VisualAge Pacbase

Middleware User's Guide

Version 3.5

Note

Before using this document, read the general information under "Notices" after the Table of Contents.

According to your licence agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Support Center at:
http://www-1.ibm.com/support/docview.wss?rs=37&context=SSEP67&uid=swg27005477

Consult the Catalog section in the Documentation home page to make sure you have the most recent edition of this document.

Third Edition (Februray 2008)

This edition applies to the following licensed programs:

- VisualAge Pacbase Versions 3..5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at: http://www.ibm.com/software/awdtools/vapacbase/support.html
or to the following postal address:
IBM France Software Laboratory, Rational Division
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.
When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Table of Contents

Notices v

Chapter 1: Introduction 7

Chapter 2: The different communication modes 9

General architecture 9

Communication in direct mode 9

Communication via a gateway 10
Particular case: relay mode 11

Chapter 3: The middleware components 13

Common files 13

Technical interfaces of communication 13

VAP Gateway module 14

VAP Relay module 14

VAP Listener module 14

Chapter 4: Available protocols by server platform 15

OS/390 - CICS 16

OS/390 - IMS 16

DOS/VSE – CICS 16

AIX 16

Solaris 16

Windows 16

AS400 16

Compaq OpenVMS 17

Compaq Tru64 UNIX 17

HP-UX 17

Linux 17

NUMA-Q DYNX (SEQUENT) 17

Host Tandem OSS-Guardian (Himalaya) 17

Host GCOS7 - TDS 17

Host GCOS8 - TP8 17

Chapter 5: Notion of Location 19

Chapter 6: How to use VapGateway 21

Start-up 21

The VapGateway version identifier 22

Chapter 7: How to use vaprelay.jar 25

Chapter 8: How to use VA Pac Listener 27

How to Launch the Listener 27

How to Use the Listener on the UNIX platform 28

Chapter 9: Protocols Description & Configuration 31

IMS Connect 31
Prerequisites 31
Mechanism of Exchanges 31
Operation Description 31
Configuration 31
Required parameters 31
Optional parameters 32

MQSeries 33
Prerequisites 33
Operation Description 33
Definition of MQSeries objects 33
Client/Server Architecture 33
Distributed Architecture 34
Configuration 36
Required parameters 36
Optional parameters 36
VA Pac Communication Monitor 37

MQSeries - CICS Bridge 37
Prerequisites 37
Operation Description 37
Definition of MQSeries objects 39
Configuration 39
Required parameters 39
Optional parameters 40

MQSeries - IMS Bridge 40
Prerequisites 40
Operation Description 40
Definition of MQSeries objects 41
Configuration 42
Required parameters 42
Optional parameters 42

EXCI 43
Prerequisites 43
Operation Description 43
Configuration 43
Required parameters 43

Socket 44
Prerequisites 44
Operation Description 44
Configuration 44
Required parameters 44
Optional parameter 45
Communication Monitor generation parameters 45

CICS Socket 45
Prerequisites 45
Operation Description 45
Configuration 47
Required parameters 47
Optional parameter 47
Communication Monitor generation parameters 48

TDS-TCP/IP 49
Architecture 49
Prerequisites 49
Mechanism of Exchanges 49

Configuration 50
Required parameters 50
Optional parameters 50

CPI-C 51
Prerequisites 51
Operation Description 51
Configuration 51
Required parameters 52
Optional parameters 52

TUXEDO 52
Prerequisites 52
Operation Description 52
Configuration 52
Required parameters 53
Optional parameters 53
Use of the BEA Jolt API 54
Launching the repository 54
Declaring the servides and buffers 54

GTEA – ECI 55

Chapter 10: How to solve the communication problems 57

Communication error messages 57

Using the trace 57
Gateway trace 57
MiddlewareAdapter trace 58
IXO trace 58

Chapter 11: Middleware deployment 59

Appendix: Customizing external software 61

IMS CPI-C 61
MVS and IMS configuration 61
VTAM definitions 61
APPC/MVS Definitions 63
IMS Definitions 64

IMS Connect 65
Example of configuration file for IMS Connect 65
Example of startup JCL for IMS Connect 65

CICS CPI-C 66
MVS and CICS Configuration 66
VTAM Definitions 66
APPC/MVS Definitions 68
CICS Definitions 68
SNA Server 3.0A for Windows NT Configuration 70

CICS ECI 86
MVS and CICS Configuration 86
VTAM Definitions 86
APPC/MVS Definitions 87
CICS Definitions 87

CICS TCP/IP Sockets Interface 91
CICS TCP/IP Configuration 91
Prerequisites 91
CICS Startup 91
Definition of CICS TCP/IP transactions 91
Definition of CICS TCP/IP programs 93
Definition of the DCT Table 96
Definitions and initializations of Configuration files (*TCP/IP Version 3.2.0) 96
Definition of the PLT table (*TCP/IP V320) 100
TCP/IP MVS/ESA Configuration 100
Modification of the TCP/IP configuration 100
TCPJOBNAME Parameter in the *hlq*.TCPIP.DATA file 101
Manual Start and Stop of CICS TCP/IP 101
Start of CICS TCP/IP 101
Stop of CICS TCP/IP 102
Cobol Compilation of the VA Pac Communication Monitor Program 102
CICS definitions for the VA Pac application 102
Definition of the Transaction Code 102
Definition of the Communication Monitor program 103
Definition of the VSAM Workfile 103

TUXEDO 104
Client 104
Server 104

MQSERIES 105
CICS Adapter 105
MQSeries Client 106

TDS-TCP/IP 108
TDS-TCP/IP Client Installation / configuration 108
Installation 108
hosts and services files 108
`ATMI` Traces 108
Example of implementation on TDS 109

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service.

The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the following address:

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM France Software Laboratory, Rational Division
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand
FRANCE

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may change this publication, the product described herein, or both.

Trademarks

IBM is a trademark of International Business Machines Corporation, Inc.
AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS, TeamConnection and VisualAge are trademarks of International Business Machines Corporation, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United Stated and/or other counties.
Microsoft, Windows, Windows NT and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.
All other company, product, and service names may be trademarks of their respective owners.

Chapter 1: Introduction

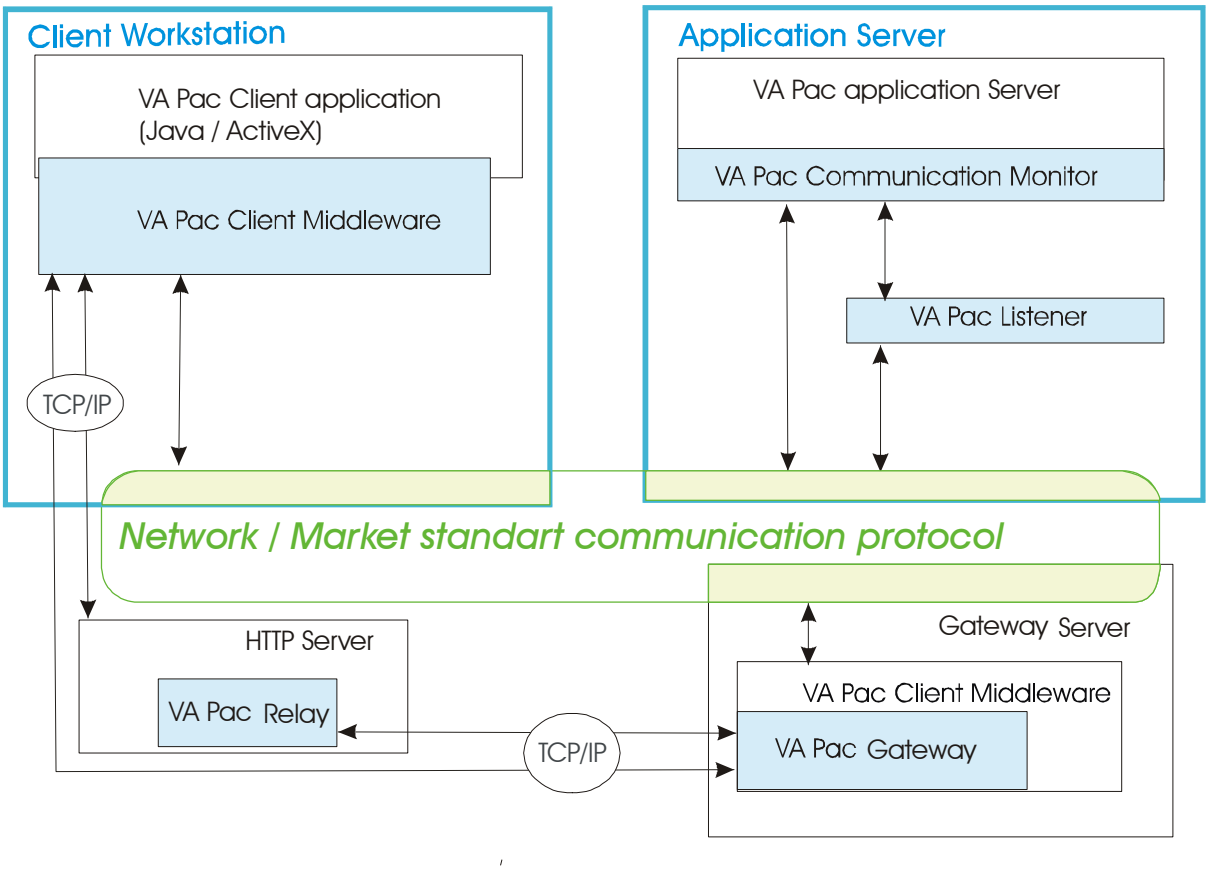
The purpose of the middleware functions is to manage the communication between the Client and Server components of an application using the communication protocols available on the market place.

This guide provides an overall information on the operation of the middleware implemented for applications generated with the eBusiness and Dialog Web Revamping modules.

Specific topics such as the middleware customizing and packaging are detailed in the 'eBusiness Applications - Graphic Presentation' manual. Middleware-specific methods and attributes are documented in the 'Public Interface of Generated Components' manual.

Chapter 2: The different communication modes

General architecture



The Client workstation hosts the Client application.
The VA Pac Client middleware is included in the VA Pac eBusiness runtime.

Communication in direct mode

In this mode of communication, the exchanges between the Client and Server components are directly managed by a market communication protocol e.g. MQ Series, TCP/IP, CICS-ECI, CPI-C….

The implementation of the communication protocol chosen is totally encapsulated.

In Java, it is represented by the **`MiddlewareAdapter`** class, provided in the VA Pac runtime (**`vaprun.jar`**). This class offers an API which enables to define the context of the communication with the Server application.

This API is documented in the '*Public Interface of Generated Components*' manual.

On each workstation hosting the Client application, you must install:

- the VA Pac Client middleware,
- the Client part of the communication protocol in use.

In this mode of communication, when the VA Pac Client application (Java or ActiveX) sends a Client component request, this latter follows the following path : it goes through the VA Pac Client middleware and is then handled by the communication protocol before it finally reaches a first Server component, the Communication Monitor.

The path to the Communication Monitor depends on the communication system architecture (Tuxedo, MQ Series, …) and the configuration implemented by the network Administrator.

You can also use the VA Pac Socket listener module which relies upon a private protocol. In this case, the Client request transits through the listener before it reaches the Communication Monitor. Refer to *Chapter 8: How to use VA Pac Listener*.

Communication via a gateway

The gateway is represented by the **`VapGateway`** component, a program which accesses the VA Pac Client middleware.

In this mode of communication, the VA Pac Client middleware is not installed on the workstation which hosts the Client application but on an intermediate Server. The Client components communicate with the VA Pac gateway which is executed on this intermediate Server.

The gateway is shared by all the Client applications and must then be run permanently.

This enables not to increase the load of tasks performed by the workstation hosting the Client application and thus to perform an optimized and centralized management of the Server accesses.

In this mode of communication, the Client component request follows the following path: it is sent by the VA Pac Client application (Java or ActiveX), and directly goes into the gateway, via TCP/IP. It then transits through the VA Pac Client middleware which is installed with the **`VapGateway`** component. Next, it passes through the Client part of the communication protocol and the network itself, before it finally reaches the Communication Monitor, the first Server component reached.

You can also use the VA Pac Socket listener module which relies upon a private protocol. In this case, the Client request transits through the listener before it reaches the Communication Monitor. Refer to *Chapter 8: How to use VA Pac Listener.*

For all non SOCKET protocols, the Client part of the communication protocol must be installed on the same machine as the one hosting the gateway. However, the TCP/IP communication protocol only is required on each machine hosting the Client application, and this protocol is generally installed by default.

Particular case: relay mode

The relay can be used for Internet-type applications. It unburdens the HTTP Server from the communication with the Servers and from the management of the connected clients contexts. It is especially used to forward all the requests which reach an HTTP Server to another Server which hosts the middleware and the gateway. This lightens the task of the HTTP Server and the Administrator does not have to manage the different communication protocols used by these applications.

The purpose of this component is also to establish a simple TCP/IP relay (such as a router) allowing its Client to connect to another TCP/IP address which it would not be able to reach directly (for example if it is located behind a firewall).

The relay receives requests which are identical to the gateway's and then forwards these requests to a gateway (or another relay). Contrary to the gateway, it does not use the middleware.

It is represented by the **`vaprelay.jar`** program.

Chapter 3: The middleware components

The files listed below can have a **.dll**, **.o**, **.so** extension or none depending on the target platform.

The files with the **.001** extension are versions enabling to activate the trace mode which helps analyze communication problems. For more information, refer to *Chapter 10: How to solve the communication problems.*

Common files

CharConv.txt	File containing codepage conversion tables
GsComMw	Middleware interface for Pacbench/PacDesign (old VA Pac Workstation)
GwAdapter	Adapter to communicate with VapGateway
(lib)JavaAdapter	JAVA/C++ middleware interface (JNI)
MwAdapter	Adapter for communication in Middleware direct mode
VapUtil	Library of C++ functions and utility classes
ixomware	Generic interface of the IXO communication layer
ixomsgen	Error labels (Windows only)

Technical interfaces of communication

These files group the execution functions of the middleware communication services for a specific communication protocol.

ixocics	CICS/ECI protocol
ixocpic	CPI-C protocol
ixocgtea	GTEA (GCOS8) protocol
ixoloc	Local Cobol DLL protocol
ixomqci	MQSeries-CICS Bridge protocol
ixomqmci	Version for MQServer
ixomqims	MQSeries-IMS Bridge protocol
ixomqmim	Version for MQServer
ixomqs	Native MQSeries protocol
ixomqm	Version for MQServer
ixosock	VAP socket protocol
ixotims	IMS Connect protocol
ixotmvs	CICS socket protocol
ixottds	TDS TCP/IP protocol
ixotux	TUXEDO protocol (if **JoltAdapter** is not used)

ixotuxmt	'Multithreads' version for Native Client Threads (if **JoltAdapter** is not used)
fieldtbl.vap	FML conversion file

VAP Gateway module

VapGateway	VAP Gateway program
VapGatewayNT	Version to be installed as a Windows/NT service

For more details, refer to *Communication via a gateway*.

VAP Relay module

vaprelay.jar	Package of VAP Relay Java classes.

This module is autonomous and as such does not need any other component of the Middleware package.

The **.jar** file contains:

- the **com.ibm.vap.relay** package,
- a **readme.txt** file,
- a DOS startup script for the relay: **vaprelay.bat**,
- a Unix startup script for the relay: **vaprelay**.

For more details, refer to *Particular case: relay mode*.

VAP Listener module

listener	VAP Listener program

Under Unix, it is only provided as a compiled object.

A link-edit with the other compiled objects of the Server application is required in order to build a listener executable module.

A makefile example is provided in the package.

dial	Dialogue agent (Windows only)
CodePageConv	Codepage conversion functions
VapUtil1	Library of C++ functions and utility classes (Unix only)

For more details, refer to *Chapter 8: How to use VA Pac Listener*

Chapter 4: Available protocols by server platform

The following lines give a few precisions regarding the Client platforms and gateway Servers which are listed in the tables below:

- **Windows**: Windows 32 bits system (98, NT, 2000, XP).
- **AIX**: v4.2. minimum required, V4.3 for GCOS7-TDS-TCP/IP and GCOS8-GTEA protocols.
- **Solaris**: v2.5.1. minimum required.
- **Unix OS390**: Unix System Services for OS390, V2R9.
- **Linux** : Two versions of the middleware are available:
  Debian GNU/Linux 2.6 with glibc 2.3.6. (to be used with the IBM JDK 5.0) and Debian GNU/Linux 2.2.19 with glibc 2.2.4-7.

☞ The `Readme.txt` file, located in the `middleware/zip` directory, lists the middleware evolutions for the various platforms.

OS/390 - CICS

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Yes				
ECI	Yes				
EXCI				Yes	
TCP/IP Socket	Yes	Yes	Yes	Yes	Yes
MQSeries	Yes	Yes	Yes		Yes
MQ-CICS Bridge	Yes	Yes	Yes		Yes

OS/390 - IMS

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Yes				
IMS Connect	Yes	Yes	Yes	Yes	Yes
MQSeries	Yes	Yes	Yes		Yes
MQ-IMS Bridge	Yes	Yes	Yes		Yes

DOS/VSE – CICS

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C	Yes				

AIX

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
ECI	Yes				
TCP/IP Socket	Yes	Yes	Yes		Yes
MQSeries	Yes	Yes	Yes		Yes
Tuxedo	Yes	Yes			

Solaris

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes
MQSeries	Yes	Yes	Yes		Yes
Tuxedo	Yes	Yes			

Windows

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
CPI-C (TX-Series)	Yes				
ECI (TX-Series)	Yes				
TCP/IP Socket	Yes	Yes	Yes		Yes
MQSeries	Yes	Yes	Yes		Yes
Tuxedo	Yes	Yes			

AS400

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

Compaq OpenVMS

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

Compaq Tru64 UNIX

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

HP-UX

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

Linux

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

NUMA-Q DYNX (SEQUENT)

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes

Host Tandem OSS-Guardian (Himalaya)

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TCP/IP Socket	Yes	Yes	Yes		Yes
Tuxedo	Yes	Yes			

Host GCOS7 - TDS

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
TDS-TCP/IP	Yes	Yes			

Host GCOS8 - TP8

Protocols / Client platform	Windows	AIX	Solaris	Unix OS390	Linux
GTEA	Yes	Yes			

Chapter 5: Notion of Location

There are two fundamental notions underlying the definition of the communication:

- the notion of location, which describes the middleware use context,
- the mode of communication,: direct or via a gateway (possible use of a relay).

Each location:

- Identifies the protocol used to access the VisualAge Pacbase Server,
- Gives the Server address according to this protocol,
- Defines the required communication protocol parameters.

The locations are grouped in the **vaplocat.ini** file. Each location name is specified in the **vaplocat.ini** file between <…>. In each location section, communication parameters are defined using the following syntax: **parameter name=parameter value**.

There are two types of parameters:

- parameters relating to the communication general functions (communication type, message length…),
- parameters specific to each communication type (project name for TCP/TDS, queue manager name for MQSERIES …). These parameters are prefixed by **IXO_** and transferred without change to the middleware layer specific to the communication type in use.

You can edit the **vaplocat.ini** file using the Location Editor tool (which can be launched independently or directly from the eBusiness module, in Developer Workbench). Refer to this tool's online help for details.

The list of parameters and their meaning is documented in *Chapter 9: Protocols Description &* , in the section *Configuration* of each protocol.

Chapter 6: How to use VapGateway

Start-up

To start the gateway, position yourself into the directory where the **VapGateway** is located and type the following command (the options order has no importance):

```
VapGateway [-h]
-s|i|d TCP_PORT_NUM [-l LOCATION_FILE] [-c CHAR_CONV_FILE]
[-t TRACE_LEVEL] [-tf TRACE_FILE]
    [-min NB_MIN_CONNECTION] [-max NB_MAX_CONNECTION]
[-clean CLEANING_INTERVAL] [-clientTimeout CLIENT_TIMEOUT]
[-connectionTimeout CONNECTION_TIMEOUT]
[-retry RETRY_INTERVAL]
```

where:

- **-h**: is the option used to display the syntax
- **-s**: starts the gateway in interactive mode
- **-i**: installs the gateway as a Windows NT service

  By default, the gateway is installed in automatic startup mode.
- **-d**: uninstalls the gateway Windows NT service
- **TCP_PORT_NUM**: sets the gateway's listening number. The default value is 5647.
- **-l LOCATION_FILE**: indicates the locations file. Default is the **vaplocat.ini** file, located in the **VapGateway** program's directory.
- **-c CHAR_CONV_FILE**: specifies the codepage conversion file

  The default is the **charconv.txt** file, located in the **VapGateway** program's directory.
- **-t TRACE_LEVEL**: sets the trace level.
  - 0: no trace
  - 1: traces of errors (default)
  - 3: general traces
  - 5: detailed traces
- **-tf TRACE_FILE**: Indicates the trace file.

  The default trace directory is **VAPTRACE**, located in the VapGateway program's directory.
- **-min NB_MIN_CONNECTION**: Specifies the minimum number of idle Server connections kept in the connection pool upon a cleaning.
  Connections open up as the need for them arises. When this mimimum number is reached, it is kept and the cleaning mechanism is activated.

  The default value is 0.
- **-max NB_MAX_CONNECTION**: Specifies the maximum number of simultaneous Server connections in the connection pool.

Before creating a new connection (when there is no idle connection in the connection pool), the gateway checks the maximum number of connections. If it has not been reached, a new connection is created. It it has been reached, the most recent idle connection (**clientTimeout** parameter) is suppressed before a new one is created. If all the connections are active, the request for a connection is withheld until a connection becomes free or until the **connection Timeout** is over.

This parameter has a major impact on performance.

The default value is unlimited.

- **-clean CLEANING_INTERVAL**: Specifies the number of seconds between two cleanings of idle Server connections in the connection pool.

  For a better performance, the middleware manages a group of associated server connections. An idle connection is a connection, belonging to the group, which has not been used since the last cleaning.

  You must assign this parameter a small value (ex: 1 second) if you do not want to keep idle connections (to limit the use of resources). On the other hand, you must assign this parameter a high value (ex: 60 seconds) if you want to obtain better performance (to reduce the number of connections/diconnections/reconnections).

  The value of the cleaning interval must not be greater than the server connection timeout, otherwise idle connections in the pool will be kept unnecessarily.

  The default value is 60.

- **-clientTimeout CLIENT_TIMEOUT**: Specifies the maximum number of seconds that an idle client connection remains open. Beyond that number, the connection is closed.

  The default value is 120.

- **-connectionTimeout CONNECTION_TIMEOUT**: Specifies the maximum number of seconds that a Client can wait to get a Server connection, when the maximum number of Server connections is reached.

  The default value is unlimited.

- **-retry RETRY_INTERVAL** : Specifies the number of seconds between two attempts of communication with the Server when a communication error occurs.
  A retry is performed only if a communication error occurs on a connection which has been idle during the number of seconds of the last **RETRY_INTERVAL**.
  The default value is **0**.
  You can inhibit this mechanism by assigning the value**-1**.

- **-pcv**: Obsolete. Kept for compatibility only. In this VapGateway version, this option is always set.

The VapGateway version identifier

To identify the version of **VapGateway**, set the trace level to 1 minimum. The version identifier is located in the header of the trace file.

Example of a trace file header created at the gateway startup in interactive mode, with a trace level equal to **1**:

```
[VapGateway 17:51:28:94
     VisualAge Pacbase (*) v3.5 VapGateway vm350v01
     Licensed Materials - Property of IBM 5655-F37
     © Copyright IBM Corp. 1983, 2006. All Rights Reserved]
```

In this example, **vm350v01** is the version identifier.

Under Windows, you can easily identify the version by consulting the Properties panel of the **VapGateway.exe** file via the Windows Explorer.

For more information about the trace, refer to *Chapter 10: How to solve the communication problems.*

Chapter 7: How to use vaprelay.jar

The **vaprelay** is a Java autonomous component, executable with a Java runtime (version 1.2 or higher recommended).

To operate correctly, the relay requires the name or IP address and the listening port of the gateway it communicates with. It must also listen for incoming connections from Clients.

An example of relay program startup is provided as two command files contained in **vaprelay.jar**:

- **vaprelay.bat** for Windows,
- **vaprelay** for Unix.

These files must be extracted and adapted to your needs.

The Java runtime must first be installed. It is not provided in this package.

In standard, you start the relay with the following command:

vaprelay <delegateHostName> [delegatePort [listeningPort]]

where:

- **delegateHostName**: is the name or IP address of the Server where the gateway is executed.
- **delegatePort**: TCP/IP port number where the gateway listens (**5647** by default).
- **listeningPort**: TCP/IP port number where the relay listens for Clients (**5647** by default).

Chapter 8: How to use VA Pac Listener

The listener is part of the VA Pac applications server. It launches the Cobol programs generated on the machine where it is installed.

The generated eBusiness application (Java or ActiveX) communicates with the listener via the socket middleware. The listener then conveys the requests of the application to the VA Pac Communication Monitor.

To install the listener, copy the following two executable programs to a directory:

- **BvpServer.exe**
- **BvpDial.exe**. This executable program is available in two versions, which correspond to the Cobol compiler in use: Acucobol or Microfocus. The Acucobol version is located in the **ACU** directory of the listener and the Microfocus version in the **MF** directory.

You must not forget to add this installation directory to the **PATH** variable.

The listener is delivered for the following platforms:

- AIX,
- AS400,
- HPUX,
- Linux,
- OSF1,
- Open VMS.
- Solaris,
- Tandem,
- Unix OS390,
- Windows.

How to Launch the Listener

You launch the listener on the server application, using the following command:

- For Windows :

```
BvpServer [-h] [-i|d|s <Port_Number>
[Environment_File][Security_program]]
```

- For any other platform :

```
Listener [-h] [-s <Port_Number>
[Environment_File][Security_program]]
```

where:

- **-h**: is the help message
- **-i<Port_Number>[Environment_File][Security_Program]**: installs the listener as a Windows NT service
- **-d <Port_Number>**: uninstalls the listener service
- **-s <Port_Number>**: enables the direct startup of the listener
- **<Port_Number>**: is the decimal value of the TCP/IP port
- **<Environment_File>**: is the environment variables setting file

- **<Security_Program>**: is the program called by the listener for security check

For platforms other than Windows, the listener executable program must be built before being used. This program needs a link-edit with libraries of your application database. Refer to the **readme** file provided in the middleware package built for your target platform.

How to Use the Listener on the UNIX platform

On the UNIX platform, you must transfer:

- the modules **bvpserver.o**, **bvppause.o**, **config.**o, **dtime.o**, **environ.**o, **general.o**, **lockdb.o**, **mems.o**, **sems.o**, **standard.o** and **tn3270.o**,
- the **Makefile** et file,
- the **CodePageConv.so** et **VapUtil1.so** libraries

to the UNIX server before performing the link-edit via the command:

make -f Makefile.

☞ The **CodePageConv.so** et **VapUtil1.so** libraries must be located in the upper directory while the modules are being built.

The result is the **listener.exe** file, located in the upper directory. You can rename it '**server**' for instance. In this case, you will launch it via the command:

server path port [-t <timeout>]&

with:

- **path**: path of the servers' executable programs,
- **port**: socket number used by the listener,
- **timeout**: maximum time for a connection without data ('0' by default for an unlimited timeout)

You can apply various trace levels:

- Level 1 : minimum trace of the listener's processing,
- Level 2 : detailed trace of the listener's processing,
- Level 4 : trace of the messages exchanged between the listener and the client workstation.

To use a trace level, you must restart the listener after setting the **SRV_TRACE** environement variable. For example:

- **export SRV_TRACE=1** for a trace with a level 1
- **export SRV_TRACE=3** for a trace with a level 1 and 2
- **export SRV_TRACE=5** for a trace with a level 1 et 4

While the debug mose is used, the traces are writtenn in the **srv<process.pid>.txt** and **dial<process.pid>.txt** files located in the **/tmp** directory.

The **SRV_DIR** environment variable enables you to create these traces in another directory. For example:

export SRV_DIR=$HOME/tmp

Chapter 9: Protocols Description & Configuration

IMS Connect

Prerequisites

IMS V6

IMS Connect V1R1

Mechanism of Exchanges

- Standard exchange

```
CLIENT QUERY            STREAM                  IMS CONNECT QUERY
SEND---------------->IRM/TRAN/DATA ------------>RECEIVE
RECEIVE<---------------LLzz/DATA<---------------SEND
...                       ...                      ...
RECEIVE<---------------LLzz/DATA<---------------SEND
RECEIVE<------------------CSM<------------------SEND
```

The connection remains active in **SOCKET PERSISTENT** mode; it must be explicitly closed at the end of the exchange.

- Error detected by IMS Connect or the exit message

```
CLIENT QUERY            STREAM                  IMS CONNECT QUERY
SEND---------------->IRM/TRAN/DATA ------------>RECEIVE
RECEIVE<------------------RSM<------------------SEND
                Connection closed
```

Operation Description

The HWSIMSO0 'user message exit' is used with the following configuration:

- Conversational mode
- Commit mode and Sync level defined by default in HWSIMSO0 (**commit mode 1**, and **sync level NONE)**

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=TCPIMS

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **SOCKET** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_TRANSID

Name of the Communication Monitor transaction (8 char.).

IXO_ADDRESS

Host IP address or logical name, followed by the IMS Connect port number, separated by a SPACE character (30 char. max.)

IXO_DATASTORE

Name of the datastore, defined to the IMS Connect (8 char. max.)

IXO_RACFGROUP

Name of the RACF group used for IMS Connect. (8 char. max.)

Optional parameters

HOST_ENCODING

Server codepage value. This value must be defined in the codepage conversion file (**charconv.txt**), provided in the middleware package.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

IXO_PERSISTENT

Persistence mode of the Socket connection.
Y persistent (default mode)
N non persistent

MQSeries

Prerequisites

MQSeries 5.2

Operation Description

MQSeries is a communication system based on message exchanges via message queues. These exchanges are managed by the MQSeries Manager (Queue Manager).

The MQSeries Middleware uses 2 message queues:

- the Request Queue

  This Queue is used to send request messages to the Server application.
- the Reply Queue

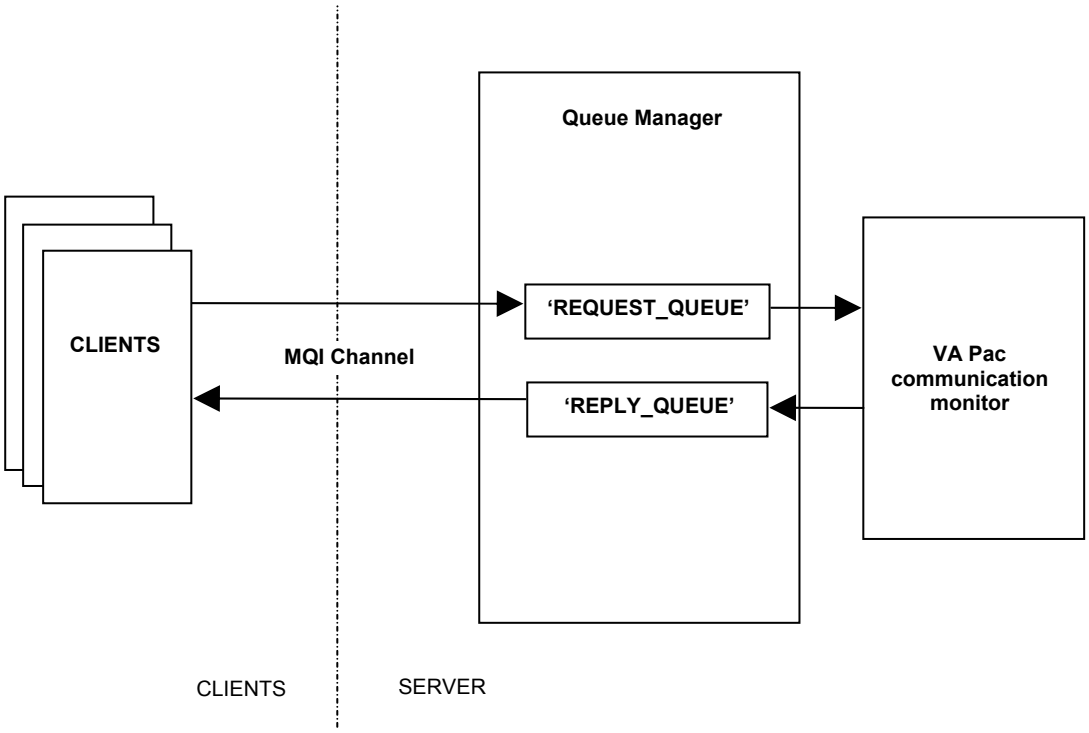
  The Client application receives the Server replies in this Queue whose name is transmitted to the Server application in the header of the request.

The request messages put on the Request Queue are read by the MQSeries Communication Monitor on the Server, and are processed on the same principle as the other communication system types. The Server programs send the replies to the Client via the Reply Queue specified in the header of each request. To make the link between a reply and its request, each reply contains the corresponding request's message identifier.

Definition of MQSeries objects

Client/Server Architecture

For this type of architecture, the Client and Server applications share the same Queue Manager. This Queue Manager and all the MQSeries objects are implemented on the Server of the application.



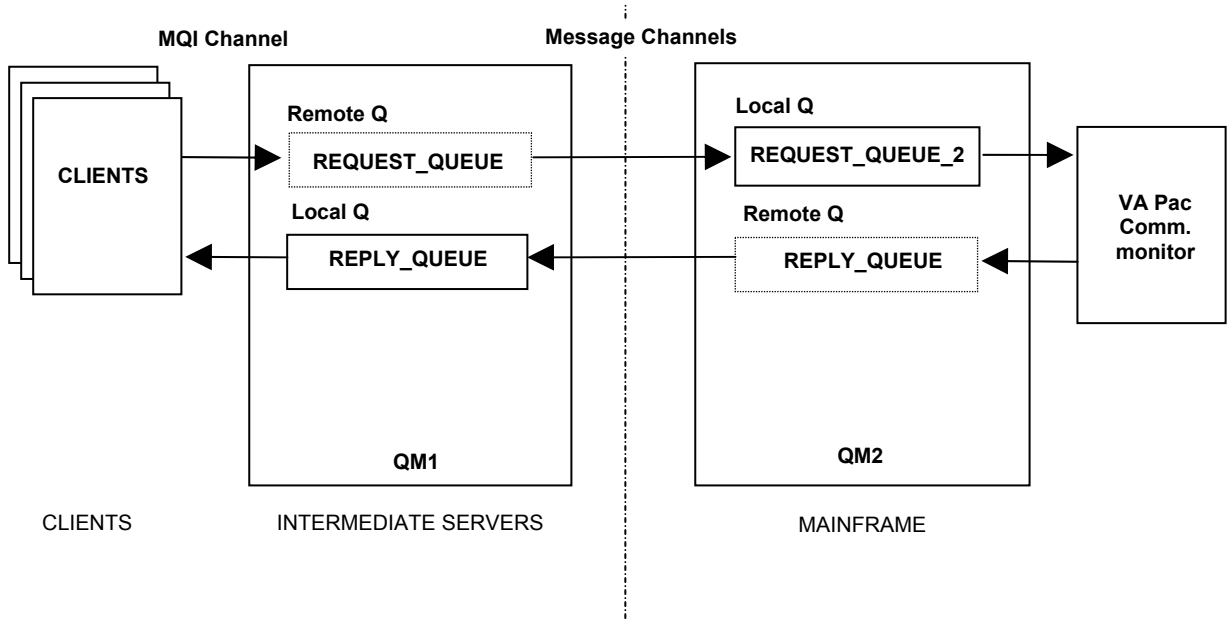
In this context, the following MQSeries objects must be defined:

- 2 `Local Queue`
  - `Request Queue`
  - `Reply Queue`
- 1 `MQI Channel SVRCONN/CLNTCONN`

Distributed Architecture

In that case, several Queue Managers are interconnected to make up a MQSeries network.

Each application, Client and Server, is connected to its own Queue Manager, known as 'Local '.



The following MQSeries objects must be defined:

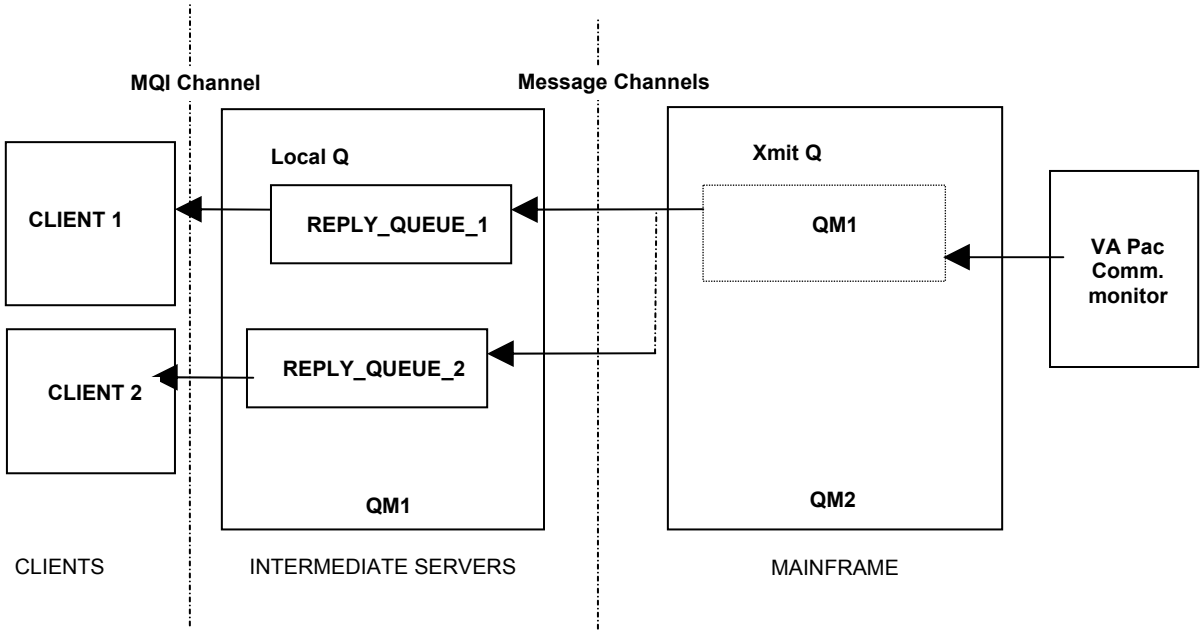
- For the local Queue Manager on the Client
  1 **Local Queue**: **Reply Queue**
  1 **Remote Queue**: **Request Queue**
  1 **MQI Channel** **SVRCONN/CLNTCONN**
  2 **Message Channel**: 1 **SDR** + 1 **RCVR**

- For the Queue Manager on the Server:
  1 **Local Queue**: **Request Queue**
  1 **Remote Queue**: **Reply Queue** (same name as the **Reply Queue** of the Client)
  2 **Message Channel**: 1 **SDR** + 1 **RCVR**

The MQSeries system internal technical objects, such as Transmit Queue, Dead Letter Queue,... are not mentioned here.

A Remote Queue is not really a message Queue; its definition only allows the Queue Manager to know the references required to route the messages to the target Queue (the names of the Queue Manager, remote Queue and Transmit Queue).
Remote Queues are required because an application can only read in a Local Queue belonging to its own Queue Manager.

Using the Name Resolution function of MQSeries:



This technique reduces considerably the number of MQSeries objects. Whatever the number of Reply Queues needed by the Client applications, the only required MQSeries object on the server side is:

1 **Transmit Queue**: same name as the Client Queue Manager

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=MQSERIES or

COMM_TYPE=MQSERVER if the middleware is installed on the same machine as MQSeries Server and if you want to use a direct link with MQSeries.

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **MQSERIES** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_QUEUEMANAGER

Name of the local Queue Manager of the Client application (48 char. max).

IXO_REQUESTQUEUE

Name of the request queue (48 char. max).

IXO_REPLYQUEUE

Name of the reply message queue (48 char. max).

Optional parameters

HOST_ENCODING

Server codepage value. This value must be defined in the codepage conversion file (**charconv.txt**), provided in the middleware package.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

IXO_DYNAMICREPLYQUEUE

When this parameter is set, the reply Queue is dynamically created by MQSeries to transmit the reply messages (refer to MQSeries documentation to use this type of Queue).

IXO_REQUESTEXPIRY

Expiry time of a request, in seconds.
The request (reply included) is considered as lost when this time is reached.
All the messages relating to this request will then be automatically purged by the Queue Manager.
Unlimited by default.

VA Pac Communication Monitor

This Monitor is generated in VA Pac by specifying the **MQSERIES** communication type.

The Monitor can be started manually or automatically (TRIGGER MQSeries) and in batch or transactional mode depending on the environment of the Server application.

Two input parameters are required to start the monitor: the **Request Queue** name followed by the **Queue Manager** name, separated by a space. If the Trigger is used to automatically start the Communication Monitor, these two parameters must be transmitted via the **USER DATA** area of the **Process Definition** object, associated with the **Request Queue**.

The VA Pac generation options are the following:

WAITINT

Waiting time (in seconds) to receive a request.

One second by default.

WAITINT1

Waiting time for the Monitor to receive a new request in the **request queue**.

Unlimited by default.

MQSeries - CICS Bridge

Prerequisites

MQSeries 5.2

Operation Description

On the Server side, the CICS Bridge module operates between MQSeries and the VA Pac Communication Monitor. It communicates with the Client application via MQSeries objects, and with the VA Pac Communication Monitor via the **EXEC CICS LINK** command.

The protocol defined between the Client component and the Communication Monitor is the same as the one implemented for any other synchronous communication system (CPI-C or ECI):

When the length of the request message is shorter than the physical limit defined by **MESSAGE_LENGTH** in the parameters file (**vaplocat.ini**), the message is sent all at once. Otherwise, it is split in segments. The Client part sends the first segment and awaits an acknowledgment from the VA Pac Communication Monitor before sending the next one and so on until the last segment of the message.

The Communication Monitor receives each segment directly in the **COMMAREA**. After receiving the last segment, it rebuilds the complete logical message and then processes the request. The reply is processed following the same operation principle, i.e. if the length of the reply is shorter than the maximum defined length, the reply is sent back to the Client all at once in the **COMMAREA**. Otherwise, the reply is split in segments.

Structure of the request messages:

<MQMD><MQCIH><TransID><COMMAREA data>

Structure of the reply messages:

<MQMD><MQCIH><TransID><COMMAREA data>

Structure of the error messages sent by the Bridge:

<MQMD><MQCIH><ErrorText>

MQMD

Which describes the request message:

- **CorrelId = MQCI_NEW_SESSION**
- **MsgType = MQMT_REQUEST**
- **Format = MQFMT_CICS**
- **Report=MQRO_EXCEPTION+MQRO_EXPIRATION+MQRO_PASS_MSG_ID + MQRO_COPY_MSG_ID_TO_CORREL_ID+MQRO_DEAD_LETTER_Q;**
- **Expiry** = IXO external parameter
- **CodedCharSetId** = IXO external parameter

MQCIH

Header of CICS Bridge of the request:

- **Version** = IXO external parameter
- **Format = MQFMT_STRING**
- **ReplyToFormat = MQFMT_STRING**
- **UOWControl = MQCUOWC_ONLY**
- **LinkType = MQCLT_PROGRAM** (DPL program)
- **Authenticator** = password
- **TransactionId** = transaction code of the Communication Monitor

TransID

CICS transaction name of the Communication Monitor.

COMMAREA

Segment of message transmitted.

ErrorText

Error label when the return code, in the **MQCIH** header (**ReturnCode**) indicates an error.

Other **MQCIH** header fields give information on the encountered error:

- **CompCode**
- **Reason**
- **Function**
- **AbendCode**

Definition of MQSeries objects

The MQSeries objects are documented in the *Definition of MQSeries objects* for the MQSeries protocol.

A Trigger (FIRST) can be defined at the Request Queue level to automatically start the Bridge Monitor.

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=MQCICS or
COMM_TYPE=MQMCICS if the middleware is installed on the same machine as the MQSeries Server and if you want to use a direct link with MQSeries.

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **MQBRIDGE** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_TRANSID

Name of the Communication Monitor transaction (8 char.).

IXO_QUEUEMANAGER

Name of the local Queue Manager (48 char. max)

IXO_REQUESTQUEUE

Name of the request queue (48 char. max)

IXO_REPLYQUEUE

Name of the reply queue (48 char. max)

Optional parameters

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

IXO_DYNAMICREPLYQUEUE

When this parameter is set, the reply Queue is dynamically created by MQSeries to transmit the reply messages (refer to MQSeries documentation to use this type of Queue).
This parameter must contain the dynamic name of the Queue (Dynamic Queue name) and the 'Queue Model' name must be specified in the parameter (48 char. max.)

IXO_REQUESTEXPIRY

Expiry time of a request, in seconds (9 char. max.)

IXO_HEADERVERSION

Version of the **MQCIH** structure: 1(default) or 2, depending on the CICS-Bridge used interface (1 char. max.)

IXO_LOCALCCSID

Code of the machine character set in local (819 by default) (9 char. max.) used by MQSeries for the message conversion.

Do not define **HOST_ENCODING**, because the transcoding is managed by MQSeries.

For the DQM distributed architecture, the sender Channels must have the **CONVERT** option set up to **YES**.

MQSeries - IMS Bridge

Prerequisites

MQSeries 5.2

Operation Description

On the Server side, the IMS Bridge operates between MQSeries and the VA Pac Communication Monitor. It communicates with the Client application via MQSeries objects, and with the VA Pac Communication Monitor via the OTMA interface.

The protocol defined between the Client component and the Communication Monitor is the same as the one implemented for the other synchronous communication systems (CPI-C or ECI).

When the length of the request message is shorter than the physical limit defined in the parameters file (**vaplocat.ini**), the message is sent all at once. Otherwise, it is split in message segments. The Client part sends the first segment and waits for an acknowledgment from the VA Pac Communication Monitor before sending the next segment and so on until the last segment of the message.

The Communication Monitor receives each message segment in the IMS Queue via the **GU** instruction. After receiving the last segment, it rebuilds the complete logical message and then processes the request. As for the request message sending, if the length of the reply message is shorter than the maximum defined length, the reply is sent back to the Client all at once with the **ISRT** instruction. Otherwise, the Monitor applies the same principle of split and segments sending as for the request message.

The used ***Format*** of MQSeries message is **MQFMT_IMS** type:

<MQIIH><LLZZ><Transcode><Application data>

LL

Length of the following segment (maximum length of an IMS segment =32764 bytes)

ZZ=00

Transcode

IMS transaction code (8 characters)

MQIIH

IMS Bridge header:

- **Format = MQFMT_IMS_VAR_STRING**
- **ReplyToFormat = MQFMT_IMS_VAR_STRING**
- **Authenticator** = password of the user specified in **MQMD**
- **CommitMode = 0 (commit then send)**
- **SecurityScope = MQISS_FULL**

  Security control at the IMS control and dependent Regions

Definition of MQSeries objects

The MQSeries objects are documented in the *Definition of MQSeries objects* for the MQSeries protocol, but they have the following specificities for the MQSeries – IMS Bridge protocol:

- **Bridge Storage Class**

  Definition of a **Storage Class** specifying the group name and the XCF member name
- **Request Queue**

  Local queue on the Server defined with the **Storage Class** of the Bridge.

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=MQIMS or
COMM_TYPE=MQMIMS if the middleware is installed on the same machine as the MQSeries Server and if you want to use a direct link with MQSeries.

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **MQBRIDGE** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_QUEUEMANAGER
Name of the local Queue Manager of the Client application

IXO_REQUESTQUEUE

Name of the request queue

IXO_REPLYQUEUE

Name of the reply queue

IXO_TRANSID

Name of the Communication Monitor transaction

Optional parameters

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

IXO_LOCALCCSID

Local codepage (819 by default), used by MQSeries for the message conversion during the reading and writing of the Queues.

IXO_HEADERVERSION

Version of the **MQIIH** structure: 1(default) or 2, depending on the IMS-Bridge used interface (1 char. max.)

IXO_DYNAMICREPLYQUEUE

When this parameter is set, the reply Queue is dynamically created by MQSeries to transmit the reply messages (refer to MQSeries documentation to use this type of Queue).

IXO_REQUESTEXPIRY

Expiry time of a request, in seconds.
The request (reply included) is considered as lost when this time is reached. All the messages relating to this request will then be automatically purged by the Queue Manager.
Unlimited by default.

Do not define **HOST_ENCODING**, because the transcoding is managed by MQSeries.

For the DQM distributed architecture, the transmitting Channels must have the **CONVERT** option set up to **YES**.

EXCI

Prerequisites

This middleware only operates on the OS390 V2R9 platform with the CICS Transaction Server 1.3.

Operation Description

The middleware uses the **EXCI Call Interface** API in its implementation.

It establishes a connection calling successively the following functions:

- **Initialise_Use**,
- **Allocate_Pipe**,
- **Open_Pipe**.

Then the messages are exchanged via the function:

- **DPL_Request**

When the dialogue with the Server is finished, the connection is closed by calling the functions:

- **Close_Pipe**,
- **Deallocate_Pipe**.

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file,see *Chapter 5: Notion of Location*).

Required parameters

COMM_TYPE=EXCI

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **CICS** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_NETNAME

EXCI user identifier or NETNAME attribute affected to the CONNECTION defined for the Pipe (8 char. max).

IXO_CICSAPPLID

Applid of the CICS region (8 char. max).

IXO_TRANSID

Name of the Communication Monitor transaction (4 char. max.).

Socket

Prerequisites

TCP/IP

Operation Description

The protocol adopted by the middleware is based on the the use of the VisualAge Pacbase Socket listener.

The listener is detailed in *Chapter 8: How to use VA Pac Listener*.

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=SOCKET

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **SOCKET** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_ADDRESS

This parameter (30 characters maximum) must have the format:

0x0002ppppaaaaaaaa, where:

0x0002: Internet standard addressing mode (AF_INET)

pppp: port number in hexadecimal

aaaaaaaa: IP address in hexadecimal

or **Host Port**, where:

Host: IP address with **a.b.c.d** format or logical name of the Server

Port: port number in decimal, 5 characters maximum

Optional parameter

IXO_LISTENERCHARCODE

Value of the character code. This parameter must be set to **EBCDIC** when the VisualAge Pacbase listener is on an AS400 platform.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

Communication Monitor generation parameters

COMM_TYPE=SOCKET

CICS Socket

Prerequisites

CICS TCP/IP Sockets Interface V3.1

Operation Description

The protocol adopted by the middleware is based on the standard CICS Socket listener using: **CSKL** transaction calling the **EZACIC02** program.

- The listener startup is defined via the **CSKE** transaction. For example, the specification of the listening port number.
- After the connection of a Client to this port, it must send a first message to the listener. This initial message must have one of the structures required by the listener (**Listener Input Format**).

  The retained structure is: **CodeTransaction,MessageAppli**
  where
  - **CodeTransaction**

    CICS transaction which starts the VA Pac Communication Monitor (4 char.)
  - **,**

    separator (1 char)
  - **MessageAppli**

    message transmitted to the VA Pac Communication Monitor (35 char. max.)

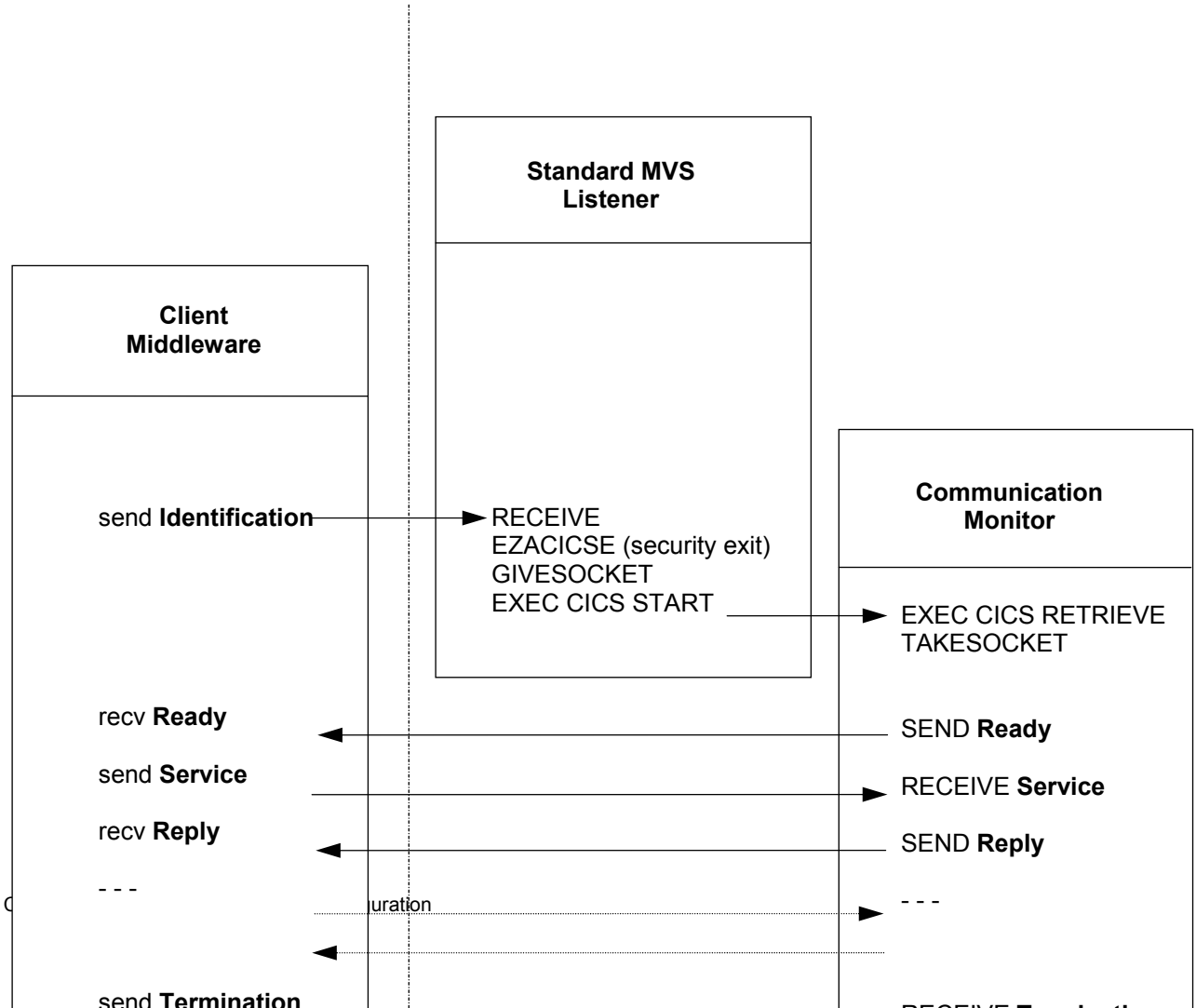
- After receiving this message, the listener starts the Server transaction, passing a Cobol area which has the following structure (**Listener Output Format**):

```
01          TCPSOCKET-PARM.
   05       GIVE-TAKE-SOCKET    PIC 9(8) COMP.
   05       LSTN-NAME           PIC X(8).
   05       LSTN-SUBNAME        PIC X(8).
   05       CLIENT-IN-DATA      PIC X(35).
   05       FILLER              PIC X(1).
   05       SOCKADDR-IN-PARM.
      10    SIN-FAMILY          PIC 9(4) COMP.
      10    SIN-PORT            PIC 9(4) COMP.
      10    SIN-ADDRESS         PIC 9(8) COMP.
      10    SIN-ZERO            PIC X(8).
```

The application message is transferred via the **CLIENT-IN-DATA** area.

- Once started, the Server transaction is directly connected to the Client application.

The security control module is not standardly delivered, but an exit point is provided to implement this function. This module must be developed as a CICS program and named **EZACICSE**. It is systematically executed by the listener, before the Server transaction startup, by an **EXEC CICS LINK** and by transmitting the security information via the **COMMAREA** (cf. CICS TCP/IP Socket Interface Guide and Reference V3R1).



WORKSTATION

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=TCPMVS

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **SOCKET** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

IXO_TRANSID

Name of the Communication Monitor transaction (4 char.)

IXO_ADDRESS

This parameter (30 characters maximum) must have the format:

0x0002ppppaaaaaaaa, where:

- **0x0002**: Internet standard addressing mode (AF_INET)
- **pppp**: port number in hexadecimal
- **aaaaaaaa**: IP address in hexadecimal

or **Host Port**, where:

- **Host**: IP address with **a.b.c.d** format or logical name of the Server
- **Port**: port number in decimal, 5 characters maximum

Optional parameter

HOST_ENCODING

Server codepage value. This value must be defined in the codepage conversion file (**charconv.txt**), provided in the middleware package.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

Communication Monitor generation parameters

COMM_TYPE=SOCKET

WAITINT

Waiting time (in seconds) before the automatic stop of the Communication Monitor
30 minutes by default.

TDS-TCP/IP

Architecture

```
                    Windows or Aix    |  GCOS 7
                                      |
Client components <-> API TDS-TCP/IP <-> OPEN7/SOCKG7 <-> TDS <-> Server components
                                      |
```

Prerequisites

The required Bull products are the following:

- on GCOS7
  - GCOS 7-V7 (minimum TS7560) or GCOS 7-V8 (minimum TS8560) or GCOS 7-V9 (minimum TS9662)
  - TDS-TCP/IP
  - SOCKG7 V4.1.0
  - GCOS 7 OPEN 7 release V5
- Windows Client
  - TDS-TCP/IP API for Workstation
- AIX (4.3) Client
  - TDS-TCP/IP API for Unix

Your TDS application must be generated with the TCP-IP option.

Mechanism of Exchanges

```
- Windows/Aix Client -                    - GCOS7 TDS TCPIP -
-------------------------------------------------------------------------------

tpconnect                          -->    Connection request, LOGON

tprecv "READY"                     <--    SEND "READY"

tpsend "<MONITOR> <PCVrequest>"    -->    Starts <MONITOR> with request in COMMAREA
         8 char.

tprecv "reply" or "error"          <--    RECEIVE,
                                          SEND "IXO&_____<PCVreply>" or "error"
                                                  5 spaces
tpsend "BYE"                       -->    LOGOUT

tprecv ""                          <--    SEND ""

tpdiscon
```

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=TCPTDS

MONITOR

Name of the VA Pac Communication Monitor program.

This Communication Monitor must be generated with the **TCPTDS** type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor (value: 32000 max.).
The maximum length defined by the **MESSAGE-LENGTH** clause in the **TDS SECTION** of **TDSGEN** must be bigger than the length of the message defined here, plus the 9 characters of the **IXO** header, to avoid a protocol error.

IXO_TRANSID

Name of the Communication Monitor transaction (8 char.).

IXO_HOSTNAME

DPS 7000 host name. The IP address is not accepted (15 char. max.).

IXO_TDSNAME

TDS name. (4 char. max.).

Optional parameters

HOST_ENCODING

Server codepage value. This value must be defined in the codepage conversion file (**charconv.txt**), provided in the middleware package.

IXO_PROJECT

Project name. Once connected, the Client application can start all the authorized transactions for this project according to the PROJECT/TDS code specified in the GCOS 7 catalog. If this parameter contains space characters only, the default GCOS 7 project is taken into account. (12 char. max.).

IXO_BILLING

The account is verified in the GCOS 7 catalog. If the parameter contains space characters only, the default GCOS 7 account, allocated to this project user, will be taken into account (12 char. max.).

IXO_DATACONVERT

Automatic conversion of data, ASCII/EBCDIC.

If this option is activated (="**Y**", by default), the ASCII/EBCDIC conversion of the buffer data is automatically managed by the ATMI interface.
The conversion of application data can also be managed by the application; example: the middleware Adapters, by setting the **HOST_ENCODING=297** parameter.

The message sent by **tpsend** is composed of the TDS transaction code, followed by a space and then by the application data. Therefore, if the automatic conversion is not activated, the IXO layer will have to manage the conversion of the header part of the message: the transID code and the Space character.

The **IXO_DATACONVERT** and **HOST_ENCODING** conversion options must be exclusive to avoid the double conversion.

CPI-C

Prerequisites

IBM Personal Communication 4.3

Operation Description

This type of communication is implemented using the standard CPI-C (Common Programming Interface - Communications) X/Open API .

Each exchange with the VA Pac Communication Monitor, on the Client side, is broken down into successive calls of the following functions:

- conversation initialization (cminit)
- conversion parameters setting (cmscst, cmscsu..)
- connection (cmallc)
- request sending (cmsend)
- reply receiving (cmrcv)
- connection closing (cmdeal)

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=CPIC

MONITOR

The 'Symbolic Destination Name' defined in the CPI-C configuration file and associated with the transaction of the VA Pac Communication Monitor.

By convention, the 'Symbolic Destination Name' must have the same name as the VA Pac Communication Monitor.

This Communication Monitor must also be generated with the CPIC type.

MESSAGE_LENGTH

Maximum message length expected by the Communication Monitor.

Optional parameters

HOST_ENCODING

Server codepage value. This value must be defined in the codepage conversion file (**charconv.txt**), provided in the middleware package.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

TUXEDO

Prerequisites

Tuxedo 8.1

Operation Description

This type of communication is implemented using the ATMI API.

Each time the Communication Monitor is called, a **tpcall** is sent with the Monitor name as the service name ; a connection to the server has been opened beforehand via a **tpinit**.

Configuration

The configuration of the communication protocol requires the setting of parameters in the **vaplocat.ini** locations file (for more details on this file, see *Chapter 5: Notion of Location*).

The parameters of the Client part of this communication protocol are documented in the *Appendix: Customizing external software*.

Required parameters

COMM_TYPE=TUXEDO for TUXEDO/WS Mono-Thread
or
COMM_TYPE=TUXEDOMT for TUXEDO/WS Multi-Thread

MONITOR

Name of the Communication Monitor defined as a TUXEDO service.

IXO_ADDRESS

This parameter (30 characters maximum) must have the following format:

0x0002ppppaaaaaaaa, with :
0x0002 : internet standard addressing mode (AF_INET)
pppp : port number in hexadecimal format
aaaaaaaa : IP address in hexadecimal format

or **Host Port**, with :
Host: IP address formatted **a.b.c.d** or logical name of Server
Port: port number in decimal format (5 characters maximum)

Optional parameters

IXO_PASSWORD

User password ('user authentication password'). 30 characters maximum.
This type of password is managed by the security system (Standard Tuxedo, Kerberos, …) which provides the authentication service and which is specified in the **UBBCONFIG** file of the application.
This parameter is required if the **USER_AUTH** security option is defined in the TUXEDO server (in **UBBCONFIG**)
This parameter should preferably be sent by the client application.

IXO_APPLPASSWORD

TUXEDO application password (30 characters maximum).
This password is generally set at the beginning and should not change. Il is defined each time the TUXEDO configuration file (**UBBCONFIG**) is generated.
This parameter is required if the **USER_AUTH** security option is defined in the TUXEDO server (in **UBBCONFIG**).

IXO_CLTNAME

Client name managed in the TUXEDO server application. (30 characters maximum.)
The **User Name / Client Name** couple identifies a client of the TUXEDO Application. It is used to check the access authorization to the application, and not to authenticate the user.
It is present or not, depending on the configuration of the security in the TUXEDO Server Application.

IXO_GRPNAME

The **Group Name** is used to associate a client of the application to a Resource Manager. (30 characters maximum.)
It is present or not, depending on the configuration of the security in the TUXEDO Server Application.

IXO_FMLCONVERSIONFILE

FML conversion
File name in FML format, used to format the input buffer.

NOTE: you must also specify the access path in the **FLDTBLDIR** environment variable and the filename in the **FIELDTBLS** environment variable.

HOST_ENCODING

Value of the server page code. This value must be defined in the file which contains the conversion table of the code pages (**charconv.txt**), delivered in the middleware package.

IXO_TIMEOUT

Waiting time to receive a reply, in seconds (30 seconds by default).

Use of the BEA Jolt API

Instead of using the classical TUXEDO API (which implies the use of the **ixotux** and **ixotuxmt** files), you can use the BEA Jolt API. This API, written in Java, can be accessed via **JoltAdapter**.

Launching the repository

Jolt uses a repository in which the services and buffers in use are described. You must then launch the Jolt repository and listener by adding the following lines in the TUXEDO configuration file:

```
JREPSVR SRVGRP=GROUP2 SRVID=6
  CLOPT="-A -- -W -P /home/ptpc/TuxServ/repository/jrepository"
JSL     SRVGRP=GROUP2 SRVID=5
  CLOPT="-A -- -n //brutus:55555 -m 1 -M 30 -x 10 -T 60"
```

In the classpath of the application which is to access Jolt, you must add the paths to **JoltRun.jar**, located under **ebusinesstolls\lib**, and the BEA **.jar** (**jolt.jar**, **joltjse.jar**).. In this case, you must set the JoltAdapter **PrimaryServerAddress** to:

```
//brutus:55555
```

Declaring the servides and buffers

The buffers used for communication are **C_ARRAY** type and are named **VAPINPUT** and **VAPOUPUT**.

A utility enables you to declare the services and the buffers in a text file. You can also declare them via a graphical editor (refer to the documentation of the BEA editor, available at:

http://edocs.beasys.com/tuxedo/tux81/jdg/dvrepos.htm)

To launch the utility, run the following script::

```
java -classpath $CLASSPATH/jolt.jar:$CLASSPATH/joltadmin.jar
bea.jolt.admin.jbld -u jpr //brutus:55555 BulkLoadFile
```

with `BulkLoadFile` which contains:

```
service=MCTXFI
export=true
inbuf=CARRAY
outbuf=CARRAY
param=VAPINPUT
type=carray
access=in
param=VAPOUTPUT
type=carray
access=out
```

GTEA – ECI

For information on these protocols, please contact the Technical Support.

Chapter 10: How to solve the communication problems

Communication error messages

The communication errors with the Server produce the **com.ibm.vap.generic.CommunicationError** error. Like any Java error, a VA Pac communication error returns no key. To know the cause of the error, you must retrieve the message associated with the error (**getMessage()** method of the class).

Three communication error messages may be displayed:

- **Open server error**
- **Call server error**
- **Close server error**

If a communication error message is displayed, inform the person in charge of the communication because the line may be blocked or defective, or a Server may be busy, etc.

You can also find the cause of the problem by consulting the trace.

Using the trace

The trace allows to analyze the communication problems that might have occurred.

The trace can be set at three levels:

- gateway, if used.,
- communication adapter (**MiddlewareAdapter** class),
- communication interface (**IXO** DLLs).

Gateway trace

You can specify the gateway trace file via the **-tf** startup command option and the trace level via the **-t** option. The different trace levels are the following:

- **0**: no trace
- **1**: traces of errors (default)
- **3**: general traces
- **5**: detailed traces

Example of trace created at the gateway startup in interactive mode, with a trace level equal to **1**:

```
[VapGateway 17:51:28:94 Parameters:
 LocationsFile=C:\TstMware\tools\vaplocat.ini,
CodePageFile=N:\MwTeam\TestMware\tools\CharConv.txt]
[VapGateway 17:51:28:94 Traces:
 File=C:\TstMware\VapTrace\VapGateway_020329_1751
 28_902.log, Level=1]
[VapGateway 17:51:28:94 Client connections: Timeout=30s]
[VapGateway 17:51:28:94 Server connections: Min=0, Max=No,
Cleaning=60s, ConnectionTimeout=Infinite]
```

```
[VapGateway 17:51:28:94 Address: Host=pc5548hd
 (9.101.40.17), Port=50000]
[VapGateway 17:51:28:94 Waiting for client connection]
```

MiddlewareAdapter trace

MiddlewareAdapter is a class provided in the VA Pac runtime (**vaprun.jar**) and is used for a communication in direct mode with the Server. This class provides an API allowing to define the communication context with the Server application, to send service requests and to receive the reply messages. As for **VapGateway**, it is possible to trace the whole processing calling the following methods of the **MiddlewareAdapter** object associated with the proxies used by the application:

- **setTraceLevel (Int TraceLevel)**,
- **setTraceFile (String TraceFile)**.

These methods are documented in the '*Public Interface of Generated Components*' manual.

IXO trace

Two versions of communication DLL are delivered: the Strip version and the Trace version.

The Strip version corresponds to the optimized version of the middleware DLLs. These DLLs do not contain the code which allows to trace the communications processing and do not interpret the environment variables reserved for this purpose.

Upon installation, the DLLs ready to be executed are the Strip ones.

The DLL files of the Trace version are delivered with the **.001** extension.
To use the Trace DLL, you must change the **.001** extension of the **IXO** DLL, which corresponds to the used communication protocol, into **.DLL**, **.SO** or **.O**, depending on the execution platform.

*For example, if you use the MQSeries protocol under Windows, you must rename the **IXOMQS.001** file in **IXOMQS.DLL**.*

When the trace DLL is in use, you can enable the traces by setting the two following environment variables:

- **IXOTRACE**:
  This variable allows to enable (**IXOTRACE=1**) or disable (**IXOTRACE=0**) the trace of the Middleware API.
- **IXOTRACE_FILE**:
  This variable allows to specify, when the trace is active (**IXOTRACE=1**), the trace file path.
  example: **IXOTRACE_FILE=c:\tmp\ixo_err.txt**

The **IXO** traces file is never re-initialized by the middleware functions. To avoid an undesirable cumulative effect, it is advised to systematically delete this file, it will automatically be re-created.

Do not forget to put the Strip version in use if you do not need any trace.

Chapter 11: Middleware deployment

The middleware is delivered, as one compressed file per target platform, in the **middleware** directory, with a **readme** file that you should read before manually installing the middleware components. It is advised to install the complete package corresponding to the platform where the middleware will be executed.

To install the middleware components, you must:

- transfer the package to the target machine,
- decompress and extract the files to a directory dedicated to the VA Pac middleware, using WinZip or the **tar** Unix command.
  Example of extraction command on Solaris:
  >zcat vm300v06_solaris.tar.z | tar -xvf -

  For Unix systems, you must change the **owner** and **group** attributes of all the extracted files. For example:
  >chown monident *
  >chgrp mongroupe *

Appendix: Customizing external software

To implement the communication of an application generated with the eBusiness and Dialog Web Revamping modules, a specific parameterizing of the prerequisite external software is required.

The following configuration examples correspond to configurations which have been tested. Therefore, they depend on a particular technical environment. For example, for the communications between Windows platforms and the MVS host, the machines located on the Token-Ring invoke an SNA Communication Manager or SNA Server, gateway, which in turn invokes a 3745 IBM controller to keep the MVS host.

These configuration solutions only suit a particular context and must then be customized to meet the technical requirements of each site.

IMS CPI-C

MVS and IMS configuration

VTAM definitions

Prerequisite (minimum): VTAM 3.3 Version

Definition of ATCSTR of the VTAM

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC TIC Address (LAN attachment) 3745 controller in NCP:

```
*********************************************
*            TIC BNN
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                       PORTADD=01,
                       LOCADD=400003172000,
                       ISTATUS=ACTIVE,
                              UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                       ADDR=01
*********************************************
```

APPC/IMS Definition

A specific APPC/IMS application for LU6.2 must be defined in VTAM (different from the APPL IMS used for the 3270 terminals).

```
*/* LIB: SYS1.VTAMLST(A0100)
*/*
A01IMS62 APPL    EAS=50,                ESTIMATED CONCURRENT SESSIONS
*
         MODETAB=MTLU62,                 name of the modes table
         DLOGMOD=LU62,                   name of the mode in the table
         APPC=YES,                       compulsory parameter
         ACBNAME=A01IMS62,       APPLID FOR ACB
         AUTH=(ACQ,BLOCK,PASS)   IMS CAN ACQUIRE & PASS TMLS
```

Mode Definition

- Definition of characteristics for the LU6.2 session.
- The SNASVCMG mode is used with the 'Parallel Sessions' support.

```
TITLE '--- "MODTABLE" RELATED TO THE LU 6.2 ---'
*
MTLU62   MODETAB
         SPACE 4
SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
               PSERVIC=X'060200000000000000000300'
LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
               PSNDPAC=X'00', SSNDPAC=X'00',
               PSERVIC=X'06020000000000000002C00'
```

SNA Definition

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*             ------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*             ------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING --> GTWTK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

When the option **DYNLU=YES** , it is not necessary to give an additional definition of the Lu 6.2 in VTAM (for the machines of the TR network).

Independent LU Definition

Despite what has been stated above, it may be worth defining an independent LU for the first communication tests:

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
IMS4349          LU      LOCADDR=0,
                 ISTATUS=ACTIVE,
                 DLOGMOD=LU62,
                 MODETAB=MTLU62
```

APPC/MVS Definitions

Prerequisite (minimum): MVS/ESA Version 4.2

Two **SYS1.PARMLIB** members are required to define the characteristics of the Local LU APPC/MVS and ASCH (APPC scheduler).

Local LU APPC/MVS Definition

- Parameterizing

```
BROWSE -- SYS1.PARMLIB(APPCPM00) - 01.12 ----------- LINE 0000
 COMMAND ===>
******************************** TOP OF DATA ****************
/*************************************************************/
/*  THE FOLLOWING PARAMETERS ARE FOR THE APPC ADDRESS SPACE. */
/*  THE APPC ADDRESS SPACE HANDLES THE ACTUAL COMMUNICATIONS.*/
/*                                                           */
/*  THESE MEMBERS PROVIDE THE LINKAGE BETWEEN LU NAMES AND   */
/*  TRANSACTION SCHEDULERS.                                  */
/*************************************************************/
LUADD
      ACBNAME(A01IMS62)  □ corresponds to APPL APPC/IMS of the VTAM
      SCHED(CGIB)
      BASE
      TPDATA(UTI.APPCTP)
      TPLEVEL(SYSTEM)
SIDEINFO DATASET(UTI.APPCSI)
```

- Startup

```
BROWSE -- SYS1.PROCLIB(APPC) ------------------------
 COMMAND ===>
******************************** TOP OF DATA ********
//APPC PROC APPC=00
//APPC EXEC PGM=ATBINITM,PARM='APPC=&APPC',REGION=0K
******************************** BOTTOM OF DATA ******
```

Scheduler APPC/MVS Definition

- Parameterizing

```
BROWSE -- SYS1.PARMLIB(ASCHPM00) - 01.00 -----------
 COMMAND ===>
******************************** TOP OF DATA ********
/*************************************************
/* THE FOLLOWING IS ADDED TO ENABLE APPC/MVS.
/*************************************************
 CLASSADD CLASSNAME(SVSAMP)
      MAX(10)
      MIN(2)
```

```
      RESPGOAL(0.02)
      MSGLIMIT(700)
 OPTIONS DEFAULT(SVSAMP)
      SUBSYS(JES2)
 TPDEFAULT REGION(4M)
      TIME(10,30)
      MSGLEVEL(1,1)
      OUTCLASS(R)
****************************** BOTTOM OF DATA ******
```

- Startup

```
BROWSE -- SYS1.PROCLIB(ASCH) -----------------------
 COMMAND ===>
******************************** TOP OF DATA ********
//ASCH PROC ASCH=00
//ASCH EXEC PGM=ASBSCHIN,PARM='ASCH=&ASCH',REGION=0K
****************************** BOTTOM OF DATA ******
```

IMS Definitions

Prerequisite (minimum): IMS 4.1 Version

IMSCTRL Macro

To generate IMS, it is necessary to use the version of the MVS/ESA libraries corresponding to the third parameter of the **SYSTEM** keyword. The minimum MVS/ESA version required is 4.2.

```
BROWSE -- EX.IMS410.SOURCE(STAGE1) - 01.28 -------------
 COMMAND ===>
*
* IMSCTRL MACRO --
*
         IMSCTRL  SYSTEM=(VS/2,CTLBLKS,4.2),
               DBRC=(YES,NO),
               DBRCNM=DBRC41,
               DLINM=DLISAS41,
               DCLWA=YES,
              IMSID=CGIB,             corresponds to parameter SCHED of
                                      the APPCPM00
               NAMECHK=(YES,S1),
               MAXIO=(,015),
               MAXREGN=(008,512K,A,A),
               MCS=(8),
               DESC=7,
               MAXCLAS=020
```

Startup

For IMS to be able to make a connection with APPC/MVS, it is necessary to specify **APPC=Y** in the IMS Startup Job (**DFSPBxxx** where **xxx** is the region suffix).

Transaction Definition

```
BROWSE -- EX.IMS410.SOURCE(STAGE1) - 01.28 ---------
 COMMAND ===>
******** TRANSACTION DB2 POUR BABY CLIENT (PTAB) *********************
APPLCTN  PSB=BABIVG
TRANSACT CODE=BABI,SEGSIZE=00000,MODE=SNGL,SEGNO=00000,
              PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
              MSGTYPE=(SNGLSEG,RESPONSE,4)
```

IMS Connect

Example of configuration file for IMS Connect

```
IMS.FRIMSCEC.PROCLIB(HWSCFG00)
*********************************************************************************************
HWS (ID=ITOC01,RACF=Y)
TCPIP (HOSTNAME=TCPIP,RACFID=RACF,PORTID=(4000),MAXSOC=300,TIMEOUT=00000,EXIT=(HWSIMSO0))
DATASTORE (ID=IMSC,GROUP=GPACMQ,MEMBER=HWSMEM,TMEMBER=FRIMSCEC)
*********************************************************************************************
```

Example of startup JCL for IMS Connect

```
IMS.FRIMSCEC.JOBS(IMSCTOC)
*********************************************************************************************
//IMSCTOC  JOB (FR9970,FRAIMSC,MM103,NAJT,JC),MSGCLASS=S,CLASS=A
//HWS      EXEC PGM=HWSHWS00,
//            PARM='BPECFG=BPECFG00,HWSCFG=HWSCFG00'
//STEPLIB  DD DSN=IMS.HWS.SHWSRESL,DISP=SHR
//         DD DSN=IMS.FRIMSCEC.RESLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.FRIMSCEC.PROCLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//HWSRCORD DD DSN=IMS.HWS.HWSRCDR,DISP=SHR
*********************************************************************************************
```

CICS CPI-C

MVS and CICS Configuration

VTAM Definitions

Prerequisite (minimum): VTAM Version 3.3

ATCSTR of the VTAM Definition

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC Address of TIC (LAN attachment) 3745 controller in NCP:

```
*********************************************
*            TIC BNN                              05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                      PORTADD=01,
                      LOCADD=400003172000,
                      ISTATUS=ACTIVE,
                      UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                      ADDR=01
*********************************************
```

CICS Definition

```
A01CICS1 APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=CICST,          APPLID FOR ACB                  *
               AUTH=(ACQ,VPACE,PASS),  CICS CAN ACQUIRE & PASS TMLS    *
*                                      CICS CAN REQUEST BLOCKED INPUT

               PARSESS=YES,                    Parallel Sessions Supports
               SONSCIP=YES,                                            *
               MODETAB=MTLU62                  name of the modes table
```

Mode Definition

- Definition of the characteristics for the LU6.2 Sessions.
- The SNASVCMG mode is used with the 'Parallel Sessions' support.

```
TITLE '--- "MODTABLE" RELATED TO THE LU 6.2 ---'
*
MTLU62   MODETAB
         SPACE 4
SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
               PSERVIC=X'060200000000000000000300'
LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
               TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
               COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
               PSNDPAC=X'00', SSNDPAC=X'00',
               PSERVIC=X'06020000000000000002C00'
```

SNA Definition

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*             -------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*             -------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING -->  GTWK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

With the option **DYNLU=YES** , it is not necessary to give an additional definition of the Lu 6.2 in VTAM (for the machines of the TR network).

Independent LU Definition

Despite what has been stated above, it may be worth defining an independent LU for the first communication tests:

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
CICSFBFB  LU      LOCADDR=0,
                  ISTATUS=ACTIVE,
                  DLOGMOD=LU62,
                  MODETAB=MTLU62
```

APPC/MVS Definitions

Prerequisite (minimum): MVS/ESA Version 4.2

There is no specific definition as we use the APPC layer delivered with the CICS version.

CICS Definitions

InterSystem Communication Parameter in the SIT Table

ISC=YES

Connection

```
 Connection     : SGFB
  Group         : GRPISC5
  DEscription   :
CONNECTION IDENTIFIERS
  Netname       : CICSFBFB     same declaration as Local LU in CM/2
                              (free code)
  INDsys        :
REMOTE ATTRIBUTES
  REMOTESystem  :
  REMOTEName    :
CONNECTION PROPERTIES
  ACcessmethod  : Vtam
  Protocol      : Appc
  SInglesess    : No           if independent LU
  DAtastream    : User
  RECordformat  : U
OPERATIONAL PROPERTIES
  AUtoconnect   : Yes
  INService     : Yes
SECURITY
  SEcurityname  : PTPD
  ATtachsec     : Verify      userid and password check at the
                              conversation
  BINDPassword  :
  BINDSecurity  : No
```

Session

```
 Sessions       : SESSIOFB
  Group         : GRPISC5
  DEscription   :
SESSION IDENTIFIERS
  Connection    : SGFB         code of the connection defined above
  SESSName      :
  NETnameq      :
  MOdename      : LU62         mode defined in the VTAM MODTABLE
SESSION PROPERTIES
  Protocol      : Appc
  MAximum       : 004 , 002
  RECEIVEPfx    :
  RECEIVECount  :
  SENDPfx       :
  SENDCount     :
  SENDSize      : 08192
  RECEIVESize   : 08192
  SESSPriority  : 000
  Transaction   :
OPERATOR DEFAULTS
  OPERId        :
  OPERPriority  : 000
  OPERRsl       : 0
  OPERSecurity  : 1
PRESET SECURITY
  USERId        :
OPERATIONAL PROPERTIES
  Autoconnect   : Yes
  INservice     :
  Buildchain    : Yes
  USERArealen   : 000
  IOarealen     : 00000 , 00000
  RELreq        : Yes
  DIscreq       : No
  NEPclass      : 000
RECOVERY
  RECOVOption   : Sysdefault
  RECOVNotify   : None
```

Transaction Definition

```
OBJECT CHARACTERISTICS                                  CICS RELEASE = 0330
   CEDB  View
    TRansaction     : VIC0              code to report in the TP name of
                                        the CPIC Side Information of CM/2
    Group           : VISUAL
    DEscription     :
    PROGram         : AVVMSV            code of the Communication Monitor
    TWasize         : 00000              0-32767
    PROFile         : DFHCICST
    PArtitionset    :
    STatus          : Enabled            Enabled ! Disabled
    PRIMedsize      : 00000              0-65520
    TASKDATALoc     : Below              Below ! Any
    TASKDATAKey     : User               User ! Cics
   REMOTE ATTRIBUTES
    DYnamic         : No                 No ! Yes
    REMOTESystem    :
    REMOTEName      :
    TRProf          :
    Localq          :                    No ! Yes
   SCHEDULING
    PRIOrity        : 001                0-255
    TClass          : No                 No ! 1-10
   ALIASES
    Alias           :
    TASKReq         :
    XTRanid         :
    TPName          :
                    :
    XTPname         :
                    :
                    :
    RECOVERY
    DTimout         : No                 No ! 1-6800
    Indoubt         : Backout            Backout ! Commit ! Wait
    RESTart         : No                 No ! Yes
    SPurge          : Yes                No ! Yes
    TPUrge          : Yes                No ! Yes
    DUmp            : Yes                Yes ! No
    TRACe           : Yes                Yes ! No
   SECURITY
    RESSec          : No                 No ! Yes
    Cmdsec          : No                 No ! Yes
    Extsec          : No                 No ! Yes
    TRANsec         : 01                 1-64
    RSl             : 00                 0-24 ! Public
```

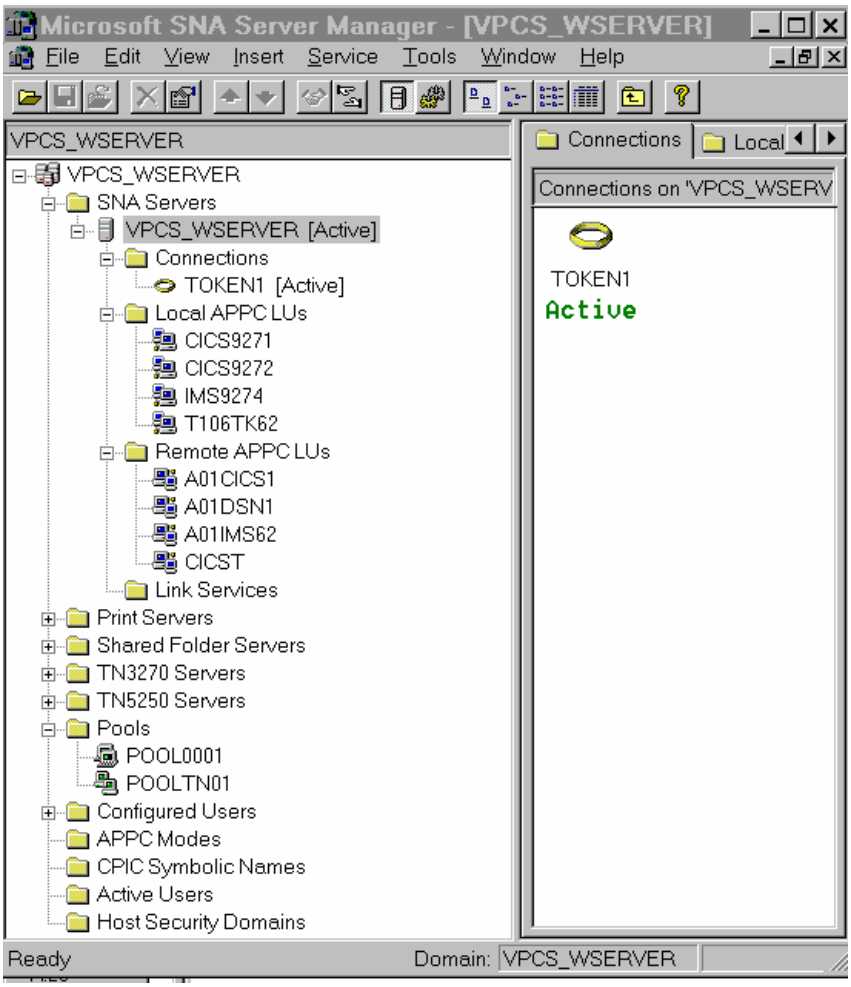
Additionally, if a transaction accesses a DB2 database, this database must be authorized and linked to the DB2 plan. The transaction code and the DB2 plan must therefore be declared in the RCT table. In this example, the transaction VIC0 and the DB2 plan ATDF are used by the Client application:

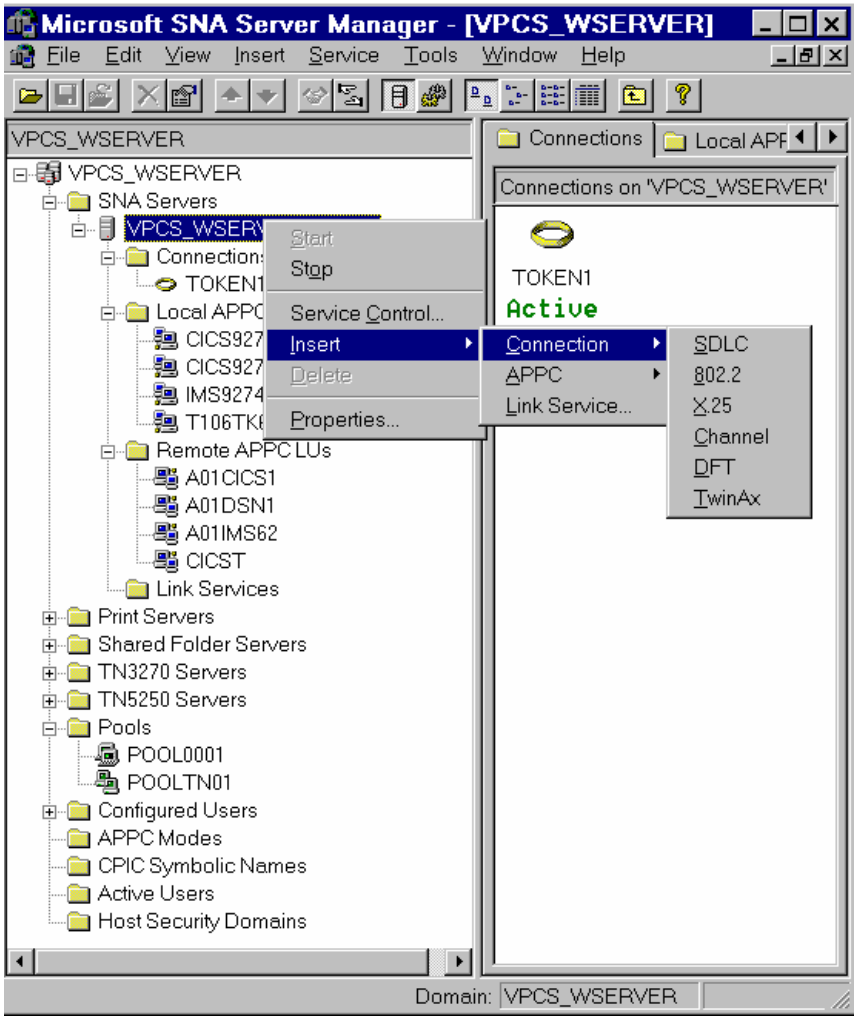
```
DSNRCT TYPE = ENTRY,TXID=(VIC0),
         THRDM=6,THRDA=6,PLAN=ATDF,AUTH=(USERID,*,*)
```

SNA Server 3.0A for Windows NT Configuration

(The Pack 2 Service must be applied for APPC corrections).

The SNA Server parameters must correspond to the VTAM, NCP and CICS definitions of the MVS central site.

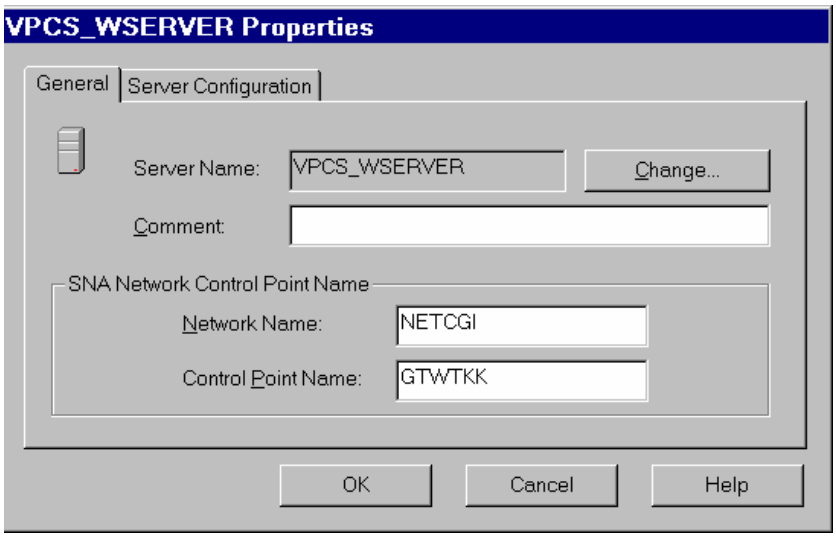

Microsoft SNA Server Manager - [VPCS_WSERVER]
File Edit View Insert Service Tools Window Help
VPCS_WSERVER
VPCS_WSERVER
SNA Servers
VPCS_WSERVER [Active]
Connections
TOKEN1 [Active]
Local APPC LUs
CICS9271
CICS9272
IMS9274
T106TK62
Remote APPC LUs
A01CICS1
A01DSN1
A01IMS62
CICST
Link Services
Print Servers
Shared Folder Servers
TN3270 Servers
TN5250 Servers
Pools
POOL0001
POOLTN01
Configured Users
APPC Modes
CPIC Symbolic Names
Active Users
Host Security Domains
Connections
Local
Connections on 'VPCS_WSERV
TOKEN1
Active
Ready
Domain: VPCS_WSERVER



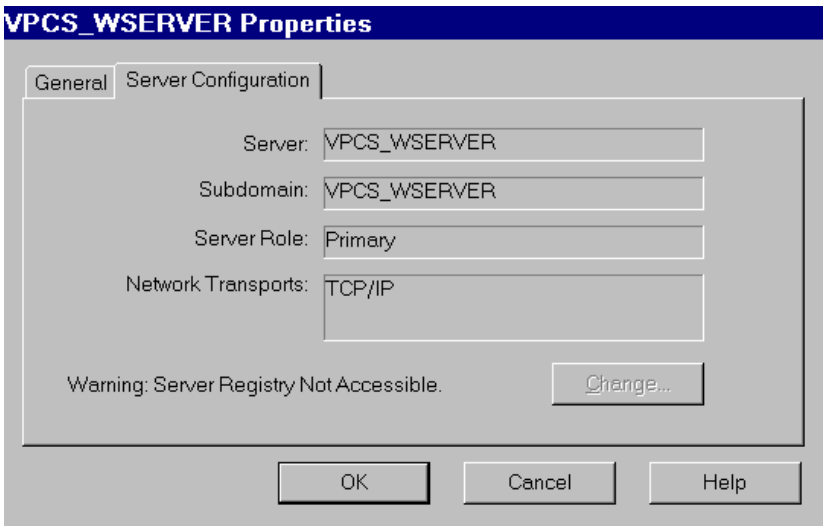
Server Properties

Network Name = identifier of the SNA network (**NETID** in **ATCSTRxx** of VTAM)

Control Point Name = corresponding to the **CPNAME** in the **PU** definition of VTAM



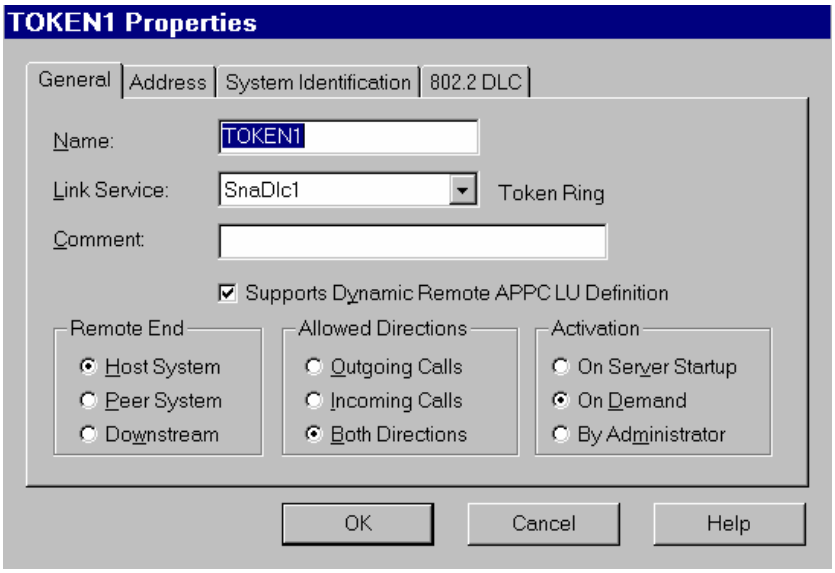
- Type of transport used between the Client and the SNA Server (example: TCP/IP protocol)



Link Properties

`Link Service`: The link type must be chosen at installation or installed later on using the Setup program. This is the component of the SNA Server which communicates with the network card driver. The SNA DLC 802.2 Link service is assigned to the communication with the central site in a LAN Token ring or Ethernet network.

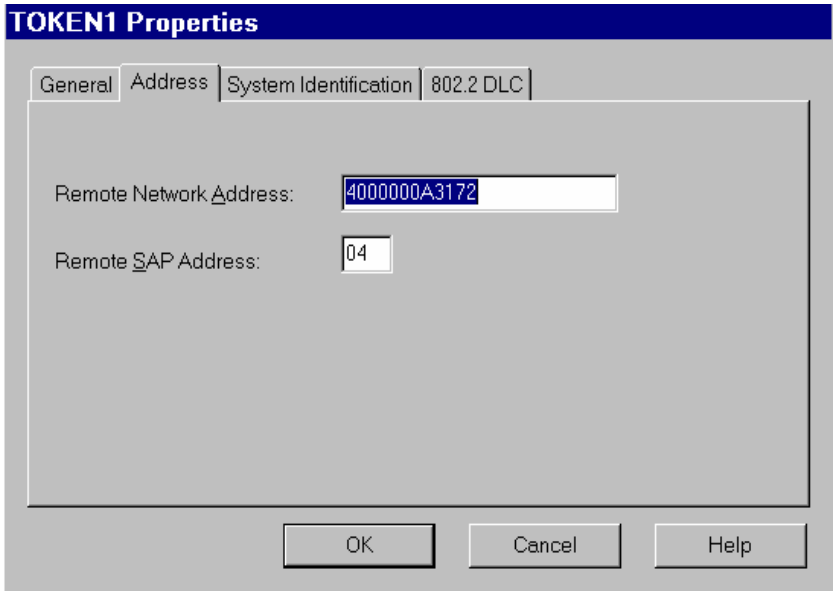
Connection type = 802.2



- Definition of the controller TIC 3745 in NCP (LAN attachment).

```
*********************************************
*            TIC BNN                                    05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
                    PORTADD=01,
                    LOCADD=4000000A3172,
                    ISTATUS=ACTIVE,
                           UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
                    ADDR=01
*********************************************
```

Remote Network Address = LOCADD (MAC Address) in the **LINE** macro of the **NCP**.



- **SSCPNAME** and **NETID** parameters of VTAM **ATCSTRxx** (required for the configuration of the Remote Node Name in SNA Server):

```
*****************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*****************************
```

- Definition of the **PU** in VTAM corresponding to the SNA Server gateway (used in the configuration of the Local Node Name for SNA Server):

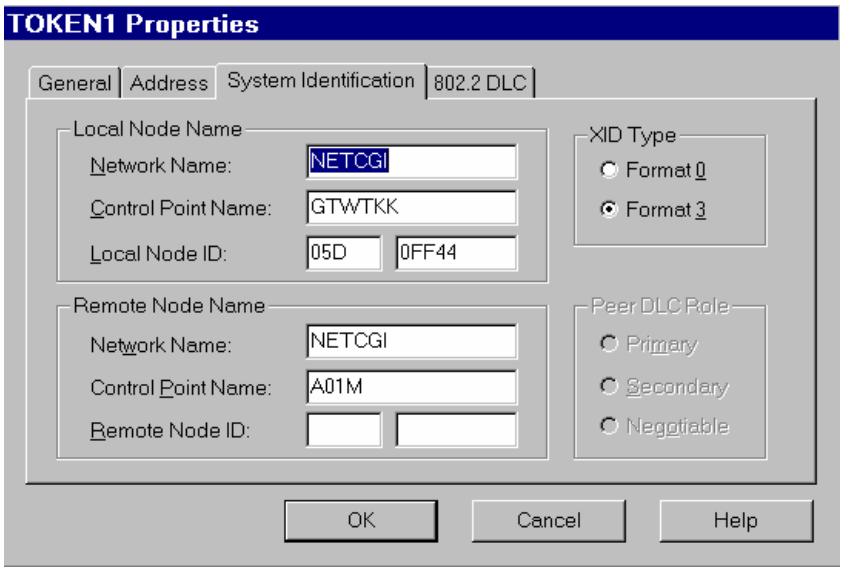
```
*******************************************
*/* SWITCHED MAJOR NODE                                  *
*******************************************
*
SW6TKR   VBUILD TYPE=SWNET,MAXNO=12,MAXGRP=06
*
W6TK00   PU    ADDR=55,
                     CPNAME=GTWTKK,
                     IDBLK=05D,
                     IDNUM=0FF44,
                     DYNLU=YES,
                     MAXPATH=1,
                     DISCNT=YES,
                     IRETRY=YES,
                     VPACING=7,
                     PACING=7,
                     SSCPFM=USSSCS,
                     USSTAB=USSTAB2,
                     MAXDATA=4096,
                     PUTYPE=2,
                     MAXOUT=7,
                     DATMODE=FULL
*
*   ==>       INDEPENDENT  LU
*
CICS9271 LU    LOCADDR=0,
                     ISTATUS=ACTIVE,
                     MODETAB=MTLU62,
                     DLOGMOD=LU62
*
************************************************
```

Local Node Name:

- **Local Node ID = IDBLK & IDNUM**
- **Control Point Name = CPNAME**
- **Network Name = NETID (ATCSTRxx)**

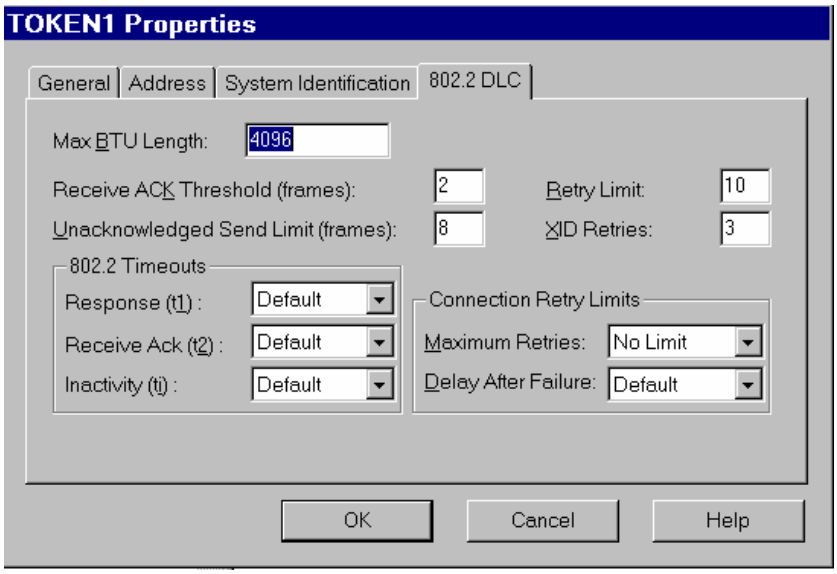
Remote Node Name:

- **Control Point Name = SSCPNAME**
- **Network Name = NETID (ATCSTRxx)**



Max BTU Length (frame size) corresponds to **MAXDATA** of the **PU** in VTAM.

- for Token ring adapter of 4 Mbps, must be smaller than or equal to 4195
- for Ethernet adapter, must be smaller than or equal to 1493.

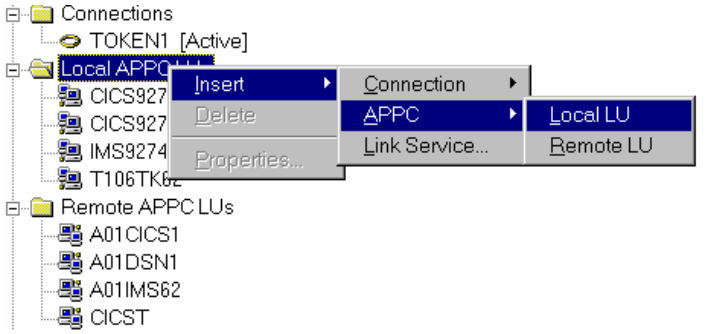


Local APPC LU

APPC uses a local LU (independent or dependent) and one or more remote LUs. The APPC sessions communicate between two LUs (Local and Remote).

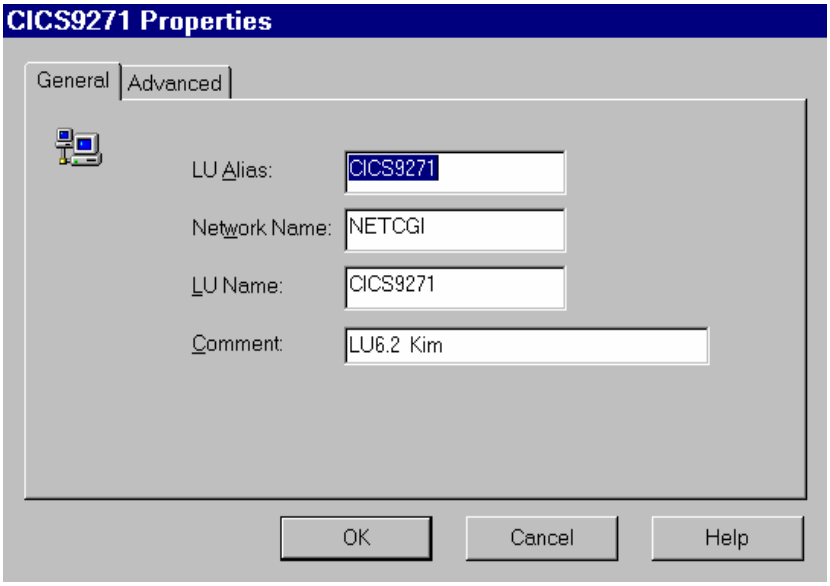
The minimum prerequisites for MVS to communicate with a TP (Transaction Program):

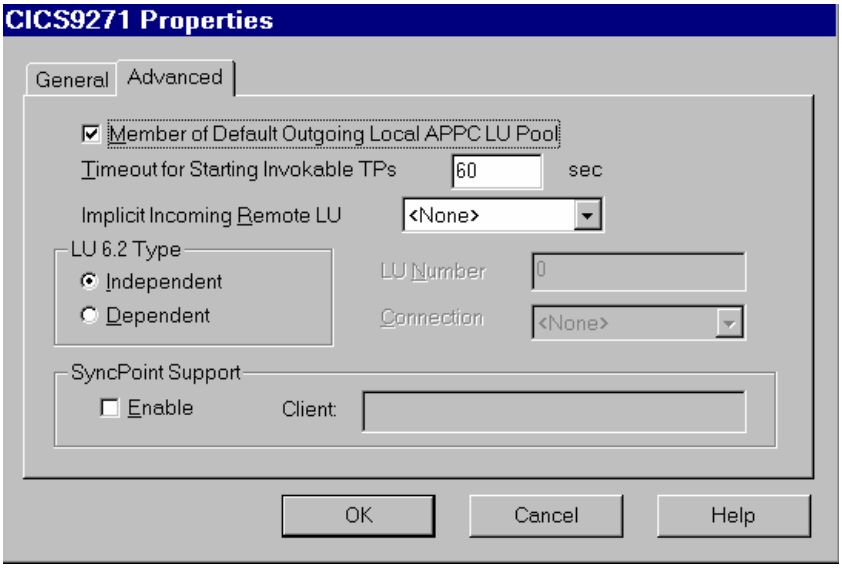
- VTAM version 3.2
- NCP version 5.2 (3745)
- NCP version 4.3 (3725)



Local LU is defined with the **PU** in VTAM as a independent LU with **LOCADDR = 0**:

- **LU Name = CICS9271**
- **Network Name = NETID in ATCSTRxx**
- **LU Alias =** Identifier for the local LU for TPs (Transaction Programs)





Remote APPC LU

- Definition of the CICS APPL in VTAM:

```
*********************************************************
*/*      THIS MEMBER CONTAINS VTAM APPLICATION DEFINITION
*/*      .      NAME         ACBNAME
*/*             ------------ -------------
*/*      .      A01CICS1     CICST
*********************************************************
A01CICS1 APPL  EAS=160,
                     ACBNAME=CICST,                APPLID FOR ACB
                     AUTH=(ACQ,VPACE,PASS),
                     PARSESS=YES,
                     SONSCIP=YES,
                     MODETAB=MTLU62
*********************************************************
```

- In the CICS SIT table, InterSystem Communication must be activated: **ISC=YES**

- Definition of the CONNECTION in CICS:

  Netname (CICS9271) corresponds to the Local APPC LU

```
------------------------------------------------------------------------
OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA  ALter
  Connection     : SG71
  Group          : GRPISC9
  DEscription  ==> CONNEXION LU6.2 SNA SERVER
 CONNECTION IDENTIFIERS
   Netname      ==> CICS9271
   INDsys       ==>
   REMOTE ATTRIBUTES
   REMOTESystem ==>
   REMOTEName   ==>
 CONNECTION PROPERTIES
 ACcessmethod ==> Vtam
 Protocol     ==> Appc
 SInglesess   ==> No
 DAtastream   ==> User
 RECordformat ==> U
 OPERATIONAL PROPERTIES
 AUtoconnect  ==> Yes
 INService    ==> Yes
 SECURITY
 SEcurityname ==> SYTD
 ATtachsec    ==> Verify
 BINDPassword ==>
 BINDSecurity ==> No
                                                    APPLID=CICST
 ------------------------------------------------------------------------
```

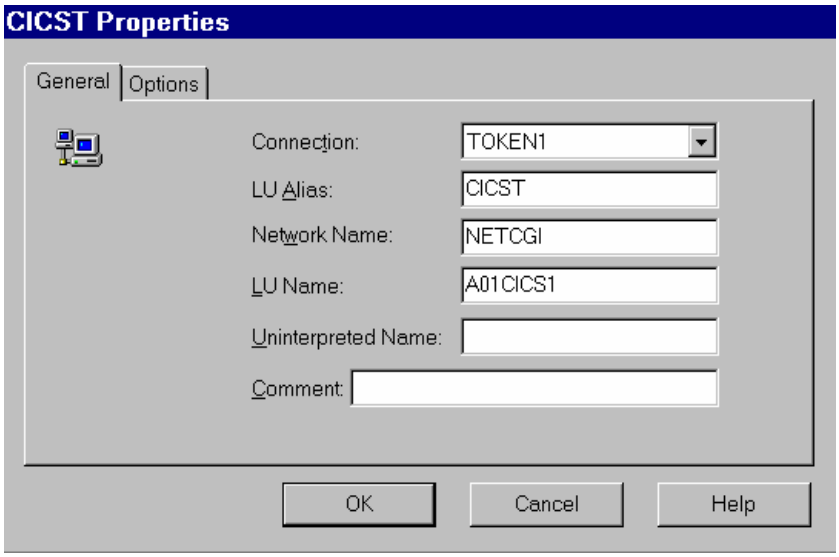
- Definition of the SESSION in CICS:
  - **Connection (SG71)** corresponds to the connection code defined above
  - **MOdename (LU62)** corresponds to the Mode name defined in SNA Server
  - **MAximum (004 , 002)** corresponds to the **Parallel Session Limit** parameters and to the **Partner Min Contention Winner** defined in the SNA Server Mode (LU62)

- **SENDSize** and **RECEIVESize** correspond to the **Max Receive RU Size** and **Max Send RU Size** parameters of the SNA Server Mode (LU62)

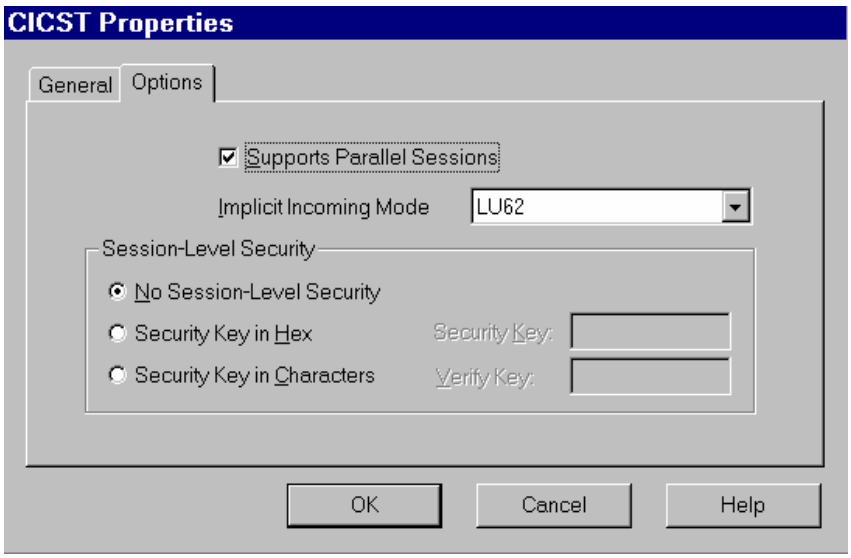
```
------------------------------------------------------------------------
OVERTYPE TO MODIFY                                CICS RELEASE = 0330
CEDA  ALter
Sessions       : SESSIO71
Group          : GRPISC9
DEscription  ==> SESSION LU6.2 SNA Server
SESSION IDENTIFIERS
Connection   ==> SG71
SESSName     ==>
NETnameq     ==>
MOdename     ==> LU62
SESSION PROPERTIES
Protocol     ==> Appc
MAximum      ==> 004 , 002
RECEIVEPfx   ==>
RECEIVECount ==>
SENDPfx      ==>
SENDCount    ==>
SENDSize     ==> 08192
RECEIVESize  ==> 08192
SESSPriority ==> 000
Transaction    :
OPERATOR DEFAULTS
OPERId         :
OPERPriority   : 000
OPERRsl        : 0
OPERSecurity   : 1
PRESET SECURITY
USERId       ==>
OPERATIONAL PROPERTIES
Autoconnect  ==> Yes
INservice      :
Buildchain   ==> Yes
USERArealen  ==> 000
IOarealen    ==> 00000 , 00000
RELreq       ==> Yes
DIscreq      ==> No
NEPclass     ==> 000
RECOVERY
RECOVOption  ==> Sysdefault
RECOVNotify  ==> None

APPLID=CICST
------------------------------------------------------------------------
```

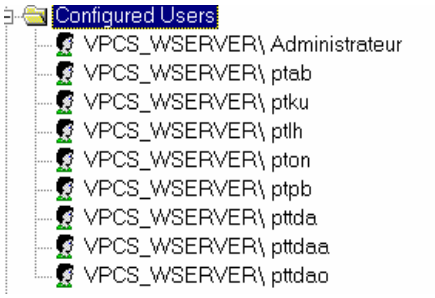
- Definition of the Remote APPC LU in SNA Server:
  - **Network Name** = **NETID** in **ATCSTRxx**
  - **LU Alias** = Identifier of the LU in local TPs (Transaction Programs)
  - **LU Name** = **APPL ID CICS** in VTAM (**A01CICS1**)

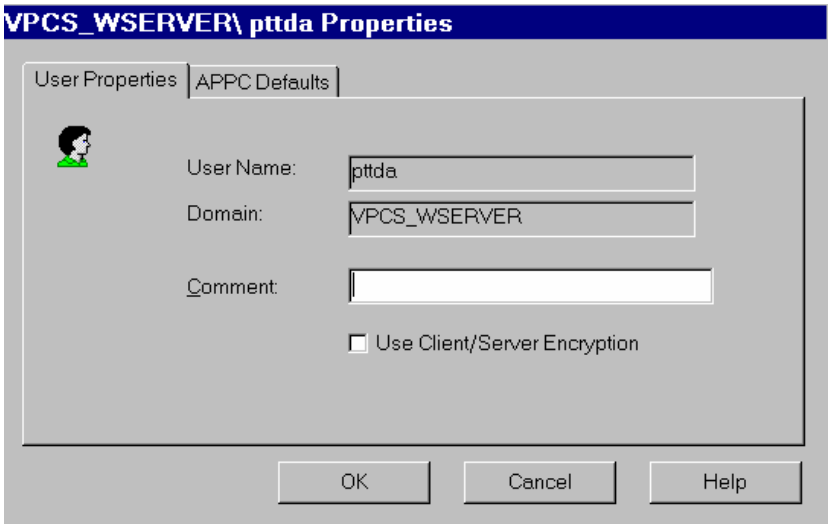


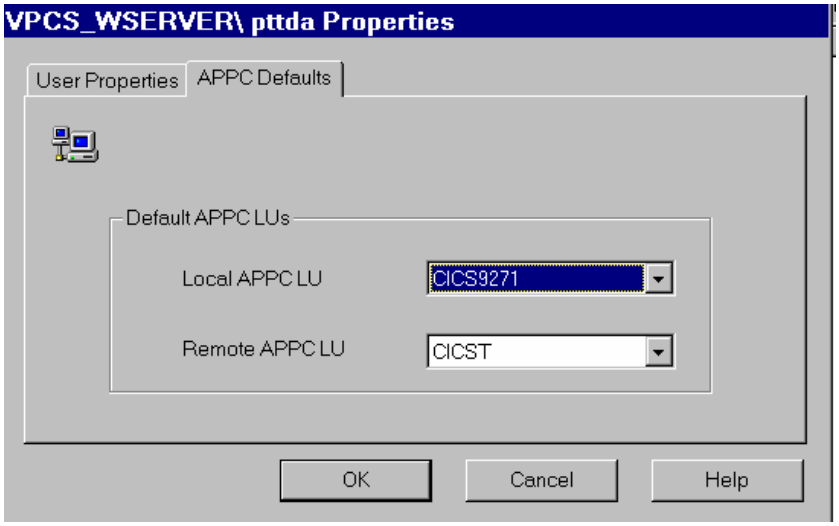
- **`Implicit Incoming Mode = Mode Name`** used for Supports Parallel Sessions and defined in the APPC Modes in SNA Server



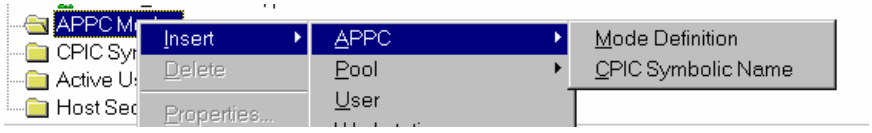
Configuration of Users







Definition of the MODE



- Definition of the LU6.2 MODES in the Modes table in VTAM:

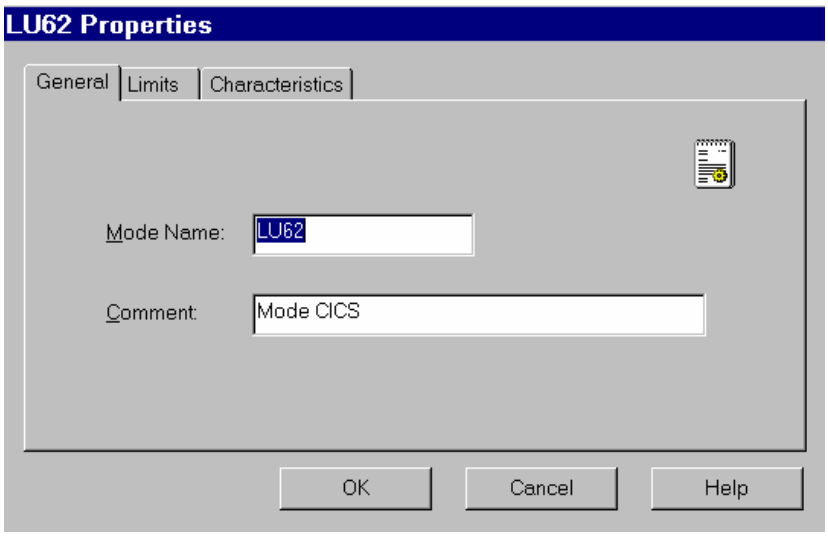
  The LU62 Mode is an example of the configured and used mode with APPC (LU6.2). The SNASVCMG mode is included in SNA Server and is used by the 'Supports Parallel Sessions'.

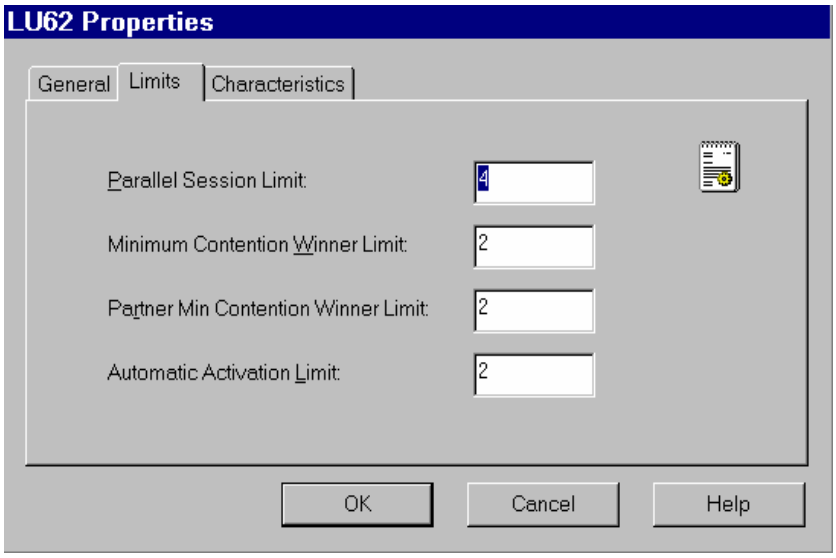
```
------------------------------------------------------------------------
*
***     TITLE '--- "MODTABLE" RELATED TO THE LU 6.2 ---'    *******
*
SNASVCMG MODEENT LOGMODE=SNASVCMG,
               FMPROF=X'13',
                TSPROF=X'07',
               PRIPROT=X'B0',
                SECPROT=X'B0',
                COMPROT=X'D0B1',
                RUSIZES=X'8585',
```

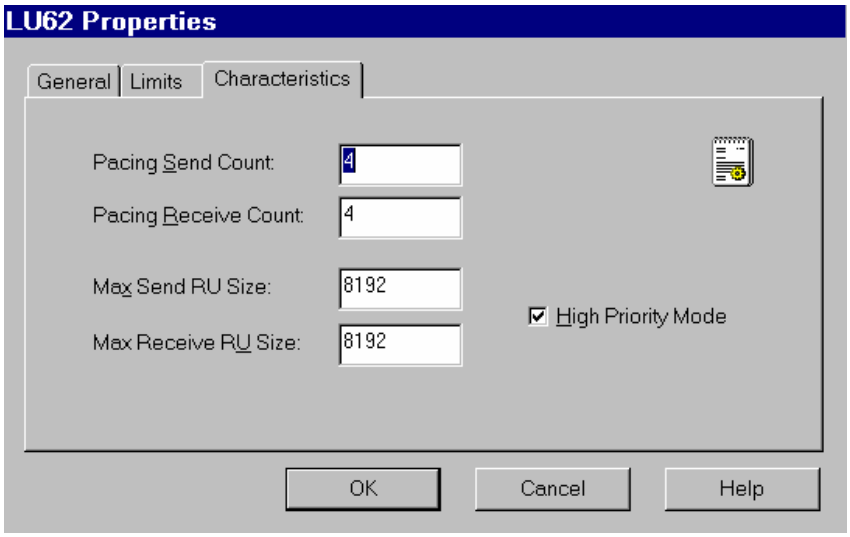
```
                ENCR=B'0000',
                PSERVIC=X'060200000000000000000300'
*
LU62      MODEENT LOGMODE=LU62,
                TYPE=X'00',
                FMPROF=X'13',
                TSPROF=X'07',
                PRIPROT=X'B0',
                SECPROT=X'B0',
                COMPROT=X'50B1',
                RUSIZES=X'8989',
                SRCVPAC=X'00',
                PSNDPAC=X'00',
                SSNDPAC=X'00',
                PSERVIC=X'06020000000000000002C00'
------------------------------------------------------------------------
```

- Definition of the Mode in SNA Server:

  Mode Name = Mode defined in the modes table in VTAM



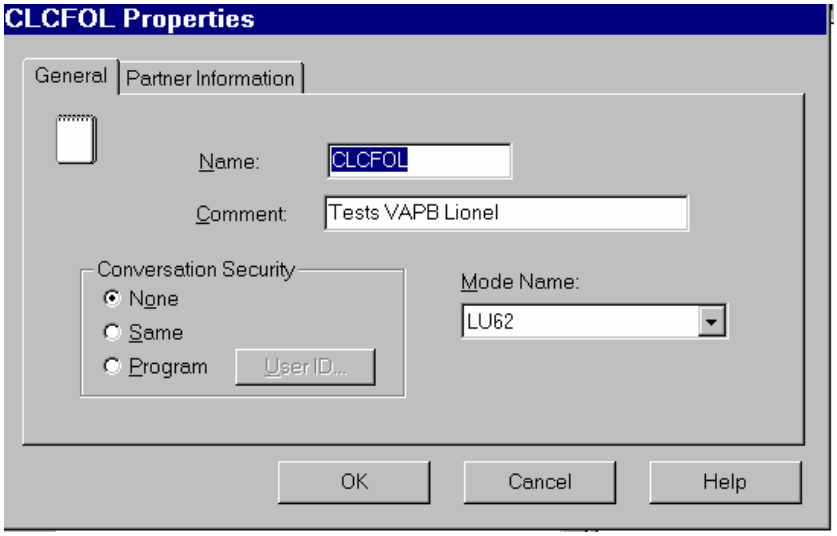




Definition of the CPI-C Symbolic Destinations Names (Side Information)

CPI-C is the interface of the communication API. It provides a standard group of function calls for all the APPC platforms that handle CPI-C.

Name corresponds to the external name of the VA Pac Communication Monitor (makes the connection between the Clients and the Server programs using APPC).



- Definition of the CICS transaction:

```
--------------------------------------------------------------------------------
OVERTYPE TO MODIFY                                      CICS RELEASE = 0330
CEDA  ALter
TRansaction     : VP20
Group           : VISUAL
DEscription  ==> VISUAL/CLIENT WIN/NT CPIC
PROGram      ==> CLCFOL
TWasize      ==> 00000
PROFile      ==> DFHCICST
PArtitionset ==>
STatus       ==> Enabled
PRIMedsize      : 00000
TASKDATALoc  ==> Below
```

```
TASKDATAKey  ==> User
REMOTE ATTRIBUTES
DYnamic      ==> No
REMOTESystem ==>
REMOTEName   ==>
TRProf       ==>
Localq       ==>
SCHEDULING
PRIOrity     ==> 001
TClass       ==> No
ALIASES
Alias        ==>
TASKReq      ==>
XTRanid      ==>
TPName       ==>
       ==>
 XTPname      ==>
                      ==>
                      ==>
RECOVERY
DTimout      ==> No
Indoubt      ==> Backout
RESTart      ==> No
SPurge       ==> No
TPUrge       ==> No
DUmp         ==> Yes
TRACe        ==> Yes
SECURITY
RESSec       ==> No
Cmdsec       ==> No
Extsec         : No
TRANsec        : 01
RSl            : 00

                                                                APPLID=CICST
-------------------------------------------------------------------------------
```

If a program uses DB2 during the transaction, the DB2 plan must be linked to the transaction of the Server monitor. The transaction must be declared in the RCT table of CICS/ESA:

```
DSNCRCT TYPE=ENTRY,TXID=(VP20),THRDM=2,
THRDA=2,PLAN=VP20,AUTH=(USERID,*,*)
```

If the application accesses DB2, the user transaction must be authorized and linked to the DB2 plan in the RCT table of the CICS area:

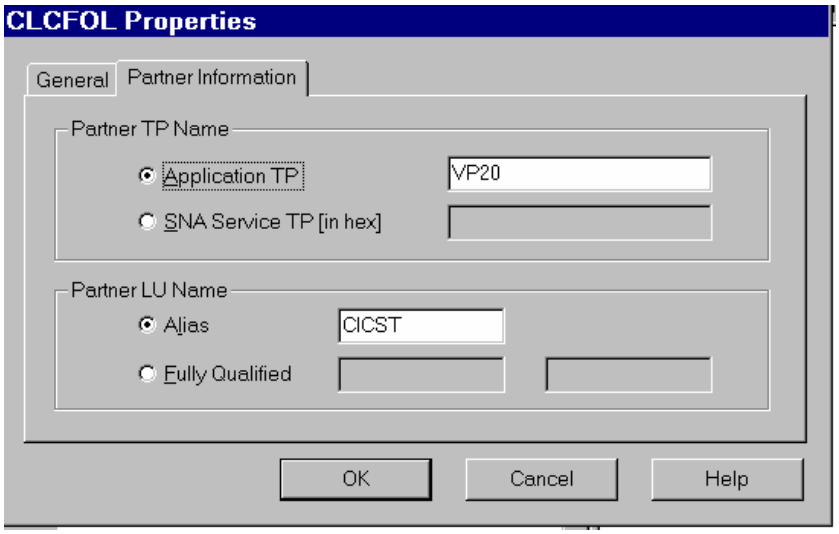
```
DSNCRCT TYPE=ENTRY,TXID=(VP20),THRDM=6,
THRDA=6,PLAN=VP20,AUTH=(USERID,*,*)
```

- Definition of **Partner TP Name**:

  Application TP = Transaction code activating the Communication Monitor on the Host and defined above

- Definition of **Partner LU Name**:

  Alias = Alias defined in Remote APPC LU


CLCFOL Properties
General
Partner Information
Partner TP Name
Application TP
VP20
SNA Service TP [in hex]
Partner LU Name
Alias
CICST
Fully Qualified
OK
Cancel
Help

CICS ECI

MVS and CICS Configuration

VTAM Definitions

Prerequisites (minimum): VTAM Version 3.3

Definition of the ATCSTR of the VTAM

```
*******************************
NOPROMPT,CONFIG=00,SSCPID=01,
MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=A01M,
SSCPORD=DEFINED,
NETID=NETCGI,
HOSTSA=1,
CRPLBUF=(550,,20,,40,40),
IOBUF=(420,182,25,,40,40),
LFBUF=(300,,0,,20,10),
LPBUF=(50,,0,,5,5),
SFBUF=(50,,0,,5,5),
SPBUF=(90,,0,,5,5),
NOTRACE,TYPE=VTAM
*******************************
```

MAC Address of the TIC (LAN attachment) 3745 controller in NCP:

```
*********************************************
*            TIC BNN                                  05742090
*********************************************
A01L1TK  LINE ADDRESS=(1088,FULL),
              PORTADD=01,
              LOCADD=400003172000,
              ISTATUS=ACTIVE,
              UACB=(X$P1AX,X$P1AR)
*
A01TK1PU PU ISTATUS=ACTIVE,
               ADDR=01
*********************************************
```

CICS Definition

```
A01CICS1 APPL  EAS=160,                ESTIMATED CONCURRENT SESSIONS   *
               ACBNAME=CICST,          APPLID FOR ACB                  *
               AUTH=(ACQ,VPACE,PASS),  CICS CAN ACQUIRE & PASS TMLS    *
               PARSESS=YES,                   Supports parallel Sessions
               SONSCIP=YES,                                           *
               MODETAB=MTLU62                 name of the modes table
```

Mode Definition

- Definition of the characteristics for the LU6.2 Sessions
- The SNASVCMG Mode is used with the 'Parallel Sessions' support.

```
 TITLE '--- "MODTABLE" RELATED TO THE LU 6.2 ---'
 *
 MTLU62   MODETAB
          SPACE 4
 SNASVCMG MODEENT LOGMODE=SNASVCMG, FMPROF=X'13',
                TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
                COMPROT=X'D0B1', RUSIZES=X'8585', ENCR=B'0000',
                PSERVIC=X'060200000000000000000300'
 LU62     MODEENT LOGMODE=LU62, TYPE=X'00', FMPROF=X'13',
                TSPROF=X'07', PRIPROT=X'B0', SECPROT=X'B0',
                COMPROT=X'50B1', RUSIZES=X'8787', SRCVPAC=X'00',
```

```
                PSNDPAC=X'00', SSNDPAC=X'00',
                PSERVIC=X'060200000000000000002C00'
```

SNA Definition

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*   --->    LIEN XCA MAJOR NODE ==> XCA1TKR  (IBM3172-3)
*/*   --->    LIEN GROUPE XCA     ==> GRP02
*             -------------------------------
*    - MODEL FOR IDBLK X'05D' - OS/2 COMMUNICATIONS MANAGER
*             -------------------------------
*
SW1TKR   VBUILD TYPE=SWNET,MAXNO=99,MAXGRP=10
*
*          ----------------------------------------
* -------> DEFINING A GATEWAY TOKEN-RING --> GTWTK1  <--------
*          ----------------------------------------
W1TK00   PU    ADDR=50,
               CPNAME=GTWTK1,
               IDBLK=05D,
               IDNUM=00002,
               DYNLU=YES,
               MAXPATH=1,
               DISCNT=NO,
               IRETRY=YES,
               VPACING=7,
               PACING=7,
               SSCPFM=USSSCS,
               MAXDATA=4096,
               PUTYPE=2
```

The **DYNLU=YES** option allows to avoid Lu 6.2 definitions at the VTAM level (for machines on the TR network).

Definition of an independent LU

Despite what has been stated above, it may be worth defining an independent LU for the first communication tests:

```
*/* LIB: SYS1.VTAMLST(SW1TKR)
*/*
*/*
*/* SWITCHED MAJOR NODE TOKEN-RING ST-MARC :            -  06/07/95.
*/*
*/*
*  INDEPENDENT LUS
*
CGI5075          LU      LOCADDR=0,
                 ISTATUS=ACTIVE,
                 DLOGMOD=LU62,
                 MODETAB=MTLU62
```

APPC/MVS Definitions

Prerequisite (minimum): MVS/ESA Version 4.2

There is no specific definition as we use the APPC layer delivered with the CICS version.

CICS Definitions

Prerequisite (minimum): CICS/ESA Version 4.1

InterSystem Communication Parameter in the SIT table

ISC=YES

Connection

```
 Connection     : SGFB
 Group          : GRPISC5
 DEscription    :
CONNECTION IDENTIFIERS
 Netname        : CGI5075       same declaration as Local LU in the CM/2
                                (free code)
 INDsys         :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
CONNECTION PROPERTIES
 ACcessmethod   : Vtam
 Protocol       : Appc
 SInglesess     : No            if independent LU
 DAtastream     : User
 RECordformat   : U
OPERATIONAL PROPERTIES
 AUtoconnect    : Yes
 INService      : Yes
SECURITY
 SEcurityname   : PTPD
 ATtachsec      : Verify        Check the userid and password at the
                                conversation
 BINDPassword   :
 BINDSecurity   : No
```

Session

```
 Sessions       : SESSIOFB
 Group          : GRPISC5
 DEscription    :
SESSION IDENTIFIERS
 Connection     : SGFB          code of the connection defined above
 SESSName       :
 NETnameq       :
 MOdename       : LU62          mode defined in the VTAM MODTABLE
SESSION PROPERTIES
 Protocol       : Appc
 MAximum        : 004 , 002
 RECEIVEPfx     :
 RECEIVECount   :
 SENDPfx        :
 SENDCount      :
 SENDSize       : 08192
 RECEIVESize    : 08192
 SESSPriority   : 000
 Transaction    :
OPERATOR DEFAULTS
 OPERId         :
 OPERPriority   : 000
 OPERRsl        : 0
 OPERSecurity   : 1
PRESET SECURITY
 USERId         :
OPERATIONAL PROPERTIES
 Autoconnect    : Yes
 INservice      :
 Buildchain     : Yes
 USERArealen    : 000
 IOarealen      : 00000 , 00000
 RELreq         : Yes
 DIscreq        : No
 NEPclass       : 000
RECOVERY
 RECOVOption    : Sysdefault
 RECOVNotify    : None
```

Transaction Definition

Definition of a user mirror transaction

```
------------------------------------------------------
                                      CICS RELEASE = 0410
   TRANSaction     : VEC1
   Group           : VPCS250
   DEscription     :
   PROGram         : DFHMIRS
   TWasize         : 00000              0-32767
   PROFile         : DFHCICSA
   PArtitionset    :
   STAtus          : Enabled            Enabled ! Disabled
   PRIMedsize      : 00000              0-65520
   TASKDATALoc     : Below              Below   ! Any
   TASKDATAKey     : User               User    ! Cics
   STOrageclear    : No                 No      ! Yes
   RUnaway         : System             System  ! 0-2700000
   SHutdown        : Disabled           Disabled ! Enabled
   ISolate         : Yes                Yes     ! No
  REMOTE ATTRIBUTES
+  DYnamic         : No                 No ! Yes
REMOTESystem    :
 REMOTEName     :
 TRProf         :
 Localq         :                    No ! Yes
SCHEDULING
 PRIOrity       : 001                0-255
 TClass         : No                 No ! 1-10
 TRANClass      : DFHTCL00
ALIASES
 Alias          :
 TASKReq        :
 XTRanid        :
 TPName         :
                :
 XTPname        :
                :
                :
RECOVERY
  DTimout        : No                 No ! 1-6800
  INdoubt        : Backout            Backout ! Commit ! Wait
  RESTart        : No                 No ! Yes
  SPurge         : No                 No ! Yes
  TPUrge         : No                 No ! Yes
  DUmp           : Yes                Yes ! No
  TRACe          : Yes                Yes ! No
  COnfdata       : No                 No ! Yes
 SECURITY
  RESSec         : Yes                No ! Yes
  CMdsec         : Yes                No ! Yes
  Extsec         : No
  TRANSec        : 01                 1-64
  RSl            : 00                 0-24 ! Public
------------------------------------------------------------
```

Moreover, if a transaction accesses a DB2 database, this database must be authorized and linked to the DB2 plan. The transaction code and the DB2 plan must therefore be declared in the RCT Table. By default, the CPMI transaction (Mirror Transaction) is used by the Client CICS. In this example, the DB2 plan used by the Client application is ATDF:

```
DSNRCT TYPE = ENTRY,TXID=(CPMI,CSPM,VICL),
         THRDM=2,THRDA=2,PLAN=ATDF,AUTH=(USERID,*,*)
```

If the CPMI mirror transaction causes an Abend ACN1, it means that the DFHCNV conversion table is not defined in the CICS area. It is required to use this transaction. Define this table, then assemble and LinkEdit using the following parameters:

```
*********************************************************************
//EXBCCNV  JOB (009),'BC',CLASS=X,MSGCLASS=X,NOTIFY=SYTD
//CICSCNV  EXEC DFHAUPLE,
// PARM.LNKEDT='RENT,REUS,LIST,XREF,LET,NCAL,AMODE=31,RMODE=ANY'
//ASM.SYSLIB DD DSN=CICS330.SDFHMAC,DISP=SHR
//ASSEM.SYSUT1 DD DSN=PT$EXP.CICST330.SOURCE(DFHCNV),DISP=SHR
//LNKEDT.SYSLMOD DD DSN=PT$PDV.PB80204.MTR8,DISP=SHR

DFHCNV   TYPE=INITIAL
DFHCNV   TYPE=ENTRY,RTYPE=PC,RNAME=TESTVP,CLINTCP=(850,437),  *
         SRVERCP=297
DFHCNV   TYPE=SELECT,OPTION=DEFAULT
DFHCNV   TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=8051,  *ES
         LAST=YES
DFHCNV   TYPE=FINAL
END
*********************************************************************
```

CICS TCP/IP Sockets Interface

CICS TCP/IP Configuration

Prerequisites

MVS/ESA:

TCP/IP Version 3, Release 1

CICS/ESA Version 3, Release 3

CICS TCP/IP Socket Interface Version 3.1

CICS Startup

Modification of the startup JCL for the CICS area

```
------------------------------------------------------------------------
//PMTCICST  JOB (008),'CICS TEST PAC',MSGLEVEL=(2,0),CLASS=O,
//   MSGCLASS=X
//CICST    PROC INDEX=CICS330,
//             UTINDX='PT$EXP.CICST330',
//             REGSZE=6M,
//             START='COLD',
//             SIP=T,
//DFHRPL   DD  DSN=&UTINDX..LNK,DISP=SHR
//                      DD  DSN=SYS1.TCPIP310.SEZALINK,DISP=SHR
//                      DD  DSN=TCPIP310.SEZATCP,DISP=SHR
//TCPDATA   DD SYSOUT=&OUTC,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
------------------------------------------------------------------------
```

Definition of CICS TCP/IP transactions

- Listener Task

```
-----------------------------------------------------------------
TRansaction    : CSKL
  Group          : TCPIPI
  DEscription    : Listener Task
  PROGram        : EZACIC02
  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
-----------------------------------------------------------------
```

*TCP/IP Version 3.1.0

- Enable the Socket Interface

```
-----------------------------------------------------------------
TRansaction    : CSKE
  Group          : TCPIPI
  DEscription    : Enable Sockets Interface
  PROGram        : EZACIC00
  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
```

```
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
--------------------------------------------------------------------
```

- Terminate the socket interface

```
--------------------------------------------------------------------
TRansaction    : CSKD
   Group          : TCPIPI
   DEscription    : Disable Sockets Interface
   PROGram        : EZACIC00
   TWasize        : 00000
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled
   PRIMedsize     : 00000
   TASKDATALoc    : Below
   TASKDATAKey    : Cics
  REMOTE ATTRIBUTES
   DYnamic        : No
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
+  Localq         :
--------------------------------------------------------------------
```

*TCP/IP Version 3.2.0

- Configure the socket interface

```
--------------------------------------------------------------------
TRansaction    : EZAC
  Group          : TCPIPI
  DEscription    : CONFIGURE SOCKETS INTERFACE
  PROGram        : EZACIC23

  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
  Localq         :
--------------------------------------------------------------------
```

- Enable the socket interface

```
--------------------------------------------------------------------
TRansaction    : EZAO
  Group          : TCPIPI
  DEscription    : ENABLE SOCKETS INTERFACE
  PROGram        : EZACIC00

  TWasize        : 00000
  PROFile        : DFHCICST
  PArtitionset   :
  STatus         : Enabled
  PRIMedsize     : 00000
  TASKDATALoc    : Below
  TASKDATAKey    : Cics
 REMOTE ATTRIBUTES
  DYnamic        : No
  REMOTESystem   :
  REMOTEName     :
  TRProf         :
```

```
  Localq         :
-----------------------------------------------------------------
```

- Terminate the socket interface

```
-----------------------------------------------------------------
TRansaction    : EZAP
   Group          : TCPIPI
   DEscription    : DISABLE SOCKETS INTERFACE
   PROGram        : EZACIC22
   TWasize        : 00000
   PROFile        : DFHCICST
   PArtitionset   :
   STatus         : Enabled
   PRIMedsize     : 00000
   TASKDATALoc    : Below
   TASKDATAKey    : Cics
  REMOTE ATTRIBUTES
   DYnamic        : No
   REMOTESystem   :
   REMOTEName     :
   TRProf         :
-----------------------------------------------------------------
```

Definition of CICS TCP/IP programs

```
-------------------------------------------------------------------------------
PROGram        : EZACIC00
    Group          : TCPIPI
    DEscription    : Connection Manager
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
      Transid        :
  EXECUtionset   : Fullapi
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
PROGram        : EZACIC02
   Group          : TCPIPI
   DEscription    : Listener
   Language       : Assembler
   RELoad         : No
   RESident       : Yes
   USAge          : Normal
   USElpacopy     : No
   Status         : Enabled
   RSl            : 00
   Cedf           : Yes
   DAtalocation   : Any
   EXECKey        : CICS
  REMOTE ATTRIBUTES
   REMOTESystem   :
+  REMOTEName     :
Transid        :
  EXECUtionset   : Fullapi
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Mapset         : EZACICM
   Group          : TCPIPI
   Description    : Mapset for Connection Manager
```

```
   REsident       : No
   USAge          : Transient
   USElpacopy     : No
   Status         : Enabled
   RSl            : 00
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC01
    Group          : TCPIPI
    DEscription    : Task Related User Exit
    Language       : Assembler
    RELoad         : No
    RESident       : Yes
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC20
    Group          : TCPIPI
    DEscription    : Initialization/termination for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC21
    Group          : TCPIPI
    DEscription    : Initialization Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACIC22
    Group          : TCPIPI
    DEscription    : Termination Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
```

```
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC23
    Group          : TCPIPI
    DEscription    : Primary Module for Transaction EZAC
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : User
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC24
    Group          : TCPIPI
    DEscription    : Message Delivery Module for CICS Sockets
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Transient
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
------------------------------------------------------------------------------

------------------------------------------------------------------------------
PROGram        : EZACIC25
    Group          : TCPIPI
    DEscription    : Cache Module for the Domain Name Server
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : User
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
```

```
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
PROGram        : EZACICME
    Group          : TCPIPI
    DEscription    : US English Text Delivery Module
    Language       : Assembler
    RELoad         : No
    RESident       : No
    USAge          : Normal
    USElpacopy     : No
    Status         : Enabled
    RSl            : 00
    Cedf           : Yes
    DAtalocation   : Any
    EXECKey        : CICS
   REMOTE ATTRIBUTES
    REMOTESystem   :
 +  REMOTEName     :
Transid        :
 EXECUtionset   : Fullapi
--------------------------------------------------------------------------------
```

Definition of the DCT Table

Definition of a TCPM transitional data queue for the listener, in the DCT table.

```
--------------------------------------------------------------------------------
TCPDATA  DFHDCT TYPE=SDSCI,        TCP/IP OUTPUT
                BLKSIZE=136,
                BUFNO=1,
                DSCNAME=TCPDATA,
                RECSIZE=132,
                RECFORM=VARUNBA,
                TYPEFLE=OUTPUT
*
TCPM     DFHDCT TYPE=EXTRA,        USED FOR MESSAGES - SEE
                DESTID=TCPM,                  INDDEST=TCPM BELOW
                DSCNAME=TCPDATA
*
TCPIN   DFHDCT TYPE=INTRA,         TCP/IP
                DESTID=TRAA,
                DESTFAC=FILE,
                TRIGLEV=1,
                TRANSID=TRAA
--------------------------------------------------------------------------------
```

Definitions and initializations of Configuration files (*TCP/IP Version 3.2.0)

EZACONFG: CICS Sockets configuration file

```
--------------------------------------------------------------------------------
File           : EZACONFG
 Group          : SOCKETS
 DEScription    : CICS SOCKETS CONFIGURATION FILE
VSAM PARAMETERS
 DSNAme         : CICS.STM9.SOCKETS.CFG
 Password       :                    PASSWORD NOT SPECIFIED
 Lsrpoolid      : 1                  1-8 ! None
 DSNSharing     : Allreqs            Allreqs ! Modifyreqs
 STRings        : 001                1-255
 Nsrgroup       :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
 RECORDSize     :                    1-32767
 Keylength      :                    1-255
INITIAL STATUS
 STAtus         : Enabled            Enabled ! Disabled ! Unenabled
Opentime       : Startup           Firstref ! Startup
 DIsposition    : Share              Share ! Old
BUFFERS
 DAtabuffers    : 00002              2-32767
 Indexbuffers   : 00001              1-32767
```

```
DATATABLE PARAMETERS
 Table          : No                No ! Cics ! User
 Maxnumrecs     :                   16-16777215
DATA FORMAT
 RECORDFormat   : V                 V ! F
OPERATIONS
 Add            : No                No ! Yes
 BRowse         : Yes               No ! Yes
 DELete         : No                No ! Yes
 REAd           : Yes               Yes ! No
 Update         : No                No ! Yes
AUTO JOURNALLING
JOurnal         : No                No ! 1-99
 JNLRead        : None              None ! Updateonly ! Readonly ! All
 JNLSYNCRead    : No                No ! Yes
 JNLUpdate      : No                No ! Yes
 JNLAdd         : None              None ! Before ! AFter ! ALl
 JNLSYNCWrite   : No                Yes ! No
RECOVERY PARAMETERS
 RECOVery       : None              None ! Backoutonly ! All
 Fwdrecovlog    : No                No ! 1-99
 BAckuptype     : Static            Static ! Dynamic
SECURITY
 RESsecnum      : 00                0-24 ! Public
--------------------------------------------------------------------------
```

EZACACHE: File required for the Domain Name Server Cache function

```
-------------------------------------------------------------------------
File           : EZACACHE
 Group          : SOCKETS
 DEScription    : DOMAIN NAME SERVER CACHE CONFIGURATION FILE
VSAM PARAMETERS
 DSNAme         : CICS.STM9.SOCKETS.EZACACHE
 Password       :                   PASSWORD NOT SPECIFIED
 Lsrpoolid      : 1                 1-8 ! None
 DSNSharing     : Allreqs           Allreqs ! Modifyreqs
 STRings        : 020               1-255
 Nsrgroup       :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
 RECORDSize     :                   1-32767
 Keylength      :                   1-255
INITIAL STATUS
 STAtus         : Enabled           Enabled ! Disabled ! Unenabled
Opentime        : Startup           Firstref ! Startup
 DIsposition    : Old               Share ! Old
BUFFERS
 DAtabuffers    : 00060             2-32767
 Indexbuffers   : 02000             1-32767
DATATABLE PARAMETERS
 Table          : User              No ! Cics ! User
 Maxnumrecs     : 00004000          16-16777215
DATA FORMAT
 RECORDFormat   : V                 V ! F
OPERATIONS
 Add            : Yes               No ! Yes
 BRowse         : Yes               No ! Yes
 DELete         : Yes               No ! Yes
 REAd           : Yes               Yes ! No
 Update         : Yes               No ! Yes
AUTO JOURNALLING
JOurnal         :  No               No ! 1-99
  JNLRead       : None               None ! Updateonly ! Readonly ! All
  JNLSYNCRead   : No                 No ! Yes
  JNLUpdate     : No                 No ! Yes
  JNLAdd        : None               None ! Before ! AFter ! ALl
  JNLSYNCWrite  : No                 Yes ! No
 RECOVERY PARAMETERS
  RECOVery      : None               None ! Backoutonly ! All
  Fwdrecovlog   : No                 No ! 1-99
  BAckuptype    : Static             Static ! Dynamic
 SECURITY
  RESsecnum     : 00                 0-24 ! Public
-------------------------------------------------------------------------
```

JCL for the definition of the **EZACONFG** VSAM file and configuration of the **EZACICD** macro for the CICS Sockets environment

```
//*******************************************************//
  //*   THE FOLLOWING JOB DEFINES AND THEN LOADS THE VSAM    *//
  //*   FILE USED FOR CICS/TCP CONFIGURATION. THE JOBSTREAM  *//
  //*   CONSISTS OF THE FOLLOWING STEPS.                     *//
  //*    1). DELETE A CONFIGURATION FILE IF ONE EXISTS       *//
  //*    2). DEFINE THE CONFIGURATION FILE TO VSAM           *//
  //*    3). ASSEMBLE THE INITIALIZATION PROGRAM             *//
  //*    4). LINK THE INITIALIZATION PROGRAM                 *//
  //*    5). EXECUTE THE INITIALIZATION PROGRAM TO LOAD THE  *//
  //*        FILE                                            *//
  //*******************************************************//
  //PTCONFIG    JOB  MSGLEVEL=(1,1)
  //*
  //* THIS STEP DELETES AN OLD COPY OF THE FILE
  //* IF ONE IS THERE.
  //*
  //DEL    EXEC  PGM=IDCAMS
  //SYSPRINT DD  SYSOUT=*
  //SYSIN    DD  *
     DELETE -
       CICS.STM9.SOCKETS.CFG -
        PURGE -
        ERASE
  //*
  //*  THIS STEP DEFINES THE NEW FILE
  //*
  //DEFILE EXEC  PGM=IDCAMS
  //SYSPRINT DD SYSOUT=*
  //SYSIN    DD *
    DEFINE CLUSTER (NAME(CICS.STM9.SOCKETS.CFG) VOLUMES(CICSVOL) -
        CYL(1 1) -
        IMBED -
        RECORDSIZE(150 150) FREESPACE(0 15) -
        INDEXED ) -
        DATA ( -
          NAME(CICS.STM9.SOCKETS.CFG.DATA) -
          KEYS (16 0) ) -
        INDEX ( -
          NAME(CICS.STM9.SOCKETS.CFG.INDEX) )
   /*
  //*
//* THIS STEP ASSEMBLES THE INITIALIZATION PROGRAM
  //*
  //PRGDEF  EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
  //SYSLIB    DD DISP=SHR,DSNAME=SYS1.MACLIB
  //            DD DISP=SHR,DSNAME=TCPV32.SEZACMAC
  //SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
  //SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSPUNCH  DD DISP=SHR,DSNAME=NULLFILE
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(400,(500,50)),
  //             DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
  //SYSTERM   DD SYSOUT=*
  //SYSPRINT  DD SYSOUT=*
  //SYSIN     DD *
       EZACICD TYPE=INITIAL,         X
             PRGNAME=EZACICDF,       X
             FILNAME=EZACONFG
       EZACICD TYPE=CICS,            X
             TCPADDR=TCPIP,       X
             NTASKS=20,                     X
             DPRTY=10,                 X
             CACHMIN=10,               X
             CACHMAX=20,               X
             CACHRES=5,                X
             ERRORTD=CSKN,             X
             APPLID=A6ECCSM9
       EZACICD TYPE=LISTENER,  X
             TRANID=CSKL,              X
             PORT=9953,         X
  BACKLOG=40,           X
```

```
         ACCTIME=30,            X
         GIVTIME=10,            X
         REATIME=300,           X
         NUMSOCK=100,           X
         WLMGN1=CICSSTM9,    X
         MINMSGL=4,             X
         APPLID=A6ECCSM9
       EZACICD TYPE=FINAL
/*
 //*
 //* THIS STEP LINKS THE INITIALIZATION PROGRAM
 //*
 //LINK    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
 //             REGION=512K,COND=(4,LT)
 //SYSPRINT  DD SYSOUT=*
 //SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
 //SYSLMOD   DD DSNAME=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
 //             SPACE=(TRK,(1,1,1)),
 //             DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760)
 //SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
 //*
 //* THIS STEP EXECUTES THE INITIALIZATION PROGRAM
 //*
 //FILELOAD EXEC PGM=*.LINK.SYSLMOD,COND=(4,LT)
 //EZACONFG DD  DSNAME= CICS.STM9.SOCKETS.CFG,DISP=OLD
```

You can also configure the CICS TCP/IP Sockets interface using the **EZAC** transaction.

JCL or the definition of the **EZACACHE** VSAM file and configuration of the **EZACICR** macro to use the DNS Cache

```
//*******************************************************************//
  //*   THE FOLLOWING JOB DEFINES AND THEN LOADS THE VSAM            *//
  //*   FILE USED FOR THE CACHE.  THE DEFINITION CONSISTS OF         *//
  //*   TWO IDCAMS STEPS TO PERFORM THE VSAM DEFINITION              *//
  //*   AND A STEP USING EZACICR TO BUILD THE FILE LOAD              *//
  //*   PROGRAM. THE FINAL STEP EXECUTES THE FILE LOAD               *//
  //*   PROGRAM TO CREATE THE FILE.                                  *//

//*******************************************************************//
  //PTCACHE  JOB  MSGLEVEL=(1,1)
  //*
  //* THIS STEP DELETES AN OLD COPY OF THE FILE
  //* IF ONE IS THERE.
  //*
  //DEL    EXEC  PGM=IDCAMS
  //SYSPRINT DD  SYSOUT=*
  //SYSIN    DD  *
     DELETE -
      CICS.STM9.SOCKETS.EZACACHE -
        PURGE -
        ERASE
  //*
  //*  THIS STEP DEFINES THE NEW FILE
  //*
  //DEFILE EXEC  PGM=IDCAMS
  //SYSPRINT DD SYSOUT=*
  //SYSIN    DD *
    DEFINE CLUSTER (NAME(CICS.STM9.SOCKETS.EZACACHE) VOLUMES(CICSVOL) -
        CYL(1 1) -
        IMBED -
        RECORDSIZE(500 1000) FREESPACE(0 15) -
        INDEXED ) -
        DATA ( -
          NAME(CICS.STM9.SOCKETS.EZACACHE.DATA) -
          KEYS (255 0) ) -
        INDEX ( -
          NAME(CICS.STM9.SOCKETS.EZACACHE.INDEX) )
/*
  //*
  //*  THIS STEP DEFINES THE FILE LOAD PROGRAM
  //*
  //PRGDEF  EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
```

```
  //SYSLIB    DD DISP=SHR,DSNAME=SYS1.MACLIB
  //            DD DISP=SHR,DSNAME=TCPV32.SEZACMAC
  //SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
  //SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
  //SYSPUNCH  DD DISP=SHR,DSNAME=NULLFILE
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(400,(500,50)),
  //             DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
  //SYSTERM   DD SYSOUT=*
  //SYSPRINT  DD SYSOUT=*


  //SYSIN     DD *
           EZACICR TYPE=INITIAL
           EZACICR TYPE=RECORD,NAME=ESSONVS1
           EZACICR TYPE=FINAL
/*
  //LINK    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
  //             REGION=512K,COND=(4,LT)
  //SYSPRINT  DD SYSOUT=*
  //SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
  //SYSLMOD   DD DSNAME=&&LOADSET(GO),DISP=(MOD,PASS),UNIT=SYSDA,
  //             SPACE=(TRK,(1,1,1)),
  //             DCB=(DSORG=PO,RECFM=U,BLKSIZE=32760)
  //SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
  //*
  //*  THIS STEP EXECUTES THE FILE LOAD PROGRAM
  //*
  //LOAD EXEC PGM=*.LINK.SYSLMOD,COND=((4,LT,ASM),(4,LT,LINK))
  //EZACICRF DD DSN= CICS.STM9.SOCKETS.EZACACHE,DISP=OLD
```

Definition of the PLT table (*TCP/IP V320)

For the automatic startup/stop of the CICS Sockets interface (*TCP/IP V320), the **EZACIC20** module must be added into the **PLT** table.

For the automatic startup, add in **PLTPI** after the **DFHDELIM** entry:

```
DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
```

For the automatic stop, add in **PLTSD** before the **DFHDELIM** entry:

```
DFHPLT TYPE=ENTRY,PROGRAM=EZACIC20
```

TCP/IP MVS/ESA Configuration

Modification of the TCP/IP configuration

For the use of CICS TCP/IP Sockets, a PORT must be defined for the CICS area in the TCP/IP configuration file (**hlq.PROFILE.TCPIP**).

```
;*********************************************
; PROFILE.TCPIP
; =============
;
; ------------------------------------------------------------------
;
; NOTES:
;
;    A port that is not reserved in this list can be used by any user.
;    If you have TCP/IP hosts in your network that reserve ports
;    in the range 1-1023 for privileged applications, you should
;    reserve them here to prevent users from using them.
;
;    The port values below are from RFC 1060, "Assigned Numbers."
;

PORT
1415 TCP CSQ9CHIN            ; MQSeries CSQ9
9011 TCP PTMBRUNT            ; TeamConnection
9950 TCP EXCICS9             ; CICS Socket for CICS9
9953 TCP CICSSTM9            ; CICS Socket for A6ECCSM9
```

```
-------------------------------------------------------------------------
```

TCPJOBNAME Parameter in the *hlq*.TCPIP.DATA file

For the initialization of CICS TCP/IP, you must know the name of the MVS TCP/IP procedure specified in the **hlq.TCPIP.DATA** file, **TCPIPJOBNAME** parameter.

```
;
;*************************************************************
;   Name of Data Set:     TCPIP.DATA                          *
;                                                             *
;   COPYRIGHT = NONE.                                         *
;                                                             *
; This data, TCPIP.DATA, is used to specify configuration     *
;   information required by TCP/IP client programs.           *
;                                                             *
;                                                             *
;   Syntax Rules for the TCPIP.DATA configuration data set    *
;       treated as a comment.                                 *
;                                                             *
*************************************************************
; TCPIPJOBNAME specifies the name of the started procedure that was
; used to start the TCPIP address space.    TCPIP is the default.
;
TCPIPJOBNAME TCPIP
```

Manual Start and Stop of CICS TCP/IP

Start of CICS TCP/IP

Execute the CICS transaction -> **CSKE** to manually start the CICS TCP/IP (*TCP/IP V310).

```
-------------------------------------------------------------------------
CSKE                                                          EZACIC00
                          CICS TASK-RELATED USER EXIT
                              CONNECTION MANAGER

                            ENABLE CICS-TCP/IP API


                         TCPIPJOBNAME tcpip--- PORT 9953-
 PF1=HELP          PF3=QUIT                                  EZAME00
-------------------------------------------------------------------------
```

Execute the CICS transaction -> **EZAO** to manually start the CICS TCP/IP (*TCP/IP V320).

```
-------------------------------------------------------------------------
EZAO,START,CICS

 APPLID=      ===> A6ECCSM9            APPLID of CICS

-------------------------------------------------------------------------
```

Next, check with **CEMT I TAS** that the **CSKL** transaction is active:

```
CEMT I TAS
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CKAM)           Sus Tas Pri( 255 )
Tas(0000024) Tra(CKTI)           Sus Tas Pri( 001 )
Tas(0000027) Tra(DSNC)           Sus Tas Pri( 255 )
Tas(0000034) Tra(ISER)           Sus Tas Pri( 254 )
Tas(0000046) Tra(CSKL)           Sus Tas Pri( 255 )
```

Stop of CICS TCP/IP

Execute the CICS transaction => **CSKD** to manually stop the CICS TCP/IP (*TCP/IP V310).

```
------------------------------------------------------------------------------
CSKD                                                          EZACIC00
                                      CICS TASK-RELATED USER EXIT
                                      CONNECTION MANAGER

                                      DISABLE CICS-TCP/IP API


                                      ===> 1  QUIESCENT DISABLE-API
                                      ===> 2  IMMEDIATE DISABLE-API

                  ENTER ONE OF THE ABOVE DISABLE-API OPTION NUMBERS: 2

PF1=HELP           PF3=QUIT                                   EZAMD00
------------------------------------------------------------------------------
```

Execute the CICS transaction => **EZAO** to manually stop the CICS TCP/IP (*TCP/IP V320).

```
------------------------------------------------------------------------------
EZAO,STOP,CICS

 APPLID    ===> A6ECCSM9             APPLID of CICS
 IMMEDIATE    ===>  Y                          Enter Yes!No
------------------------------------------------------------------------------
```

Cobol Compilation of the VA Pac Communication Monitor Program

Modification of the JCL for the compilation and link-edit to integrate the CICS Socket TCP/IP interface.

```
//PTTDASOC  JOB (661),LH,CLASS=X,MSGCLASS=X,MSGLEVEL=(0,0)
//*JCLLIB ORDER=DSNY220.JCLLIB
//*********************************************
//*  JCL to Compil/link VAP Communication Monitor *
//*   with the CICS SOCKET TCP/IP interface       *
//*********************************************
//CBL EXEC DB2SOCK,MEMBER=CLTMVS,LOAD=PT$VIC.VIC.MTR8,
//          DBRMLIB=PT$VIC.VIC.DBRMLIB,SOURCE=PT$VIC.CICS.SOURCE
//LKED.SYSLIB  DD DSN=PT$VIC.TCPIP310.SEZATCP,LIB=SHR
//LKED.SYSIN DD *
    INCLUDE  SYSLIB(EZACICAL)
    INCLUDE  SYSLIB(EZACIC04)
    INCLUDE  SYSLIB(EZACIC05)
    INCLUDE  DB2LOAD(DSNCLI)
    NAME     CLTMVS(R)
 //
```

CICS definitions for the VA Pac application

Definition of the Transaction Code

```
------------------------------------------------------------------------------
TRansaction    : VSO1
   Group          : VISUAL
   DEscription    : VISUAL/CLIENT Transaction SOCKET CICS TCP/IP
```

```
   PROGram         : CLTMVS
   TWasize         : 00000
   PROFile         : DFHCICST
   PArtitionset    :
   STatus          : Enabled
   PRIMedsize      : 00000
   TASKDATALoc     : Below
   TASKDATAKey     : User
  REMOTE ATTRIBUTES
   DYnamic         : No
   REMOTESystem    :
   REMOTEName      :
   TRProf          :
+  Localq          :
--------------------------------------------------------------------------------
```

Definition of the Communication Monitor program

```
--------------------------------------------------------------------------------
PROGram         : CLTMVS
    Group           : VISUAL
    DEscription     : Communication Moniteur CICS TCP/IP SOCKET
    Language        : CObol
    RELoad          : No
    RESident        : No
    USAge           : Normal
    USElpacopy      : No
    Status          : Enabled
    RSl             : 00
    Cedf            : Yes
    DAtalocation    : Below
    EXECKey         : User
   REMOTE ATTRIBUTES
    REMOTESystem    :
 +  REMOTEName      :
+  Transid         :
    EXECUtionset    : Fullapi
--------------------------------------------------------------------------------
```

Definition of the VSAM Workfile

```
--------------------------------------------------------------------------------
File            : FRVABI
    Group               : VISUAL
    DEScription     :
   VSAM PARAMETERS
    DSNAme          : PT$VIC.CICS.FRBIS
    Password            :
    Lsrpoolid            : 1
    DSNSharing      : Allreqs
    STRings             : 001
    Nsrgroup            :
   REMOTE ATTRIBUTES
    REMOTESystem    :
    REMOTEName      :
    RECORDSize      :
    Keylength           :
   INITIAL STATUS
 +  STAtus          : Enabled
--------------------------------------------------------------------------------
```

TUXEDO

Client

Tuxedo /WS version 6.x

The only configuration to be considered consists in indicating the Server address and the port associated with the application via the **WSNADDR** environment variable.

WSNADDR must respect the following syntax:

```
WSNADDR=0X0002ppppaaaaaaaa
          |    |    ↳ 8-char. long IP address in hex
          |    ↳ 4-char. long port number in hex.
          ↳ AF_INET domain.
```

example: **WSNADDR=0X00020BB8C0060A5D**

```
                |   ↳ 192.6.10.93
                ↳ 3000
```

If the Tuxedo /WS version allows it (version 6.4 or higher), this address must be specified as follows:

```
WSNADDR=//serveur:port
                  |   ↳ Port number
                  ↳Logical name or Server IP address
```

example: **WSNADDR=//9.143.96.178:3005**

Server

Refer to the TUXEDO manual.

The services' names in the Tuxedo Server must correspond to the programs' external names.

MQSERIES

CICS Adapter

Example of a simple configuration for a CICS application triggered by the Trigger Monitor MQSeries.

- **REQUEST QUEUE** Definition

```
Queue name  . . . . . . . . . . VAP.REQUEST.Q
Description . . . . . . . . . : REQUEST QUEUE

Put enabled . . . . . . . . . : Y  Y=Yes,N=No
Get enabled . . . . . . . . . : Y  Y=Yes,N=No
Usage . . . . . . . . . . . . : N  N=Normal,X=XmitQ
Storage class . . . . . . . . : DEFAULT
Creation method . . . . . . . : PREDEFINED
Output use count  . . . . . . : 0
Input use count . . . . . . . : 0
Current queue depth . . . . . : 0
Default persistence . . . . . : Y  Y=Yes,N=No
Default priority  . . . . . . : 5  0 - 9
Message delivery sequence . . : F  P=Priority,F=FIFO
Permit shared access  . . . . : Y  Y=Yes,N=No
Default share option  . . . . : S  E=Exclusive,S=Shared
Index type  . . . . . . . . . : N  N=None,M=MsgId,C=CorrelId
Maximum queue depth . . . . . : 100000     0 - 999999999
Maximum message length  . . . : 4194304  0 - 4194304
Retention interval  . . . . . : 999999999  0 - 999999999 hours
Creation date . . . . . . . . : 1999-06-23
Creation time . . . . . . . . : 12.59.47

Trigger Definition

Trigger type  . . . . . . . . : E  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . . . : Y  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . . : 1          1 - 999999999
   Trigger data . . . . . . . :

   Process name . . . . . . . : VAP.PROCESS.DEF
   Initiation queue . . . . . : CICS.INITQ
```

- **REPLY QUEUE** Definition

```
   Queue name  . . . . . . . . . . VAP.REPLY.Q
   Description . . . . . . . . . : output QUEUE


   Put enabled . . . . . . . . . : Y  Y=Yes,N=No
   Get enabled . . . . . . . . . : Y  Y=Yes,N=No
   Usage . . . . . . . . . . . . : N  N=Normal,X=XmitQ
   Storage class . . . . . . . . : DEFAULT
   Creation method . . . . . . . : PREDEFINED
   Output use count  . . . . . . : 0
   Input use count . . . . . . . : 0
   Current queue depth . . . . . : 0
   Default persistence . . . . . : Y  Y=Yes,N=No
   Default priority  . . . . . . : 0  0 - 9
   Message delivery sequence . . : F  P=Priority,F=FIFO
   Permit shared access  . . . . : Y  Y=Yes,N=No
   Default share option  . . . . : S  E=Exclusive,S=Shared
   Index type  . . . . . . . . . : N  N=None,M=MsgId,C=CorrelId
   Maximum queue depth . . . . . : 999999999  0 - 999999999
   Maximum message length  . . . : 4194304  0 - 4194304
   Retention interval  . . . . . : 999999999  0 - 999999999 hours
```

```
Creation date . . . . . : 1999-06-23
Creation time . . . . . : 13.00.10

Trigger type  . . . . . : N  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . : N  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . : 1          1 - 999999999
   Trigger data . . . . . :

   Process name . . . . . :
   Initiation queue . . . . :
```

- **INITIATION QUEUE** Definition

```
Queue name  . . . . . . . . . CICS.INITQ
Description . . . . . : CKTI initiation queue

Put enabled . . . . . : Y  Y=Yes,N=No
Get enabled . . . . . : Y  Y=Yes,N=No
Usage . . . . . : N  N=Normal,X=XmitQ
Storage class . . . . . : SYSTEM
Creation method . . . . . : PREDEFINED
Output use count  . . . . . : 0
Input use count . . . . . : 1
Current queue depth . . . . : 0
Default persistence . . . . : Y  Y=Yes,N=No
Default priority  . . . . . : 5  0 - 9
Message delivery sequence . : F  P=Priority,F=FIFO
Permit shared access  . . . : Y  Y=Yes,N=No
Default share option  . . . : E  E=Exclusive,S=Shared
Index type  . . . . . : N  N=None,M=MsgId,C=CorrelId
Maximum queue depth . . . . : 100        0 - 999999999
Maximum message length  . . : 4194304    0 - 4194304
Retention interval  . . . . : 999999999  0 - 999999999 hours
Creation date . . . . . : 1999-05-07
Creation time . . . . . : 18.30.18
Trigger Definition

Trigger type  . . . . . : N  F=First,E=Every,D=Depth,N=None

   Trigger set  . . . . . : N  Y=Yes,N=No
   Trigger message priority : 0  0 - 9
   Trigger depth  . . . . . : 1          1 - 999999999
   Trigger data . . . . . :

   Process name . . . . . :
   Initiation queue . . . . :
    Event Control
```

- **PROCESS** Definition

```
Process name . . . . . VAP.PROCESS.DEF
Description  . . . . . : VAP Process

Application type . . . . . : CICS
Application ID . . . . . : AMQM  ← Server application transaction
User data  . . . . . : VAP.REQUEST.Q QMGR
```

MQSeries Client

The MQSeries Client component must be installed on each Client application workstation and configured via the following system environment variables:

- **MQSERVER**

  This environment variable is a simple way to indicate the **only** path (MQI Channel) to reach the Server.

This variable must be defined as follows, depending on the used transport protocol:
. TCP/IP:
<ChannelName>/TCP/<ServeurName>(<PortNumber>)
The default port number is 1414

. LU 6.2:
<ChannelName>/LU62/<SymbolicDestName>/<ModeName>/<TPName>

Example under Unix:
```
set MQSERVER = CHANNEL.TO.VAP/tcp/9.134.12.87(1414)
export MQSERVER
```

- **MQCHLLIB** and **MQCHLTAB**

  On a workstation, when the MQSeries applications must use more than one Channel, the **MQSERVER** environment variable cannot be used. The MQSeries network administrator must generate and provide a Channels description file (**amqclchl.tab** by default) and make it accessible to the Client workstations.

  The **MQCHLLIB** and **MQCHLTAB** environment variables are used to indicate the location path and the name of this file respectively..

TDS-TCP/IP

TDS-TCP/IP Client Installation / configuration

Installation

Under **Windows NT**, you must install:

- the **Atmi32.dll** file in a directory which is accessible through the PATH (ex: VAP middleware directory).
- the **Atmi.ini** file in the Windows system directory (ex: c:\winnt).

Under **AIX 4.3**, you must:

- Transfer the **XATMI** file to AIX,
- Specify the name of the directory where **XATMI** is located, in the **LIBPATH** environment variable

  Ex: **export LIBPATH=$LIBPATH:/vap/middleware**

hosts and services files

They are located in the **c:\winnt\system32\drivers** directory for Windows NT and **/etc** for Unix.

In the **hosts** file, you must define an entry for the GCOS7 Server (ex: **213.62.98.213 bc0e** #GCOS7)

In the **services** file, you must define the service with a name obtained by concatenating the **host** name with the TDS name (ex: **bc0evpd5 51000/tcp** #TCP/IP Access TDS)

ATMI Traces

ATMI.INI *file for Windows*

This file must be located in the Windows system directory (ex: c:\winnt), and contains the following parameters:

```
[ATMI]

; PATH:  for the log and trace files (the current path by default)

; CRYPT: required, must not be modified
CRYPT=YES

; DEBUG: echo of warning messages
   0: no display of the messages
   1: display of all the messages, excepted the disconnection messages
   2: display of all the messages
DEBUG=0

; TRACE_API: user trace: input and output parameters of the XATMI called
   functions
   0: no trace
```

1: parameters trace but without the content of the buffer
2: parameters trace and buffer content

```
TRACE_API=2
```

; TRACE_SOC: internal trace for the debugging interface with Windows Sockets
0: no trace
1: trace of called functions and of the input/output parameters
2: 1 + content of the header exchanged with TDS

```
TRACE_SOC=0
```

; TIMEOUT: connection time in milliseconds
0: no timeout

```
TIMEOUT=30000
```

Environment variables for AIX

The trace file is: `ATMITDS.TRC_<process id>.`

The following environment variables must be set up:

ATMI_TRACE_API: trace of API calls.
0: no trace (default).
1: trace of the buffers' 32 first bytes.
2: trace of all the buffers content.

☞ The value 0 is generally recommended. The value 1 is recommended in case of problem for which the content of sending/receiving buffers is not necessary. The value 2 must be reserved in case of problem for which the content of sending/receiving buffers must be analyzed.

ATMI_TRACE_SOC: trace of socket verbs. This variable is reserved for the system debugging.
0: no trace (default).
1: trace of socket verbs.
2: trace of sending/receiving headers exchanged with TDS.

These environment variables must be standardly exported so that the application can read their values.

Moreover, independently of the trace file mechanism, you can get the log of the error messages via the ATMI_DEBUG environment variable.
0: no console display (default).
1: the abnormal disconnection messages only are displayed on the console.
2: display of all the errors on the console.

Example of implementation on TDS

Example of TDS source

TDS name: VPD5

TDS source: STDS member of the VPD5.SLLIB library

The using of the TCP/IP Socket must be declared in the **STDS** member under **VPD5.SLLIB** and the TDS (**TP7GEN**) must be re-generated.

You can modify the **STDS** member of **VPD5.SLLIB** as follows:

```
 TDS SECTION.
 PROGRAM-ID. VPD5.
* BTNS                IS BTNS.
 NUMBER TERMINALS       15.
 NUMBER OF DUMMY CORRESPONDENT IS 1 MAXIMUM IS 3.
 SIMULTANEITY           10.
 RESERVE                280 AREAS.
 ATTACH SHARABLE MODULE H_SM_DCM.
 NUMBER MODULES         10.
 MESSAGE-LENGTH         32001.
 TPR-TIME-LIMIT         500000.
 TCP-IP PROTOCOL USED WITH OPEN7.
 USE "MENU"     TRANSACTION-MENU.
 USE ZAR990.
 USE ZARS12.
 USE FORMS.
 USE CLCLNT.
 USE CLPROD.
 SERVICE-MESSAGE HEADER IS "27F1C3"
     TRAILER IS "4040".
 CANCELCTX AT RECONNECTION.
 ...
```

Then you define in **STDS** the transaction which starts the Communication Monitor:

```
 TRANSACTION SECTION.
 MESSAGE "VTCP" ASSIGN    CLSOCK
     IMPLICIT COMMITMENT
     PAGES                 50
     WITH TPR ACCOUNTING
     AUTHORITY-CODES      31
     TRANSACTION-STORAGE  SIZE  500.
 ...
```

Reference : DDOVM000353A