

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.

This chapter contains the following topics:

- Overview
 - Installation Procedures for CICS
-

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 dataset. 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 datasets 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 cipherng 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.
- 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 NATPARAMs 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 DYNDEM 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. 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.

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.
			<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	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>Example: 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>
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.
			<p>Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNRECV:</p> <pre>STACK=(LOGON SYSCNT2;YIX9000;F),PROGRAM=ctend</pre>

Keyword	M	O	Description
DYNDEM		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.
			<p>Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNDEM:</p> <pre>STACK=(LOGON SYSCNT2;YIX7000 %;F),PROGRAM=ctend</pre> <p>Note: DYNDEM is to be specified only if remote database access is used.</p>
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.
			<p>Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNWTCH:</p> <pre>STACK=(LOGON SYSCNT2;YFWTCH %;F),PROGRAM=ctend</pre>
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.
			<p>Example: If "ctend" is the name of the transport service session termination program, the following parameters could be contained in DYNTERM:</p> <pre>STACK=(LOGON SYSCNT2;YFSTOP;F),PROGRAM=ctend</pre>

Keyword	M	O	Description
DYNAPPL		X	<p>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:</p> <p>\$ Natural Application Name & Natural Program Name (specify '&&' for one '&') / transport service Node Name % transport service Application Name</p> <p>Example: If "ctend" is the name of the transport service session termination program, DYNAPPL may contain the following parameters (among others):</p> <pre>STACK=(LOGON \$;&& /,%;F),PROGRAM=ctend</pre> <p>Note: The above example is suitable for a Con-nect external mail application</p>
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 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-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.
			<p>Example: OP and CL commands will be directed to database 2 with the following:</p> <pre>ADADBID=2</pre> <p>OP and CL commands will be directed to databases 10 and 15 with the following:</p> <pre>ADADBID=(10,15)</pre> <p>In this example, OP and CL commands will be directed to databases 10 and 15:</p> <pre>ADADBID=(5,10,15), ADAOPRB=(UPD=3,6.,,ACC.)</pre>
ADAOPRB			Specify the record buffer contents for the OP command. If this parameter has not been specified, OP commands will not be issued.
			<p>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> <p>Example: An OP command will be 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 will be 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 will not be directed to database 10.</p>
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 dataset 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 dataset and the transport service source dataset 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, DYNDEM, 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.

