

CA Gen

Distributed Processing - Communications Bridge User Guide

Release 8.5



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Gen

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction 9

Audience	11
Visual Studio Support	11
Related Information	11

Chapter 2: Communications Bridge Overview 13

Concepts and Definitions	13
Distributed Processing Application	13
Client Workstation	13
Server Workstation	14
Cooperative Flow	14
Common Format Buffer	14
DPC	14
DPS	14
CA Gen DP Application Network	15
Comm. Bridge Client Environments	15
Comm. Bridge Server Environments	16
Comm. Bridge Features	16
Client and Server Connections	18
Comm. Bridge Client Connections	18
Comm. Bridge Server Connections	18
Comm. Bridge Security	19
Comm. Bridge as a Windows System Service	19
Confirming DPC/DPS Communications	19
Transaction Statistics	20
Customizable User Exits	20
External Trace Log Enabling	20
Default Configuration and Log file Locations	21

Chapter 3: Comm. Bridge General Configuration 23

Comm. Bridge Installation	23
Start Comm. Bridge After Install	23
Executing the Comm. Bridge	24
Using the Desktop Start, All Programs Menu	24
Using a Desktop Shortcut Icon	24
From an MS-DOS Command Prompt	25

As a Windows System Service	26
Comm. Bridge Setup Dialog	26
Description	27
File Description List	28
Browse	28
Rename	28
File Browser.....	28
Auto-Reset Server Connection	29
Trace Logging	29
Statistics	30
Dump I/O Buffers	30
Max Log Size (M.B.)	30
Client Monitor	31
Service Config Button.....	32
Configuring Comm. Bridge Client Connections	32
Configuring TCP/IP (Socket) Client Connections	32
Configuring a Client Connection for TCP/IP (Sockets)	33

Chapter 4: Configuring Comm. Bridge Server Connections 35

Server Configuration	35
Server Name.....	35
Description	35
Transport API	36
Transport API – Additional Details	36
LU6.2 CPI-C Connections.....	37
TCP/IP/Socket Connections.....	40
z/OS CICS External Call Interface (ECI)	48
NonStop RSC/MP Connections	51
Other API.....	55

Chapter 5: Communications Bridge Security 57

CFB Security.....	57
Decrypting the CFB.....	59
Comm. Bridge DECRYPT User Exit	59
Translating UserID and Password	60

Chapter 6: Configuring Multiple Comm. Bridges 61

Creating Multiple Comm. Bridges Using Icons	62
Creating Multiple Comm. Bridges Without Using Icons.....	62
Creating Multiple Comm. Bridge System Services	63

Chapter 7: Configuring a Comm. Bridge as a Windows System Service **65**

Special User Access Privileges	65
Logging Comm. Bridge Events	65
Interactive and Non-interactive Modes	66
System Service Configuration.....	67
Display Name	68
Service Name	68
Service Description.....	68
User ID.....	68
Password.....	68
Automatic.....	68
Manual	68
Disabled.....	69
Install Service	69
Remove Service.....	69
Parameter File.....	69
File Directories	69
CODEPAGE.INI	70
IEFCBN.INI	70
IEFCBN.SRV and IEFCBN.LOG	70
Registering Using the Comm. Bridge GUI Interface	70
Registering Using the Comm. Bridge Command Line.....	72
Removing Using the GUI Interface	73
Removing Using the Comm. Bridge Command Line.....	74
Comm Bridge Service Targeting RSC/MP	75

Chapter 8: Saving Configuration Files **77**

Saving Configuration File Names.....	77
Changing Configuration File Names	78

Chapter 9: Testing Comm. Bridge Connections **79**

Testing Connectivity Using a DPC Application.....	79
Testing Connectivity Using the Client Manager	80

Chapter 10: Comm. Bridge Transaction Statistics **83**

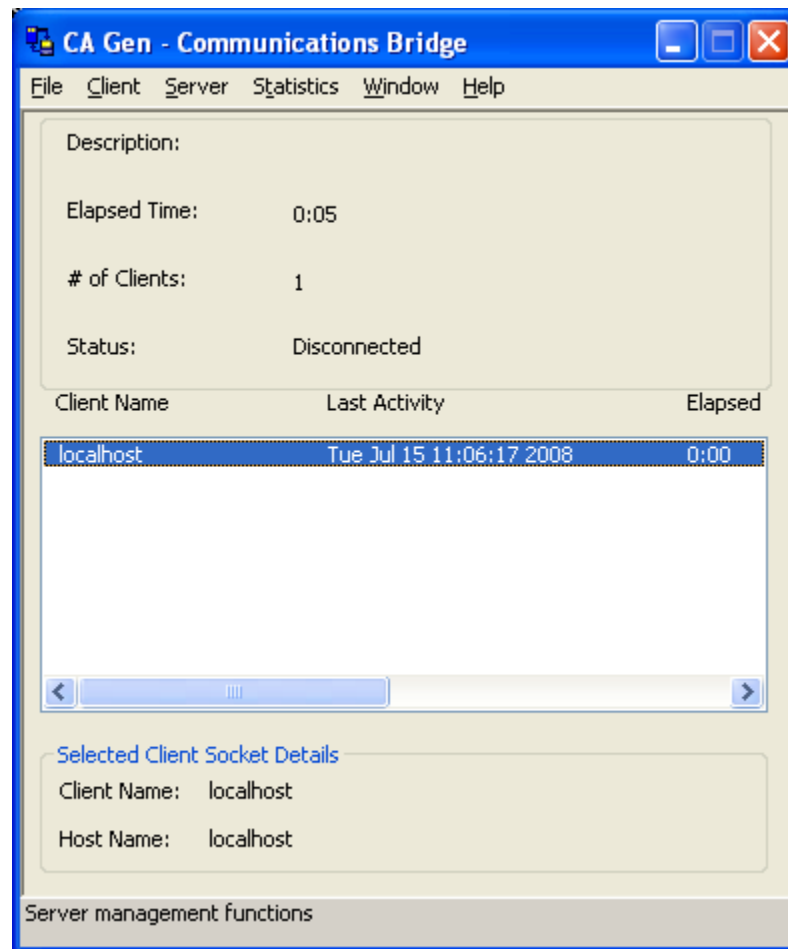
Statistics - Summary Dialog.....	84
----------------------------------	----

Appendix A: Error Messages	87
Setting the Logging Level to Tracing.....	87
Appendix B: Comm. Bridge User Exits	89
Index	91

Chapter 1: Introduction

This guide provides an overview and guidelines for using and setting up a CA Gen Communications Bridge (Comm. Bridge). The Comm. Bridge is a CA Gen communications utility program that can be used as part of a communications infrastructure.

The following diagram shows the Comm. Bridge main GUI window.

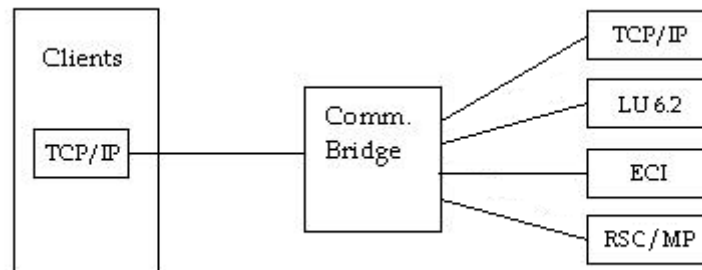


The Comm. Bridge is a generated Windows application. It serves as a gateway component that is capable of converting the communications protocol used by its clients to the protocol supported by its target server. The Comm. Bridge can also be used in a TCP/IP network to handle large number of users, funneling many clients to a target server execution environment. The Comm. Bridge supports multiple, concurrent cooperative flows from the set of clients it serves to the target server it represents.

The supported outbound protocols and APIs include:

- TCP/IP (Sockets)
- LU6.2 (CPI-C)
- CICS External Call Interface (ECI)
- NonStop Remote Server Call (RSC/MP)

This diagram depicts client and server transports supported by Comm. Bridge.



The following items are discussed in further detail within this document:

- Conceptual information about the CA Gen Comm. Bridge in a CA Gen Client/Server Distributed Process application network
- General configuration options used to define a Comm. Bridge
- Information about configuring the Comm. Bridge for connections from CA Gen clients
- Information about configuring the Comm. Bridge for connection to a CA Gen server execution environment
- Comm. Bridge security capabilities
- Details for creating multiple Comm. Bridges on a single workstation
- Procedures to use for configuring, installing, and removing Comm. Bridges that run as Windows system services
- Test procedures for verifying proper client to server communications when using a Comm. Bridge
- Information regarding the statistics kept by the Comm. Bridge
- User exits invoked by a Comm. Bridge. User exits allow certain runtime behavior to be customized.

Audience

This guide is intended for CA Gen administrators or users who need to configure CA Gen Clients to communicate with CA Gen Servers. To get the most from this guide, you should be familiar with the components required for successful deployment of CA Gen Client/Server distributed processing applications. The Comm. Bridge is an optional part of the networking infrastructure used to support a CA Gen Distributed Processing application. Knowledge of network topology and communications administration requirements is also needed. In-depth knowledge of specific network transports/protocols is generally not required. This guide is written for:

- Communications specialists
- System administrators
- Server administrators
- Application Integrators
- Application developers

Visual Studio Support

CA Gen supports a Communications Bridge that has been built using Visual Studio.

The %GENxx%Gen\VSabc folder contains a collection of files that support the Communications Bridge with Visual Studio. A set of user exit rebuild procedures is also present in the VSabc folder and must be used to rebuild any necessary Visual Studio designated user exits. Add %GENxx%Gen\VSabc to PATH when working with the Communications Bridge.

Note: VSabc refers to the supported version of Visual Studio. Replace VSabc with VS100 for Visual Studio 2010 and VS110 for Visual Studio 2012. xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

Related Information

Along with this document, third party vendor operation and configuration documents applicable to specific communications transports should also be reviewed.

To complete configuration tasks, administrators must communicate information specific to the protocols in use. Some protocols may require special hardware and software configurations, discussed fully in your vendor documentation.

The following guides provide additional information for other CA Gen products used within a distributed processing application:

- *Distributed Processing - Overview Guide*
- *Distributed Processing - Client Manager User Guide*
- *Distributed Systems Installation Guide*
- *Technical Requirements Document*
- *Transaction Enabler User Guide*
- *Tuxedo User Guide*
- *z/OS Implementation Toolset User Guide*

The following third party documents provide additional information that you may find useful in configuring network components:

- Microsoft documentation for Host Integration Server Administration
- IBM documentation for z/OS Communications Server
- IBM Communications Server for Windows
- IBM CICS Transaction Gateway documentation for External Call Interface (ECI)
- HP NonStop Remote Server Call (RSC/MP) Installation and Configuration Guide

Chapter 2: Communications Bridge Overview

The Communications Bridge (Comm. Bridge) is a separately installable, standalone CA Gen utility application. It is an optional communications gateway application that resides between a CA Gen Distributed Processing Client and a Distributed Processing Server.

Concepts and Definitions

This section discusses the various concepts and definitions in CA Gen distributed processing client/server application.

Distributed Processing Application

A CA Gen Distributed Processing (DP) client/server application is software that is comprised of two or more separate executables: Distributed Processing Clients (DPC) and Distributed Processing Servers (DPS). Each executable performs a specific function for the overall application.

The DPC communicates with the DPS by transmitting request and reply byte streams across a network connection. These byte streams are transmitted over supported transport protocols provided by CA Gen using a mechanism known as a cooperative flow.

For a detailed description of CA Gen Distributed Processing applications, see the *Distributed Processing – Overview Guide*.

Client Workstation

In this discussion, the client workstation is considered to be the machine hosting the client application part of a Distributed Processing application. As an option, the Comm. Bridge can reside on the client workstation, depending on the desired configuration. Although not generally done, the client and server applications can also reside on the same machine. For the purpose of discussion, the client workstation is considered the machine where the client application resides even if the server application resides on the same machine.

Server Workstation

In this discussion, the server workstation is considered to be the machine hosting the server application part of a Distributed Processing application. As an option, the Comm. Bridge can reside on the server workstation, depending on the desired configuration. Although not generally done, the client and server applications could reside on the same machine. For the purpose of discussion, the server workstation is considered to be the machine where the server application resides, even if the client application resides on the same machine.

Cooperative Flow

A cooperative flow is the generated set of instructions that implement the invocation of a target server procedure step from a client procedure step. A cooperative flow provides the means by which a client procedure step passes control and data to and from a server procedure step.

Common Format Buffer

The Common Format Buffer (CFB) is an encoded byte stream that CA Gen uses to exchange view data during the processing of a cooperative flow. In addition to import and export view data, a CFB contains control data used in processing a cooperative flow. For more information about CFB, see the *Distributed Processing – Overview Guide*.

DPC

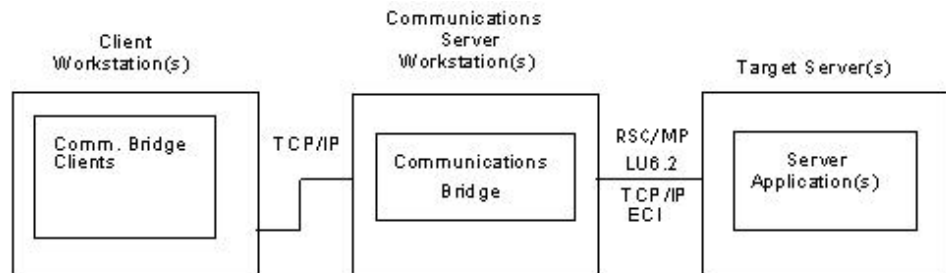
The Distributed Processing Client (DPC) application resides on a client workstation. The role of the DPC is to handle the GUI presentation and the logic associated with that presentation.

DPS

The Distributed Processing Server (DPS) application resides on a target server. The role of the DPS is to perform the business logic and database processing activities.

CA Gen DP Application Network

The following diagram depicts a CA Gen Distributed Process Application Network Environment:



Comm. Bridge Client Environments

Various types of CA Gen client applications can be clients of a Comm. Bridge. When the Comm. Bridge discussion refers to Comm. Bridge clients, it includes the following supported Comm. Bridge clients:

- CA Gen Windows GUI clients
- CA Gen Java Internet clients
- CA Gen ASP .NET Internet clients
- CA Gen C proxy clients
- CA Gen COM proxy clients
- CA Gen Java proxy clients
- CA Gen .Net proxy clients
- CA Gen Client Manager
- CA Gen Java EJB servers
- CA Gen .NET Servers

In general, potential clients of a Comm. Bridge are:

- Application components of a Cooperatively packaged CA Gen Distributed Processing application that are capable of performing a cooperative flow using TCP/IP (IPv4 or IPv6 protocol) as a transport.
- User-written applications that make use of a generated proxy and are capable of performing a cooperative flow using TCP/IP (IPv4 or IPv6 protocol) as a transport.

Comm. Bridge Server Environments

A CA Gen Comm. Bridge can serve as a gateway application to various types of CA Gen server execution environments. When the Comm. Bridge discussion refers to a Comm. Bridge server, it includes one of the following supported Comm. Bridge server environments:

- z/OS CICS using LU6.2 (CPI-C)
- z/OS IMS using LU6.2 (CPI-C)
- UNIX: Transaction Enabler using TCP/IP (IPv4 or IPv6 protocol)
- UNIX: Tuxedo Proxy Client using TCP/IP (IPv4 or IPv6 protocol)
- Windows: Transaction Enabler using TCP/IP (IPv4 or IPv6 protocol)
- z/OS CICS: CICS Socket Listener (TCP/IP - IPv4 or IPv6 protocol)
- z/OS IMS: IMS TCP/IP Direct Connect (IPv4 or IPv6 protocol)
- z/OS CICS: External Call Interface (ECI)
- NonStop (Pathway) using Remote Server Call (RSC/MP)

Comm. Bridge Features

The Comm. Bridge is an optional utility application that provides communications support for a Distributed Process application by acting as a communications link or gateway between the client applications and target server environments. The Comm. Bridge accepts message requests from the clients and forwards the request to a target server. Server response data is routed back to the client that made the request.

Using the Comm. Bridge with your Distributed Process application provides a centralized place for configuring host-specific communications hardware and software. Each Comm. Bridge client side interface is identified by its TCP/IP host name and a well-known port number. The Comm. Bridge communicates with the target server using a configured transport protocol, either LU6.2, TCP/IP, CISC External Call Interface (ECI), or RSC/MP. Each Comm. Bridge is configured to communicate with a specific target server environment.

Depending on its configuration, the Comm. Bridge can operate as a multiplexing funnel, directing multiple client connections to a single server environment. This capability helps CA Gen application networks expand as the number of client workstations grows beyond the capacity of the target server execution environment.

Alternatively, Comm. Bridge can operate as a communications gateway, providing protocol conversion to enable clients communicating using one communications protocol to connect to servers using a different protocol, such as:

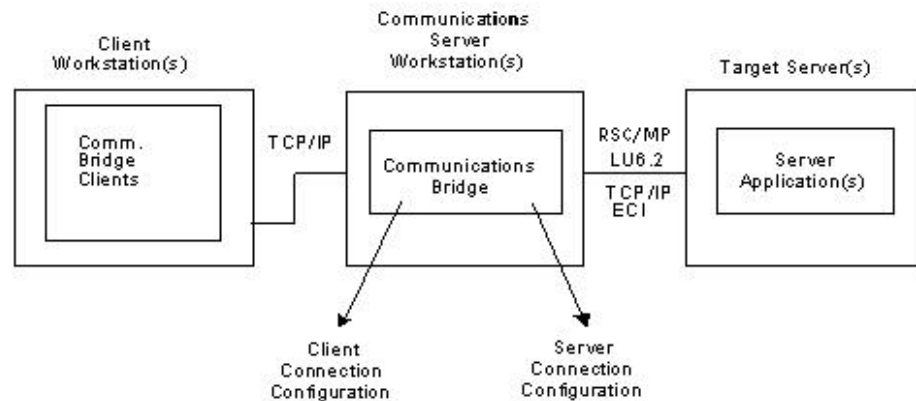
- TCP/IP to LU6.2
- TCP/IP to CICS External Call Interface (ECI)
- TCP/IP to RSC/MP

Other operational features of the Comm. Bridge include:

- The Comm. Bridge configuration can be modified in one of two ways:
 - Using the Comm. Bridge GUI interface
 - Using an ASCII editor to modify the configuration text files
- Configuration of the Comm. Bridge to run as a Windows system service.
- Separate client and server configurations.
- A single Comm. Bridge can concurrently handle up to 255 simultaneous Comm. Bridge client connections (although the practical client limit may be lower).
- More than one Comm. Bridge can reside on the same workstation.
- Multiple Comm. Bridge instances can be configured within a single directory
- More than one Comm. Bridge can access the same target server; however, each Comm. Bridge must be uniquely identified by its client side configuration.
- Periodic logging of transmission statistics.
- Selectable amount of trace data can be written to the Comm. Bridge log file to assist in problem determination.
- Configurable log file size and wrapping, eliminates large log files when gathering trace data.
- Configurable automatic disconnection of idle clients.
- Customizable ASCII file browser for viewing Comm. Bridge configuration and log files.
- Automatic reset of server connections after a configuration change has been made.

Client and Server Connections

You must set configuration parameters for both client and server within the Comm. Bridge. The client side of the Comm. Bridge contains the connection support for communicating with a Client Manager or other CA Gen clients. The server connection contains the connection support for communicating with a target server environment.



Comm. Bridge Client Connections

For the client connection, the Comm. Bridge communicates with clients using TCP/IP (IPv4 or IPv6 Sockets API).

More Information:

[Comm. Bridge General Configuration](#) (see page 23)

Comm. Bridge Server Connections

For the server connection, the Comm. Bridge can communicate using LU6.2, TCP/IP (IPv4 or IPv6 Sockets API), ECI, or RSC/MP. For more information about configuring a Comm. Bridge server connection, see the chapter "Configuring Comm. Bridge Server Connections".

More Information:

[Configuring Comm. Bridge Server Connections](#) (see page 35)

Comm. Bridge Security

The Comm. Bridge passes a Common Format Buffers (CFB) that it receives from its clients to the server execution environment being served by the Comm. Bridge. The Comm. Bridge does not perform any security validation of its own. Additionally, the Comm. Bridge does not supply or add security data to a CFB. The originating client must specify the security data that is used. If security data is to be used for a given cooperative flow, the corresponding CFB must contain it such that it can be available for use by the Comm. Bridge.

The choice of which CFB security data the Comm. Bridge is to use depends on the type of CFB received, standard or enhanced. If the CFB is an enhanced CFB and the CFB header has the *Use Client Manager Security flag* set, then the security data comes from the CFB security offset area. Otherwise, the security data comes from the CFB header.

For more information about Comm. Bridge security, see the chapter "Communications Bridge Security." For more information about security within a Distributed Processing application, see the *Distributed Processing – Overview Guide*.

More Information

[Communications Bridge Security](#) (see page 57)

Comm. Bridge as a Windows System Service

A Comm. Bridge can be configured to run as a Windows system service. This may be required if the Comm. Bridge is to run unattended on a system with no logged in user.

The service can be configured to:

- Start automatically upon system reboot or wait to be started manually
- Display or not display, the Comm. Bridge GUI interface upon startup
- Execute under a specific user or authorized system user

For more information, the chapter "Configuring a Comm. Bridge as a Windows System Service."

Confirming DPC/DPS Communications

The chapter "[Testing Comm. Bridge Connections](#) (see page 79)" provides details that can be used to confirm proper Comm. Bridge to server communications.

Transaction Statistics

Certain transaction statistics are gathered automatically. From the Comm. Bridge main GUI window, you can display a dialog containing a summary of the Comm. Bridge statistics. Once displayed, the statistics dialog can be continuously refreshed based on a configurable time period.

More Information:

[Comm. Bridge Transaction Statistics](#) (see page 83)

Customizable User Exits

The Comm. Bridge contains user exits that can influence certain default behavior.

More Information:

[Comm. Bridge User Exits](#) (see page 89)

External Trace Log Enabling

The Comm. Bridge allows external enabling and disabling of Trace level logging. This capability activates the collection of trace data without the need to use the Comm. Bridge GUI interface. The logging level change is triggered by either of the two notification files that exist in the Comm. Bridge install directory.

The file CBTRACE_ALL triggers a log level change for all active Comm. Bridges that find this file in their %APPDATA%\CA\Gen xx\cb directory. This is a way to enable logging for multiple Comm. Bridges at the same time.

The second file is CBTRACE_nnnn, where nnnn is the port number of a client. Placing this file in the %APPDATA%\CA\Gen xx\cb directory triggers a log level change but only for the client on that port.

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

In both the cases, logging level is changed from the current level to Trace. When a trace notification file is removed, the logging level is reset to its previous value. This is particularly useful when the Comm. Bridge is installed as a system service or when multiple instances of a Comm. Bridge have been configured and simultaneous logging is needed for all active Comm. Bridges.

The Comm. Bridge checks for the existence of these notification files every five seconds.

Default Configuration and Log file Locations

To allow support of the User Account Control (UAC) mechanism featured with the Windows 7 operating system this default location has been changed. With UAC enabled a non administrative user is not allowed to write into the Program Files subdirectory. As this is the recommended default Comm Bridge installation directory these user writeable files have been moved into the "%APPDATA%" directory.

This is a per user directory location. Thus, if multiple users have Comm Bridge execution authority, each user will maintain separate configuration and log file locations.

The locations of these files can be overridden using configuration changes with the Files – Setup dialog accessible from the Comm Bridge main window.

The following locations are the default locations for these configuration files:

For the .ini and .srv configuration files:

%APPDATA%\CA\Gen xx\cfg\cb

For the .log file:

%APPDATA%\CA\Gen xx\logs\cb

For the transaction mapping file used when the server configuration is set to support RSC/MP:

%APPDATA%\CA\Gen xx\cfg\cb\RSCMP\<Path-Mon>\tirtmt.tbl

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

<Path-Mon>

Specifies the RSC/MP configured Pathway Monitor configuration setting.

The default file names have not changed, only the default directory locations.

Chapter 3: Comm. Bridge General Configuration

This chapter discusses miscellaneous topics required for installing and configuring a Comm. Bridge. This chapter includes the following topics:

- Installing a Comm. Bridge
- Customizing a Comm. Bridge during the first startup following an install
- Comm. Bridge execution startup methods
- Details of the configuration items found on the Comm. Bridge File, Setup dialog
- Information for configuring Comm. Bridge client connections

Configuring a Comm. Bridge server connection is not covered in this chapter.

More Information:

[Configuring Comm. Bridge Server Connections](#) (see page 35)

Comm. Bridge Installation

The Comm. Bridge is installed using the CA Gen product install procedure. For more information, see the *Distributed Systems Installation Guide*. The Comm. Bridge has no dependency on any other CA Gen component, so the Comm. Bridge can be installed as a separate component.

Start Comm. Bridge After Install

During the first execution of a Comm. Bridge following its installation, a configuration dialog may be presented. This dialog allows you to choose the language used when displaying text in pop-up message boxes. If an existing configuration file is used during this first start and that file already contains a message language choice, then this dialog will not be presented. The default language is U.S. English.

If the Comm. Bridge configuration file (by default, iefcbn.ini) does not exist during the first startup after installation, you are prompted with a choice for executing the Setup dialog. If you accept the setup prompt, the File - Setup dialog is displayed giving you the opportunity to continue the configuration of the Comm. Bridge.

Subsequent executions do not require this first start configuration step.

Executing the Comm. Bridge

There are several ways in which a Comm. Bridge can be started:

- Using the Windows Start menu
- Using a desktop shortcut icon
- From a Command Prompt window
- As a Windows system service

Using the Desktop Start, All Programs Menu

To start a Comm. Bridge from the Windows desktop start menu, select Start, All Programs, CA, Gen <version>, Communications Bridge where, <version> is the CA Gen product version installed on your system.

Using a Desktop Shortcut Icon

A Comm. Bridge can be started from a desktop shortcut after the icon is created.

Follow these steps:

1. Right-click an empty area of the desktop
2. Select from the popup menu New, Shortcut
3. In the Create Shortcut dialog, click browse
4. Using the Browse for Folder dialog, select down the folder tree into the Comm. Bridge installation directory
5. Click the Comm. Bridge executable file IEFCBxxN.EXE.
Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.
6. Click Ok.
7. Click Next.
8. Change the name of the shortcut if desired, then click Finish. This should create a shortcut and place an icon on the desktop.
9. Locate the icon, right-click the icon and select the properties menu item.

10. In the edit field labeled Target, append to the file name the following parameters:

`iefcb startup /initfile=iefcbn.ini`

iefcb

Specifies the transaction code associated with the initial procedure step. This is a required parameter.

startup

Specifies the initial command executed by the initial procedure step. This is a required parameter.

/initfile=

Is followed by the name of the initialization file for this Comm. Bridge instance. Use the filename `iefcbn.ini` for the first execution. In later executions, use the name you give the initialization file during configuration. This parameter is optional if only one instance of the Comm. Bridge is needed. If not present, the default initialization file used is `iefcbn.ini`.

11. Click OK on the properties dialog to complete the shortcut configuration.

To Use the Comm. Bridge Shortcut:

To start the Comm. Bridge, double click on the shortcut icon.

From an MS-DOS Command Prompt

To start the Comm. Bridge from a command prompt, change to the directory where you installed the Comm. Bridge and enter the following command:

`IEFCBxxN.exe iefcb startup /initfile=[filename]`

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

IEFCBxxN.exe

Specifies the name of the Comm. Bridge executable

iefcb

Specifies the transaction code associated with the initial procedure step. This is a required parameter.

startup

Specifies the initial command executed by the initial procedure step. This is a required parameter.

/initfile=

Is followed by the name of the initialization file for this Comm. Bridge instance. Use the filename iefcbn.ini for the first execution. In later executions, use the name you gave the initialization file during configuration. This parameter is optional if only one instance of the Comm. Bridge is needed. If not present, the default initialization file used is iefcbn.ini.

As a Windows System Service

The Comm. Bridge can execute as a Windows System Service. As with any other Windows Service, a Comm. Bridge configured to run as a service can be started automatically when the Windows operating system is booted or it can be started manually. For more information about how a Comm. Bridge can be set up to run as a Windows System Service, see "[Configuring a Comm. Bridge as a Windows System Service](#) (see page 65)."

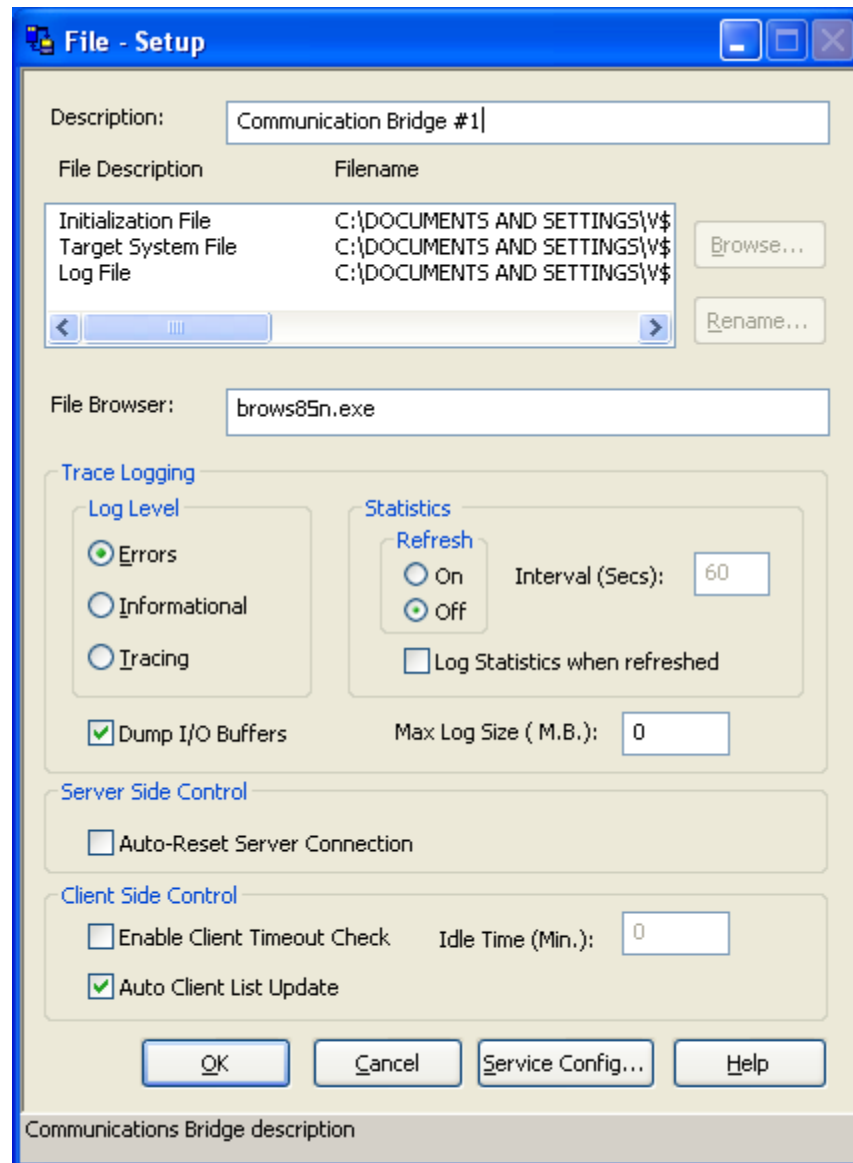
More Information:

[Configuring a Comm. Bridge as a Windows System Service](#) (see page 65)

Comm. Bridge Setup Dialog

This section covers those configuration items found on the Comm. Bridge File - Setup dialog. While some items may be discussed elsewhere within this document, this section is intended to provide the details about the parameters in one location.

From the Comm. Bridge main GUI window, select the File - Setup menu item. The following figure shows the Comm Bridge File - Setup dialog:



Description

This text data field allows entry of the name for the current Comm. Bridge configuration. The value you enter here appears in the Description field in the main window.

File Description List

This list contains the names of Comm. Bridge configuration and log files. Each file in this list can be given a custom name using the Rename... push button. The Browse... push button is used to view the contents of the file currently highlighted in the list.

Initialization File

The default is IEFCBN.INI. This file contains the current configuration data for those items found on the File - Setup dialog.

Target System File

The default is IEFCBN.SRV. This file contains configuration data needed to establish communications with a target server.

Log File

The default is IEFCBN.LOG. This file contains messages encountered during execution of the Comm. Bridge.

Browse

When you select a row from the File List and click Browse..., the file browse utility displays the selected file.

Rename

When you select a row from the File List and click Rename..., the Comm. Bridge displays the Setup - Change Filename dialog box. This dialog allows the user to customize the name of the selected file.

File Browser

A text entry field used to specify the name of the utility program used to view text files. Any viewer capable of reading text files may be used. The full pathname of the file viewer must be used if it is not found within the execution environment PATH variable.

Auto-Reset Server Connection

The Auto-Reset Server Connection check box, if checked, allows an automatic disable followed by enable of the server connection after the server configuration has been modified.

Disabling a server

- Disconnects the server connection.
- Removes all data pertaining to that connection.
- Makes the server unavailable for use.

Enabling a server makes the server available for use.

If the check box is not checked, then the server connection must be manually reset before any server configuration changes are put into use.

Trace Logging

Following are the information fields in the trace log file:

Log Level

These three radio buttons control the level of information that is output to the trace log file.

Errors

Use this setting for normal operations. Only information pertaining to error events will be logged to the log file.

Informational

Select the Informational radio button if you want to see error events and informational messages concerning transaction flow.

Tracing

Use this setting if you are attempting to debug a communications error. This setting shows all of the error events, informational messages, and message data. This setting can create rather large log files.

Statistics

These configuration parameters control enabling, frequency and logging of periodic updates of the statistical data displayed in the Statistics - Summary dialog. The displayed data includes data bytes transferred and transaction counts for the server and any connected clients.

More Information:

[Comm. Bridge Transaction Statistics](#) (see page 83)

Refresh On/Off

These radio buttons enable or disable the periodic update of the Statistics – Summary dialog. The default value is Off. Select the On button to enable the periodic update feature. The Summary – Statistics dialog will not be automatically updated if the refresh feature is not active. You can still manually update the statistics by selecting the Statistics, Refresh now Comm. Bridge main window menu items.

Interval

The Interval entry field controls the length of time in whole seconds between refreshes of the Summary – Statistics dialog. The default value is 60 seconds.

Log Statistics When Refreshed

If this check box is selected statistical data is written to the trace log file periodically based on the setting of the refresh interval described above. Statistical data can also be written to the log file if manual update of the statistics is selected using the Statistics, Refresh now Comm. Bridge main window menu item.

Dump I/O Buffers

This check box enables additional I/O buffer data to be output to the trace log file.

Max Log Size (M.B.)

This entry field controls the maximum size of the trace log file. After the trace log reaches this size, the log file will be renamed to a backup name. Trace logging then continues to the original file name. There can be at most two trace log files, the currently active log and the previously saved log. The log size entered in this field is to be specified in units of megabytes. However, the corresponding MAXLOGSIZE property as stored in the iefcbrn.ini file is expressed as bytes.

Client Monitor

The Client Monitor Enable and Idle Time fields control how long a client remains connected to the Comm. Bridge after a completed cooperative flow. A flow is considered complete when the target server has returned its response back to the client. If the target server has not yet responded, possibly due to a long running process, its associated client will not be subjected to the idle time check. Only completed flows are subject to idle time checks.

Any client connections that have been idle longer than the Idle Time period are subject to automatic disconnection. To reduce timeout thread processing overhead, the Comm. Bridge is not using individual timer events per connection to drive the timeout expiration process. Periodically, (partially derived from the configured timeout value) the Comm. Bridge runs through its list of client connections looking for those that meet the timeout criteria. There is a trade off between timely disconnect of idle clients and constant processing of the client connection list. Small values of timeout should not be allowed to consume a large percentage of CPU processing time, nor should there be an extremely long disconnect lag time when a large timeout value is specified.

Depending upon the current idle time of a particular client, when the client list is processed, client disconnect can happen at two extreme time intervals or at any time between those intervals. These time values are somewhat dependent upon the configured timeout interval.

The minimum and maximum disconnect times are:

Configured Timeout Value	Min. Timeout period	Max. Timeout period
1 to 2 Min.	Timeout	Timeout + 1 Min.
2 to 20 Min.	Timeout	Timeout +1/2 of Timeout
20 + Min.	Timeout	Timeout + 10 Min.

Enable

This check box enables the idle client timeout feature. This allows any idle clients to be automatically disconnected after the defined timeout period.

Idle Time (Min)

This field specifies the number of minutes a client must be inactive before being subjected to automatic disconnection by the Comm. Bridge.

Note: If the idle time is 0, no client connection checks will be done.

Auto Client List Update

If checked, the refresh client name list checkbox enables the refreshing of the client name list found in the main Comm. Bridge window. Disabling refresh may improve performance. However, real time update of the list of connected clients would be lost. This can be enabled or disabled dynamically and does not require a Comm. Bridge restart. The checkbox is checked by default.

Service Config Button

The Service Config pushbutton allows access to the Service Configuration Details dialog box. This dialog box is used to define and install the Comm. Bridge to run as a Windows system service.

Configuring Comm. Bridge Client Connections

The client side of the Comm. Bridge contains the connection support for communicating with clients using TCP/IP (Sockets API). A Comm. Bridge can manage multiple, simultaneously connected clients.

The Comm.Bridge supports connections from GUI or Proxy clients to facilitate connections to backend servers. As a result the Comm.Bridge acts as a target server for these types of clients.

Configuring TCP/IP (Socket) Client Connections

The Comm. Bridge TCP/IP implementation makes use of stream sockets. A stream socket provides a full-duplex, sequenced, reliable transmission mechanism, over which a DPC cooperative flow byte stream can flow. The socket provides the TCP communications endpoint, used by the Comm. Bridge, to gain access to the IP network.

A Comm. Bridge is connected to the IP network and listens for inbound connections using its well-known port number. This well-known port number is configured within the Comm. Bridge. The Comm. Bridge port listening subsystem will handle concurrent connections originating from either IPv4 or IPv6 clients.

When a client successfully connects to a Comm. Bridge the client is added to the client list found on the Comm. Bridge main GUI window. This scrollable list displays an entry for each connected client. Each client entry displays the client's host name or IP address. The IP address can be either an IPv4 or IPv6 address depending on the protocol used for a successful connection. Each entry also includes a time/date stamp of the last flow activity to or from the client as well as a running byte count totals. Selecting a client from the client list will result in appropriate connection configuration details being displayed in the Selected Client Details area just below the list. A client is removed from the list when it becomes disconnected from the Comm. Bridge.

By default, a domain name service (DNS) host name lookup is performed to obtain the name of the client workstation. Optionally, if desired for performance reasons, the DNS host name lookup can be skipped and a client IP address displayed instead. This behavior can be modified using the Comm. Bridge IEFNET TCP/IP user exit. For more information about this user exit, see "[Comm. Bridge User Exits](#) (see page 89)."

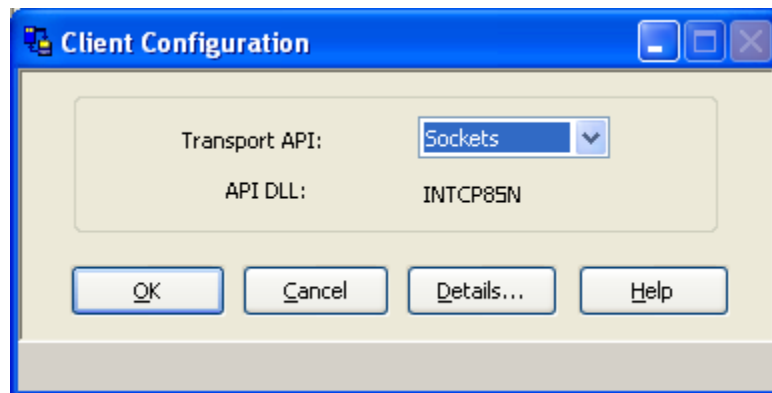
The following procedure can be used to configure the Comm. Bridge well known client port number.

Configuring a Client Connection for TCP/IP (Sockets)

Follow these steps:

1. Start the Comm. Bridge.
2. Navigate to the Client Configuration dialog. From the Comm. Bridge main GUI window, select menu item Client,Config.

The Client Configuration dialog appears.



3. Select Sockets from the Transport API selection list.
The Comm. Bridge automatically supplies the API DLL name.
4. Click Details to display the TCP/IP communication parameters.

5. Provide the TCP/IP source port number that this Comm. Bridge can advertise as its well-known port address and click OK.
6. Click OK button from the Client Configuration window to return to the Comm. Bridge's main GUI window.
7. Save the current configuration. For more information about saving configurations, see "[Saving Configuration Files](#) (see page 77)".

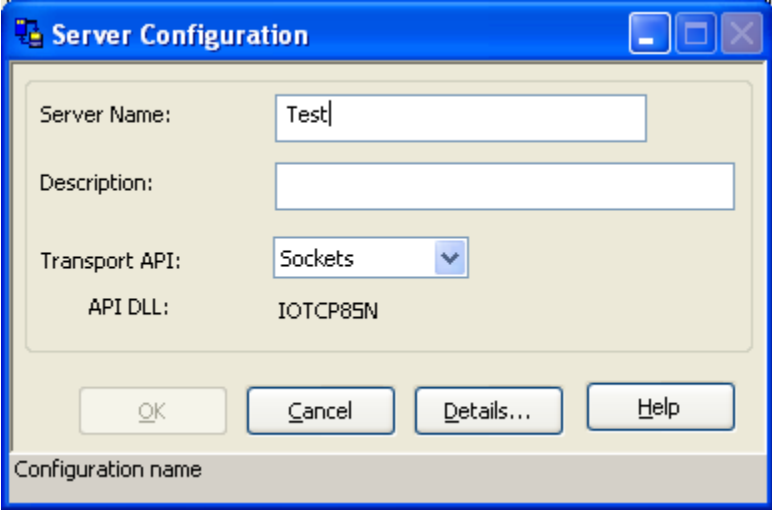
Note: You must restart the Comm. Bridge for the newly configured client transport parameters to take effect. The Comm. Bridge will report an error at startup time if the port chosen is already in use by another application on the workstation. When an error occurs a message box will be displayed. After pressing OK on the message box, the Comm. Bridge will continue to start. Once started the conflicting port number can be modified. The above procedure, Configuring a Client Connection for TCP/IP (Sockets), provides details for modifying the client port number.

Chapter 4: Configuring Comm. Bridge Server Connections

The Communications Bridge (Comm. Bridge) has the capability of communicating with a CA Gen server environment using one of four supported communications protocols. This chapter details the process of configuring the Comm. Bridge to communicate with the various target server execution environments using LU6.2, TCP/IP, CICS External Call Interface (ECI), or Remote Server Call (RSC/MP).

Server Configuration

The following dialog is presented by the Comm. Bridge for detailing the server configuration. The Server Configuration dialog is launched from the Comm. Bridge main GUI window by selecting the Server, Config... menu item.



Server Configuration

Server Name: Test

Description:

Transport API: Sockets

API DLL: IOTCP85N

OK Cancel Details... Help

Configuration name

Server Name

A label used to refer to the target server within the Comm. Bridge. The label is typically used when building log messages.

Description

The description field is an optional text field that provides a text description of the associated target server.

Transport API

The Transport API designates which communications protocol will be used to communicate with the associated target execution environment. The Comm. Bridge supports the following transport/protocols serving the listed target server environments:

- LU6.2 (CPI-C)
 - z/OS CICS
 - z/OS IMS
- TCP/IP (Sockets)
 - UNIX: CA Gen Transaction Enabler
 - UNIX: Tuxedo (using CA Gen Tuxedo Proxy Client)
 - Windows: CA Gen Transaction Enabler
 - Java EJB (using CA Gen EJB CFB Server)
 - z/OS CICS (using CA Gen CICS Socket Listener)
 - z/OS IMS: (using CA Gen IMS TCP/IP Direct Connect)
- ECI—CICS External Call Interface
 - z/OS CICS
- RSC/MP—Remote Server Call
 - NonStop Pathway

Transport API – Additional Details

The Comm. Bridge can use one of the following provided transport protocols when communicating with a target server execution environment.

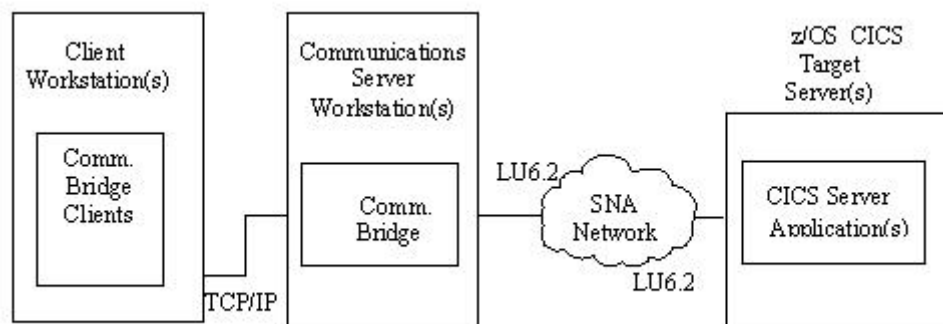
The following sections provide a description of the configuration details of:

- LU6.2 CPI-C Connections
- TCP/IP Socket Connections
- ECI—CICS External Call Interface
- NonStop RSC/MP Connections
- Other—(available for use only when directed by Technical Support)

LU6.2 CPI-C Connections

Cooperative flows from a client application can pass through a Comm. Bridge and then on to a CICS or IMS target server environment using an SNA Independent LU (ILU). Each cooperative flow results in an LU6.2 conversation being established between the Comm. Bridge and the target server environment (CICS or IMS). Each Common Format Buffer (CFB) is transmitted over its own LU6.2 conversation. The conversation is completed when a corresponding response is returned to the Comm. Bridge.

The following diagram illustrates a Distributed Processing application in an LU6.2 communications environment.



Configuring VTAM and SNA Communication Parameters

Configuring an LU6.2 transport requires a usable SNA network configuration. Defining a Comm. Bridge to use LU6.2 requires certain information from VTAM and other components that define the SNA Network such as NCP definitions. You need to be familiar with the task required to configure the third party products that provide the SNA support to the Comm. Bridge.

For a list of third party products offering SNA supported for use with the Comm. Bridge, see the *CA Gen Technical Requirements* document.

You will need to obtain parameters related to the z/OS host's SNA environment. For a CICS or IMS target server, these parameters are found in the VTAM and NCP configuration files. They include:

- For CICS, the VTAM application LU name of the region
- For IMS, the VTAM application LU name for the z/OS APPC address space (for IMS)
- For certain SNA products, the Local LU Alias the Comm. Bridge will use when allocating its LU6.2 conversations.
- The VTAM Log Mode table entry of a logon mode entry that allows for parallel LU6.2 sessions

The details for setting VTAM, NCP, CICS, and z/OS APPC address space parameters can be found in your IBM or Microsoft documentation associated with their respective SNA product.

Configuring for an SNA Target Server

After the VTAM, NCP, CICS, or APPC z/OS configurations has been completed on the host, definition values from these configurations are used to configure the SNA products on the workstation to support LU6.2 communications to z/OS (CICS or IMS).

After the third party SNA product that will provide the Independent Logical Unit (ILU) to the Comm. Bridge is setup and is capable of connecting to the SNA network, the Comm. Bridge can complete its configuration of a target server. The Comm. Bridge needs to configure its target server to use CPI-C as its transport API.

Details for Configuring for LU6.2

When configuring a Comm. Bridge to use CPI-C for a target server, the following parameters from the SNA network configuration are required:

- Partner LU name of the application LU name for CICS or z/OS APPC address space (for use with IMS)
- Local LU Alias

Note: This field may or may not be used depending on the SNA service provider. Certain SNA providers allow an application to specify the Local LU Alias prior to allocating a conversation. Others select the LU based on their own defined convention. This field is ignored if the SNA service provider does not allow the Local LU Alias to be specified by the application program.

For more information about how a Local LU is selected for use by a CPI-C application program, see the third party vendor documentation.

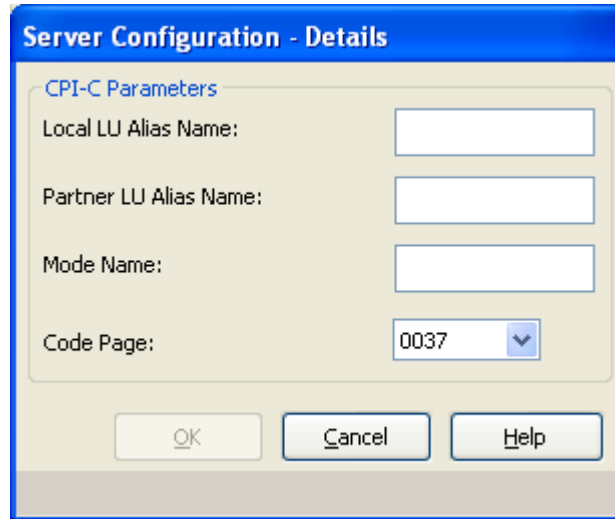
- Mode Table Entry name
- Code Page number

To set the Comm. Bridge server configuration parameters:

Follow these steps:

1. Start the Comm. Bridge and select Server > Config...
2. Enter a server name and description for this target server connection.

3. Select CPI-C from the Transport API selection list. The Comm. Bridge automatically supplies the API DLL name.
4. Click Details to display the LU6.2 (CPI-C) Server Configuration – Details dialog.



Server Configuration - Details

CPI-C Parameters

Local LU Alias Name:

Partner LU Alias Name:

Mode Name:

Code Page: ▼

5. Provide the LU6.2 parameter values listed above.
6. Select an appropriate Code Page value. The Code Page field allows you to select from a list of supported code pages for translation. Only the header portion of the Common Format Buffer (CFB) is translated. The view data contained in the cooperative flow CFB is not subject to translation. The supported code pages include:

0037

Australia, Brazil, Canada, The Netherlands, Portugal, New Zealand, South Africa, United States of America (USA)

0273

Austria, Germany

0277

Denmark, Norway

0278

Finland, Sweden

0280

Italy

0284

Spain

0285

Ireland, United Kingdom (UK)

0297

France

0500

Belgium, Switzerland

Click OK to return to the Server Configuration dialog.

7. Click OK to return to the main menu.
8. Save the current configuration. For more information about saving configurations, see "[Saving Configuration Files](#) (see page 77)."

TCP/IP/Socket Connections

For TCP/IP server connections, you must set server configuration parameters within the Comm. Bridge corresponding to the target server environments:

- UNIX Transaction Enabler
- UNIX Tuxedo Proxy Client
- Windows Transaction Enabler
- z/OS CICS Socket Listener
- z/OS IMS TCP/IP Direct Connect
- CA Gen EJB Converter Service

Configuring for a TCP/IP Target Server

The Comm. Bridge TCP/IP implementation makes use of stream sockets. A stream socket provides a full-duplex, sequenced, reliable transmission mechanism, over which a cooperative flow request can be transmitted. The socket provides the TCP communications endpoint used to gain access to the IP network.

A target server environment that is connected to the IP network listens for inbound connections and data transmissions using a socket that is located on the system hosting the target server.

Used by the Comm. Bridge, a socket consists of two parts, each defining application endpoints in the TCP communications. One of the application endpoints is the local application (that is, the Comm. Bridge). The other endpoint is the remote application. That is, the TCP front-end process of the target server environment. Examples include the Transaction Enabler AEFUF, Tuxedo Proxy Client, z/OS CICS Socket Listener or z/OS IMS Direct Connect, EJB Converter Service. These application endpoints are also referred to as the source and destination, respectively.

Each connection that a client TCP application establishes on a workstation is identified by its unique TCP port address. When a connection to a target server is requested, the source port address is selected by TCP from those port numbers not already in use by other TCP connections. The destination port address is the target server environment's well-known port address. The IP address of the target machines, along with its port address, is used by the Comm. Bridge in the construction of the destination portion of the TCP socket.

The Comm Bridge uses a system level socket support API to obtain all available IP addresses for the target server host machine. The Comm Bridge server connection subsystem will then iterate through all available IP addresses until a valid connection is made. The ordering of the reported IP addresses is determined by the Comm Bridge's host operating system configuration.

The Comm Bridge makes no attempt to prefer IPv4 connections over IPv6 or vice versa. The IP addresses are iterated through in the order received from the underlying operating system.

No special user interaction is required at the Comm Bridge to enable this support.

To configure a Comm. Bridge to communicate with a TCP/IP server environment you need:

- The target server machine host name or its IP address.
- The target server configured/well-known port address (destination port number)

The network address associated with the machine name of the target server environment can be resolved using either the workstation hosts file or by way of a Domain Name Services (DNS).

To modify the hosts' file or the DNS configuration on the client workstation, see the appropriate vendor documentation for instructions on using hosts file or DNS setup.

Target Server Environment Communication Styles

The Comm. Bridge supports 3 distinct styles of communications to target servers using the TCP/IP Sockets protocol. The three styles are configurable through check boxes on the Sockets Server Configuration – Details dialog. The state of the check boxes depends on the target server environment requirements.

All three use the same features of the underlying TCP/IP protocol. Two of the styles differ in the length of the life span of the connection, the third in how the Common Format Buffer (CFB) data is packaged.

Connection Life Time

Two different connection life time modes are supported. The mode chosen, persistent vs. non persistent, is dependent on the needs of the target server environment.

Persistent socket connections are those connections that are long lived. Once a connection is established that same connection object is used for multiple client to server transactions. The connection is maintained until the client or server application performs a connection disconnect or exits. This type of connection is used by the Transaction Enabler (TE) on UNIX and Windows, z/OS IMS TCP/IP Direct Connect, z/OS CICS TCP/IP Direct Connect, EJB CFB Server and the Tuxedo Proxy Client on UNIX.

Non persistent socket connections are maintained for the duration of a single client to server transaction. After the server returns a response buffer back to the client the server will close its connection. A subsequent client request to the same server requires that a new connection be established. This type of connection is supported only when connecting with the z/OS CICS Socket Listener.

Common Format Buffer Packaging

The third TCP/IP communication style concerns how the Common Format Buffer data is delivered to the target server. There are two choices, wrapped for IMS support or unwrapped.

Communications to z/OS IMS TCP/IP Direct Connect require the CFB to be wrapped inside of IMS header data prior to it being sent to the IMS target server environment. Only z/OS IMS Direct Connect has this requirement. All other TCP/IP target server environments require a normal or non wrapped CFB.

The three styles are configurable through check boxes on the Sockets Server Configuration – Details dialog. The state of the check boxes depends on the target server environment and is detailed in the following section.

Details for Configuring TCP/IP (Sockets)

To set the Comm. Bridge configuration files for TCP/IP transport:

Follow these steps:

1. Start the Comm. Bridge and select Server, Config...
2. At the Server Configuration dialog, enter a server name and description for this target server connection.

3. Select Sockets from the Transport API selection list. The Comm. Bridge automatically supplies the API DLL name.
4. Click Details to display the TCP/IP Server Configuration – Details dialog.

The following figure shows the details dialog for socket configuration:

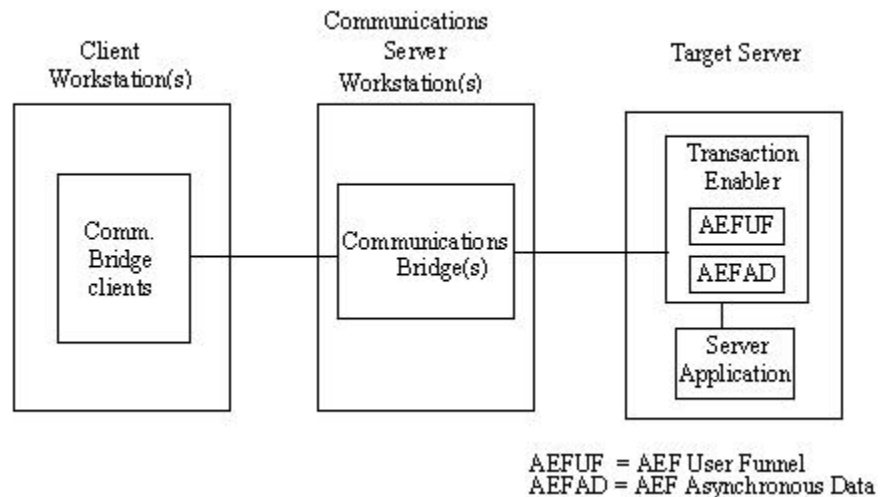
5. Provide the TCP/IP host and port parameters that identify the target server environment. An IP address may conform to either the IPv4 or IPv6 formats.
6. Select one, if appropriate, of the socket connection style check boxes according to the target server environment requirements as listed below.

Target Server Environment	CICS Socket Listener check box	IMS TCP/IP Direct Connect Host check Box
Transaction Enabler (TE - Windows or UNIX)	Unchecked	Unchecked
UNIX: Tuxedo Proxy Client	Unchecked	Unchecked
z/OS CICS TCP/IP Direct Connect	Unchecked	Unchecked
EJB CFB Server	Unchecked	Unchecked
z/OS IMS TCP/IP Direct Connect	Unchecked	Checked
z/OS CICS Socket Listener	Checked	Unchecked

7. Click OK to return to the Server Configuration dialog.
8. Click OK from the Server Configuration dialog to return to the main window.
9. Save the current configuration. For more information about saving configurations, see "[Saving Configuration Files.](#)" (see page 77)

Windows/UNIX Transaction Enabler

The Transaction Enabler AEFUF process listens for input from client connections using its well-known port address. AEFUF passes messages to the AEFAD process, which, if necessary, activates the requested DPS application. When the Distributed Process Server (DPS) application processing is complete, a response message is passed back to the Comm. Bridge and then to the originating Distributed Process Client (DPC) application.

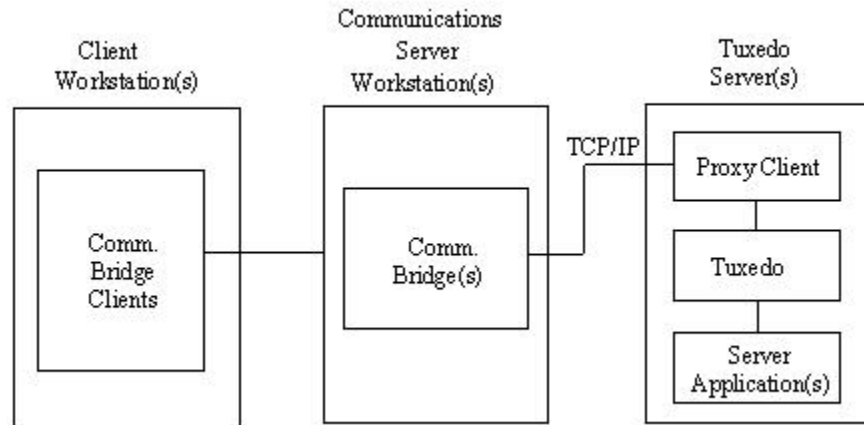


To configure for TCP/IP support for a Transaction Enabler, you must configure AEFUF. Cooperative flows must make use of the AEFUF process when connecting to a UNIX or a Windows AEFAD transaction processor monitor.

For configuration information of both the AEFUF and AEFAD components, see the *Transaction Enabler User Guide*.

UNIX: Tuxedo Proxy Client

CA Gen offers a TCP/IP sockets communications interface to Tuxedo based servers. This processing makes use of the CA Gen Tuxedo Proxy Client interface. The proxy client listens for input from clients using its well-known port address. Messages are passed to the Tuxedo Proxy client. The Proxy client passes the message to the Tuxedo TP monitor for processing. When the DPS application processing is complete, a response message is passed back to the Comm. Bridge and on to the DPC application.



To configure TCP/IP support for Tuxedo, you must configure the CA Gen Tuxedo Proxy Client for connection to the Tuxedo transaction processor monitor.

For Installation and configuration information about the CA Gen Tuxedo Proxy Client component, see the *Tuxedo User Guide*.

For information about configuring and starting Tuxedo, see your vendor documentation.

z/OS CICS TCP/IP

CA Gen offers two connection implementations that use CICS TCP/IP Sockets:

- CICS Socket Listener

This implementation uses the CICS Socket Listener program TISRVLIS. TISRVLIS passes the connection socket to the Gen server application. The application server manages the socket and closes the socket when execution completes. This is known as a non persistent socket connection as the socket connection is not maintained from one client request to the next.

- Direct Connect for CICS

This implementation uses a persistent socket connection where the single socket connection is maintained throughout the life of the application.

Note: Direct Connect for CICS support is no longer supported. Earlier versions of this implementation will be supported as long as the software release in which it was delivered is supported.

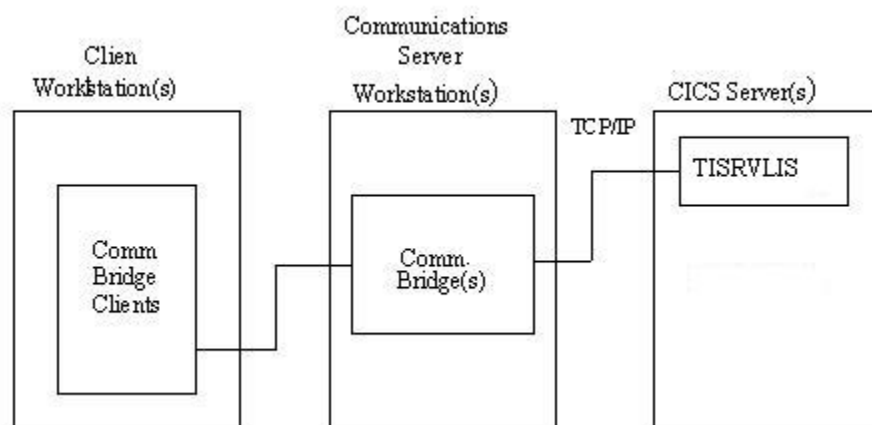
z/OS CICS TCP/IP – CICS Socket Listener

- The CICS Socket Listener is the preferred CICS TCP/IP connection mechanism. This version supports a non persistent socket which means a new socket connection is created for each client/server request/response. All supported client side runtimes have been modified to allow support for this non persistent behavior.
- If this listener is used with the Comm Bridge the server configuration details must be set with the CICS Socket Listener check box selected as mentioned in the Details for Configuring TCP/IP (Sockets) section above.

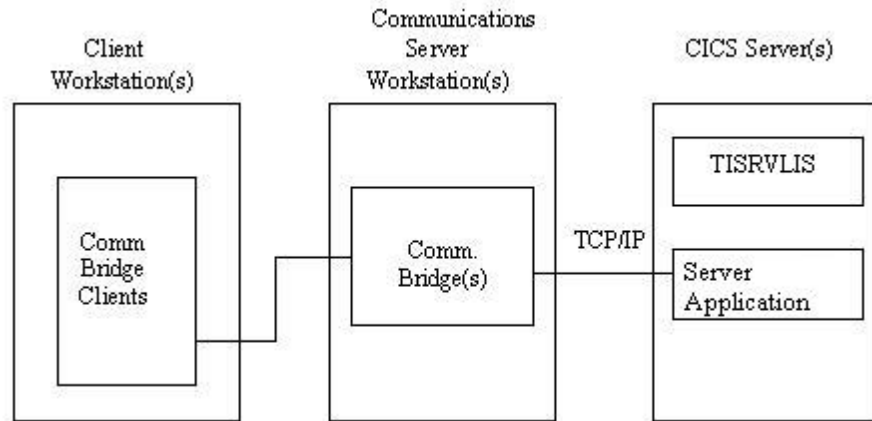
The CICS Socket Listener implementation uses the CICS program TISRVLS. TISRVLS listens for connection requests from the Comm Bridge using its well-known port address. When a request is accepted TISRVLS passes the connection request to the appropriate CA Gen application server. The CA Gen application server then assumes responsibility for creating, managing, and closing the socket connection.

When DPS application processing is complete, a response message is passed from the server application to the Comm. Bridge and on to the DPC application.

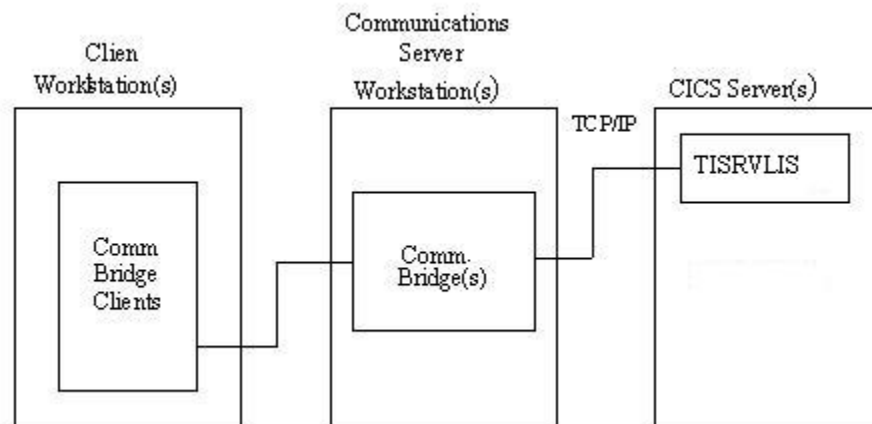
The following diagram illustrates a Distributed Process application prior to a DPC request being serviced by the target server environment TISRVLS process. The Comm. Bridge initiates a bind socket request to the TISRVLS process's well-known port address.



The following diagram illustrates a Distributed Process application at the time the Comm. Bridge request has been accepted and the application server process has taken responsibility for the connection. The TISRVLS process is now available to listen for additional connections from the Comm. Bridge.



The following diagram illustrates a Distributed Process application after the server application has completed. The socket connection used for the transaction has been closed. The TISRVLS process is handling another connection request from the Comm Bridge and the cycle repeats.



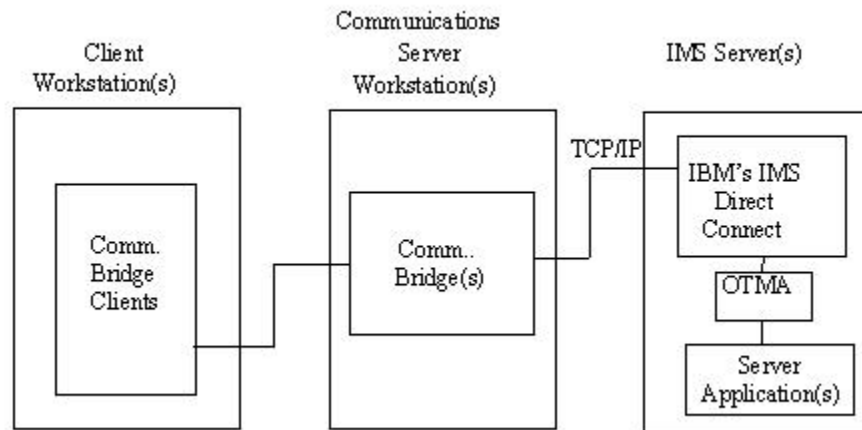
Installation and configuration information for the TISRVLS component appears in the *Installation Guide for Host Encyclopedia and Host Construction Installation Guide* and in the *z/OS Implementation Toolset Installation Guide for z/OS*.

z/OS CICS TCP/IP Direct Connect

Note: This implementation of CICS TCP/IP Direct Connect support using TILSTNR and TICONMGR is no longer supported. Earlier versions of this implementation will be supported as long as the software release in which it was delivered is supported.

z/OS IMS: IMS TCP/IP Direct Connect

TCP/IP support for z/OS IMS uses the CA Gen IMS TCP/IP Direct Connect product and the IBM IMS Connect product on the z/OS machine. The IMS Connect product listens for input from the Comm. Bridge. IMS Connect receives the input message and invokes a CA Gen message exit to start the IMS DPS application using the IMS OTMA interface. When the DPS application processing is complete, a response message is passed back to IMS Connect that returns it to the Comm. Bridge and on to the DPC application.



Installation and configuration information for the IMS TCP/IP Direct Connect components appears in the *Host Encyclopedia and Host Construction Installation Guide* and in the *z/OS Implementation Toolset Installation Guide*.

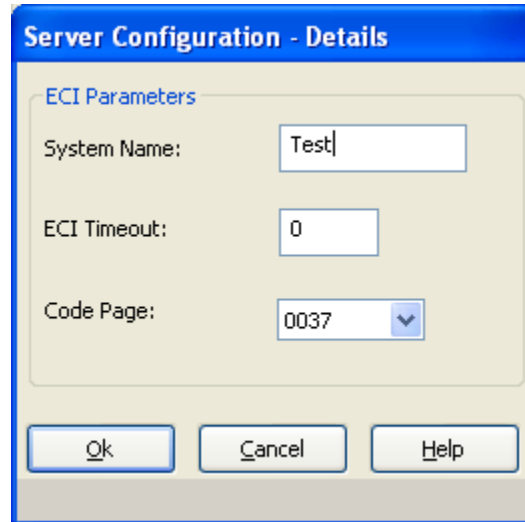
The hardware and software vendor documentation will assist in the setup and testing for communications with TCP/IP.

z/OS CICS External Call Interface (ECI)

The CA Gen ECI communications runtime provides the interface mechanisms that CA Gen DPC applications can use to externally call a CICS DPS application using a CICS Distributed Program Link (DPL). The Target DPS appears as if it were called using an EXEC CICS LINK with the COMMAREA option.

3. Select ECI from the Transport API selection list. The Comm. Bridge automatically supplies the API DLL name.
4. Click Details to display the ECI Server Configuration – Details dialog.

The following figure shows the details dialog for ECI configuration:



5. Provide the name of the CICS region containing the target server as defined in the server section of the IBM CICS Transaction Gateway's ctg.ini file. (The ctg.ini is provided as part of the Transaction Gateway product, and is used by the Universal Client).
6. Set the desired ECI Timeout value to specify the maximum time in seconds that an ECI request is allowed to take. A timeout occurs if the servicing of a request takes longer than the specified time. The value must be in the range 0 through 9999. A value of zero, the default, indicates that the request will not be timed out.

Note: A non-zero value is not recommended unless you have read and understood the advice given in the IBM CICS Transaction Gateway for Windows documentation.

7. Select the appropriate Code Page value. The Code Page field allows you to select from a list of supported code pages for translation. Only the header portion of the Common Format Buffer (CFB) is translated. The cooperative flow view data is not subject to translation. The supported code pages include:

0037

Australia, Brazil, Canada, The Netherlands, Portugal, New Zealand, South Africa, United States of America (USA)

0273

Austria, Germany

0277

Denmark, Norway

0278

Finland, Sweden

0280

Italy

0284

Spain

0285

Ireland, United Kingdom (UK)

0297

France

0500

Belgium, Switzerland

8. Click OK to return to the Server Configuration dialog
9. Click OK from the Server Configuration dialog to return to the main window.
10. Save the current configuration using the procedures in the chapter "[Saving Configuration Files](#) (see page 77)".

NonStop RSC/MP Connections

Communications targeting NonStop servers relies on a third party product called HP NonStop Remote Server Call (RSC/MP). You must obtain this product from the vendor as it is not included as part of the CA Gen installation package.

Installing and Configuring RSC/MP

After you install RSC/MP, ensure the path to its /bin directory is included in the %PATH% environment variable. This is required as the Comm Bridge transport DLL depends on RSC/MP DLL's installed within the bin directory. For more information about installing and configuring RSC/MP, see the *HP NonStop Remote Server Call (RSC/MP) Installation and Configuration Guide*.

The following examples show how to customize the RSC/MP files for use within the CA Gen environment.

RSC.ini customization

A sample content from a typical <RSC Install Dir>\bin\RSC.ini file is shown next. These configuration items should be merged within the existing RSC.ini file as this list is not all inclusive.

The following example defines a new section called GEN_RSC, which is also the name used for the Comm Bridge Initialization Section Name configuration as shown next.

Note: Use [] when defining sections within the RSC.ini file.

```
[GEN_RSC]
;
; Set the fully qualified RSC/MP error file name.
; Assume RSC/MP is installed in C:\rsc
error_file      = c:\rsc\bin\RSC.ERR

; The following options are required for Piccolo connections.
; set them to appropriate values.

; this is the default
subsystem_name  = RSCPIPE      ;( default = RSCPIPE )

; the host_pipename is typically the host name of the target NonStop
; machine. Replace SOME_NONSTOP_HOST with the target server name
host_pipename   = RSC@SOME_NONSTOP_HOST
; The value of the writeread_pathmon entry should be that of the
; name of the Pathmon process for the CA Gen application that
; has or will be installed on the NonStop server. This name must
; match the name entered in the Setup Tool when the application was
; or will be installed.
writeread_pathmon = $TEST
```

Note: RSC/MP test programs use the default [RSC] section within RSC.ini for testing client/server communications. This named entry must remain within the RSC.ini file if you want to successfully test the base RSC/MP installation. Some of values used in the RSC.ini file configuration rely on RSC/MP server side configuration settings. For more information about your RSC/MP installation at your site, contact your system administrator.

Pipe.ini customization

A sample content from a typical <RSC Install Dir>\bin\PIPE.INI file is shown next. These configuration items should be merged within the existing pipe.ini file as this list is not all inclusive. This data is used internally by RSC/MP. Since %RSC_INSTALL_DIR% is NOT an environment variable that is created when RSC is installed, it should not be referred as one. Any field that has % should or a \$ in front of it is assumed to be an existing environment variable.

```
[PIPEMAN]
```

```
; SystemName is the name of the client workstation where  
; the RSC/MP client and Gen Client Manager is installed.  
SystemName=Client Workstation
```

```
; DomainName is the domain where the SystemName machine resides  
DomainName=CA.COM
```

For more information about configuration and basic communication testing, see the *HP NonStop Remote Server Call (RSC/MP) Installation and Configuration Guide*.

Configuring the Comm. Bridge for NonStop RSC/MP

For RSC/MP server connections, you must set server configuration parameters within the Comm. Bridge corresponding to the NonStop target server environment.

Note: Asynchronous flows, to a NonStop server or from a NonStop server, are not supported through the Comm Bridge.

You can select the RSC/MP target server transport type for connecting to the NonStop host platforms.

Follow these steps:

1. Start the Comm. Bridge and select Server, Config...
The Server Configuration dialog is displayed.
2. Enter a server name and add a brief description for this target server connection.
Select RSC/MP in the Transport API list.

3. The Comm. Bridge automatically supplies the API DLL name.

Click Details... to display the following Server Configuration – Details dialog:

Server Configuration - Details

RSC/MP Parameters

Initialization Filename: RSC.ini

Initialization Section Name: GEN_RSC

User Exit DLL: RSCUX85N.dll

OK Cancel Help

Enter the name of the User Exit DLL

4. You can specify the RSC/MP configuration properties in this dialog.

Initialization Filename

Identifies the name and location of RSC/MP communications product's initialization file. This file is located in the RSC/MP install bin directory. If this directory has been added to the PATH environment variable, a full path name is not required. For example, RSC.ini.

Initialization Section Name

Identifies the section name within the RSC.INI file that contains the target server configuration information. For example, GEN_RSC.

User Exit DLL

Identifies the Comm. Bridge RSC/MP user exit DLL name. If a name is not specified, then the User Exit will not be called.

Note: Check your NonStop RSC/MP Transport API documentation for case-sensitivity requirements for parameter fields.

5. Click OK to return to the Comm. Bridge main window.
6. Save the configuration.

For more information about saving a configuration, see the chapter "[Saving Configuration Files](#) (see page 77)."

7. The Comm. Bridge is now configured for NonStop RSC/MP.

Transaction Mapping Table

When connected to a server transaction mapping data is transferred from the host and is stored in a local file. This file is located in the following directory:

`%APPDATA%\CA\Gen xx\cfg\db\RSCMP\<Path-Mon>\tirtmt.tbl`

Where %APPDATA% corresponds to the directory location the Windows operating system uses for the user id which is executing the Comm Bridge.

<Path-Mon> corresponds to the name of the configured Pathway Monitor on the NonStop system. The content of the file is used internally by the Comm Bridge and should not be modified.

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

Verifying RSC/MP Communications

Communications between the Windows workstation and the NonStop server must be established. RSC/MP is not operational without a working transport. RSC/MP supplies a file, `rsctestw.exe`, which can be used to test the RSC/MP configuration on the workstation. Do not continue if you have not successfully executed this program. For more information about executing this program, see the *HP Remote Server Call (RSC/MP) Installation and Management Guide*.

Other API

The Other API protocol is a server side transport that is intended for use under the guidance of Technical Support.

Chapter 5: Communications Bridge Security

The Comm. Bridge transmits the Common Format Buffer (CFB) that it receives from the DPC on to the DPS without performing any security validation of its own. However, the Comm. Bridge does parse the CFB to extract security data (UserID and Password) when processing a cooperative flow. The Comm. Bridge sets up the same interface to each of its server transports. The passing of security data to the communications runtime is done regardless of type of server side transport.

For a general discussion covering CA Gen security, see the chapter "Security" in the *Distributed Processing – Overview Guide*.

CFB Security

The security content contained within a CFB is influenced by a client side security user exit. The name of the client side user exit depends on the type of client. The following is a list of clients and their respective client side security user exits:

- Windows GUI clients
WRSECTOKEN (wrexitn.c)
- C/COM Proxy
WRSECTOKEN (proxyxit.c)
- Java Proxy
CFBDynamicMessageSecurityExit
(CFBDynamicMessageSecurityExit.java)
- .Net Proxy
CFBDynamicMessageSecurityExit
(CFBDynamicMessageSecurityExit.cs)

The client side security exit directs the client runtime to construct a CFB formatted as either a Standard security CFB or an Enhanced security CFB. The user exit returns a value signifying the desired security mode of either Standard or Enhanced.

- **Standard Security**

Indicates that a client's cooperative flow request CFB "does not" contain the optional security offset section. The CLIENT_USERID and CLIENT_PASSWORD attribute values are included in the CFB header area. Standard security is enabled using the client side security user exit. The CFB header area is not part of the CFB that is eligible to be encrypted.

- **Enhanced Security**

Indicates that a client's cooperative flow request CFB contains the optional security offset section. The security offset section will contain the CA Gen CLIENT_USERID and CLIENT_PASSWORD values as defined by the client application. Additionally, the security offset can contain an optional security token. Enhanced security is enabled using the client side security user exit. The data added to the security offset section of the CFB is eligible to be encrypted by the client runtime by an encryption user exit.

The client side security user exit also indicates from where within the CFB the Comm. Bridge should extract the UserID and Password data. The user exit is passed a pointer to an integer field that can be set to a value of either TRUE or FALSE. The input parameter bClntMgrSecurity points to the integer field. The content of the integer field only has meaning if the CFB is directed to contain Enhanced Security data. Setting the integer field to TRUE causes the client runtime to set a flag byte within the CFB header. A setting of FALSE causes the flag byte in the CFB to remain unset. FALSE is the default value.

If it is desired that the Comm. Bridge use the Enhanced Security data when it processes an inbound cooperative request buffer, the CFB containing the request must have the flag byte set accordingly. This is accomplished by making sure that the client side security exit set the integer field, pointed to by the bClntMgrSecurity parameter, to TRUE.

When the Comm. Bridge processes a CFB that contains Enhanced Security data, the content of the flag byte is used to direct the Comm. Bridge as to which security data it should use when processing the associated cooperative flow request. If the CFB header flag byte is set, the Comm. Bridge will use the Enhanced Security data. If the flag byte is not set, the Comm. Bridge will use the security data that is provided in the CFB header.

Decrypting the CFB

The CFB data transmitted from DPC applications can optionally be encrypted. A flag byte in the CFB header signifies the data has been encrypted. This flag byte is used to notify the receiver of the CFB that it has been encrypted. It is the responsibility of the receiver to decrypt the CFB prior to using it.

With respect to the Comm. Bridge, the data in the CFB only needs to be decrypted if the security data located in the security offset area is to be used when sending the cooperative flow request to a target server. The Comm. Bridge uses the CFB CMUseSecure CFB flag to determine if the data in the security offset should be used. The CFB CMUseSecure CFB flag is set on by the client runtime if its invocation of the client side security exit returns a TRUE for the bCIntMgrSecurity flag.

If the DPC calls for an encrypted CFB, a flag byte signifying the data has been encrypted is placed in the header portion of the CFB. This flag byte is used to notify the receiver of the CFB that encryption has been used. It is then the receiver's responsibility to decrypt the CFB prior to using it.

Comm. Bridge DECRYPT User Exit

Depending on the state of the CFB, the Comm. Bridge may be required to extract data from the security offset. If the CFB being processed has been encrypted, the Comm. Bridge must decrypt the CFB in order to obtain the security data.

The DECRYPT Comm. Bridge user exit is intended to provide a user-written decryption routine that must be able to correctly decrypt the CFB previously encrypted by the client runtime's encryption user exit.

The Comm. Bridge passes the encrypted portion of the CFB to the DECRYPT exit routine. The exit returns a decrypted version of the data it was passed as input.

On return from the DECRYPT user exit, the Comm. Bridge parses out the user ID and password. The Comm. Bridge uses this retrieved security data when servicing the associated cooperative flow request.

The Comm. Bridge decrypts the CFB for its internal use only. When the CFB is forwarded on to the target server environment, it remains in the encrypted state as received from the originating DPC.

The DECRYPT user exit is discussed in more detail in the *User Exit Reference Guide*. Also, see the chapter *Distributed Processing - Overview Guide* for a detailed discussion on encryption and security within a DP application.

Translating UserID and Password

The Comm. Bridge provides an optional Conversation Instance Data user exit to facilitate any required *pre-translation* of the UserID and Password prior to the request being sent to the transport layer. *Pre-translation* of a UserID and/or Password is required in cases where the ASCII characters entered by user do not translate to their corresponding EBCDIC values.

CIDE_INIT()

This entry point is invoked only once during initialization of the Comm. Bridge. Future calls to CIDE_PROC() will be enabled or disabled depending on the return value from this exit.

CIDE_PROC()

This entry point can be used to perform the "pre-translation" of the UserID and Password fields before they are passed to the transport protocol layer.

The Conversation Instance Data user exit is discussed in more detail in the *User Exit Reference Guide*.

Chapter 6: Configuring Multiple Comm. Bridges

Multiple instances of a Comm. Bridge can reside within the same directory on any given communications workstation.

Each instance of a Comm. Bridge is defined by its set of configuration files:

- Initialization file (the default name is iefcbn.ini)
- Server configuration file (the default name is iefcbn.srv)
- Log file (the default name is iefcbn.log)

It is recommended to use a single filename prefix with the .ini, .srv, and .log extensions so that the configuration files associated with different Comm. Bridge instances are easily distinguished from others residing in the same directory. For example, test1.ini, test1.srv, and test1.log represent a different Comm. Bridge configuration than test2.ini, test2.srv, and test2.log.

For more information about current default locations of the log and configuration files, see Default Configuration and Log File Locations in the chapter "[Communications Bridge Overview](#)" (see page 13)".

You can create as many sets of configuration files as you need. You can start a specific instance of a Comm. Bridge from either a system prompt, an icon, or by using the system services console. Specify the name of its corresponding initialization file on the Comm. Bridge startup command. The initialization file name is specified using the /initfile= command argument. For example, if the initialization file has been named test2.ini, the command for starting a Comm. Bridge would be:

```
IEFCBnnN.exe iefcb startup /initfile=test2.ini
```

Note: The nn in the above command designates the CA Gen release number.

By adding new configuration files, you can create Additional Comm. Bridge instances. There is no need to install additional copies of the software. Multiple instances of the software can be run concurrently, each using their own set of configuration files.

Creating Multiple Comm. Bridges Using Icons

Use this procedure to create multiple Comm. Bridge instances in the same directory.

Follow these steps:

1. Open the CA Gen folder on the desktop.
2. Copy the Comm. Bridge program object (icon).
3. Access the property settings for the copied program object (icon) and change the initialization file name option (/initfile =) to reference the .ini file for the desired instance. Also, change the program object title of the newly created Comm. Bridge instance to something appropriate for the new instance.
4. Save the program object (icon).
5. Start the Comm. Bridge using the newly created icon then continue configuring the Comm. Bridge as required.

Note: You must configure the new Comm. Bridge and save the settings in the initialization and server files before use. The settings in these files create the additional Comm. Bridge. You can configure as many Comm. Bridges as you need.

6. Save the current configuration. For more information about saving configurations, see "[Saving Configuration Files](#) (see page 77)."

Creating Multiple Comm. Bridges Without Using Icons

Follow these steps:

1. To start the Comm. Bridge from a command prompt, change to the directory where you installed the Comm. Bridge and enter the command:

```
IEFCBxxN.exe iefcb startup /initfile=[filename]
```

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

IEFCBxxN.exe

Specifies the name of the Comm. Bridge executable

iefcb

Specifies the transaction code associated with the initial procedure step. This is a required parameter.

startup

Specifies the initial command executed by the initial procedure step. This is a required parameter.

/initfile=

Is followed by the name of the initialization file for this Comm. Bridge instance. Use the filename iefcbn.ini for the first execution. In later executions you will use the name you give the initialization file during configuration. This parameter is optional if only one instance of the Comm. Bridge is needed. If not present the default initialization file used is iefcbn.ini.

2. Configure the Comm. Bridge client and server connections as desired.

Note: You must configure the new Comm. Bridge instance and save the settings to the initialization and server files before they can be used on subsequent activation commands. The set of configuration files is what creates the additional Comm. Bridge instance. You can create as many Comm. Bridges as you need.

3. Save the current configuration using the procedures in the chapter "[Saving Configuration Files](#) (see page 77)."

Creating Multiple Comm. Bridge System Services

Multiple Comm. Bridge system services can be concurrently active. Each active Comm. Bridge system service must have its own configuration.

More information:

[Configuring a Comm. Bridge as a Windows System Service](#) (see page 65)

Chapter 7: Configuring a Comm. Bridge as a Windows System Service

A Comm. Bridge can be registered to execute as a Windows system service under control of the Windows Control Panel Services dialog. Registering as a Windows system service requires several configuration parameters that must be contained within the Comm. Bridge initialization file (IEFCBN.INI by default). These parameters can be set using the Comm. Bridge GUI interface or a text editor provided the required format is adhered to.

Registration and removal of a specific Comm. Bridge service instance is accomplished using either the Comm. Bridge GUI interface or command line options. The command line option *instserv* is used to register a service, while the *remserv* option removes a service.

Special User Access Privileges

Installing an application as a Windows system service requires special system access privileges. The currently logged in User ID must have Windows Administrator privilege to manipulate its defined set of Windows system services.

Logging Comm. Bridge Events

From the time a Comm. Bridge is registered as a Windows system service until it is removed, important status change events are logged to the standard Windows system logs. These events can be viewed using the Windows Event Viewer. Most events are logged in the Windows Application Log. However, a few fatal errors may be logged in the Windows System Log as well. Events that cause a log record to be created include Comm. Bridge startup, shutdown, registration, removal, and any errors encountered during these processes.

Interactive and Non-interactive Modes

A Comm. Bridge running as a Windows system service can be run in either interactive or non-interactive mode. By default, the Comm. Bridge runs as a non-interactive service. Non-interactive services can be started without the system having an active desktop. This allows the service to run without requiring a user to log on. This in turn allows the services to restart automatically in an unattended mode should the system experience a reboot. When in non-interactive mode, no Comm. Bridge GUI dialogs or windows are displayed. Therefore, statistic gathering, client/server configuration, and log file viewing are not available using the Comm. Bridge GUI interface.

If a Comm. Bridge is run in interactive mode, the desktop must be active, which means that a user must be logged in before the service can start successfully. If the service is configured to start automatically then the Comm. Bridge GUI interface will pop up immediately when a user logs in. The service will not be usable until the user login activates the desktop.

If the Comm. Bridge GUI interface is exited, the service stops. If the Comm. Bridge GUI interface is not to be displayed, then the Comm. Bridge must be restarted from the Windows local services console interface window in non-interactive mode.

Interactive startup mode can be enabled (or disabled) using a user selectable check box found in the services properties dialog. To enable (or disable) the interactive mode of operation:

Follow these steps:

1. Open the system control panel.
2. Select Administrative Tools and double-click the Services item.
3. Select the appropriate service.
4. Click the right mouse button, and then select Properties.
5. Click the Log On tab on the service properties dialog.
6. Select the Allow Service to Interact with Desktop check box to enable interactive startup. (Uncheck the check box to disable interactive startup).
7. Click OK to dismiss the properties dialog.

The service is now ready to be restarted using the interactive mode selected.

Note: If running the Comm. Bridge as a Windows system service you must be sure the path to locate the CODEPAGE.INI file has been properly established. See the CODEPAGE.INI section below for additional details.

System Service Configuration

The Comm. Bridge application's Service Configuration dialog allows you to configure a Comm. Bridge as a Windows system service. This dialog can be displayed by selecting the File, Setup menu, then clicking Service Config in the displayed dialog. The following figure shows the Service Configuration - Details dialog.

Service Configuration - Details

Display Name:

CB Service #1

Service Name:

CB_SERVICE_ONE

Service Description:

This is a Comm Bridge service. This is #1 of the possible 7 services defined for the host.

User ID:

admin

Password:

••••••

SERVICE START MODE

☐ Automatic

☒ Manual

☐ Disabled

Install Service

Remove Service

Parameter File:

N DATA\CA\GEN 8 5\CFG\CB\IEFCBN.INI

OK

Cancel

Help

The Service Configuration – Details dialog contains the following fields and buttons:

Display Name

Enter the display name to be used for this Comm. Bridge service. The name is alphanumeric and can contain up to 32 characters with embedded spaces.

Service Name

Enter the service name to be used for this Comm. Bridge service. The name is alphanumeric and can contain up to 32 characters with embedded spaces.

Service Description

This text describes the purpose of the service. This text will be displayed within the system services dialog for those systems that support the description field.

User ID

Enter the Windows operating system user name that is to be used when executing this service. This ID must conform to the user name format requirements of the Windows operating system. Enter the ID in the form of domain Name\UserID. If the ID belongs to the built-in domain, you can use the form of \UserID. The User ID/Password is required only if the service is to run under a specific user ID.

Password

Enter the Windows system password associated with the specified User ID that is to execute this service. This password must conform to the password format requirements of the Windows operating system. The User ID/Password is required only if the service is to run under a specific user ID.

Automatic

If selected, the Comm. Bridge service will be configured to start automatically upon system reboot.

Manual

If selected, the Comm. Bridge service must be started manually after each system reboot.

Disabled

If selected, the Comm. Bridge service will be disabled. Before executing service, the service must be enabled using the control panel services dialog.

Install Service

If pressed, the Comm. Bridge is installed as a Windows system service using the configuration defined within the remaining dialog box fields. This establishes the configuration within the Windows Services control Environment. The service is not to be started at this time.

Remove Service

This pushbutton will attempt to remove the Comm. Bridge system service as defined within the service name field.

Parameter File

This field displays the location of the Comm. Bridge initialization file to be used by this instance of the Comm. Bridge system service. This file location cannot be changed from this dialog. It must be changed from the File - Setup dialog. This file name will be used when the Comm. Bridge is started, thus it is imperative that this file be a valid and accessible location at startup.

File Directories

The Comm. Bridge startup software must be able to locate the startup files in order to initialize the options for each Comm. Bridge. You must properly designate the location of the following files:

- CODEPAGE.INI
A CA Gen support file.
- IEFCBN.INI
The Comm. Bridge initialization file.
- IEFCBN.SRV and IEFCBN.LOG
The Comm. Bridge configuration file and log file.

CODEPAGE.INI

When a Comm. Bridge is run as a Windows system service, you must either copy the CODEPAGE.INI file into the system's system32 directory or specify the fully qualified path to the Comm. Bridge install directory in the System Path variable. In order to activate the latter choice, the system must be restarted.

IEFCBN.INI

When a Comm. Bridge is run as a Windows system service, the Comm. Bridge initialization file (IEFCBN.INI by default) is found by looking in the fully qualified path that was used when the Comm. Bridge was configured as a Windows system service.

IEFCBN.SRV and IEFCBN.LOG

The location of the other Comm. Bridge files (.SRV and .LOG) is specified in the Comm. Bridge initialization file (.INI). A relative or fully qualified path (a path that contains .\ , ./ or :) may be used.

The current path to which a relative path is applied is one of the following:

- The Comm. Bridge install directory, if started using the Windows program group.
- The directory in the Start in property, if started using a shortcut.
- The current directory, when started using a DOS prompt.
- The system32 directory, if the Comm. Bridge is defined to run as a Windows system service.

If the relative or full path is not specified, it is assumed the files are in the directory where the initialization file was found.

Note: If the Comm. Bridge is running in non-interactive mode and the Comm. Bridge configuration file does not exist or cannot be located, the Comm. Bridge will not start up.

Registering Using the Comm. Bridge GUI Interface

Following are the steps to do service registration using the Comm. Bridge GUI interface:

Follow these steps:

1. Start the Comm. Bridge using one of the following methods:
2. From an Icon, open the CA Gen folder and double-click the Comm. Bridge icon.

3. From a command prompt, change to the directory where you installed the Comm. Bridge and enter the following command:

IEFCBxxN.exe iefcb startup /initfile=filename

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

IEFCBxxN.exe

Specifies the name of the Comm. Bridge executable

iefcb

Specifies the transaction code associated with the initial procedure step. This is a required parameter

startup

Specifies the initial command executed by the initial procedure step. This is a required parameter

/initfile=

Is followed by the name of the initialization file for this Comm. Bridge instance. Use the filename iefcbn.ini for the first execution. In later executions, you use the name you gave the initialization file during configuration. This parameter is optional if only one instance of the Comm. Bridge is needed. If not present the default initialization file used is iefcbn.ini.

4. Select File, Setup menu items
5. On the File – Setup dialog click Service Config. This brings up the Service Configuration Details dialog.
6. Fill in the required Display and Service Name fields, the optional User ID and Password fields if the service is to run under a non-administrator ID, and the desired start mode. The Service Description field can contain any descriptive text desired. Standard CA Gen application on line help has been provided for each of these items.
7. Ensure the displayed parameter file location is correct and accessible to the specified User ID at the time the service is to be started. The parameter file location can be changed on the File – Setup dialog.
8. The service information is stored into the Windows Operating system services repository when the Install Service is pressed.
9. A status dialog will report the success or failure of the service installation operation.

Note: The service will not be started at this time.

If successfully installed the service can now be started and controlled using the Windows Control Panel Services utility.

Registering Using the Comm. Bridge Command Line

Following are the steps to do service registration using the Comm. Bridge command line:

1. Prior to registering the service using the instserv command line option the targeted Comm. Bridge initialization file (IEFCBN.INI) must be created and properly formatted. This can be accomplished by using the Comm. Bridge in GUI mode to define the service configuration and create the file. This properly formatted file can be used as a template for future versions of Comm. Bridge configuration files.

The following is an example of a properly formatted .ini file. The items used for system service definition are those items between the start/end of service configuration items. The SERVICE_PASSWORD value is encrypted prior to writing the value to the log file. If a password is required then the method detailed in the section Registering Using the GUI Interface must be used to enter the password.

```
# Communications Bridge Parameters...
DESCRIPTION = sample system service
LOGFILE     = IEFCBN.LOG
LOGLEVEL    = 3
DUMPIOBUFFERS = 1
REFRESH     = 0
REFRESH_INTERVAL = 1
IDLE_CLIENT_TIMEOUT = 0
ENABLE_CONNECTION_MONITOR = 0
# start of service configuration items
SERVICE_STARTMODE = AUTOMATIC
SERVICE_NAME = TestService
DISPLAY_NAME = Test the Service
SERVICE_USERID =
SERVICE_PASSWORD =
# end of service configuration items
SRVRS       = IEFCBN.SRV
# Client Transport parameters...
COMMDLL     = INTCP80N
TRANSPORT   = 2
IP_SRCEPORT = 5002
# End of Communications Bridge Parameters
```

2. Register the service by executing the following command:

```
IEFCBxxN.exe instserv /initfile=[filename]
```

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

Where:

- IEFCBxxN.exe
Specifies the name of the Comm. Bridge executable

- **instserv**

Specifies the transaction code associated with the initial procedure step. This is a required parameter when installing a Comm. Bridge as a system service

- **/initfile=**

Is followed by the name of the initialization file to be used for this Comm. Bridge instance. If not present the default initialization file used is iefcbn.ini

3. A status dialog will report the success or failure of the service installation operation.

Note: The service will not be started at this time.

Removing Using the GUI Interface

Following are the steps to remove service registration using the Comm. Bridge GUI interface:

Follow these steps:

1. Start the Comm. Bridge using one of the following methods:

From an Icon, open the CA Gen folder and double-click the Comm. Bridge icon.

From a command prompt, change to the directory where you installed the Comm. Bridge and type the following command:

```
IEFCBxxN.exe iefcb startup /initfile=[filename]
```

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

IEFCBxxN.exe

Specifies the name of the Comm. Bridge executable

iefcb

Specifies the transaction code associated with the initial procedure step. This is a required parameter.

startup

Specifies the initial command executed by the initial procedure step. This is a required parameter.

/initfile=

Is followed by the name of the initialization file to be used for this Comm. Bridge instance. If possible, use the filename used when the service was previously registered. If that file is not available, the name of the service to be deleted can be entered into the file manually.

Prior to starting the Comm. Bridge, ensure the target service has been stopped using the Windows Control panel Services utility.

2. From the Comm. Bridge main window Select File, Setup menu items
3. On the File – Setup dialog click Service Config. . This brings up the Service Configuration Details dialog.
4. Within this dialog ensure the required Service Name field is correct if populated, or else enter the Service Name to be removed. The remaining items are not used for the remove service operation.
5. Click Remove Service to remove the service information from the Windows Operating system services repository.
6. A status dialog will report the success or failures of the service remove operation.

Removing Using the Comm. Bridge Command Line

Following are the steps to remove service registration using the Comm. Bridge command line:

Follow these steps:

1. Prior to removing the service registration using the remserv command line option the targeted Comm. Bridge initialization file (IEFCBN.INI) must be created and be properly formatted. See the section Registering Using the Command Line above for an example of a properly formatted .ini file. This can be accomplished by using the Comm. Bridge in GUI mode to define the service configuration and to create the file. This properly formatted file can be used as a template for future versions of Comm. Bridge configuration files.
2. Remove the service registration by executing the following command line options:

```
IEFCBxxN.exe remserv /initfile=[filename]
```

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

- IEFCBxxN.exe

Specifies the name of the Comm. Bridge executable

- remserv

Specifies the transaction code associated with the initial procedure step. This is a required parameter when removing a Comm. Bridge previously installed as a system service

- /initfile=

Is followed by the name of the initialization file to be used for this Comm. Bridge instance. If possible, use the filename used when the service was previously registered. If that file is not available, the name of the service to be deleted can be entered into the file manually.

A status dialog reports the success or failures of the service remove operation.

Comm Bridge Service Targeting RSC/MP

A Comm Bridge running as a system service configured to target a NonStop server environment has special requirements not seen with other target server environments.

The Comm Bridge uses a third party product, HP RSC/MP, as the transport mechanism when communicating with NonStop target servers. When the Comm Bridge is configured to run as a system service the RSC/MP product must also be configured to run as a system service.

At the time of this writing the RSC/MP product contains two separate but dependent processes, RSC and Piccolo. For proper operation of the Comm Bridge as a system service these products must also be installed as system services. This can be accomplished through the RSC/MP products rscserv.exe command. (rscserv.exe –install).

For proper operation of the Comm Bridge to NonStop communications both the RSC and Piccolo system services must be running prior to a transaction being attempted. For more information, see the RSC/MP product documentation.

During normal operation transaction mapping data is down loaded from the NonStop host and stored in a local file within the Comm Bridge's host machines file system. If the Comm Bridge is operating as a system service then this mapping file will be located in:

`%APPDATA%\CA\Gen xx\cfg\cb\RSCMP\<Path-Mon>\tirtmt.tbl`

Where <Path-Mon> is the RSC/MP configured Pathway Monitor configuration setting.

The value of %APPDATA% is dependent upon the Windows User Id under which the Comm Bridge is executing. If the Comm Bridge service was installed without specifying a User Id, the default User Id is the Local System account. In this case %APPDATA% would equate to LocalService\AppData which would translate the above to a path similar to:

`C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\CA\Gen xx\cfg\cb\RSCMP\<Path-Mon>\tirtmt.tbl`

Note: xx refers to the current release of CA Gen. For the current release number, see the *Release Notes*.

Chapter 8: Saving Configuration Files

The configuration parameters for a Comm. Bridge are saved in two distinct files, an initialization file, and a target server file. You can save these configuration files using either the default file names or choose your own file names.

The default name of the initialization file is IEFCBN.INI. This file contains:

- The log file name
- Current log level
- Name of server configuration (.srv) file
- Statistic refresh state and interval
- Message Language dll name used to determine which of the supported languages is used when writing status information to the log file.
- Idle client timeout state and interval
- Windows Service configuration data, if registered as a Windows Service
- Configuration parameters required for the client connections

The default name of the target server configuration file is IEFCBN.SRV. This file contains all the configuration parameters required for server connections

Retaining the .INI and .SRV extensions aids in keeping track of the descriptive nature of the files' functions. It is also recommended that you use the same file name prefix for a given instance as it makes it easier to distinguish the set of files related to a particular Comm. Bridge.

Saving Configuration File Names

Use the following steps to save configuration files:

1. From the Comm. Bridge main window, Select File, Save
2. At the File – Save Configuration dialog verify that check boxes for the Initialization file and Server Configuration file are marked
3. Click OK to return to the main dialog

4. The Comm. Bridge must be stopped and restarted before the saved configuration files will be used. The Comm. Bridge reads the configuration files once during its startup processing.

Note: If you change file names from within the File - Save Configuration dialog, the current configuration is saved under these names. This is similar to the Save as option in other programs.

Changing Configuration File Names

To change the names of the Comm. Bridge configuration files:

Follow these steps:

1. Select File, Setup from the Comm. Bridge main window.
2. At the File – Setup dialog select the file name you want to change.
3. Click Rename.
This displays the Setup - Change Filename dialog.
4. Change the name of the file as desired and click OK.
5. Click OK at the Change Filename Confirmation dialog
This propagates the changed names to the File > Setup dialog.
6. Repeat steps 2 through 7 for the remaining file names to be changed.
7. Click OK on the File – Setup dialog.
8. On the main window, select File, Save and verify that both check boxes on the File – Save Configuration dialog are checked.
9. Click OK to return to the main dialog.

Note: Changing the name of a Comm. Bridge configuration file, effectively creates a new instance of the file. The saved initialization files can be used to start a new Comm. Bridge instance.

Chapter 9: Testing Comm. Bridge Connections

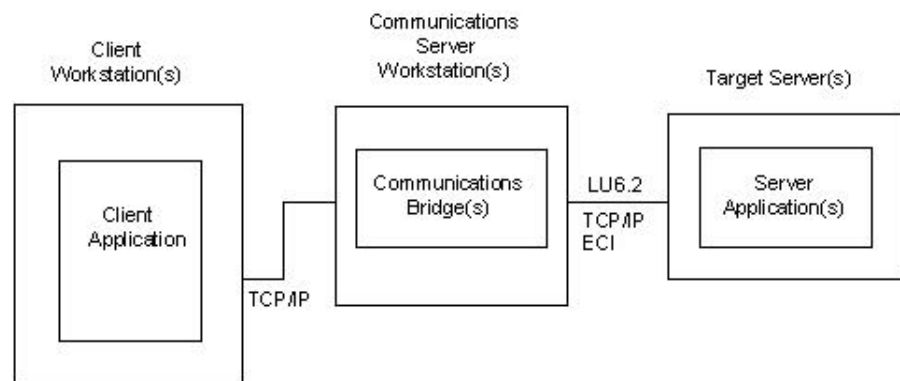
The Comm. Bridge creates a connection to its target server, if a connection does not exist, whenever it processes a cooperative flow request. Therefore, any DPC configured to communicate to its target server using a Comm. Bridge can be used to test the connectivity to the configured server execution environment.

Alternately, the CA Gen Client Manager can be used to drive the testing of a Comm. Bridge by making use of its Send Test Tran capability.

Testing Connectivity Using a DPC Application

The testing of a particular Comm. Bridge can be accomplished using a DPC application that targets a DPS that is deployed to the specific target server execution environment being served by that Comm. Bridge. The DPC would need to be generated for, or be configured to use, TCP/IP (Sockets) as its transport mechanism. The DPC initiates a test request to a particular Comm. Bridge by designating that Comm. Bridge's Host and Port information as defined by the target Comm. Bridge client side configuration.

The following diagram illustrates using a DPC to drive the testing of a Comm. Bridge:



The following steps can be used to drive the testing of a particular Comm. Bridge.

1. The target DPS must be installed to the specific server the Comm. Bridge being tested is configured to serve. See the Comm. Bridge Server Side configuration information to determine the specific target server execution environment the Comm. Bridge is configured to serve.

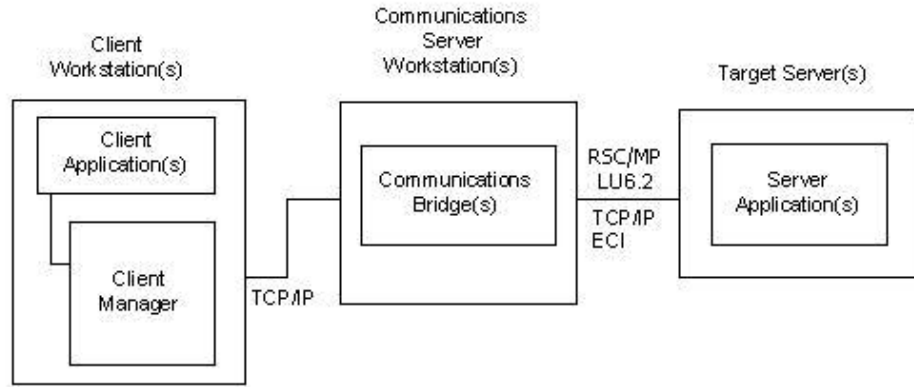
2. Configure the Comm. Bridge client side configuration. See the chapter "Comm. Bridge General Configuration" in this guide. Note the Comm. Bridge client configuration is a well-known Port number.
3. On the DP client workstation, configure the DPC application to access the Comm. Bridge by configuring the DPC to communicate to the Host name and well-known port of the target Comm. Bridge. The socket information associated with the Comm. Bridge being tested can be configured into a generated DPC or be specified using one of the CA Gen support commcfg files. See the *Distributed Processing – Overview Guide* for details of how to override a generated client configuration at runtime.
4. On the client workstation, activate the client application and exercise the application as required to initiate a cooperative flow that will target the Comm. Bridge being tested.
5. The cooperative flow request should result in a connection request to the Comm. Bridge. The Comm. Bridge will process the inbound cooperative flow request. If the Comm. Bridge already has a connection to its associated target server execution environment the connection is used, otherwise a new connection will be created from the Comm. Bridge to the target server environment.
6. Both the client and server connection status can be monitored from the Comm. Bridge main window. The Status field reflects the status of the server connection. The Client Name contains a list of those clients currently connected to the Comm. Bridge.
7. Verify the test cooperative flow worked as expected.

If errors are encountered, use the information found in the Comm. Bridge log file to determine the cause of the failure. Access the log file by selecting File,Browse,Log File from within the Comm. Bridge application. Log files created by the client and/or the target server may also be needed to fully determine the cause of a communications failure.

Testing Connectivity Using the Client Manager

Similar to the previous discussion, the Client Manager can make use of a generated GUI application to drive the testing of a particular Comm. Bridge.

The following diagram illustrates a generated GUI client application being used to drive the testing of a Comm. Bridge. The Client Manager provides the network connectivity from the DPC to the Comm. Bridge.



Alternately, the Client Manager is able to simulate a cooperative flow request without the use of a DPC application. The Client Manager Send Test Tran operation builds a simulated CFB using the Test Tran name that is associated with a selected target server.

To test a specific Comm. Bridge the user of a Client Manager selects its associated server entry from its list of configured servers. After selecting, the user can request the Client Manager to Send a Test Tran. This mechanism allows the Client Manager to simulate a cooperative flow request that results in the Client Manager sending a Test transaction to the selected target server for processing.

CA Gen provides various implementations of an ECHO server application that can be used as a designated test transaction. The various implementations of ECHO exist for the various server execution environments a Client Manager is capable of servicing such as CICS, IMS, Transaction Enable and Tuxedo.

See the *Distributed Processing - Client Manager User Guide* for steps used to execute a Test Transaction.

A successful execution of the Test Transaction from the Client Manager through the Comm. Bridge to the target server environment confirms the communication infrastructure is working and is able to support subsequent cooperative flow requests.

Chapter 10: Comm. Bridge Transaction Statistics

The Comm. Bridge has a statistic gathering capability that tracks the number of cooperative flows as well as the total number of bytes sent and received. A dialog displaying a summary of the statistics can be displayed.

The statistics can be updated on demand or periodically refreshed. The periodic refresh can be enabled or disabled. If it is enabled, the time between refreshes is set by the time parameter that is configured as part of the Comm. Bridge Setup dialog.

The Comm. Bridge can optionally be directed to write its statistical data to the Comm. Bridge log when it processes its periodic refresh operation.

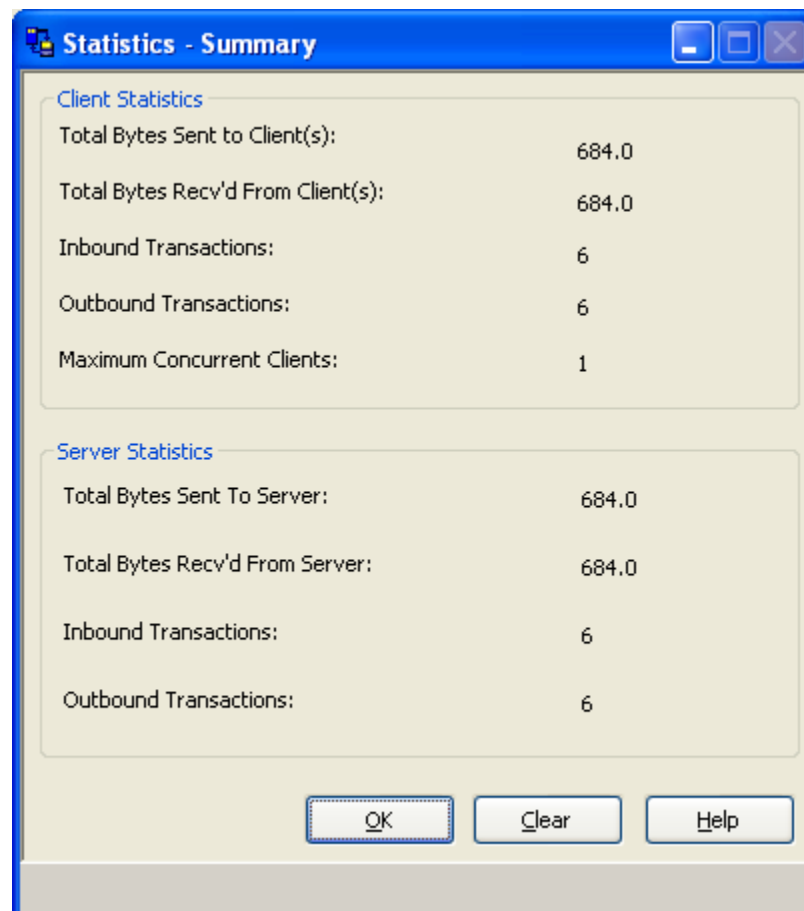
Note: The user interface for establishing the statistics refresh values has been moved to the main Comm. Bridge setup dialog.

More Information:

[Comm. Bridge General Configuration](#) (see page 23)

Statistics - Summary Dialog

The Comm. Bridge Statistics-Summary dialog displays real-time byte transfer statistical data for both the client and target server transports. To display the statistics dialog from the Comm. Bridge main window select the Statistics, Summary menu items. The following figure shows the Statistics Summary dialog.



Note: The fields that provide total byte counts of data routed through the Comm. Bridge have a special multiplier field that follows the data field. The first letter of each of these abbreviations denotes the multipliers: Kilo, Mega, Giga, and Tera.

The statistical counts are restarted at zero when:

- The Comm. Bridge is first started
- The clear pushbutton on the Statistics – Summary dialog is pressed.
- The Statistics, Clear All main window menu item is selected

The display-only fields on this dialog include:

Total Bytes Sent to Client

The total number of bytes sent to all Comm. Bridge clients.

Total Bytes Received from Client

The total number of bytes received from all Comm. Bridge clients.

Inbound Transactions – Client

The total number of cooperative flow requests received from all Comm. Bridge clients.

Outbound Transactions – Client

The total number of cooperative flow responses returned to all Comm. Bridge clients.

Maximum Concurrent Clients

Maximum number of concurrently connected clients.

Total Bytes Sent to Server

The total number of bytes sent to the target server by the Comm. Bridge.

Total Bytes Received from Server

The total number of bytes received by the Comm. Bridge from the target server.

Outbound Transactions – Server

The total number of cooperative flow requests sent to the target server by the Comm. Bridge.

Inbound Transactions – Server

The total number of cooperative flow responses received by the Comm. Bridge from the target server.

Appendix A: Error Messages

For messages representing communication errors that occur most often with CA Gen Distributed Processing applications, see the *Host Encyclopedia Construction Guide* and *z/OS Implementation Toolset User Guide*.

Additional messages providing details of communications failure can be found in the Comm. Bridge log file. You can view the log files with an ASCII text editor or from within the Comm. Bridge main window using the File,Browse, Log File menu items.

Setting the Logging Level to Tracing

The Comm. Bridge logging level can be changed to alter the amount of data dumped to the log file. To change the log level:

Follow these steps:

1. Select File, Setup from the main Comm. Bridge window.
2. Set the desired tracing level. The Trace Logging group box contains the trace level controls. Tracing is the most verbose level while Errors is the least verbose level.
3. Depending upon the amount of activity, enabling tracing may result in rather large log files. If the size of the log file is an issue, its size can be limited by using the Max Log Size Comm. Bridge setup option. See the Comm. Bridge General Configuration chapter within this document for details.
4. The Comm. Bridge log level change takes affect immediately. It is not necessary to restart the Comm. Bridge.
5. Re-execute the failing cooperative flow.

Note: Using the Trace logging level may produce large log files.

More Information:

[Comm. Bridge General Configuration](#) (see page 23)

[Default Configuration and Log file Locations](#) (see page 21)

Appendix B: Comm. Bridge User Exits

This chapter lists the Comm. Bridge user exits. The collection of Comm. Bridge user exits offer users of the Comm. Bridge a mechanism for customizing certain default behaviors. The Comm. Bridge currently supports the following exits:

User Exit Name	Source Code	Description
GetTCPHostName	inetipux.c	Disables host name lookup that a Comm. Bridge uses to obtain the host name of a connected client by using its IP address.
DECRYPT	decrexit.c	Decrypts the CFB from a client if the data in the Enhanced Security offset area is to be used and the CFB data is encrypted.
CIDE_INIT	cidexit.c	Conversation Instance Data – Initialize. Used to disable or enable subsequent CIDE_PROC calls.
CIDE_PROC	cidexit.c	Conversation Instance Data – Process. Used to modify certain fields of the Conversation Instance data (UserId and Password), prior to the conversation supporting a cooperative flow being created.
eci_client_exit	ioeciclx.c	Allows ECI arguments to be overridden at runtime. These arguments include the name of the target CICS System (as it is known to the CICS Universal Client), the specified ECI timeout value, and the Transaction Name associated with the CICS Mirror application DFHMIRS.
RSCUserEntry()	iorscclx.cxx	An optional user exit which will provide access to and modification of CA Gen user and application data. Note: While routines supported by this exit do allow data to be modified, the total length of the data buffer cannot be changed. It is up to the user to maintain data integrity.

Notes:

- If the Comm. Bridge is installed into a system protected file system area and you need to rebuild user exits, you may need to elevate your system access privilege to system admin level. If the user id does not have the proper authority the user will be notified via an authorization denied error message displayed at the command window.
- For a detailed description of each of the above user exits, see the *User Exit Reference Guide*.

Index

C

- CA Gen • 13, 14, 15, 80
 - Common Format Buffer • 14
 - Distributed Processing (DP) client/server application • 13
 - DP application network environment • 15
 - ECHO server application • 80
- client and server connections • 18
- Client Manager, network connectivity • 80
- CODEPAGE.INI file • 70
- Comm. Bridge • 13, 15, 16, 18, 19, 20, 23, 24, 26, 28, 32, 57, 59, 61, 62, 65, 70, 79, 83, 89
 - CFB, parsing • 57
 - CFB, Security • 57
 - client connection, configuring • 18
 - client connection, configuring • 32
 - client environments • 15
 - configuration file • 23
 - configuring as a Windows system service • 65
 - configuring multiple bridges • 61
 - executing • 24
 - Installing • 23
 - Installing multiple bridges • 62
 - log file • 28
 - logging events • 65
 - operational features • 16
 - potential clients • 15
 - Resides on • 13
 - security • 19
 - server connection, configuring • 18
 - server environments • 16
 - service registration • 70
 - Setup dialog • 26
 - statistics • 20
 - TCP/IP host name • 16
 - testing logical network connections • 79
 - transaction statistics • 83
 - user exit • 59
 - user exits • 89
 - Windows system service, run as • 19
- Common Format Buffer (CFB) • 14, 19, 57, 59
 - Comm. Bridge, parsing • 57
 - decrypting • 59
 - enhanced CFB • 19

- enhanced security • 57
 - standard CFB • 19
 - standard security • 57
- communication • 13, 79, 87
 - byte streams • 13
 - error messages • 87
 - failure, log file • 79
- configuration file names • 77, 78
 - changing • 78
 - saving • 77
- configuring • 61, 65
 - as a Windows system service • 65
 - multiple Comm. Bridges • 61
- cooperative flow • 13, 14, 79
 - connection request • 79

D

- Distributed Processing application • 13, 14
 - client workstation • 13
 - server workstation • 14
- Distributed Processing Client (DPC) application • 14
- Distributed Processing Server (DPS) application • 14

E

- ECHO server application, implementation • 80
- error communication • 87

F

- file name, saving configuration file • 77

H

- host-specific communications hardware, configuring • 16

I

- IEFCBN.INI, initialization file • 70
- IEFCBN.LOG, log file • 70
- IEFCBN.SRV, configuration file • 70
- interactive Mode • 66

L

- log files, tracing • 29
- logging Comm. Bridge events • 65

- logging level • 87
 - error messages • 87
 - setting to tracing • 87

M

- messages, communication error • 87
- modes of operating • 66
- multiple Comm. Bridges • 61, 62
 - configuring • 61
 - installing • 62

N

- network environment, Comm Bridge • 16
- non-interactive Mode • 66

O

- operating modes • 66

S

- saving configuration file • 77
- security user exits, client side • 57
- service registration, Comm. Bridge • 70, 72, 73, 74
 - registering using command line • 72
 - registering using GUI interface • 70
 - removing using command line • 74
 - removing using GUI interface • 73
- sockets, configuring client connection • 33
- Statistics-Summary dialog, Comm. Bridge • 84

T

- TCP/IP socket, configuring client connection • 33
- testing, Comm. Bridge connections • 79
- tracing • 20, 29, 87
 - external enabling • 20
 - log files • 29
 - setting the logging level • 87
- transaction statistics • 20
- transaction statistics, Comm. Bridge • 83

U

- user access privileges • 65