, the type of external node (telex, telefax, or teletex) and the node parameters.

The user exit may call a user application program to obtain the address information and return it to the Send program. Depending on the returned parameters, the Send program decides whether the "Send Telex/Teletex/Telefax to `Node-name`" screen appears or not, and whether the subject is filled in the "Subject" line and the user ID is filled in the "From" line.

Parameters

Parameters	Format	Tlx Driver Supplied	Out	Remarks
Code	N2		X	
Node-name	A8	X		Node name of external mail node
Node-type	A8	X		Node type of external mail node
Context	A251	X		Contains the fields described below
Filler	A1			Not used
Tlx-num-1	A16	X	X	First 16 digits of the telex/telefax/ teletex number
Tlx-ans	A16	X	X	Answer back (only telex and teletex)
Tlx-time1	N4	X		From time (read only)
Tlx-time2	N4	X		To or send time (read only)
Tlx-retry	N2	X	X	Retry counter
Tlx-lang	N1	X		Language code (read only)
Tlx-cancel	A1	X	X	Modifiable flag. Controls whether or not the "Send Telex/Teletex/Telefax to `Node-name`" screen is displayed. See the *Con-nect Administration* documentation.
Tlx-to-pfx	A9	X	X	Text for the "To:" prefix
Tlx-to	A56	X	X	Text for the "To:" line
Tlx-from-pfx	A9	X	X	Text for the "From:" prefix
Tlx-from	A56	X	X	Text for the "From:" line
Tlx-subj-pfx	A9	X	X	Text for the "Subj:" prefix
Tlx-subj	A51	X	X	Text for the "Subj:" line
Device-ID	A8	X	X	Name of the TOPCALL system
TLX-num-2	A8	X	X	Last 8 digits of the telex/telefax/teletex number

Return Codes for Parameter Code

0	The contents of the "Subj:" (subject) and "From:" lines (sender's user ID) is automatically created.
1	The contents of the "Subj:" (subject) and "From:" lines (sender's user ID) is not automatically created.

Source Code to Adapt Existing Telex Systems

The spool file DDM and a description of how the external send functions will be distributed with the release medium. This may be used as a basis from which users may create their own telex, telefax, or teletex interface (including non-standard text).

The Command TELEX

The editor command TELEX checks the telex text, as long as it is on the Con-nect "Modify *Document-name*" screen, for characters that are not permitted for telex transmission.

Even if the command TELEX is not used, the characters are automatically translated when the telex is sent, according to the Special Characters table, and the lines are automatically adjusted to the permitted line length (this does not, however, affect the text of the telex message stored in the Outbasket).

IV Part 4. Con-nect SNADS

This part describes how to install Con-nect SNADS. It is subdivided into the following topics:

Note: For further information regarding Con-nect SNADS, see the *Con-nect Administration* documentation.

13 General Information for Con-nect SNADS

This chapter contains general information for installing Con-nect SNADS.

Prerequisite Software

Before installing Con-nect SNADS, ensure that the following software is already installed:

- Con-nect Version 3.4.3 or above with its prerequisite software (e.g. Adabas database software);
- Natural Version 8.2.2 or above with the appropriate Natural CICS or Natural Com-plete interface.
- EntireX Broker Services, or CICS (for the currently supported version, see the Natural documentation).

SNADS Network Terms and Naming Concepts

The following terms are used when discussing SNADS networks:

Acronym	Description
SNA	System Network Architecture
SNADS	System Network Architecture Distribution Services
DSU	Distribution Services Unit (node in network)
DU	Distribution User (person/application in network)
DIU	Distribution Interchange Unit (distribution object; i.e. message, document)

The following terms define the SNADS network naming concept:

Acronym	Description
DSUN	Distribution Services Unit Name
	The name of a node in the network; each node within a network must have a unique name.
	The DSUN consists of a RGN and a REN separated by a period as follows:
	DSUN=RGN.REN
	RGN Routing Group Name
	An optional value (usually blank) used to distinguish duplicate REN values by routing group.
	REN Routing Element Name
	A required value that must be unique within a routing group. If RGN is blank (i.e. DSUN=blank.REN),then the REN value is equivalent to the node name (DSUN) and must be unique within the network.
DUN	Distribution User Name

Acronym	Description
	The name of a user in the network. The DUN consists of a DGN and a DEN separated by a period as follows:
	DUN=DGN.DEN
	DGN Distribution Group Name
	A value identifying a group of users; typically a node or application name.
	DEN Distribution Element Name
	A value identifying a specific user within a group of users; typically a user ID or a name.

A possible configuration for a user's SNADS address can consists of the DSUN (RGN + REN) and the DUN (DGN + DEN) together as follows:

```
  DSUN              DUN
RGN + REN   +   DGN + DEN
blank.node      node.user
```

Communicating with Systems that are not SNADS Nodes

Con-nect users may exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Con-nect users may also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) when gateway functionalities such those provided by DISOSS are used.

CICS 31-BIT Addressing

Con-nect SNADS fully supports CICS 31-bit addressing mode. If you are installing Con-nect SNADS for the first time, or upgrading your Con-nect SNADS to Version 1.5.3, CICS 31-bit addressing mode is automatically available. For further information, see ***Installation Steps***.

14 Functional Overview Of Con-nect SNADS

This chapter provides a functional overview of Con-nect SNADS, which can be either run under Com-plete using the EntireX Broker Services LU6.2 API (refered to in this chapter as the LU6.2 API), batch mode and the LU6.2 API, or CICS.

Note: Con-nect SNADS cannot handle documents larger than 1 MB.

Con-nect SNADS in Com-plete

The following topics are covered below:

- Con-nect SNADS Concepts
- Con-nect SNADS and the SNADS Network
- Con-nect SNADS and Con-nect
- Con-nect SNADS and Natural
- Con-nect SNADS and Natural Security

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. The implementation of Con-nect SNADS in Com-plete with the LU6.2 API provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS is written primarily in Natural and invokes the communication services of the LU6.2 API with the Natural `PROCESS` statement.

Con-nect SNADS implements SNA Distribution Services functions by means of a program or set of programs called "queue servers". A queue is an area where Distribution Interchange Units (DIUs) reside while awaiting further processing by a queue server. The Con-nect SNADS programs DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND process DIUs. The following is a brief description of each program:

DS_RECEIVE

The DS_RECEIVE program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. A Con-nect SNADS "demon" process continuously listens for conversation requests from remote SNADS partner nodes. When a request is caught, the demon process executes the DS_RECEIVE program as an inline subroutine.

The demon process itself runs as an attached task in the Com-plete environment and uses a dedicated Con-nect SNADS queue called the dummy queue for task control purposes.

The Con-nect SNADS supplied startup routine, CPSEND, leads the flow of control into the appropriate Natural environment.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the router component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the director component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

DS_ROUTER_DIRECTOR is invoked by Con-nect SNADS with the Com-plete ATTACH function. The Con-nect SNADS supplied startup routine, CPSEND, leads the flow of control into the appropriate Natural environment.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program of an adjacent node.

Con-nect SNADS invokes the DS_SEND process with the Com-plete ATTACH function. The Con-nect SNADS supplied startup routine, CPSEND, leads the flow of control into the appropriate Natural environment.

All Con-nect SNADS nodes have one and only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke SNADS functions.

Installations also have one or several outbound queues which are used to direct DIUs to other nodes. In addition to the outbound queues, installations have a dummy queue which is used to control the execution of the demon process.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) whenever gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of the Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The hop count is specified in SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" along which the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the hop count for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note: Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Since Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note: Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool and the number of Com-plete threads should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see ***Modify the "queue server" front-end parameter module CONSNADZ***. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second setup are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, there are three options for where and when you can set the required NATPARMs:

1. You can use one nucleus for both the "server programs" application and the "end user" application with a NATPARM that enforces the restrictive "server usage" requirements.
2. You can choose to generate a separate nucleus solely for the use of the "server programs" application.
3. You can use one nucleus for both the "end user" and "server programs" applications with a NATPARM that enforces only the less restrictive "end user" requirements and the Natural session parameters to enforce the more restrictive "server programs" requirements.

This example can be set up by means of Natural dynamic parameters at the beginning of the respective Natural sessions. The dynamically set NATPARMs can be specified in the respective keywords in the CONSNADZ module (e.g. DYNRECV).

The CONSNADZ Parameter Module

The CONSNADZ module is linked with the front-end programs, CPSEND, CPINIT, CPTERM, and the back-end program CPEND. The purpose of these 370 assembler programs is twofold:

- They permit the Natural "server programs" application to be invoked by a single Com-plete transaction code;
- They permit Natural dynamic parameters to be passed to the nucleus to be used by the "server programs" application.

If required, CPDBAS can be linked to the front- and back-end programs so that Adabas OPEN (OP command) and CLOSE (CL command) commands occur at the beginning and end of an asynchronous Natural task.

You can satisfy the NATPARM requirements by using dynamic overrides specified within the CONSNADZ module. This eliminates the need for a separate nucleus while satisfying the NATPARM requirements of the "server programs" application.

For information about the CONSNADZ parameter module see ***Modify the "queue server" front-end parameter module CONSNADZ***.

Note: Since errors in this portion of the system are difficult to debug, it is recommended that you determine the method you will use to satisfy the NATPARM requirements before you begin the installation process. Any modifications to the Natural dynamic parameters should be verified in the interactive mode.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires LU6.2 support for the physical transmission of messages and documents between nodes. The Com-plete ATTACH function invokes the Con-nect SNADS "server programs" application. Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running under Com-plete.

Since the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from a different TP monitor even though the "server programs" application runs from Natural under Com-plete.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application under Com-plete and the "end user" application. The "end user" application can run in another Com-plete or a different system such as CICS. Any TP monitor which is supported by the installed Natural version is acceptable for the "end user" application.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the Com-plete environment and the other for the "end-user" environment.

  - The Com-plete Natural nucleus must satisfy all installation requirements for the "server programs" application.
  - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link can be created:
  - For each Con-nect system file you have, a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenace*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file number 251.
  - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.
  - If you move a Con-nect system file to a different database or file number, you must re-initialize the "home node record" with the new setting before mail can be sent or received.
  - You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function on the Con-nect "SNADS Administration Menu" screen. For further information, see the *Con-nect Administration* documentation, section *Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS and Natural Security

Natural assigns a user ID to both non-interactive and interactive sessions. By default, Natural passes the current user ID to the Adabas nucleus. The user ID allows Adabas to access any end of transaction (ET) data that can be stored for that particular user ID. However, Adabas cannot accept simultaneous sessions with the same user ID. This restriction can cause problems for the DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND programs. If Natural Security is implemented, more than one instance of the programs can be active simultaneously and all of the programs may attempt to logon with the same Natural Security user ID.

To avoid conflicts resulting from more than one active session assigned to the same user ID, assign a value of blanks to the Natural keyword ETID, i.e. ETID=' ' (blank). Since the Con-nect SNADS server programs do not use the ET data facility, data integrity problems will not occur.

Note: The ETID setting can be modified in the Natural Security definition or in the dynamic parameter settings in CONSNADS.

Additionally, Natural does not issue OP and CL commands at the beginning or end of a session whenever ETID is set to blanks. Nevertheless, the module CPDBAS can be linked to the front- and back-end programs to issue OP and CL commands against databases which are specified in the CONSNADZ parameter module.

Con-nect SNADS in Batch Mode

The following topics are covered below:

- Con-nect SNADS Concepts
- Con-nect SNADS and the SNADS Network
- Con-nect SNADS and Con-nect
- Con-nect SNADS and Natural

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. Software AG's LU6.2 API provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS, which is written in Natural, invokes the communication services of the LU6.2 API with the Natural PROCESS statement.

Con-nect SNADS implements SNA Distribution Services functions by means of a program or a set of programs called "queue servers". A queue is an area where Distribution Interchange Units (DIUs) reside while awaiting further processing by a queue server. The Con-nect SNADS programs DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND process DIUs. The following is a brief description of each program:

DS_RECEIVE

The DS_RECEIVE program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. A Con-nect SNADS "demon" process continuously listens for conversation requests from remote SNADS partner nodes. When a request is caught, the demon process executes the DS_RECEIVE program as an inline subroutine.

The demon process itself runs in batch mode where it exclusively occupies a continuously executing Natural task called the "input handler" and uses a dedicated Con-nect SNADS queue called the dummy queue for task control purposes.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the "router" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the "director" component of the DS_ROUTER_ DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

DS_ROUTER_DIRECTOR is executed by Con-nect SNADS with a continuously running Natural task called the "queue server". The DS_ROUTER_DIRECTOR can share this batch task with the DS_SEND program of one or multiple outbound queues but not with the demon process.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program an adjacent node.

Con-nect SNADS executes the DS_SEND process in a continuously running Natural task called the "queue server". The DS_SEND can share this batch task with the DS_ROUTER_DIRECTOR program of one or multiple outbound queues but not with the demon process.

All Con-nect SNADS nodes have one and only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke SNADS functions.

Installations also have one or several outbound queues which are used to direct DIUs to other nodes. In addition to the outbound queues, installations have a dummy queue which is used to control the execution of the demon process.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) when gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of the Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The "hop count" is specified in SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" along which the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the "hop count" for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note: Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Since Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note: Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see ***Installing Con-nect SNADS in Batch Mode***. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second example are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, you have two options for where and when you can set the required NATPARMs:

1. You can choose to generate a separate Natural batch nucleus solely for the use of the "server programs" application.
2. You can specify the more restrictive parameters for the "server programs" application as dynamic overrides in the z/OS JCL.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires SNA LU6.2 support for the physical transmission of messages and documents between nodes. Additionally, two batch tasks, the "input handler" and the "queue server" are required to schedule the "server programs". Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running in batch mode.

Because the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from any TP monitor even though the "server programs" application runs from Natural in batch mode.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application in batch mode and the "end user" application. The "end user" application can run in a TP system such as Com-plete.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the batch tasks and the other for the "end-user" environment.
  - The Natural batch task nucleus must satisfy all installation requirements for the "server programs" application.
  - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link can be created:
  - For each Con-nect system file you have a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file number 251.
  - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.
  - If you move a Con-nect system file to a different database or file number, you must re-initialize the home node record with the new setting before mail can be sent or received.
  - You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function on the Con-nect "SNADS Administration Menu" screen. For further information about the Control Maintenance function, see the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS in CICS

The following topics are covered below:

- Con-nect SNADS and Natural Security

Con-nect SNADS Concepts

Con-nect SNADS is an implementation of the IBM SNA Distribution Services (SNADS) architecture. The implementation of Con-nect SNADS under the CICS runtime environment is written in Natural and IBM 370 Assembler. CICS provides the support for application-to-application communication between Con-nect and other SNADS nodes. Con-nect SNADS invokes the communication services of CICS with a 370 Assembler language routine using CICS commands to access the corresponding functions provided by CICS.

Con-nect SNADS implements SNA Distribution Services functions by means of a program, or a set of programs, called "queue servers". A queue is an area where the Distribution Interchange Units (DIUs) reside while awaiting further processing by a "queue server". The Con-nect SNADS programs DS RECEIVE, DS_ROUTER_DIRECTOR,and DS SEND process DIUs. The following is a brief description of each function:

DS_RECEIVE

The DS_RECEIVE program processes DIUs which are received from an adjacent node and inserts them into the inbound queue of the current node. DS_RECEIVE is invoked by CICS whenever an incoming conversation request is received from a remote partner. The Con-nect SNADS supplied startup routine, CSRECV, leads the flow of control into the appropriate Natural environment.

DS_ROUTER_DIRECTOR

If the recipients of the DIUs are elements (addressees) of a remote SNADS node, the "router" component of the DS_ROUTER_DIRECTOR program processes the DIUs from the inbound queue into one or several of the local system's outbound queues.

Whereas, if the recipients of the DIUs are elements (addressees) of the local Con-nect system, the "director" component of the DS_ROUTER_ DIRECTOR program processes the DIUs from the inbound queue into the appropriate multi-node routine. This routine delivers the DIUs into the recipients' Inbasket.

Con-nect SNADS invokes the DS_ROUTER_DIRECTOR program by means of the CICS interval control `START` command. The Con-nect SNADS supplied startup routine, CSSEND, leads the flow of control into the appropriate Natural environment.

DS_SEND

The DS_SEND program processes DIUs from an outbound queue of the current node into the DS_RECEIVE program of an adjacent node.

Con-nect SNADS invokes the DS_SEND process by means of the CICS interval control `START` command. The Con-nect SNADS supplied startup routine, CSSEND, leads the flow of control into the appropriate Natural environment.

All Con-nect SNADS nodes have only one inbound queue. This queue is used for both inbound DIUs and DIUs which originate from the local Con-nect system.

All DIUs make their initial entry to, or begin their departure from, the Con-nect node through the inbound queue. Thus, there is a common starting point for Con-nect to invoke the SNADS functions.

Installations also have one or several outbound queues. These queues are used to direct DIUs to other nodes.

Con-nect users can exchange information with users on other Con-nect nodes and with users of other office systems that provide SNADS implementations (e.g. DISOSS). Additionally, Con-nect users can also exchange information with users of office systems that do not provide SNADS implementations (e.g. PROFS) whenever gateway functionalities such as those provided by DISOSS or Soft-Switch Central are used.

Con-nect SNADS and the SNADS Network

You need to choose a suitable location for Con-nect SNADS in your SNADS network by determining which existing SNADS node(s) the new system will communicate with directly. Con-nect SNADS should then be located adjacent to that node.

Since the existence and location of Con-nect SNADS must be properly reflected in the other SNADS network nodes (i.e. in their routing tables), ensure that the Con-nect SNADS location you have chosen can be reasonably implemented in the SNADS routing tables of both existing SNADS nodes and the new Con-nect SNADS.

Con-nect SNADS uses SNA logical unit type 6.2 (LU6.2) links to exchange data with neighboring nodes in the SNADS network. These links must be established before Con-nect SNADS can be used.

If routing tables are not consistently specified, DIUs may be forwarded endlessly through the SNADS network. To prevent this from occurring, the SNADS architecture uses a "hop count" to limit node-to-node forwarding operations.

The "hop count" is specified in the SNADS Distribution Interchange Units (DIUs) and defines the maximum number of "legs" the DIU can travel. Con-nect SNADS uses a value specified at installation (though it can be changed later) as the "hop count" for all DIUs that originate from this location.

Con-nect SNADS and Con-nect

Each Con-nect SNADS system uses only one Con-nect spool file as a data repository. However, that spool file can service more than one Con-nect system file.

Note: Each Con-nect system file serviced by the Con-nect SNADS system must be assigned at least one SNADS Distribution Service Unit Name (DSUN) or SNADS "node name".

Because Con-nect SNADS uses features of multi-node Con-nect internally to convey information between the Con-nect spool file and the associated Con-nect system file(s), each Con-nect system file must have a unique multi-node node name stored in the spool file.

Note: Node names are stored in the spool file using the Initialize Node File function of the Con-nect ADMIN program.

The Con-nect system file node name (the name specified as your Con-nect system during the Initialize Node File function) is used as the Distribution Group Name (DGN) along with the Con-nect user ID (as Distribution Element Name (DEN)) to form the default Distribution User Name (DUN):

```
DUN=DGN.DEN
DUN=node.user
```

Con-nect SNADS and Natural

The size of the Natural buffer pool and the number of Natural threads and Natural roll files should reflect the increased system load that will be created by the Con-nect SNADS routines.

Requirements for the Natural Nucleus Parameter (NATPARM)

As part of the installation process, you will need to ensure that certain Natural parameters are set correctly. For a listing of the Natural parameters, see *Modify the "queue server" front-end parameter module CONSNADS*. Con-nect SNADS can use the Natural nucleus with either of the following setups:

1. In the "end user" application of Con-nect (i.e. the Send function).
2. In the "server programs" application of Con-nect SNADS (e.g. the DS_SEND program).

The setups above impose a number of requirements for the NATPARM settings. However, the requirements for the second example are more specific and restrictive.

Many sites will find the required NATPARMs for the "server programs" application unacceptable in normal Natural operations. Depending upon how the Natural nuclei are used, you have three options for where and when you can set the required NATPARMs:

1. You can use one nucleus for both the "server programs" application and the "end user" application with a NATPARM that enforces the most restrictive "server usage" requirements.
2. You can choose to generate a separate nucleus solely for the use of the "server programs" application.
3. You can use one nucleus for both the "end user" and "server programs" applications with a NATPARM that enforces only the less restrictive "end user" requirements and the Natural session parameters that enforces the restrictive "server programs" requirements.

This can be set up by means of Natural dynamic parameters at the beginning of the respective Natural sessions. The Natural session parameters can be specified in the respective keywords in the CONSNADS module (e.g. DYNRECV).

The CONSNADS Parameter Module

The CONSNADS module is linked with the front-end programs, CSSEND, CSRECV, CSINIT, and CSTERM. The purpose of these 370 assembler programs is twofold:

- They permit the Natural "server programs" application to be invoked by a single CICS transaction code;
- They permit Natural dynamic parameters to be passed to the nucleus to be used by the "server programs" application.

If required, CSDBAS can be linked to the front- and back-end programs so that Adabas OPEN (OP command) and CLOSE (CL command) commands occur at the beginning and end of an asynchronous Natural task.

You can satisfy the NATPARM requirements by using dynamic overrides which are specified within the CONSNADS module. This eliminates the need for a separate nucleus while satisfying the NATPARM requirements of the "server programs" application.

For detailed information about the CONSNADS parameter module, see *Modify the "queue server" front-end parameter module CONSNADS*.

Note: Since errors in this portion of the system are difficult to debug, it is recommended that you determine the method you will use to satisfy the NATPARM requirements before you begin the installation process. Any modifications to the Natural dynamic parameters must be verified in the interactive mode.

TP Monitor and Natural Nucleus Considerations

Con-nect SNADS requires CICS LU6.2 support for the physical transmission of messages and documents between nodes. Additionally, CICS interval control facilities such as the CICS `START` command to schedule the "server programs" are required. Therefore the "server programs" application of Con-nect SNADS must be executed by Natural running under CICS.

Because the "end user" application does not perform the physical transmission of data, you can configure your Con-nect SNADS system so that users are supported from a TP monitor even though the "server programs" application runs from Natural under CICS.

This can be achieved by using a single Con-nect spool file that is shared by the "server programs" application under CICS and the "end user" application. The "end user" application can run in another CICS or a different system such as Com-plete.

For this configuration alternative, the following are required:

- Two Natural nuclei, one for the CICS environment and the other for the "end-user" environment:
  - The CICS Natural nucleus must satisfy all installation requirements for the "server programs" application.
  - The Natural nucleus on the "end user TP monitor" must satisfy all installation requirements for the "end user" application.
- Optionally, a single Natural system file (FNAT) can be shared between the nuclei.

In addition, the following requirements must be satisfied for the Con-nect system files and spool (telex) file:

- A single Con-nect "spool" (telex) file must be shared between the Natural nuclei (logical file number 223 in the NATPARM) for the "end user" and "server programs" applications.
- A link between the Con-nect system file(s) and the shared Con-nect spool file must be created. The following is a description of how this link is created:
  - For each Con-nect system file you have a "home node record" must be defined to the spool file using the Initialize Node File function on the Con-nect "Administration - External Mail Nodes" screen. For further information on the Initialize Node File function, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*. This process creates a record with a name (used as the DGN for SNADS), the current physical DBID, and the current physical FNR of the logical file 251.
  - If separate logical 251 files are used, you must invoke Natural with a different logical file 251 setting each time you write these records in order to properly value the home node record. This can be achieved by using the Natural dynamic parameter LFILE.
  - If you move a Con-nect system file to a different database or file number, you must re-initialize the home node record with the new setting before mail can be sent or received.

- You can verify the correlation of DSUNs to DGNs and their physical locations (DBID,FNR) by selecting the Control Maintenance function in the Con-nect "SNADS Administration Menu" screen. For further information about the Control Maintenance function, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Control Maintenance*.

Con-nect SNADS and Natural Security

Natural assigns a user ID to both non-interactive and interactive sessions. By default, Natural passes the current user ID to the Adabas nucleus. The user ID allows Adabas to access any end of transaction (ET) data that can be stored for that particular user ID. However, Adabas cannot accept simultaneous sessions with the same user ID. This restriction can cause problems for the DS_RECEIVE, DS_ROUTER_DIRECTOR, and DS_SEND programs. If Natural Security is implemented, more than one instance of these programs can be active simultaneously and all of the programs may attempt to logon with the same Natural Security user ID.

Therefore, if Natural Security is installed in the environment in which Con-nect SNADS is to be implemented, user IDs must be reserved for the Con-nect SNADS "server programs". There are two methods that can be used to avoid conflicts resulting from more than one active session assigned to the same user ID.

The first and most convenient method is to prevent Natural from passing user IDs to Adabas. This is accomplished by assigning a value of blank to the Natural keyword ETID, i.e. ETID=' ' (blank). Because the Con-nect SNADS "server programs" application does not use the ET data facility, data integrity problems will not occur. In this case, only one user ID is necessary for the "server programs" application.

Note: The ETID setting can be modified in the Natural Security definition or in the dynamic parameter settings in CONSNADS.

However, when ETID is set to blank, Natural does not issue OP and CL commands at the beginning or end of a session. Nevertheless, the module CSDBAS can be linked to the front- and back-end programs to issue OP and CL commands against all databases which are specified in the CONSNADS parameter module.

The second and less convenient method is to allow the appropriate Con-nect SNADS server programs to manage a pool of user IDs. With this method, a user ID which is currently inactive is selected and assigned to a particular function. User IDs consist of a one-to-six character prefix and a two-digit number from 01 through 99 and are specified in the CONSNADS parameter module during the installation process. The two-digit number represents the number of user IDs and should reflect the expected amount of parallel activity.

For example, if you specify XYZ as the prefix and 02 for the number of user IDs, the "server programs" will use either XYZ01 or XYZ02 as a user ID. All user IDs must be defined to the Natural Security system. Also, a password must be associated with each user ID; the same password can be assigned to all user IDs.

This method is not recommended for use. It is supported for compatibility reasons only and may be withdrawn in future versions.

15 The Con-nect SNADS Installation Medium

This chapter contains information about the Con-nect SNADS installation medium.

Installation Medium Description

The installation medium for Con-nect SNADS uses a standard label, 1600 BPI tape with volume serial number CSL*nnn*. If you have been shipped a special tape, the tape can have a different volume serial number. The installation medium contains the following data sets in the order listed: where *nnn* equals the version number.

1. CSL*nnn*.SRCE - Con-nect SNADS Source Library

   The z/OS version of this data set is an unloaded partitioned data set that was created using the IBM IEBCOPY utility.

   CSL*nnn*.SRCE is the Con-nect SNADS source library; it contains the following members:

Member Name	Content
CFSNADS	A macro that is referred to by source programs CPINIT, CPSEND, CPTERM, CPEND, CSINIT, CSRECV, CSSEND and CSTERM.
CONSNADS	A sample Con-nect SNADS parameter module for systems running under CICS.
CONSNADZ	A sample Con-nect SNADS parameter module for systems running under Com-plete.
CSDBAS	A CICS command level program used to issue Adabas OP (open) and CL (close) commands.
CPDBAS	A Com-plete command level program used to issue Adabas OP (open) and CL (close) commands.
CPEND	A Com-plete command level program used to terminate functions when the execution of an asynchronous server program is completed.
CSINIT	A CICS command level program used as a front-end program for Natural to invoke the Con-nect SNADS queue initialization function.
CPINIT	A Com-plete command level program used as a front-end program for Natural to invoke the Con-nect SNADS queue initialization function.
CSNADS	A macro used for generating the Con-nect SNADS parameter module.
CSRECV	A CICS command level program used as a front-end program for Natural to invoke the Con-nect SNADS DS_RECEIVE program.
CSSEND	A CICS command level program used as a front-end program for Natural to invoke either the Con-nect SNADS DS_SEND or the DS_ROUTER_DIRECTOR program.
CPSEND	A Com-plete command level program used as a front-end program for Natural to invoke either the Con-nect SNADS DS_SEND, DS_RECEIVE, or the DS_ROUTER_ DIRECTOR program.
CSTERM	A CICS command level program used as a front-end program for Natural to invoke the Con-nect SNADS queue termination function.

Member Name	Content
CPTERM	A Com-plete command level program used as a front-end program for Natural to invoke the Con-nect SNADS queue termination function CSINIT.

2. CSL*nnn*.INPL

   DCB=(RECFM=VB,BLKSIZE=4628,LRECL=4624)

   A data set used as input to the Natural INPL utility. CSL*nnn*.INPL contains a number of Natural programs, subprograms, maps and local data areas to be loaded into the application SYSCNT2 in the Natural system file.

   Generated with IEBGENER.

16

Installing Con-nect SNADS in Com-plete

This chapter describes step by step how to install Con-nect SNADS for execution under Com-plete using the EntireX Broker Services LU6.2 API (refered to in this section as the LU6.2 API).

Note: The LU6.2 API uses the Natural VIEW COMMUNICATE-LU62. To install Con-nect SNADS, Adabas and Natural DBA skills are required and you must be familiar with the Con-nect administration facilities as well.

Overview

Before you begin the installation process, you need the following information:

- The LU name which the VTAM system programmer assigned to the local and adjacent systems. See step 9.
- The VTAM log mode name which is used for APPC sessions to the adjacent systems. See step 9.

Installation Steps for Com-plete

1. Load the Con-nect SNADS programs from data set CSLnnn.INPL

The CSL*nnn*.INPL data set contains a number of Natural programs that supplement the Con-nect system programs in the SYSCNT2 application. The Natural utility program INPL must be used to load the contents of the data set into the appropriate Natural system file. Objects are loaded only into the SYSCNT2 system libraries.

Note: All objects begin with the characters X-F or XF.

In addition, program XFSEC, which is distributed as source code, must be configured to your system's requirements.

The purpose of program XFSEC is to customize the use of security options in the LU6.2 API. The supplied model program contains detailed comments so that you can understand the logic implemented. Knowledge of the security facilities of the LU6.2 API, however, is necessary. For further information, see the *EntireX Broker Services Reference Manual*.

2. Relink your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) which will be used by the "end user" and "server programs" applications of Con-nect SNADS. The requirements for the "end user" application are less stringent than those for the "server programs" application.

- Following are the requirements for the Con-nect SNADS "end user" application:
  - The environment must include a fully operational Con-nect system Version 3.4.3.
  - The environment must contain a valid reference to a Con-nect spool file. The spool file identifier number is 223. This reference can be specified in either the Natural parameter module with a NTLFILE macro call, or dynamically with the Natural keyword LFILE.

Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The Con-nect SNADS system programs must be loaded into the appropriate Natural system file.

Note: The following requirements only apply if event-driven scheduling is used for the Con-nect SNADS inbound queue; that is, if the DS_ROUTER_DIRECTOR program is to be immediately invoked when a send request is submitted or when the DS_RECEIVE program receives a SNADS message from another node within the network.

- The Natural nucleus must run in a Com-plete environment.
- The Com-plete function, ATTACH, must be cataloged in the Com-plete library so that it can be accessed by the Natural nucleus.

Following are the requirements for the Con-nect SNADS "server programs" application:

- All of the requirements for the "end user" application also apply to the "server programs" application.
- The statement NTDB PROCESS,148 must be added to the Natural parameter module.
- The Natural PROCESS auxiliary buffer, ASIZE, must reflect the requirments of Software AG's XCOM Communication Module. The recommended value is 28.
- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
RELO	OFF
MADIO	0
MAXCL	0
AUTO	OFF
ETID (see note below)	Blank

Note: The setting ETID=' ' (blank) is recommended but not required. However, if the Natural keyword ETID is set to blank, and if more than one instance of a Con-nect SNADS server program becomes active at one time with more than one of the instances using the same Natural user ID, Adabas will not issue an error message.

3. Modify the "queue server" front-end parameter module CONSNADZ

Note: Always check that the CONSNADZ parameters are valid. If not, you will have to re-execute this step and the following three steps.

The CONSNADZ module is an Assembler program that uses the keyword macro CSNADS. It is linked to the four Con-nect SNADS queue server front-end programs in order to provide the information required to handle the Natural front-end logic, e.g. the name of the Com-plete transaction code assigned to Natural; and specify the dynamic parameters that must be passed to the Natural sessions.

Natural parameters that were not included as NATPARMs for the nucleus can be specified dynamically in the CONSNADZ parameter module.

- In the CONSNADZ parameter module, you must specify:
  - the name used for the Con-nect SNADS front-end program while setting up the Initialize Queue and watchdog functions. The supplied front-end program is CPINIT;
  - the Natural transaction used for the "server programs" application; and
  - the appropriate values for the CONSNADZ keyword parameters that are used by the CSNADS macro (see the table below).
- Assemble the CONSNADZ parameter module using the Con-nect SNADS source data set as a macro library.

Required CONSNADZ Keyword Parameters

The following table lists the required CONSNADZ keyword parameters and describes the value associated with each one:

<table>
<tr><th>Keyword</th><th>Value</th></tr>
<tr><td>DYNINIT</td><td>A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the initialize queues program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FMRUN I" (if you are using the watchdog function, replace the command "X-FMRUN I" with "X-FWTCH %"), and terminate the Natural session. Example stack parameter:
<code>STACK=(LOGON SYSCNT2;X-FMRUN I;FIN)</code>
Example program parameter:
<code>PROGRAM=cpend</code>
where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYNINIT parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.</td></tr>
<tr><td>DYNROUT</td><td>A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the DS_ROUTER_DIRECTOR program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FRPP", and terminate the Natural session. Example stack parameter:
<code>STACK=(LOGON SYSCNT2;X-FRPP;FIN)</code>
Example program parameter:
<code>PROGRAM=cpend</code>
where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYNROUT parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.</td></tr>
<tr><td>DYNSEND</td><td>A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the DS_SEND program. The parameter string must contain the Natural commands required to log on to application SYSCNT2, execute the command "X-FOP000 %", and terminate the Natural session. The percent sign (%) is substituted at execution time by the identifier that represents the queue to be processed. Example stack parameter: STACK=(LOGON SYSCNT2;X-FOP000 % I;FIN) Example program parameter: PROGRAM=cpend where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYNSEND parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.</td></tr>
<tr><td>DYMDEMN</td><td>A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the "demon" process. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FDEM", and terminate the Natural session. Example stack parameter:</td></tr>
</table>

Keyword	Value

	`STACK=(LOGON SYSCNT2;X-FDEM ;FIN)`
	Example program parameter:
	`PROGRAM=cpend`
	where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYMDEMN parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.

DYNTERM	A dynamic parameter string in a Natural stack parameter that is passed to the Natural nucleus in order to execute the "terminate queues" program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FMRUN T", and terminate the Natural session. Example stack parameter:
	`STACK=(LOGON SYSCNT2;X-FMRUN T;FIN)`
	Example program parameter:
	`PROGRAM=cpend`
	where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYNTERM parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.

NPGRMID	The name of the non-resident portion of the Natural Com-plete driver as it has been cataloged in the Com-plete library.

4. Specify keyword parameters (if required) to perform periodic supervision ("watchdog") of the Con-nect SNADS tasks

The "watchdog" program is generated to periodically supervise the status of the system and attempt to recover it automatically when necessary. When a temporary system failure is detected, the watchdog program issues a message to the operator console and records the event in the Con-nect SNADS log records. Then the watchdog program re-adjusts the status queues and re-starts the queue servers. However, the watchdog program cannot acquire inactive APPC connections.

The following table lists the required keyword parameters and describes the value associated with each one:

Keyword	Value
WATCH	A value of YES or NO. When the value is YES, the watchdog program is generated and replaces the Initialize Queues function. The default value is NO.
WATCHIN	The length of time (in minutes) between subsequent executions of the watchdog program. The default value is 15.

If the watchdog program is generated, be sure that "X-FWTCH %" is specified in the Stack portion of the CONSNADZ DYNINIT keyword parameter (instead of "X-FMRUN I").

5. Assemble and link (if required) the Con-nect SNADS Adabas interface module

The Con-nect SNADS Adabas interface module is a subroutine which permits Adabas OPEN (OP command) and CLOSE (CL command) commands to be executed from within the front- and back-end programs. The interface module is a member of the source data set CPDBAS and must be link-edited with each of the front-end programs.

It is not necessary to use the Adabas interface module unless your Con-nect SNADS front- and back-end programs are to issue OP and CL commands.

The following table lists the required keyword parameters and describes the value associated with each one:

Keyword	Value

ADADBID	Specify the database number against which OP and CL commands are to be issued. If files in multiple databases are to be opened and closed (using the OP and CL commands), the database numbers must be specified in subparameter notation. For example:
	`ADADBID=2`
	OP and CL commands will be directed to database 2.
	`ADADBID=(10,15)`
	OP and CL commands will be directed to databases 10 and 15.

ADAUID	Specify whether the startup programs will generate an Adabas user ID when OP commands are issued. Usually, Adabas user IDs are not used when ETID= ' ' (blank). Therefore this keyword should not be specified. Valid values are YES and NO. The default value is NO.
ADALINK	Specify whether the Con-nect SNADS Adabas interface module is entered with the Com-plete COLINK function (cataloged in the Com-plete library), or if it is link-edited with each of the startup programs. The default value is NO. Change the default value to YES, if the module is to be entered with the Com-plete COLINK function.
ADAPGM	The name of the Con-nect SNADS Adabas interface module. The default value is CPDBAS.
ADAOPRB	Specify the record buffer contents for the OP command. If this parameter has not been specified, OP commands will not be issued. If OP commands are to be directed to multiple databases, multiple record buffer contents must be specified in subparameter notation. All Natural and Con-nect system file numbers which are used must be specified. For example:

Keyword	Value

	`ADADBID=2, ADAOPRB='UPD=4,8,12.'`
	In this case, an OP command will be directed to database number 2 with the record buffer contents 'UPD=4,8,12.'
	`ADADBID=(5,10,15), ADAOPRB=('UPD=3,6.',,'ACC.')`
	An OP command, in this example, will be directed to databases 5 and 15 with the record buffer contents 'UPD=3,6.' and 'ACC.'. OP commands will not be directed to database 10.

ADACLSE	Specify whether the back-end program CPEND will issue CL commands at the end of a session. Valid values are YES and NO. The default value is NO.

6. Assembling and link-editing the "queue server" front-end programs

The names of the supplied front-end programs for the "queue server" functions are as follows:

Program	Queue Server Function
CPEND	Terminates asynchronous server programs.
CPINIT	Initializes queues.
CPSEND	DS_SEND, DS_ROUTER_DIRECTOR, and DEMON.
CPTERM	Terminate queues.

Note: If OP and CL commands are to be issued, the Con-nect SNADS Adabas interface module CPDBAS must be link-edited with each of the front-end programs.

- Assign the appropriate Com-plete source data set and the Con-nect SNADS source data set as macro libraries, and assemble the queue server front-end programs and CPDBAS (if required).
- Reference the Com-plete library, link-edit the queue server front-end programs and CPDBAS (if required) with the CONSNADZ parameter module created in the previous step.
- Ensure that an execution will start with the first instruction of the queue server front-end programs (by placing the queue server front-end programs at the beginning of the lode modules, or by specifying the appropriate ENTRY statements).
- Catalog the queue server front-end programs to the Com-plete library. No specifications are required for the region size and the programs do not need to be marked as privileged.

7. Define local node for each Con-nect system file serviced

Note: If you change the DBID or FNR of your Con-nect system file, you will have to re-execute this and the following steps.

Con-nect SNADS uses spool method routines for transferring data from the Con-nect system file (for local node data) to the Con-nect spool file (for external node data) and vice versa. Thus, a local node for the spool file method must be defined.

- Select the System Maintenance function from the "Con-nect Administration Menu".
- Invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen.

Caution: The following sub-step must be completed before you begin step 8.

- Select the Define Local Node function and, if you have not already done so, define a name for your spool file in the "Define Local Node" field and press ENTER. You are then prompted to define a local node for the transport service, although it is not required for Con-nect SNADS. Define a name in the "Define Local Node" field and press ENTER. For further information, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance*, sub-section *Define Local Node*.

By default, Con-nect SNADS uses the name you enter in the "Define Local Node" field for the spool file method as the Distribution Group Name (DGN) when mapping local Con-nect user identifiers to the SNADS Distribution User Names (DUNs).

8. Define control information for nodes (system files) - SNADS Initialization

You must define the control information for all SNADS Distribution Service Units (DSUs) that will be serviced from the current Con-nect spool file as local SNADS nodes.

To define Con-nect system files as nodes in the SNADS network, perform the following steps:

- Select the System Maintenance function from the Con-nect Administration - Main Menu.
- Invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen.
- Select the Maintain Mail Nodes function and mark "F Snads" in the resulting Maintain Mail Node window.
- Invoke the Initialization function from the "SNADS Administration Menu" screen. The "Control Maintenance" screen is displayed. Enter the following information:
    - Specify the DSUN (RGN and REN) of the Con-nect system file in the "Local Node Name" fields. In this context, a "Con-nect System" is the system file.
    - Specify the Natural PROCESS ID for the desired LU6.2 API nucleus in the "NPR Node" field.

  - Specify the Con-nect SNADS send front-end program in the "Com-plete Send" field. Specify the name under which the CPSEND program has been cataloged in the Com-plete library.

For more information about the "Control Maintenance" screen refer to the *Con-nect Administration* documentation, section *Con-nect SNADS*, sub-section *Maintaining Con-nect SNADS*, and sub-section *Control Maintenance*.

9. Create one or multiple outbound queues for the storing of Distribution Interchange Units (DIUs) intended for other nodes

Note: This and the following three steps define the path messages take from Con-nect to other nodes. If changes are made in your SNA network, you may have to re-execute this step.

At least one outbound queue must be created for each adjacent node in the SNADS network. When an outbound queue is created, the attributes of the LU6.2 link between EntireX Broker Services and the adjacent SNADS node must be defined. To create an outbound queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu. The "Queue Maintenance" screen is displayed. Press PF4 to add a queue. The "Queue Info" screen is displayed. Enter the following information:
  - In the "Queue ID" field, enter a one-to-eight character name to be used as the queue ID.
  - In the "Connection ID" field, enter the fully qualified LU name used by the VTAM system programmer to define the adjacent node to the SNA network.
  - In the "Mode-Name" field, enter the VTAM log mode name that the VTAM system programmer set up for the APPC sessions to that node.
  - Press PF5 (Confirm) to add the new outbound queue.

For more information about the "Queue Info" screen, refer to the *Con-nect Administration* documentation, section *Configuring Con-nect SNADS* , sub-section *Outbound Queues*.

10. Set the "Reset" and "Input" statuses for the outbound queue

When you initially test Con-nect SNADS, the input status of the outbound queue must be set to active and the reset status to inactive. To set the input and reset statuses of the outbound queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu to display the "Queue Maintenance" screen.
- Enter `MO` (modify) in the "Cmd" column for the outbound queue and press ENTER. The "Queue Info" screen is displayed. Modify the following fields:
  - In the "Reset Status" field, enter `I` (inactive).

  - In the "Input Status" field, enter `A` (active).
  - Mark the "Reset" field with any character and press PF5.

For more information about the "Queue Maintenance" screen refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS* , sub-section *Queue Maintenance*.

11. Create a dummy queue for the "demon" process

An additional queue must be created which will be used to start the Con-nect SNADS "demon" process. When you create the dummy queue, define the attributes of the fully qualified LU name assigned to the LU6.2 API. To create the dummy queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu to display the "Queue Maintenance" screen. Press PF4 to add a queue. The "Queue Info" screen is displayed.
- In the this screen, enter the following:
  - In the "Queue ID" field, enter "***DMY**" as the name to be used as the "queue ID".
  - In the "Connection ID" field, enter the fully qualified LU name used by the VTAM system programmer to define the LU6.2 API to the SNA Network.
  - In the "Mode-Name" field, enter the VTAM log mode name that the VTAM system programmer set up for the APPC sessions to that node.
  - Press PF5 (Confirm) to add the new outbound queue.

12. Define routing specifications for all other nodes with which you intend to communicate

You must define a routing entry for each node in the network with which you wish to communicate. The routing entry determines how the SNADS DIUs are to be routed to a particular node.

To define routing specifications, perform the following steps:

- Select the Routing Entry Maintenance function from the SNADS Administration Menu. Press PF4 (Add) to add a routing entry. The following must be specified in the "Routing Entry" screen:
  - In the "Recipient Node" field, enter the DSUN (RGN.REN) for each node.
  - In the "Next Queue" field, enter the name of an outbound queue which was defined. See step 10.
  - Press PF5 (Confirm) to add the new routing entry.

For more information about the "Routing Entry" screen, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS* , sub-section *Routing Entry Maintenance* .

13. Add a Mail Node, type F (SNADS), for sending purposes

You must define an external mail node in Con-nect for sending mail.

To define an external mail node, perform the following steps:

- Select the System Maintenance function in the Con-nect Administration - Main Menu to display the "Administration - System Maintenance" screen.
- From the "Administration - System Maintenance" screen, select the External Mail Nodes function and press ENTER to display the "Administration - External Mail Nodes" screen. Specify the node name in the "Mail Node/Type" field (this node name will be used in the "Addressee" field in the Con-nect "Send" screen), select the Add Mail Node function and press ENTER.
- In the "Administration - Add Mail Node" screen, specify the node type F (SNADS) in the "Node Type" field.
- Enter the RGN (Group) and REN (Element) for each external SNADS node (i.e. addressee) to form the DSUN (Node) portion of the SNADS address.

When an object (document or memo) is sent to this addressee, the value you entered for the DSUN (RGN and REN) is automatically provided by the system. The user must supply the DUN (DGN and DEN) for a specific user in order to complete the SNADS address. REN and RGN must be provided in case a generic specification of the DSUN in the mail node definition is given.

14. Test the installation of Con-nect SNADS.

When testing, use the following settings for all queues:

Input Status: A

Output Status: I

Reset Status: I

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you start a queue server when either the outbound or inbound queue is in the "hold" status, the queue server task will immediately exit without processing any information.

If you are unsuccessful in sending messages between nodes with Con-nect queues set to either "event" or "timer" status, use the following procedure to manually step through the process of sending and receiving a Distribution Interchange Unit (DIU):

- Initialize the node. A routing entry, inbound queue, and one outbound queue must be defined to the node.
- Send a mail item from Con-nect. Then check the inbound queue to see if the DIU is there.

- Start the inbound queue from the "Queue Maintenance" screen. This activates the "router", which should route the DIU to the outbound queue. Since the outbound queue is inactive, the DIU should remain in the outbound queue.
- Check to see if the DIU was routed to the outbound queue.
- Check the log records. The log records can be accessed using the Log Information Maintenance function in the SNADS Administration Menu. A list of log records written by the SNADS driver programs are displayed. If the "router" was executed, log records indicating successful or unsuccessful routing are displayed. The exact format of these records is not available; however, you can verify whether or not the tasks have been executed. Typically, a message, e.g. "routing error", which corresponds to a particular log record indicates that an error occurred. Up to this point, LU6.2 has not been involved. The following steps allow you to test sending a DIU to another SNADS node.
- Set the outbound queue with the routed DIU to the "inactive" status and start the outbound queue from the "Queue Maintenance" screen.
- Check the log records. The log records are accessed from the Log Information Maintenance function in the "SNADS Administration Menu". If the DIU was successfully sent, records from the DS_SEND program (X-FON000) are displayed.
- Check for log records which were created when a status request was received from the target node. Records from the DS_RECEIVE program (X-FIN000) should be displayed.
- Check to see whether the DIU is delivered to the recipient node (the Inbasket, if Con-nect is the recipient node). In addition, there should be a "status" DIU generated and routed for delivery to the sender at the origin node.
- If you are still experiencing problems, the following commands and functions can be useful for debugging purposes:
  - VTAM Operator Console Commands
  - LU6.2 API Operator Commands
  - The following functions are offered on the Con-nect SNADS Administration Menu:

    Con-nect SNADS Administration Functions

    Queue Maintenance - Provides a DIU count, time stamps for recent activity, and the ability to manually start the queue server.

    Control Maintenance - Provides the correlation between a DSUN and a system file (DBID and FNR).

    Messages Awaiting Confirmation of Delivery - Displays the number of undelivered DIUs sent from Con-nect.

    Log Information Maintenance - Provides debugging information that can be helpful to Software AG in resolving any problems you are experiencing with Con-nect SNADS.

17 Installing Con-nect SNADS in Batch Mode

This chapter describes step by step how to install Con-nect SNADS for execution in batch mode using the EntireX Broker Services LU6.2 API (refered to in this section as the LU6.2 API).

Note: The LU6.2 API uses the Natural VIEW COMMUNICATE-LU62.

Overview

Before you begin the installation process, you need the following information:

- The LU name which the VTAM system programmer assigned to the local and adjacent systems (See step 5).
- The VTAM log mode name which is used for APPC sessions to the adjacent systems (See step5).

Installation Steps for Batch Mode

1. Load the Con-nect SNADS programs from data set CSL nnn.INPL

The CSL *nnn*.INPL data set contains a number of Natural programs which supplement the Con-nect system program in the SYSCNT2 application. The Natural utility program INPL must be used to load the contents of the data set into the appropriate Natural system file. Objects are loaded only into the SYSCNT2 system libraries.

2. Relink your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs

Note: If you change the DBID or FNR of a Con-nect system file, you have to re-execute this and the next step.

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) which will be used by the "end user" and "server programs" applications of Con-nect SNADS. The requirements for the "end user" application are less stringent than the requirements for the "server programs" application.

Following are the requirements for the Con-nect SNADS "end user" application:

- The environment must include a fully operational Con-nect system Version 3.4.3.
- The environment must contain a valid reference to a Con-nect spool file. The spool file identifier number is 223. This reference can be specified in the Natural parameter module with a NTLFILE macro call, or dynamically with the Natural keyword LFILE.

Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The Con-nect SNADS system programs must be loaded into the appropriate Natural system file.

Following are the requirements for the Con-nect SNADS "server programs" application:

- All of the requirements for the "end user" application also apply to the "server programs" application.
- The statement NTDB PROCESS, 148 must be added to the Natural parameter module.
- The Natural PROCESS auxiliary buffer, ASIZE, must reflect the requirements of Software AG's XCOM Communication Module. The recommended value is 28.
- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
MADIO	0
MAXCL	0
AUTO	OFF
ETID (see note below)	Blank

Note: The setting ETID=' ' (blank) is recommended but not required. However, if the Natural keyword ETID is set to blank, and if more than one instance of a Con-nect SNADS server program becomes active at one time with more than one of those instances using the same Natural user ID, Adabas will not issue an error message.

3. Define local node for each Con-nect system file serviced

Con-nect SNADS uses spool file method routines for transferring data from the Con-nect system file (for local node data) to the Con-nect spool file (for external node data) and vice versa. Thus, a local node for the spool file method must be defined.

- Select the System Maintenance function from the Con-nect Administration Menu.
- Invoke the External Mail Nodes function on the "Administration - System Maintenance" screen. The "Administration - External Mail Nodes" screen is displayed.

Caution: The following sub-step must be completed before you begin step 4.

- Select the Define Local Node function and, if you have not already done so, define a name for your spool file in the "Define Local Node" field and press ENTER. You are then prompted to define a local node for the transport service, although it is not required for Con-nect SNADS. Define a name in the "Define Local Node" field and press ENTER. For further information, see the *Con-nect Administration* documentation, section *System Maintenance*, sub-section *Define Local Node*.

By default, Con-nect SNADS uses the name you enter in the "Define Local Node" field for the spool file method as the Distribution Group Name (DGN) when mapping local Con-nect user identifiers to the SNADS Distribution User Names (DUNs).

4. Define control information for nodes (system files) - SNADS Initialization

You must define the control information for all SNADS Distribution Service Units (DSUs) that will be serviced from the current Con-nect spool file as local SNADS nodes.

To define Con-nect system files as nodes in the SNADS network, perform the following steps:

- Select the System Maintenance function from the Con-nect Administration - Main Menu.
- Invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen.
- Select the Maintain Mail Nodes function and mark "F Snads" in the resulting Maintain Mail Node window.
- Invoke the Initialization function from the SNADS Administration Menu. The "Control Maintenance" screen is displayed.
- On the "Control Maintenance" screen, specify the DSUN (RGN and REN) of the Con-nect system file. In this context, a "Con-nect System" is the system file.
- Specify the Natural PROCESS ID for the desired LU6.2 API nucleus in the "NPR Node" field.

Note: Specifying a time interval in the "Retry Delay" field is not needed since the queues are initiated with a batch task.

For more information about the "Control Maintenance" screen, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS* , sub-section *Control Maintenance*.

5. Create one or multiple outbound queues for the storing of Distribution Interchange Units (DIUs) intended for other nodes

Note: This and the following three steps define the path messages take from Con-nect to other nodes. If changes are made in your SNA network, you may have to re-execute this step.

At least one outbound queue must be created for each adjacent node in the SNADS network. When an outbound queue is created, the attributes of the LU6.2 link between the EntireX Broker Services and the adjacent SNADS node must be defined. To create an outbound queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu. The "Queue Maintenance" screen is displayed. Press PF4 to add a queue. The "Queue Info" screen is displayed.
- In the "Queue Info" screen, enter the following:

In the "Queue ID" field, enter a one-to-eight character name to be used as the queue ID.

In the "Connection ID" field, enter the fully qualified LU name used by the VTAM system programmer to define the adjacent node in the SNA network.

In the "Mode-Name" field, enter the VTAM log mode name that the VTAM system programmer setup for APPC sessions to that node.

Press PF5 (Confirm) to add the new outbound queue.

Note: The status of the "Reset" field must be either I (Inactive) if the queue is to be processed regularly or H (held) if regular processing is to be suspended. Specifying a time interval in the "Time Interval" field is not necessary since the queues are activated with a Natural task.

For more information about the "Queue Info" screen, refer to the *Con-nect Administration* documentation, section *Configuring Con-nect SNADS* , sub-section *Outbound Queues*.

6. Set the "Reset", and "Input" statuses for the outbound queue

When you initially test Con-nect SNADS, the input status of the outbound queue must be set to active, and the reset status to inactive. To set the input and reset statuses of the outbound queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu to display the "Queue Maintenance" screen.
- Enter MO (modify) in the "Cmd" column for the outbound queue and press ENTER. The "Queue Info" screen is displayed. Modify the following fields:

In the "Reset Status" field, enter I (inactive).

In the "Input Status" field, enter A (active).

Mark the "Reset" field with any character and press PF5.

For more information about the "Queue Maintenance" screen, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Queue Maintenance*.

7. Create a dummy queue for the demon process

An additional queue must be defined which will be used to start the Con-nect SNADS "demon" process. When you create the dummy queue, define the attributes of the fully qualified LU name assigned to the LU6.2 API. To create the dummy queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu. The "Queue Maintenance" screen is displayed. Press PF4 to add a queue. The "Queue Info" screen is displayed.
- In the this screen, perform the following steps:

  In the "Queue ID" field, enter "***DMY**" as the name to be used as the queue ID.

  In the "Connection ID" field, enter the fully qualified LU name used by the VTAM system programmer to define the LU6.2 API to the SNA Network.

  In the "Mode-Name" field, enter the VTAM log mode name that will be used for the APPC connections allocated by the adjacent nodes.

  Press PF5 (Confirm) to add the new outbound queue.

8. Define the JCL server task procedures

The "server tasks" consist of two different JCL procedures, the "input handler" and the "queue server", which run simultaneously. At least one input handler and queue server must be set up.

- The input handler is used as an interface to the LU6.2 API and initiates the demon process. When the demon process is initiated, it listens for transaction requests from remote SNADS partner nodes. When a request is caught, the demon process executes the DS_RECEIVE program. The input handler procedure consists of the execution of the Natural nucleus in batch mode. The following is an example of a JCL input handler procedure:

  ```
  //CMSYNIN DD *
  LOGON SYSCNT2
  X-FDEM
  FIN
  /*
  ```

  To stop the input handler procedure you must set the status of the dummy queue to H (hold).
- The queue server processes the inbound queue and the outbound queues. A queue server can be specified to handle one or a multiple of queues. The following is an example of a JCL queue server procedure:

```
//CMSYNIN DD *
LOGON SYSCNT2
X-FSCHED 2
***INB**
*
/END/
FIN
/*
```

where "2" is the length of time (in minutes) between subsequent processing of the specified queues.

Whenever you want to process the inbound queue, that queue must be specified as "***INB**", whereas the outbound queues can be specified with a name or a wildcard.

If you want to specify the outbound queues with a wildcard, you can either:

- enter an asterisk (*) so that all outbound queues are processed, or
- enter the first couple of letters of a queue's name followed by an asterisk (*) so that only those outbound queues whose names begin with the specified letters are processed.

To stop the queue server procedure you must set the statuses of all queues, specified within the procedure, to H (hold).

9. Define routing specifications for all other nodes with which you intend to communicate

You must define a routing entry for each node in the network with which you wish to communicate. The routing entry determines how the SNADS DIUs are to be routed to a particular node.

To define routing specifications, perform the following steps:

- Select the Routing Entry Maintenance function from the SNADS Administration Menu to display the "Routing Entry Maintenance" screen. Press PF4 to add a routing entry.

  The following must specified in the "Routing Entry" screen:

  In the "Recipient Node" field, enter the DSUN (RGN and REN) for each node.

  In the "Next Queue" field, enter the name of a defined outbound queue. See step 6.

  Press PF5 (Confirm) to add the new routing entry.

For more information about the "Routing Entry" screen, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS* , sub-section *Routing Entry Maintenance* .

10. Add a Mail Node, type F (SNADS), for sending purposes

You must define an external mail node in Con-nect for sending mail.

To define an external mail node, perform the following steps:

- Select the System Maintenance function in the Con-nect Administration - Main Menu.
- From the "Administration - System Maintenance" screen, invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen. Specify the node name in the "Mail Node/Type" field (this node name will be used in the "Addressee" field in the Con-nect "Send" screen), and select the Add Mail Node function. Press `ENTER`.
- In the "Node Type" field specify F for SNADS.
- Enter the RGN (Group) and REN (Element) for each external SNADS node (i.e. Addressee) to form the DSUN (Node) portion of the SNADS address.

When an object (document or memo) is sent to this addressee, the value you entered for the DSUN (RGN and REN) is automatically provided by the system. The user must supply the DUN (DGN and DEN) for a specific user in order to complete the SNADS address.

11. Test the installation of Con-nect SNADS

When testing, use the following settings for all queues:

Input Status: A

Output Status: I

Reset Status: I

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you try to start a queue server when either the outbound or inbound queue is in the "hold" status, the queue server task will immediately exit without processing any information.

If you are unsuccessful in sending messages between nodes with the Con-nect queues set to either "event" or "timer" status, use the following procedure to manually step through the process of sending and receiving a Distribution Interchange Unit (DIU):

- Initialize the node. A routing entry, inbound queue, and one outbound queue must be defined to the node.
- Start the input handler procedure. This activates the demon process so that it listens for transmission requests from remote SNADS partners.
- Send a mail item from Con-nect. Then check the inbound queue to see if the DIU is there.

- Start the queue server procedure for the inbound queue (***INB**). This activates the "router", which should route the DIU to the outbound queue. Since the outbound queue is inactive, the DIU should remain in the outbound queue.
- Check to see if the DIU was routed to the outbound queue.
- Check the log records. The log records can be accessed using the Log Information Maintenance function in the SNADS Administration Menu. A list of log records written by the SNADS driver programs are displayed. If the router was executed, log records indicating successful or unsuccessful routing are displayed. The exact format of these records is not available; however, you can verify whether or not tasks have been executed. Typically, a message, e.g. "routing error", which corresponds to a particular log record indicates that an error occurred. Up to this point, LU6.2 has not been involved. The following steps allow you to test sending a DIU to another SNADS node.
- Start the queue server process for the respective outbound queue.
- Check the log records. The log records are accessed from the Log Information Maintenance function in the SNADS Administration Menu. If the DIU was successfully sent, records from the DS_SEND program (X-FON000) are displayed.
- Check for log records which were created when a status request was received from the target node. Records from the DS_RECEIVE program (X-FIN000) should be displayed.
- Check to see whether the DIU is delivered to the recipient node (the Inbasket, if Con-nect is the recipient node). In addition, there should be a "status" DIU generated and routed for delivery to the sender at the origin node.
- If you are still experiencing problems, the following commands and functions can be useful for debugging purposes:

  VTAM Operator Console Commands

  LU6.2 API Operator Commands

  Con-nect SNADS Administration Functions

  The following functions are offered in the Con-nect SNADS Administration Menu:

  Control Maintenance - Provides the correlation between a DSUN and a system file (DBID and FNR).

  Messages Awaiting Confirmation of Delivery - Displays the number of undelivered DIUs sent from Con-nect.

  Log Information Maintenance - Provides debugging information that can be helpful to Software AG in resolving any problems you are experiencing with Con-nect SNADS.

18 Installing Con-nect SNADS Running Under CICS

This chapter describes the steps that must be performed to install Con-nect SNADS Version 1.5.3 under CICS.

Note: To install Con-nect SNADS, Adabas and Natural DBA skills are required and you must be familiar with the Con-nect administration facilities as well.

Overview

Before you begin the installation process, you need the following information:

- The "connection" names used by the CICS system programmer when defining the adjacent nodes to CICS using CEDA for TCT generation (See step **11**).
- The VTAM log mode names used by the CICS system programmer when defining sessions for the above mentioned connections.

Installation Steps for CICS

1. Load the Con-nect SNADS programs from data set CSLnnn.INPL

The CSL*nnn*.INPL data set contains a number of Natural programs that supplement the Con-nect system program in the SYSCNT2 application. The Natural utility program INPL must be used to load the contents of the data set into the appropriate Natural system file. Objects are loaded only into the SYSCNT2 system libraries.

2. Use the local macro libraries and the CICS preprocessor, DFHEAP1$, to assemble and link-edit CXLU62 if not already done for transport service

CXLU62 is the Con-nect external communication module and is located on the CNT*nnn*.SRCE data set. The CXLU62 module is written in Assembler and uses CICS commands.

- Process the CXLU62 communication module using the CICS command language processor DFHEAP1$.
- Assemble the translated output using the appropriate CICS, Natural, and Natural CICS source data sets as macro libraries.
- Link-edit the CXLU62 module. The CXLU62 module is referenced by the CSTATIC list of a Natural parameter module. Since a BALR instruction must be used to branch to the CXLU62 module (rather than by a CICS LINK or XCTL command), the system administrator can use one of the following three options to link-edit the CXLU62 module:
  - Option One: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is part of a Natural CICS nucleus, and CXLU62 is linked to the Natural CICS

  nucleus. In this case, the CXLU62 module uses the CICS exec interface stub included with the Natural CICS nucleus.

  With this option, the CXLU62 module should not be linked with the CICS exec interface stub modules.

  - Option Two: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is part of a Natural CICS nucleus, and CXLU62 is not linked to the Natural CICS nucleus. In this case, the CXLU62 module is loaded by CICS each time a Natural session is initialized by using the Natural RCA (Resolve CSTATIC Addresses) feature. The setting RCA=ON is used to activate the RCA parameter.

    With this option, the CXLU62 module must be linked with the CICS exec interface stub modules. Since Natural will branch to the beginning of the CXLU62 load module, the CSECTs of the load module must be linked in the following sequence: CXLU62, DFHEAI, and then DFHEAI0. The name of the resulting load module must be CXLU62.

    Next, the CXLU62 module must be copied into one of the step libraries of the CICS nucleus and referenced by the appropriate PPT entry. The PPT entry must indicate that CXLU62 is resident. Additionally, if you are running CICS/ESA 3.3 or above and this module is intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.

  - Option Three: The CXLU62 module is referenced by the CSTATIC list from a Natural parameter module that is not part of a Natural CICS nucleus, but is, nevertheless, loaded at Natural session initialization by the Natural keyword PARM.

    With this option, both the CXLU62 module and CICS exec interface stub modules must be linked with the Natural parameter module. Because Natural assumes the parameter module referenced in the PARM keyword is located at the beginning of the load module, the CSECTs that comprise the Natural parameter module must be located at the beginning of the resulting load module.

    Next, the module must be copied into one of the step libraries of the CICS nucleus and referenced by the appropriate PPT entry. The PPT entry must indicate that the module is resident. Additionally, if you are running CICS/ESA 3.3 or above and this module is intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.

- The linkage editor may flag a few unresolved external references when you link-edit a Natural parameter module that is designed to be loaded dynamically. This is unavoidable and will not cause any problems. If CICS refuses to load a module that has been marked as "not executable", specify either the linkage editor NCAL or the LET option.

3. Relink your Natural nucleus (or nuclei) to satisfy the requirements for NATPARMs, the interface module CXLU62, and the driver module NATTTY

Certain requirements must be satisfied when generating the Natural nucleus (or nuclei) to be used by the "end user" and "server programs" applications of Con-nect SNADS. The requirements for the "end user" application are less stringent than the requirements for the "server programs" application.

Following are the requirements for the Con-nect SNADS "end user" application:

- The environment must include a fully operational Con-nect system Version 3.4.3.
- The environment must contain a valid reference to a Con-nect spool file. The spool file identifier number is 223. This reference can be specified in the Natural parameter module with a NTLFILE macro call, or dynamically using the Natural keyword LFILE.

Note: Do not use Adabas passwords or ciphering for either the Con-nect spool file or the Con-nect system file.

- The Con-nect SNADS system programs must be loaded into the appropriate Natural system file.
- The Natural keyword CSIZE which specifies the size of the Con-nect buffer, must be assigned a value of 2 (this is the minimum value that can be assigned).

Note: The following requirements only apply if event-driven scheduling is used for the Con-nect SNADS inbound queue; that is, if the DS_ROUTER_DIRECTOR program is to be immediately invoked when a send request is submitted or when the DS RECEIVE program receives a SNADS message from another node within the network.

- The Natural nucleus must run in a CICS environment.
- The CSTATIC list from the Natural parameter module must specify the module name CXLU62.
- The CXLU62 module must be either linked to or accessible to the Natural nucleus.

- Following are the requirements for the Con-nect SNADS "server programs" application:

- All of the requirements for the "end user" application also apply to the "server programs" application.

- The Natural system keywords listed below must be set as follows:

Natural Keyword	Required Setting
WH	ON
PSEUDO	OFF
RELO	OFF
MADIO	0
MAXCL	0
AUTO	OFF
ETID (see note below)	Blank

Note: The setting ETID=' ' (blank) is recommended but not required. However, if more than one instance of a Con-nect SNADS server program becomes active at the same time, and more than one of those instances uses the same Natural user ID, Adabas will not issue an error message if the Natural keyword ETID is set to blank.

4. Modify the "queue server" front-end parameter module CONSNADS

Note: Always check that the CONSNADS parameters are valid. If not, you have to re-execute this and the following three steps.

The CONSNADS module is an Assembler program that uses the keyword macro CSNADS. It is linked to the four Con-nect SNADS "queue server" front-end programs in order to provide the information required to handle the Natural front-end logic, e.g. the name of the resident Natural CICS nucleus, and to specify the dynamic parameters that must be passed to the Natural sessions.

Natural parameters that were not included as NATPARMs for the nucleus can be specified dynamically in the CONSNADS parameter module.

- In the CONSNADS parameter module, you must specify:
  - the names used for each of the four Con-nect SNADS front-end programs. The supplied front-end programs are: CSINIT, CSRECV, CSSEND, and CSTERM;
  - the names used for each of the four Con-nect SNADS front-end programs. The supplied front-end programs are: CSINIT, CSRECV, CSSEND, and CSTERM;
  - the Natural nucleus used for the "server programs" application; and
  - the appropriate values for the CONSNADS keyword parameters that are used by the CSNADS macro (see the table below).
- Assemble the CONSNADS parameter module using the Con-nect SNADS source data set as a macro library.

Required CONSNADS Keyword Parameters

The following table lists the required CONSNADS keyword parameters and describes the value associated with each one:

Keyword	Value
DYNINIT	A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the "initialize queues" program. The parameter string must contain the Natural commands required to log on to the SYSCNT2 application, execute the command "X-FMRUN I" (if you are using the watchdog function, replace the command "X-FMRUN I" with "X-FWTCH %"), and terminate the Natural session.
	Example stack parameter: `STACK=(LOGON SYSCNT2;X-FMRUN I;FIN)`
	In addition, the DYNINIT parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.
DYNRECV	A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the DS_RECEIVE program. The parameter string must contain the Natural commands required to log on to the SYSCNT2 application, execute the command "X-FIX000", and terminate the Natural session.
	Example stack parameter: `STACK=(LOGON SYSCNT2;X-FIX000;FIN)`
	If the Natural keyword ETID is not assigned a value of blanks, the user identifier must be represented in the parameter string by a dollar sign character ($).
	Example stack parameter: `STACK=(LOGON SYSCNT2,$,PASSWORD;X-FIX000;FIN)`
	In addition, the DYNRECV parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.
DYNROUT	A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the DS_ROUTER_DIRECTOR program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FRP", and terminate the Natural session.
	If the Natural keyword ETID is not assigned a value of blanks, the user identifier must be represented in the parameter string by a dollar sign character ($).
	Example stack parameter: `STACK=(LOGON SYSCNT2,$,PASSWORD;X-FRP;FIN)`
	In addition, the DYNROUT parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.
DYNSEND	A dynamic parameter string in a Natural stack that is passed to the Natural nucleus in order to execute the DS_SEND program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FOX000 %", and terminate the Natural session. The percent sign (%) is substituted at execution time by the identifier that represents the queue to be processed.

Keyword	Value
	If the Natural keyword ETID is not assigned a value of blanks, the user identifier must be represented in the parameter string by a dollar sign character ($).
	Example stack parameter: `STACK=(LOGON SYSCNT2,$,PASSWORD;X-FOX000 %;FIN)`
	In addition, the DYNSEND parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.
DYNTERM	A dynamic parameter string in a Natural stack parameter that is passed to the Natural nucleus in order to execute the "terminate queues" program. The parameter string must contain the Natural commands required to log on to the application SYSCNT2, execute the command "X-FMRUN T", and terminate the Natural session.
	Example stack parameter: `STACK=(LOGON SYSCNT2;X-FMRUN T;FIN)`
	In addition, the DYNTERM parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.
INITPGM	The name of the front-end program, e.g. CSINIT, used to invoke the "initialize queues" function. Make a note of the value you assign as you will need it when defining the CICS PPT.
NPGRMID	The name of the resident Natural CICS nucleus previously defined in the CICS PPT. If the "OS Core" load option is used, this must not be the same name as the Natural CICS nucleus load module.
NTRANID	The name of the CICS transaction code previously assigned to the Natural CICS nucleus in the CICS PPT.
RECVPGM	The name of the front-end program, e.g. CSRECV, used to invoke the DS_RECEIVE program. Make a note of the value you assign as you will need it when defining the CICS PPT.
SENDPGM	The name of the front-end program, e.g. CSSEND, used to invoke the DS_SEND and DS_ROUTER_DIRECTOR programs. Make a note of the value you assign as you will need it when defining the CICS PPT.
TERMPGM	The name of the front-end program, e.g. CSTERM, used to invoke the "terminate queues" function. Make a note of the value you assign as you will need it when defining the CICS PPT.

Optional CONSNADS Keyword Parameters

The following table lists the optional CONSNADS keyword parameters and describes the value associated with each one.

Note: These parameters are available with Con-nect SNADS 1.5.3, but may be withdrawn in future versions.

Keyword	Value
RNTASKS	A value from 1 through 99 which represents the number of user identifiers available to Con-nect SNADS. This keyword applies only to the DS_RECEIVE program, and if the value of the Natural keyword ETID is not blank.
RTRYCNT	A value from 0 through 32767 which represents the maximum number of attempts that can be made to obtain an available user identifier. If the value of RTRYCNT is exceeded, the respective Con-nect SNADS server program will abend (code CS00).
	A value of 0 indicates that no attempts can be made. If no value is specified, 3 is the default.
	This keyword applies only to the DS_RECEIVE program and only if the value of the Natural keyword ETID is not blank.
RTRYDLY	A value from 1 through 32767 which represents the number of seconds between attempts to obtain an available user identifier.
	If no value is specified, 20 is the default.
	This keyword applies only if the value of the Natural keyword ETID is not blank.
RUIDPFX	A one-to-six character string used as the prefix for user identifiers. This value is concatenated with a two-digit number which has a value from 1 through 99 to form a user identifier.
	Note: The value specified for the RUIDPFX parameter must not be the same value specified for the SUIDPFX parameter.
	If no value is specified, the character string RSNADS is the default.
	This keyword applies only to the DS RECEIVE program and if the value of the Natural keyword ETID is blank.
SNTASKS	A value from 1 through 99 which represents the number of user identifiers available to Con-nect SNADS.
	This keyword applies only to the DS_SEND and DS_ROUTER_DIRECTOR programs, and if the value of the Natural keyword ETID is blank.
SUIDPFX	A one-to-six character string used as the prefix for user identifiers. This value is concatenated with a two-digit number which has a value from 1 through 99 to form a user identifier.
	Note: The value specified for the SUIDPFX parameter must not be the same value specified for the RUIDPFX parameter.
	If no value is specified, the character string SSNADS is the default. This keyword applies only to the DS_SEND and DS_ROUTER_DIRECTOR programs, and if the value of the Natural keyword ETID is not blank.

5. Specify keyword parameters (if required) to perform periodical supervision ("watchdog") of the Con-nect SNADS tasks

The watchdog program is generated to periodically supervise the status of the system and attempt to recover it automatically when necessary. When a temporary system failure is detected, the watchdog program issues a message to the operator console and records the event in the Con-nect SNADS log records. Then the watchdog program re-adjusts the status queues and re-starts the queue servers. However, the watchdog program cannot acquire inactive APPC connections.

The following table lists the required keyword parameters and describes the value associated with each one:

Keyword	Value
WATCH	A value of YES or NO. When the value is YES, the watchdog program is generated and replaces the Initialize Queues function. The default value is NO.
WTRANID	The CICS transaction code which is assigned to the CSINIT program.
WATCHIN	The length of time (in minutes) between subsequent executions of the watchdog program. The default value is 15.
WATCHRQ	A one-to-eight character string used by the watchdog. The watchdog program occupies one CICS request ID, which is specified here. The request ID specified must not conflict with any other CICS request ID.
	The default value is WATCH.

If the watchdog program is generated, apply the following modification to the CONSNADS keyword parameter DYNINIT:

Replace the command "X-FMRUN I" with "X-FWTCH %".

6. Assemble and link (if required) the Con-nect SNADS Adabas interface module

Use the default options PROLOG and EPILOG of the CICS command language translator DFHEAP1$.

The Con-nect SNADS Adabas interface module is a subroutine which permits Adabas OPEN (OP command) and CLOSE (CL commands) commands to be executed from within the front-end programs. The interface module is a member of the source data set CSDBAS.

It is not necessary to use the Adabas interface module unless your Con-nect SNADS front-end programs are to issue OP and CL commands. The functionality can be useful if the value of the Natural keyword ETID is set to blank.

If the Adabas interface module is entered via the CICS LINK command, CSDBAS must be assembled and link-edited as a standard CICS application program (i.e. with the CICS exec interface stub modules DFHEAI and DFHEAI0). CONSNADS must be included when link-editing CSDBAS. A PPT entry is required in this instance and must indicate that the module is resident. The name of the PPT must match the value specified as ADAPGM. If CICS/ESA 3.3 or above is installed and this module is intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.

If OP and CL commands are to be issued by the front-end programs and the Con-nect SNADS Adabas interface module is not entered via the CICS LINK command, CSDBAS must be link-edited with each of the front-end programs. In this case, the CICS exec interface stub modules and the PPT entry are not required.

The following table lists the required keywords and describes the value associated with each one:

Keyword	Value
ADADBID	Specify the database number which OP and CL commands are to be issued against. If files in multiple databases are to be opened and closed, the database numbers must be specified in subparameter notation. For example the OP and CL commands will be directed to database 2: `ADADBID=2`
	The OP and CL commands will be directed to databases 10 and 15: `ADADBID=(10,15)`
ADAUID	Specify whether the startup programs will generate an Adabas user ID when issuing OP commands. Usually the intention of using ETID=' ' (blank) is not to use Adabas user IDs. Therefore this keyword should not be specified.
	The default value is NO.
ADANAME	The name of the Adabas Link module in your CICS environment. The default value is Adabas.
ADALINK	Specify whether the Con-nect SNADS Adabas interface module is entered via the CICS LINK command (in which case the interface module will occupy a PPT entry of its own) or if it is link-edited with each of the startup programs.
	The default value is NO. Change the default value to YES if you want the module to be entered via the CICS LINK command.
ADAPGM	The name of the Con-nect SNADS Adabas interface module.
	The default value is CSDBAS.
ADAOPRB	Specify the record buffer contents for the OP commands. If this parameter has not been specified, the OP commands are not issued.
	If OP commands are to be directed to multiple databases, multiple record buffer contents must be specified in subparameter notation. The number of all Natural and Con-nect system files in use must be specified. For example the OP commands will be directed to database number 2 with the record buffer contents 'UPD=4,8,12.':

Keyword	Value
	`ADADBID=2, ADAOPRB='UPD=4,8,12.'`
	The OP commands will be directed to databases 5 and 15 with the record buffer contents 'UPD=3,6.' and 'ACC.'. OP commands will not be directed to database 10: `ADADBID=(5,10,15), ADAOPRB=('UPD=3,6.',,'ACC.')`
ADACLSE	Specify whether the startup programs will issue CL commands at the end of the session.
	The default value is NO.

7. Assemble and link-edit the "queue server" front-end programs

The names of the supplied front-end programs for the "queue server" functions are as follows:

Program	Queue Server Function
CSINIT	Initialize queues
CSRECV	DS_RECEIVE
CSSEND	DS_SEND and DS_ROUTER_DIRECTOR.
CSTERM	Terminate queues

The CSSEND and CSRECV front-end programs are required. Depending on how you have defined the DIU queues within the Con-nect Administration facility, the CSSEND and CSRECV programs are started either asynchronously at specified time intervals, or when a particular event occurs.

The CSINIT and CSTERM front-end programs are optional. CSINIT is invoked when CICS is initialized and CSTERM is invoked when CICS is terminated. If these front-end programs are not generated, queue server activity is activated when an event occurs or at specified time intervals, unless the Con-nect system administrator uses the Con-nect ADMIN program to terminate or initialize the queue server activity.

Note: The names of the load modules created in this step must correspond to the values specified for the INITPGM, RECVPGM, SENDPGM, and TERMPGM keywords of the CONSNADS parameter module.

- Use the CICS command language translator DFHEAP1$ with the default options PROLOG and EPILOG to process the queue server front-end programs.
- Assign the appropriate CICS source data set and the Con-nect SNADS source data set as macro libraries, and assemble the queue server front-end programs.
- Link-edit the queue server front-end programs with the CICS exec interface stub modules DFHEAI and DFHEAI0, and the CONSNADS parameter module created in the previous step. When link-editing the queue server front-end programs, place the CICS exec interface stub program DFHEAI first.

- Copy the queue server front-end programs into an appropriate step library of the CICS procedure.

8. Update the CICS PPT and PCT using CEDA (the CICS resource definition transaction)

- Define the queue server front-end programs in the CICS PPT and mark them as resident.

  The name you specify for the queue server front-end programs in the CICS PPT must be the same as the values specified in the INITPGM, RECVPGM, SENDPGM, and TERMPGM keywords of the CONSNADS parameter module. If CICS/ESA 3.3 or above is installed and these programs are intended to be used in 31-bit addressing mode, it is recommended that DATALOCATION=ANY is specified in the PPT entry.
- Assign a CICS transaction code to each of the queue server front-end programs in the CICS PCT. The size of the Transaction Work Area (TWA) must be a minimum of 256 bytes. The actual TWA space requirement is dependent on the length of the dynamic parameter string specified in the CONSNADS parameter module. In most cases, a TWA size of 512 bytes is adequate. If the TWA size is not adequate, the queue server front-end programs will abend (code CS01). If CICS/ESA 3.3 or above is installed and these programs are intended to be used in 31-bit addressing mode, it is recommended that TASKDATALOC=ANY is specified in the PCT entry.
- Assign an external transaction identifier (XTRANID), in the appropriate PCT entry for the front-end programs used with the DS_RECEIVE program and the DS_SEND and DS_ROUTER_DIRECTOR programs. The external transaction IDs that must be used are:
  - 21F0F0F1 for the queue server front-end program that invokes the DS SEND and DS_ROUTER_DIRECTOR programs;
  - 21F0F0F2 for the queue server front-end program that invokes the DS_RECEIVE program.

When the values 21F0F0F1 and 21F0F0F2 are entered, the CICS CEDA utility may issue a warning indicating that these transaction codes are reserved for IBM use. This message can be ignored.

9. Define local node for each Con-nect spool file serviced

Note: If you change the DBID or FNR of a Con-nect system file, you have to re-execute this and the following step.

Con-nect SNADS uses spool file method routines for transferring data from the Con-nect system file (for local node data) to the Con-nect spool file (for external node data) and vice versa. Thus, a local node for the spool file method must be defined.

- Select the System Maintenance function from the Con-nect Administration Menu.
- Invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen.

Caution: The following sub-step must be completed before you begin step 10.

- Select the Define Local Node function and, if you have not already done so, define a name for your spool file in the "Define Local Node" field and press ENTER. You are then prompted to define a local node for the transport service, although it is not required for Con-nect SNADS. Define a name in the "Define Local Node" field and press ENTER. For further information, see the *Con-nect Administration* documentation, section *External Mail Nodes*, sub-section *Define Local Node*.

By default, Con-nect SNADS uses the name you enter in the "Define Local Node" field for the spool file method as the Distribution Group Name (DGN) when mapping local Con-nect user identifiers to the SNADS Distribution User Names (DUNs).

10. Define control information for nodes (system files) - SNADS Initialization

You must define the control information for all SNADS distribution service units (DSUs) that will be serviced from the current Con-nect spool file as local SNADS nodes.

To define Con-nect system files as nodes in the SNADS network, perform the following steps:

- Select the System Maintenance function from the Con-nect Administration - Main Menu.
- Invoke the External Mail Nodes function to display the "Administration - External Mail Nodes" screen.
- Select the Maintain Mail Nodes function and mark "F Snads" in the resulting Maintain Mail Node window.
- Select the Initialization function from the SNADS Administration Menu to display the "Control Maintenance" screen.
- On the this screen, specify the DSUN (RGN and REN) of the Con-nect system file. In this context, a "Con-nect system" is the system file.

For more information about the "Control Maintenance" screen refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Control Maintenance*.

11. Create one or multiple outbound queues for the storing of Distribution Interchange Units (DIUs) intended for other nodes

Note: This and the following three steps define the path messages take from Con-nect to other nodes. If changes are made in your SNA network, you may have to re-execute this step.

At least one outbound queue must be created for each adjacent node in the SNADS network. When an outbound queue is created, the attributes of the LU6.2 link between CICS and the adjacent SNADS node, e.g. a Soft-Switch node, must be defined. To create an outbound queue, perform the following steps:

- Select the Queue Maintenance function from the SNADS Administration Menu. The "Queue Maintenance" screen is displayed. Press PF4 to add a queue. The "Queue Info" screen is displayed.
- In the this screen, enter the following:
  - In the "Queue ID" field, enter a one-to-eight character name to be used as the queue ID.
  - In the "Queue ID" field, enter a one-to-eight character name to be used as the queue ID.
  - In the "Connection ID" field, enter the connection name used by the CICS system programmer to define the adjacent node to CICS.
  - Enter the log mode name that the CICS system programmer designated to be used with the connection specified above.
  - Press PF5 (Confirm) to add the new outbound queue.

For further information about the "Queue Info" screen, refer to the *Con-nect Administration* documentation, section *Configuring Con-nect SNADS*, sub-section *Outbound Queues*.

12. Set the "Reset" and "Input" statuses for the outbound queue

When you initially test Con-nect SNADS, the input status of the outbound queue must be set to active, and the reset status to inactive. To set the input and reset statuses of the outbound queue, perform the following steps:

- Select the Queue Maintenance function from the "SNADS Administration Menu" screen to display the "Queue Maintenance" screen.
- Enter MO (modify) in the "Cmd" column for the outbound queue and press ENTER. The "Queue Info" screen is displayed. Modify the following fields:
  - In the "Reset Status" field, enter I (inactive).
  - In the "Input Status" field, enter A (active).
  - Mark the "Reset" field with any character and press PF5.

For more information about the "Queue Maintenance" screen refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS* , sub-section *Queue Maintenance*.

13. Define routing specifications for all other nodes with which you intend to communicate

You must define a routing entry for each node in the network with which you wish to communicate. The routing entry determines how the SNADS DIUs are to be routed to a particular node.

To define routing specifications, perform the following steps:

- Select the Routing Entry Maintenance function from the SNADS Administration Menu to display the "Routing Entry Maintenance" screen. Press PF4 to add a routing entry. The following must be specified in the "Routing Entry" screen:
  - In the "Recipient Node" field, enter the DSUN (RGN.REN) for each node.

- In the "Next Queue" field, enter the name of an outbound queue defined in step 12.
- Press PF5 (Confirm) to add the new routing entry.

For more information about the "Routing Entry" screen, refer to the *Con-nect Administration* documentation, section *Maintaining Con-nect SNADS*, sub-section *Routing Entries*.

14. Add a mail node, type F (SNADS), for sending purposes

You must define an external mail node in Con-nect for sending mail.

To define an external mail node, perform the following steps:

- Select the System Maintenance function on the Con-nect Administration - Main Menu to display the "Administration - System Maintenance" screen.
- Select the External Mail Nodes function and press ENTER to display the "Administration - External Mail Nodes" screen.
- Specify the node name in the "Mail Node/Type" field (this node name will be used on the "Addressee" field in the Con-nect "Send" screen), select the Add Mail Node function and press ENTER.
- Enter F (SNADS) in the "Node Type" field.
- Enter the RGN (Group) and REN (Element) for each external SNADS node (i.e. addressee) to form the DSUN (Node) portion of the SNADS address.

When an object (document or memo) is sent to this addressee, the value you entered for the DSUN (RGN and REN) is automatically provided by the system. The user must supply the DUN (DGN and DEN) for a specific user in order to complete the SNADS address.

15. Test the installation of Con-nect SNADS

When testing, use the following settings for ALL queues:

Input Status: A

Output Status: I

Reset Status: I

This guarantees that the queue servers will be activated only when they are started from the "Queue Maintenance" screen.

If you try to start a queue server when either the outbound or inbound queue is in the hold status, the queue server task will immediately exit without processing any information.

If you are unsuccessful in sending messages between nodes with the Con-nect queues set to either "event" or "timer" status, use the following procedure to manually step through the process of sending and receiving a Distribution Interchange Unit (DIU):

- Initialize the node. A routing entry, an inbound queue, and one outbound queue must be defined to the node.
- Send a mail item from Con-nect. Then, check the inbound queue to see if the DIU is there.
- Start the inbound queue from the "Queue Maintenance" screen. This activates the router, which should route the DIU to the outbound queue. Since the outbound queue is inactive, the DIU should remain in the outbound queue.
- Check to see if the DIU was routed to the outbound queue.
- Check the log records. The log records can be accessed using the Log Information Maintenance function in the "SNADS Administration Menu" screen. A list of log records written by the SNADS driver programs are displayed. If the router was executed, log records indicating successful or unsuccessful routing are displayed. The exact format of these records is not available; however, you can verify whether or not tasks have been executed. Typically, a message, e.g. "routing error", which corresponds to a particular log record indicates that an error has occurred.

Up to this point, LU6.2 has not been involved. The following steps allow you to test sending a DIU to another SNADS node.

- Set the outbound queue with the routed DIU to "inactive" status; then start the outbound queue from the "Queue Maintenance" screen.
- Check the log records. The log records are accessed from the Log Information Maintenance function in the "SNADS Administration Menu" screen. If the DIU was successfully sent, records from the DS_SEND program (X-FON000) are displayed.
- Check for log records which were created when a status request was received from the target node. Records from the DS_RECEIVE program (X-FIN000) should be displayed.
- Check to see whether the DIU is delivered to the recipient node (the Inbasket, if Con-nect is the recipient node). In addition, there should be a "status" DIU generated and routed for delivery to the sender at the origin node.
- If you are still experiencing problems, the following commands and functions can be useful for debugging purposes:
    - CICS Supplied Transactions CEMT, CEDC and CEDF

      For example, use the `CEMT I CONN` command to verify whether or not an LU6.2 session has been acquired (where "conn" is the connection ID). Use the `CEDC VIEW CONNECTION(conn) GROUP(isc)` command to examine the characteristics of an LU6.2 connection. Use the `CEDF` command to display the online trace utility.
    - Con-nect SNADS Administration Functions

      The following functions are offered in the Con-nect SNADS Administration Menu:

      Queue Maintenance - Provides a DIU count, time stamps for recent activity, and the ability to manually start the queue server.

Control Maintenance - Provides the correlation between a DSUN and a system file (DBID and FNR).

Messages Awaiting Confirmation of Delivery - Displays the number of undelivered DIUs sent from Con-nect.

Log Information Maintenance - Provides debugging information that can be helpful to Software AG in resolving any problems you are experiencing with Con-nect SNADS.

19 Overcoming Common Difficulties When Installing Con-nect

SNADS Under CICS

This chapter is designed to assist the system administrator to verify that Con-nect SNADS has been installed properly.

Although the following procedures are not the only means of verification and problem analysis, the intent here is to provide the administrator with some guide lines that will insure that the system is operational.

Verifying the Installation of the Asynchronous Server Programs

The asynchronous server programs are the Con-nect SNADS implementation of DS_ROUTER_DIRECTOR, DS_SEND and DS_RECEIVE. Although any of the three programs may not run for a common reason, the procedures for installation, verification and problem analysis vary from program to program. See the *Con-nect Administration* documentation, section *Con-nect SNADS*, sub-section *Implementation of Con-nect SNADS* for a brief description of these programs.

Test Set-up for DS_ROUTER_DIRECTOR and DS_SEND

Before testing DS_ROUTER_DIRECTOR and/or DS_SEND, set all Con-nect SNADS queues to manual operation as described in ***Installation Steps for Com-plete***, ***Installation Steps for Batch Mode*** , or ***Installation Steps for CICS***. For example, set the Reset Status for inbound and all outbound queues to I (inactive) and change the status to output by marking the "Reset Status" field in the "Queue Info" screen (do not use the timer driven scheduling (T) unless you are certain that the system is completely operational). Verify that the input status is active.

Using the Con-nect Send function, send a message to a SNADS addressee. The status message "Waiting for Distribution" appears in your Outbasket along with the list of addressees. The actual message sent appears as an item in the Con-nect SNADS inbound queue. You can begin testing at this point.

Note: If testing must be interrupted, re-establish the inactive status for inbound and outbound queues.

Standard Testing Techniques for DS_ROUTER_DIRECTOR

Initialize the inbound queue (using the DS_ROUTER_DIRECTOR program) from the "Queue Maintenance" screen. Observe the number of items in the queue and the changes in the output status associated with this queue by occasionally pressing ENTER.

If the system is running properly, the following sequence of events occur:

1. The output status of the queue changes from I (inactive) to "scheduled" (indicated by a blank). When the status changes to "scheduled", the end-user task issues a CICS START command which invokes the DS_ROUTER_DIRECTOR program but does not activate it.

2. The output status of the queue changes from " " (scheduled) to A (active). This status change is made by the DS_ROUTER_DIRECTOR program when executed.
3. The number of items in the queue is decremented to 0. If the items in the queue are addressed to multiple addressees, the queue items appear as members of the outbound queues. The items do not appear in the outbound queues if the only specified SNADS addressee is a local Con-nect user or if the specified addressee DSUN could not be matched to a SNADS routing entry.
4. The output status of the queue changes back to I when the DS_ROUTER_DIRECTOR completes its execution.

A sequence of events which does not follow the above indicates a problem with the installation of Con-nect SNADS, usually resulting in one of two events:

1. The queue output status changes to H (held) which indicates that Con-nect SNADS detected a fatal error (e.g. a program check during execution). If this occurs, check the Con-nect SNADS log records for further information.

   If the problem is a Natural error condition, it appears in the log records as a message that originates from the X-FE program. This message contains the Natural error number, line number and the name of the affected Natural program which runs in conjunction with Con-nect SNADS.
2. The queue output status changes from I (inactive) to " " (scheduled) but the system is "hung". A hung system indicates that an error occurred before the DS_ROUTER_DIRECTOR program was initialized properly. See ***Problem Analysis When the DS_ROUTER_DIRECTOR is "hung" in the Scheduled Status*** for further information.

Note: Always reset the queue output status to I (inactive) before continuing the test procedures.

Problem Analysis When the DS_ROUTER_DIRECTOR is "hung" in the Scheduled Status

If the output status queue changed from I (inactive) to " " (scheduled) but the system is "hung" in the Scheduled status, find the CICS transaction code which is assigned to the Con-nect SNADS CSSEND program. That transaction code in this section will be referred to as the parameter *tran*.

Execute the DS_ROUTER_DIRECTOR program by using the CICS CECI utility and issue the following command:

```
CECI START TR(tran) FR(_ _ _ _ _ _ _ _)
```

Make sure that eight underscores are entered in the FR parameter and press ENTER.

The three most common results when using the CECI utility are:

1. The installation problem is obvious.
2. The program seems to be working properly. If so, examine the PCT entry by which you defined the parameter "tran". Con-nect SNADS does not internally use "tran" but rather the external

transaction ID, X'21F0F0F1'. Verify that 21F0F0F1 is specified as an external transaction ID in that PCT entry and that it does not appear as an external transaction ID in other PCT entries.

3. The DS_ROUTER_DIRECTOR is still hung in the Scheduled status. Using the CICS terminal ID (taken from the EIB display in CECI) assign your terminal to the transaction in question (terminal ID in this section is referred to as the parameter "term") and issue the following command:

```
CECI START TR(tran) TE(term) FR(_ _ _ _ _ _ _ _)
```

One of the following occurs:

- The installation problem is obvious.
- The program seems to work but screen I/Os are displayed (for example, a Natural Security mailbox). Asynchronous Natural tasks should never attempt screen I/Os. When screen I/Os are requested in such a task, the Natural CICS driver forces that task to abend with NT06 and a message is directed to the operator console. If this happens, change the system set up so that screen I/Os cannot occur and restart the test from the beginning.
- The program seems to work and screen I/Os are not apparent except for the Natural code NAT9995. Check for device type dependencies. Make sure NATTTY was added in the Natural nucleus. If that is not the problem, verify that the Natural Security logon procedure or user exit which is based on terminal IDs or types are correct (when asynchronous Natural tasks are running under CICS the value of "*DEVICE" is set to ASYNC).
- The DS_ROUTER_DIRECTOR program is still hung. Use the CICS CEDF utility as a debugging aid. To use the CICS utility, start the DS_ROUTER_DIRECTOR program by issuing the following command:

  ```
  CECI START TR(tran) TE(term) FR(_ _ _ _ _ _ _ _) I(30)
  ```

Before the program begins, the CICS Interval Control pauses thirty seconds, and then executes the transaction. During the pause issue the following command:

```
CEDF
```

If the DS_ROUTER_DIRECTOR program runs successfully, the number of items in the queue is decremented to 0. If the items in the queue are addressed to multiple addressees the queue items appear as members of the outbound queues. The items do not appear in the outbound queues if the only specified SNADS addressee is a local Con-nect user or if the specified addressee DSUN could not be matched to a SNADS routing entry.

Testing Techniques for DS_SEND

Testing techniques for DS_SEND are similar to the testing procedures for the DS_ROUTER_DIRECTOR program. However, before you begin there are several important steps that you must observe.

Verify that the APPC connection is operational and is properly defined to Con-nect SNADS. If the connection is operational, the CICS CEMT utility indicates that the connection is in the acquired state and the log mode is defined to CICS. The "Available Session Count" must be greater than 0 and SNASVCMG must not be used as a log mode.

The logical link to the CICS connection and log mode definitions must be set up in the appropriate outbound queue definition. Be aware that the Connection ID in the "Queue Info" screen must not be specified as an LU name, but as the four-character CICS name assigned to that connection. The log mode must be specified exactly as it is defined to the network or left blank (if you do not care which one is used).

When issuing the CICS START command, specify the Con-nect SNADS queue ID in the FR parameter (which must be eight characters long, using trailing blanks if necessary).

If "queue-id" is the Con-nect SNADS queue ID, the CECI commands should be issued in the order below:

```
CECI START TR(tran) FR(queue-id)
```

```
CECI START TR(tran) TE (term) FR(queue-id)
```

```
CECI START TR(tran) TE (term) FR(queue-id) I(30)
```

If you are using the CICS CEDF utility as a diagnostic aid, you can trap the CICS LU6.2 commands They are useful in pointing out the installation problem(s). If you do trap the commands, keep in mind that they all begin with EXEC CICS GDS (at least for the unmapped conversation as it is used for SNADS).

An additional reason why the output status changes to H (held) when testing DS_SEND is that the neighboring SNADS node is rejecting Con-nect messages. If that is the case, they could be syntactically or semantically incorrect. An indication of this problem are log records which originate from the X-FOX-N8 program. In this case, see ***Neighboring Node Does Not Accept Messages***.

Test Set-up for DS_RECEIVE

Begin by setting the Con-nect SNADS inbound queue to manual operation (i.e. to status I for inactive). If a SNADS message from Con-nect is successfully transmitted to another node in the network, a SNADS status message appears.

As an alternative test, log on to one of the other SNADS nodes in the network and send a message to an addressee residing at the local Con-nect node. If this message appears in your inbound queue within a reasonable amount of time (about five or ten minutes), proceed with ***Checking For Inconsistent SNADS Definitions***.

Testing Technique for DS_RECEIVE

If messages sent from another SNADS node (within the SNADS environment) do not appear in your inbound queue, perform the following steps:

1. Verify that the connections to the neighboring nodes are operational and are in the acquired state.
2. If they are, verify that the usage count of the Con-nect SNADS CSRECV program is incremented (an indication that a neighboring SNADS node is attempting to invoke the DS_RECEIVE program (CSRECV) to transmit messages to Con-nect). If the parameter "csrecv" is the PPT name of your local CSRECV program, you can observe the usage count when using the CICS CEMT utility and issuing the command:

   ```
   CEMT I PROG(csrecv)
   ```

   If the usage count is not incremented, check that X'21F0F0F2' is specified as an external transaction ID in the PCT entry which is associated with CSRECV. If it is, check that a duplicate 21F0F0F2 does not appear as an external transaction ID in other PCT entries. If you have verified all of this, then the problem is with the other SNADS nodes. Verify that the APPC connections are properly set up and that the routing tables are set up correctly. The remote system requires definitions not only of local but also of remote users in order to communicate with them. Check the set up of the remote systems.
3. If CSRECV is invoked but the items do not appear in the inbound queue, look at the Con-nect SNADS log records. The log records display information relating to Natural error conditions (program names beginning with X-FE) or they indicate that the DS_RECEIVE program has become active (log records which display program names beginning with X-FIX).
4. If CSRECV is invoked but fails before becoming operational and you are unable to figure out why, use CEDF for further testing.

When testing with CEDF, it is helpful to know that each possible APPC session is associated to its own TCT terminal entry (CICS generates the TCT entries automatically). Typically, the name of a TCT entry begins with a minus sign (-) e.g. "-99F".

If you know the TCT entry name of the APPC session you are using to transmit a SNADS message to Con-nect, trapping the respective CSRECV execution is possible when issuing the following command:

```
CEDF -99F (or corresponding ID)
```

If you do not know the TCT entry name and you are using single session connections (usually not the case), let "lu-name" be the LU name of the neighboring node by issuing the command:

```
CEMT I NET(lu-name)
```

The result is the display of one TCT terminal entry. Use the ID from this entry.

If you are using parallel session connections, issue the command:

```
CEMT I NET(lu-name)
```

The result is a list of TCT entries. Try each of the TCT entries until you "catch" the conversation with CEDF.

If CSRECV abends after the initialization phase, check the log records resulting from programs beginning with X-FIX. The terminal ID (called "conversation ID" here) is listed in these log records.

Checking For Inconsistent SNADS Definitions

When a SNADS message from a remote sender is received on the local Con-nect SNADS inbound queue, insure that the SNADS definitions match on both sides.

To verify the definitions, check the queue item display and look at the originator's SNADS node and user ID. If there is a status message, the user ID may be blank. In this case, try with a real message (i.e. DSUN and DUN). Verify that the names are the same you assumed when setting up the local system.

Additionally, look at the addressees queue item display. Verify that the names match with your prior assumptions. If you are not clear about this, use the Con-nect Send function and send a message to a SNADS addressee or a SNADS node. The window which appears shows your and the addressee's SNADS node and user IDs. If necessary, change the definitions for consistent naming.

If the naming is consistent, start the DS_ROUTER_DIRECTOR program and verify that the message appears in the appropriate Con-nect Inbasket or, if it is a status report, check the Con-nect Outbasket. If the message cannot be found, consult the Con-nect SNADS control information and verify that the DS_ROUTER_DIRECTOR program marked the appropriate Con-nect system as accessible.

Furthermore, verify that the database ID and file number of the Con-nect system file which the DS_ROUTER_DIRECTOR program entered during its last execution are correct. Basically, there are two reasons for problems in this area.

1. The name of the Con-nect system entered in the Con-nect SNADS control information does not match the name which is specified in the Initialize Node File function. For further information, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance* .
2. The contents of the Con-nect spool file are not consistent. For further information, see ***Inconsistency in the Con-nect Spool File***.

Note: If the Con-nect timezones are not set up properly, incoming messages may remain "invisible" for some time after delivery in the Con-nect Inbasket.

Neighboring Node Does Not Accept Messages

If upon execution of DS_SEND, the output status of the respective outbound queue changes to H (held), check for one of two problems:

1. There may be a communication problem; check the connection status to the neighboring system.
2. The neighboring system may find the message is syntactically or semantically incorrect. An indication of this situation are log records that originate from a program with the name X-FOX-N8.

If the SNADS messages from a Con-nect node are incorrect, the problem is one of the following:

- The Origin Node DSUN and/or the Origin Name DUN are not built according to SNADS conventions. In particular, the only component of DSUN and DUN that can be blank is the group entry of the DSUN (RGN). Check the Origin Node DSUN and the Origin Name DUN by looking at the "Data Interchange Unit" screen. If the DSUN is wrong, then the SNADS control information is defined incorrectly. If the first part of the SNADS user ID (DUN, the first-group-component of it is called DGN) is blank, then the contents of the Con-nect spool file are probably inconsistent.
- The definitions that are set up for Con-nect in the neighboring system are incorrect.

Inconsistency in the Con-nect Spool File

The Home Node Record in the Con-nect Spool File

For each Con-nect system file that is to be serviced by one Con-nect spool file, there is one so called "Home Node Record" stored in the spool file. That record is used for all external mailing methods of Con-nect, not just Con-nect SNADS. The Home Node Record is a logical link between the spool file and the system files. The Home Node Record contains two major pieces of information, both of which are defined as descriptors: the name of the respective Con-nect system (up to eight characters in length), and the database ID and file number of the system file.

Con-nect SNADS uses the Home Node Record for two purposes:

1. During execution of the Send function, Con-nect SNADS determines the database ID and file number of the Con-nect system file. The database ID and file number are then used to retrieve the name of the respective system from the Home Node Record. This name is used as the default DGN when building the SNADS DUN of the message originator. The name is also used as a key to locate the appropriate SNADS control information entry when building the SNADS DSUN of the message originator.
2. The Home Node Record is also used when an incoming SNADS message is directed to the local Con-nect system. Con-nect SNADS determines the name of the local system using the SNADS recipient DSUN and the information kept in the Con-nect SNADS control information. Then for a physical transfer of the message to that system, Con-nect SNADS consults the Home Node Record to retrieve the appropriate database ID and file number.

Creating a Home Node Record

A Home Node Record is created when executing the Initialize Node File function in the Con-nect ADMIN program. For further information, see the *Con-nect Administration* documentation, section *Con-nect System Maintenance* .

Common Inconsistencies and their Consequences

Inconsistencies may occur within the Con-nect Home Node Record. Below are some of the most common inconsistencies.

As in the Con-nect system described in scenario number one (outgoing SNADS messages) of the previous section, ***The Home Node Record in the Con-nect Spool File***, there are two common problems:

- The Home Node Record is missing altogether. This occurs when the Initialize Node File function is not executed.

- The Home Node Record is obscured by other database records that have the same "number" key value. Normally all "noise" records have a name descriptor which is similar to the real Home Node Record. (They are in effect partial copies of the Home Node Record.) This "normal" case, however, persists only if the Initialize Node File functions are executed for each serviced system file immediately after you create your Con-nect spool file and before any other Administration Functions. What usually happens in this case is that Con-nect SNADS builds syntactically invalid SNADS DSUNs (node IDs) and DUNs (user IDs) Con-nect message originators. Check the SNADS DSUN and DUN which Con-nect assigns to you when issuing the Send function with a SNADS address (it is displayed in the window which prompts for the completion of the addressee information). Check the outgoing messages in the Con-nect SNADS queues. A neighboring system usually rejects such messages. Be sure that you have set up the SNADS control information properly, invalid DSUNs may have been specified.

As in the Con-nect system described in scenario number two (incoming SNADS messages) of the previous section, ***The Home Node Record in the Con-nect Spool File***, there are five common inconsistencies.

- The Home Node Record is missing altogether. This occurs when the Initialize Node File function is not executed.
- The name descriptor of the Home Node Record is corrupt. In this case, the system behaves as if the Home Node Record is missing.
- The Home Node Record exists, but the database ID and file number specified are incorrect. Verify that the database ID and file number of your Con-nect system file was changed after executing the Initialize Node File function.
- The Home Node Record is not unique. For example, more than one Home Node Record exists with the same name descriptor. The same name may have been used while executing the Initialize Node File function from different Con-nect systems, referencing the same spool file. Or, an attempt was made to fix the inconsistency resulting from changing the database ID and file number of the Con-nect system file by implementing another Initialize Node File function.
- If any of the above inconsistencies occur, the incoming SNADS messages and their status will not transfer to the appropriate Con-nect system. Check the Con-nect SNADS control information. The DS_ROUTER_DIRECTOR program records the database ID and file number of each defined local Con-nect system as they are found in the respective Home Node Records at the beginning of each run.

Making the Spool File Consistent

If you find that the contents of your spool file are inconsistent, try to solve the problem by executing the Initialize Node File function.

If the problem persists, the only recommended solution is to scratch the spool file and create a new one (execute the Initialize Node File function immediately after creating the spool file). Not only Con-nect SNADS, but all other external mailing methods will have trouble with your old, corrupted spool file. Keep in mind that you have to re-enter all Con-nect SNADS system definitions (not the SNADS nodes and addressees) as they are defined in the Con-nect spool file.