

# 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.

This chapter contains the following topics:

- Overview
  - Installation Procedures for Com-plete
-

## 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 dataset 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 dataset 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 steps 2 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 Connect 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.

**Note:**

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 NATPARAMs 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 CTCPSST;
- 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 DYNDEM 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. Tying 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	<p>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.</p> <p>Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNROUT:</p> <pre>STACK=(LOGON SYSCNT2;YR;F),PROGRAM=ctend</pre>
DYNSEND		X	<p>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.</p> <p><b>Example:</b> If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNSEND:</p> <pre>STACK=(LOGON SYSCNT2;YOX0000 %;F),PROGRAM=ctend</pre>
DYNDEM N		X	<p>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.</p> <p><b>Example:</b> If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNDEM N:</p> <pre>STACK=(LOGON SYSCNT2;YIX8DEM %;F),PROGRAM=ctend</pre>

Keyword	M	O	Description
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:  <code>STACK=(LOGON SYSCNT2;YFWTCH %;F),PROGRAM=ctend</code>
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:  <code>STACK=(LOGON SYSCNT2;YFSTOP;F),PROGRAM=ctend</code>
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):  <code>STACK=(LOGON \$;&amp;&amp; /,%;F),PROGRAM=ctend</code>  <b>Note:</b> 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 CTCPSST).

<b>Keyword</b>	<b>M</b>	<b>O</b>	<b>Description</b>
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:

<b>Keyword</b>	<b>Value</b>
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 datasets 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 dataset 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.
			<p>Example: OP and CL commands are directed to database 2 with the following:</p> <pre>ADADBID=2</pre> <p>OP and CL commands are directed to databases 10 and 15 with the following:</p> <pre>ADADBID=(10,15)</pre>
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.
			<p>Example: An OP command is directed to database number 2 with the record buffer contents 'UPD=4,8,12.' with the following:</p> <pre>ADADBID=2,ADAOPRB='UPD=4,8,12.'</pre> <p>However, an OP command is directed to databases 5 and 15 with the record buffer contents 'UPD=3.6.' and 'ACC.' with the following:</p> <pre>ADADBID=(5,10,15),ADAOPRB=('UPD=3,6.', , 'ACC.')</pre> <p>OP commands are not directed to database 10.</p>
ADACLSE		X	Specify whether the back-end (session termination) program CTCPTTE will issue CL commands at the end of a session. Default value: No

## 5. Assembling the CTPARMZ parameter module using the transport service source dataset 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 dataset and the transport service source dataset 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 CTCADB, if required, with the CTPARMZ parameter module.

### Note:

If OP and CL commands are to be issued, the transport service Adabas interface module CTCADB 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, DYNDEM, 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 dataset 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 LU6.2 protocol.
TPNAME=	<p>The TPN which identifies the corresponding receiving programs at the adjacent node.</p> <ul style="list-style-type: none"> <li>● If the adjacent node uses EntireX Broker LU6.2 ACI in Complete, this is the name of the general startup routine.</li> <li>● If the adjacent node uses EntireX Broker LU6.2 ACI in batch mode, this is the unique name used to identify the server.</li> <li>● 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).</li> </ul>
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.
SYNLEVEL=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 *Connect 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 *Connect 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 dataset 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.
SYNLEVEL=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 *Connect 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 Connect), you can test the installation of the transport service.

The following test uses Connect 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 Connect 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.