

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 (referred to in this section as the LU6.2 API).

It covers the following topics:

- Overview
- Installation Steps for Com-plete

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 dataset CSL nnn .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 3.4.1.
 - 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 cipherng 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 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
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 NATPARAMs 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 dataset 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:

Keyword	Value
DYNINIT	<p>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:</p> <pre>STACK=(LOGON SYSCNT2;X-FMRUN I;FIN)</pre> <p>Example program parameter:</p> <pre>PROGRAM=cpend</pre> <p>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.</p>
DYNROUT	<p>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:</p> <pre>STACK=(LOGON SYSCNT2;X-FRPP;FIN)</pre> <p>Example program parameter:</p> <pre>PROGRAM=cpend</pre> <p>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.</p>
DYNSEND	<p>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: <pre>STACK=(LOGON SYSCNT2;X-FOP000 % I;FIN)</pre> <p>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.</p> </p>

Keyword	Value
DYMDEM	<p>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:</p> <pre>STACK=(LOGON SYSCNT2;X-FDEM ;FIN)</pre> <p>Example program parameter:</p> <pre>PROGRAM=cpend</pre> <p>where "cpend" is the name of the load module which was created from the source program CPEND. In addition, the DYMDEM parameter string must satisfy the NATPARM requirements for the "server programs" application, e.g. WH=ON.</p>
DYNTERM	<p>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:</p> <pre>STACK=(LOGON SYSCNT2;X-FMRUN T;FIN)</pre> <p>Example program parameter:</p> <pre>PROGRAM=cpend</pre> <p>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.</p>
NPGRMID	<p>The name of the non-resident portion of the Natural Com-plete driver as it has been cataloged in the Com-plete library.</p>

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	<p>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:</p> <p>ADADBID=2</p> <p>OP and CL commands will be directed to database 2.</p> <p>ADADBID=(10,15)</p> <p>OP and CL commands will be directed to databases 10 and 15.</p>
ADAUID	<p>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.</p>
ADALINK	<p>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.</p>
ADAPGM	<p>The name of the Con-nect SNADS Adabas interface module. The default value is CPDBAS.</p>
ADAOPRB	<p>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:</p> <p>ADADBID=2, ADAOPRB='UPD=4,8,12.'</p> <p>In this case, an OP command will be directed to database number 2 with the record buffer contents 'UPD=4,8,12.'</p> <p>ADADBID=(5,10,15), ADAOPRB=('UPD=3,6.', , 'ACC.')</p> <p>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.</p>
ADACLSE	<p>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.</p>

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.



Warning:

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.