



RPC BROKER SYSTEMS MANUAL

Version 1.1

Revised May 2002

Department of Veterans Affairs
VISTA System Design & Development (SD&D)
Information Infrastructure Service (IIS)

Document Revision History

The following table displays the revision history for this document. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author
05/08/02	3.0	Revised Version for Patch 26.	Thom Blom, Oakland OIFO
04/08/02	2.0	Revised Version for Patch 13.	Thom Blom, Oakland OIFO
09/97	1.0	Initial RPC Broker Version 1.1 software release.	Thom Blom, San Francisco OIFO



For a complete list of patches released with the RPC Broker V. 1.1 software, please refer to "Appendix A—Patch Revision History."

Document Revision History

Contents

Orientation	ix
How to Use this Manual	ix
Commonly Used Terms.....	x
How to Obtain Technical Information Online.....	x
Assumptions About the Reader	xi
Reference Materials.....	xii
1. Introduction	1-1
Overview	1-1
How Does It All Work?.....	1-2
System Overview	1-4
2. System Features	2-1
Client Features.....	2-1
RPC Broker Client Agent	2-1
"Connect To" Dialog.....	2-2
Edit Broker Servers Program	2-3
Standalone Programs and their Associated Help Files	2-5
HOSTS File.....	2-6
What Happened to the Client Manager?	2-8
What Happened to the VISTA.INI File?	2-9
Server Features	2-11
Menu for System Managers	2-11
Broker Listeners and Ports.....	2-11
Starting And Stopping Listeners	2-12
RPC BROKER SITE PARAMETERS File.....	2-14
Integrated Auto Signon for Multiple User Sessions	2-15
RPC Broker Message Structure	2-17
Client/Server Timeouts	2-18
Load Balancing on Alpha Systems	2-19
MSM for NT 4.3.0 MSERVER Replaces RPC Broker Listener Process	2-20

3. Security	3-1
Security Features	3-1
Validation of Connection Request.....	3-1
Validation of Users.....	3-1
<i>VISTA</i> Sign-on Dialog.....	3-1
<i>VISTA</i> Division Selection Dialog.....	3-3
Users Can Customize <i>VISTA</i> Sign-on Dialog	3-4
Change <i>VISTA</i> Verify Code Component.....	3-7
Validation of RPCs.....	3-8
Sample Security Procedures	3-9
Security Features Tasks Summary	3-9
4. Troubleshooting	4-1
Test the Broker Using the RPC Broker Diagnostic Program	4-1
Verify and Test the Network Connection.....	4-3
Signon Delays.....	4-4
Glossary	Glossary-1
Appendix A—Patch Revision History	Appendix A-1
Index	Index-1

Figures

Table 1: Documentation symbol descriptions.....	ix
Table 2: Commonly used RPC Broker terms.....	x
Figure 1: <i>VISTA</i> RPC Broker system overview diagram.....	1-4
Figure 2: RPC Broker Client Agent dialog.....	2-1
Figure 3: Server and port configuration selection dialog.....	2-2
Figure 4: Add Server dialog.....	2-2
Figure 5: Edit Broker Servers dialog.....	2-3
Figure 6: Sample error message when adding a new server entry.....	2-4
Table 3: Standalone RPC Broker programs and their associated help files.....	2-5
Table 4: HOSTS file location.....	2-6
Figure 7: Sample HOSTS file.....	2-7
Table 5: VISTA.INI entries and Microsoft Windows Registry disposition table.....	2-10
Figure 8: Automatically starting the Listener(s) when TaskMan is restarted.....	2-13
Table 6: Listener site parameter entries description table.....	2-14
Table 7: Multiple and Auto Signon Settings table.....	2-16
Figure 9: Sample <i>VISTA</i> Sign-on security dialog.....	3-2
Figure 10: Sample Select Division dialog.....	3-3
Figure 11: Sign-on Properties on the System Menu.....	3-4
Figure 12: Sign-on Properties dialog.....	3-5
Figure 13: Sample Font dialog.....	3-7
Figure 14: Change <i>VISTA</i> Verify Code dialog.....	3-7
Table 8: Security Tasks.....	3-9
Figure 15: RPC Broker connection diagnostic program.....	4-2
Table 9: RPC Broker V. 1.1 patch revision history (in reverse sequence order).....	Appendix-7

Figures

Orientation

How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of the RPC Broker V. 1.1 and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VISTA) and commercial off-the-shelf (COTS) software products.

There are no special legal requirements involved in the use of the RPC Broker's Interface.

This manual uses several methods to highlight different aspects of the material:

- Various symbols are used throughout the documentation to alert the reader to special information. The following table gives a description of each of these symbols:

Symbol	Description
	Used to inform the reader of general information including references to additional reading material
	Used to caution the reader to take special notice of critical information

Table 1: Documentation symbol descriptions

- Descriptive text is presented in a proportional font (as represented by this font).
- "Snapshots" of computer online displays (i.e., roll-and-scroll screen captures/dialogs) and computer source code are shown in a *non*-proportional font and enclosed within a box. Also included are Graphical User Interface (GUI) Microsoft Windows images (i.e., dialogs or forms).
 - User's responses to online prompts will be boldface type.
 - The "<Enter>" found within these snapshots indicate that the user should press the Enter or Return key on their keyboard.
 - Author's comments are displayed in italics or as "callout" boxes.



Callout boxes refer to labels or descriptions usually enclosed within a box, which point to specific areas of a displayed image.

- Object Pascal code uses a combination of upper- and lowercase characters. All Object Pascal reserved words are in boldface type.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).

Commonly Used Terms

The following is a list of terms and their descriptions that you may find helpful while reading the RPC Broker documentation:

Term	Description
Client	A single term used interchangeably to refer to a user, the workstation (i.e., PC), and the portion of the program that runs on the workstation.
Component	A software object that contains data and code. A component may or may not be visible.  For a more detailed description, see the "Borland Delphi for Windows User Guide."
GUI	The Graphical User Interface application that is developed for the client workstation.
Host	The term Host is used interchangeably with the term Server.
Server	The computer where the data and the RPC Broker remote procedure calls (RPCs) reside.

Table 2: Commonly used RPC Broker terms



Please refer to the "Glossary" for additional terms and definitions.

How to Obtain Technical Information Online

Exported file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.



Methods of obtaining specific technical information online will be indicated where applicable under the appropriate topic.

Help at Prompts

VISTA software has online help and commonly used system default prompts. In roll-and-scroll mode users are strongly encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of **VISTA** software.

To retrieve online documentation in the form of Help in *VISTA* roll-and-scroll software:

- Enter a single question mark ("?") at a field/prompt to obtain a brief description. If a field is a pointer, entering one question mark ("?") displays the HELP PROMPT field contents and a list of choices, if the list is short. If the list is long, the user will be asked if the entire list should be displayed. A YES response will invoke the display. The display can be given a starting point by prefacing the starting point with an up-arrow ("^") as a response. For example, **^M** would start an alphabetic listing at the letter M instead of the letter A while **^127** would start any listing at the 127th entry.
- Enter two question marks ("??") at a field/prompt for a more detailed description. Also, if a field is a pointer, entering two question marks displays the HELP PROMPT field contents and the list of choices.
- Enter three question marks ("???") at a field/prompt to invoke any additional Help text that may be stored in Help Frames.

Obtaining Data Dictionary Listings

Technical information about files and the fields in files is stored in data dictionaries. You can use the List File Attributes option on the Data Dictionary Utilities submenu in VA FileMan to print formatted data dictionaries.



For details about obtaining data dictionaries and about the formats available, please refer to the "List File Attributes" chapter in the "File Management" section of the "VA FileMan Advanced User Manual."

Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the following:

- *VISTA* computing environment (e.g., Kernel Installation and Distribution System [KIDS])
- VA FileMan data structures and terminology
- Microsoft Windows
- M programming language

No attempt is made to explain how the overall *VISTA* programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web for a general orientation to *VISTA*. For example, go to the System Design & Development (SD&D) Home Page at the following web address:

<http://vista.med.va.gov/>

This manual does provide, however, an explanation of the RPC Broker, describing how it can be used in a client/server environment.

Reference Materials

Readers who wish to learn more about the RPC Broker should consult the following:

- "RPC Broker Getting Started with the Broker Development Kit (BDK)" (written for programmers)
- "RPC Broker Developer's Guide" (i.e., BROKER.HLP, online help designed for programmers, distributed in the BDK)
- "RPC Broker Technical Manual"
- "RPC Broker Installation Guide"
- "RPC Broker Release Notes"
- RPC Broker Home Page at the following web address:

<http://vista.med.va.gov/broker/>

This site provides announcements, additional information (e.g., Frequently Asked Questions [FAQs], advisories), documentation links, archives of older documentation and software downloads.

Broker documentation is made available online, on paper, and in Adobe Acrobat Portable Document Format (.PDF). The .PDF documents must be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following web address:

<http://www.adobe.com/>



For more information on the use of the Adobe Acrobat Reader, please refer to the "Adobe Acrobat Quick Guide" at the following web address:

<http://vista.med.va.gov/iis/acrobat/index.html>



DISCLAIMER: The appearance of external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Health Administration (VHA) of this Web site or the information, products, or services contained therein. The VHA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of this VHA Intranet Service.

1. Introduction

Overview

The Remote Procedure Call (RPC) Broker (also referred to as "Broker") is a client/server system within VA's Veterans Health Information Systems and Technology Architecture (VISTA) environment. It establishes a common and consistent foundation for client/server applications being written as part of VISTA. It enables client applications to communicate and exchange data with M Servers.

The RPC Broker is a bridge connecting the client application front-end on the workstation (e.g., Delphi GUI applications) to the VISTA M-based data and business rules on the server. It links one part of a program running on a workstation to its counterpart on the server. Therefore, the RPC Broker assists in opening the traditionally proprietary VISTA software to Commercial Off-the-Shelf (COTS) and Hybrid Open Systems Technology (HOST) products.

This manual provides descriptive information and instructions on the use of the RPC Broker client/server software. The emphasis is on the use of Borland's Delphi software. However, the RPC Broker does support other client environments.

This document is intended for the VISTA development community, the Information Resource Management (IRM) staff, and clinicians using Broker-based client/server applications. A wider audience of technical personnel engaged in operating and maintaining the Department of Veterans Affairs (VA) software may also find it useful as a reference.

The RPC Broker includes the following:

- A common communications driver interface that handles the device-specific characteristics of the supported communications protocol.
- An interface component separate from the communications driver that interprets the message, executes the required code, and eventually returns data to the communications driver.
- A common file that all applications use to store the information on the queries to which they respond (i.e., REMOTE PROCEDURE file [#8994]).
- Architecture that supports multiple GUI and client front-ends.

This version of the Broker also includes the Broker Development Kit (BDK). The BDK provides VISTA application programmers with the following features:

- The capability to create GUI client/server VISTA applications using Borland's Delphi software. The BDK provides the TRPCBroker, TSharedRPCBroker, and TXWBRichEdit components, which developers use in Delphi applications to execute remote procedure calls (RPCs) on VISTA M servers.
- Support for COTS/HOST client/server software using the Broker Dynamic Link Library (DLL).

The RPC Broker:

- Operates in a 32-bit environment while supporting *VISTA* applications previously developed in the 16-bit environment (e.g., PCMM). The client workstation can be running any of the following Microsoft operating systems:
 - Windows 95
 - Windows 98
 - Windows NT V. 3.51 or greater
 - Windows 2000
- Provides support for Auto Signon. Users need only sign on once when accessing both a *VISTA* roll-and-scroll (e.g., Laboratory, Pharmacy) and a *VISTA* client/server GUI-based application (e.g., CPRS, NOIS, PCMM) on the same workstation, regardless of which application is started first.



For more information on Auto Signon, please refer to the "Integrated Auto Signon for Multiple User Sessions" topic in Chapter 2, "System Features" in this manual.

•

- Provides new and enhanced Broker management and configuration tools (e.g., new debugging tools, new RPC BROKER SITE PARAMETERS file (#8994.1), enhanced Broker Listener).



For more information on troubleshooting the Broker, please refer to Chapter 4, "Troubleshooting" in this manual.

How Does It All Work?

The process begins on a user's workstation (i.e., PC), running Microsoft Windows, which is either connected directly or remotely via a modem to a site's local area network (LAN). The workstation must be able to run some version of Transmission Control Protocol/Internet Protocol (TCP/IP).



For more specific environment requirements, please refer to the "RPC Broker Installation Guide."



Currently only Winsock-compliant TCP/IP protocol is supported on the LAN or remotely as Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP). You must use RAS (Remote Access Service) or Dialup Networking to connect to the server using PPP or SLIP. For the setup of RAS or Dialup Networking, please refer to the appropriate operating system's documentation.

When a user starts a *VISTA* program on the client, the program requests a connection with a server. The server is continuously running at least one Broker "Listener" job in the background whose sole purpose is to establish connections with clients.

Once the Listener receives a connection request, it does the following:

1. Validates the message.
2. Creates (spawns, jobs off) another process "Handler." The Handler process does the work to satisfy the client's requests.
3. Goes back to listening.

When the connection to the server is established, users who are not already logged into the server are asked to identify themselves by logging in with their Access and Verify codes. With the implementation of Auto Signon, users are considered already logged in to the server if they have previously logged in to a **VISTA** GUI or roll-and-scroll application that is still running on their workstation. After a successful login, the application is active on both the server and the client.



For more information on Auto Signon, please refer to the "Integrated Auto Signon for Multiple User Sessions" topic in Chapter 2, "System Features" in this manual.

As you manipulate the interface, your client process is reading and writing data to the server. The reading and writing is carried out as messages traveling over the TCP/IP link. In the message sent to the server, client applications will include the name of the requested RPC to be activated and its associated parameters. These RPCs will be written in M and registered in a file containing available and authorized RPCs (i.e., REMOTE PROCEDURE file [#8994]). Upon receipt by the server, the message is decoded, the requested remote procedure call is activated, and the results are returned to the calling application.

The server receives a message from the client and parses out the name of the remote procedure call and its parameters. The Broker module on the server looks up the remote procedure call in the REMOTE PROCEDURE file (#8994), verifies that the RPC is allowed to run in the context of the application, and executes the RPC using the passed parameters. At this point, the server side of the application processes the request and returns the result of the operation. The result of the call contains either several values or a single value. If the operation is a query, then the result is a set of records that satisfy that query. If the operation is to simply file the data on the server or it is unnecessary to return any information, then, typically, notification of the success of the operation will be returned to the client.



This version of the RPC Broker supports messaging for non-Delphi client applications (e.g., Borland C++, Microsoft Visual Basic, or other COTS Microsoft Windows-based products). The RPC Broker Version 1.1 supplies a set of functions providing a Dynamic Link Library (DLL) interface that allows non-Delphi applications to conform to the client side interface of the Broker.

For more specific information about the Broker DLLs, please refer to the "RPC Broker Developer's Guide" (i.e., BROKER.HLP, online help in the BDK).

System Overview

The following diagram gives an overview of the *VISTA*/Broker environment:

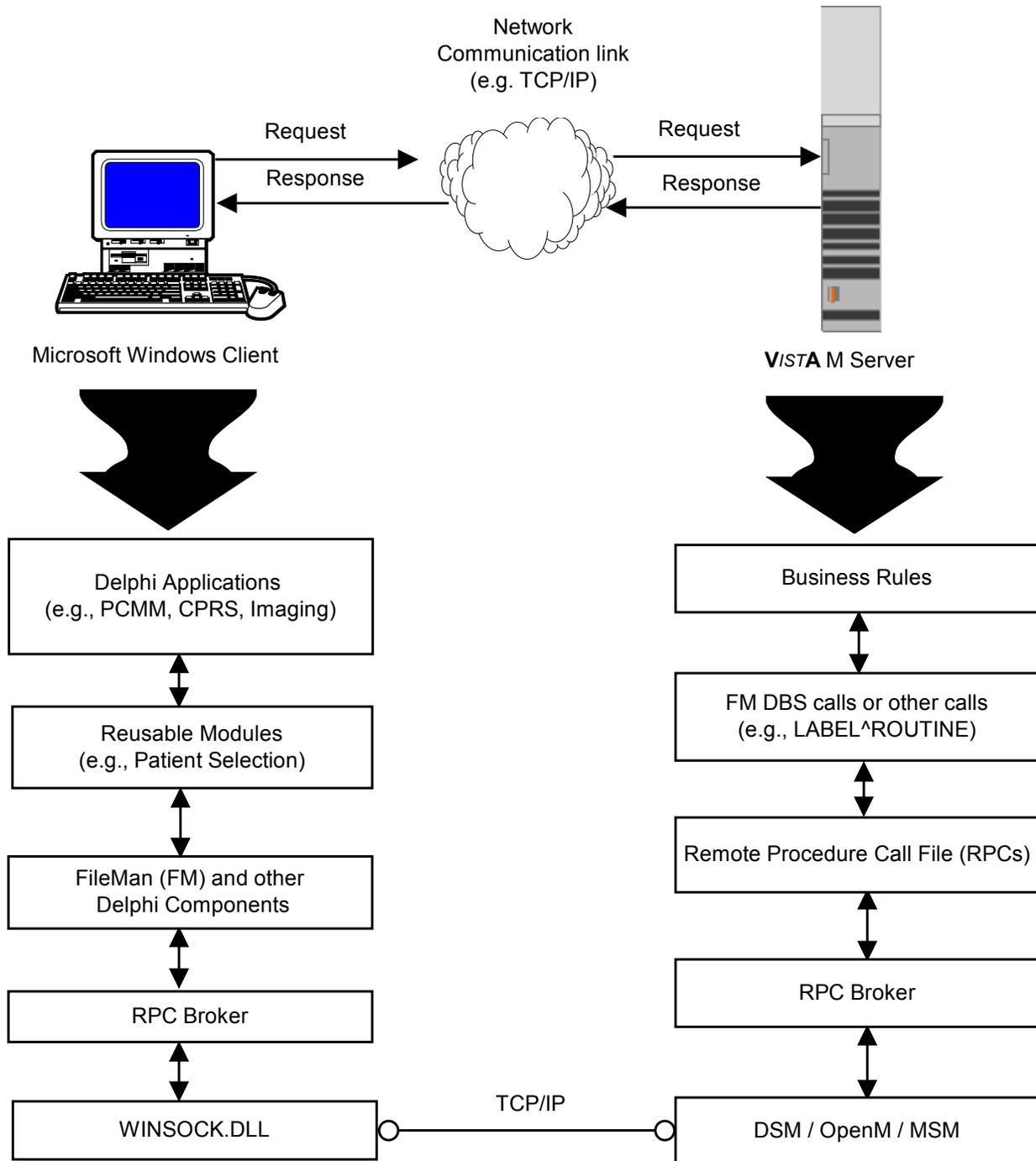


Figure 1: *VISTA* RPC Broker system overview diagram

2. System Features

Client Features

RPC Broker Client Agent

The RPC Broker Client Agent program (i.e., CLAGENT.EXE) runs in support of the Auto Signon process. This program automatically and continuously runs in the background on the client workstation and normally should *not* be closed or shut down by the user. A satellite dish icon will be displayed in the System Tray indicating the Broker Client Agent is running. The icon will change when an active connection is made to the server—a green line indicating an active connection will emanate from the satellite dish.

By double clicking on the Client Agent icon, you can see how many active connections are currently open, as shown below. However, the "Active connections" count may include "orphan" connections that are no longer active. Use this count as an approximate count only.



Figure 2: RPC Broker Client Agent dialog

i The "Start Client Agent with Windows" checkbox should be checked so that Auto Signon, if allowed, will be operational. By default, this box is checked. However, if a particular workstation is not always connected to the network upon startup, you may wish to prevent the Client Agent from starting automatically. You can always reset it to start automatically by starting the Client Agent manually first and re-checking this checkbox.

i For more information on Auto Signon, please refer to the "Integrated Auto Signon for Multiple User Sessions" topic that follows in this chapter.

The RPC Broker Client Agent is installed with the End-User Client Workstation installation of the RPC Broker and is *not* included with RPC Broker Development Kit (BDK).

i For more information on the End-User Client Workstation and Client Agent installation, please refer to the "RPC Broker Installation Guide."

"Connect To" Dialog

Upon logging in to a *VISTA* client/server application, users may be presented with the "Connect To" dialog, as shown below:



Figure 3: Server and port configuration selection dialog

This server and port configuration selection dialog can be used by Delphi *VISTA* client/server applications that wish to allow users to:

- Select an existing server name and associated port from a list of servers entered into the Microsoft Windows Registry
- Enter a new server name, Internet Protocol (IP) address, and associated port number.

For example, this can be useful when you want to run the application in either a Test or Production account.

To add a new server and associated port number to the Microsoft Windows Registry, press the New button (see Figure 3 above). You are presented with the "Add Server" dialog, as shown below:



Figure 4: Add Server dialog

You can also add additional server names and ports to the Microsoft Windows Registry by using the Edit Broker Servers program.

 For more information on adding new servers, please refer to the "Edit Broker Servers Program" topic that follows.

Edit Broker Servers Program

If someone in IRM wishes to add, modify, or delete servers and ports to be used by the Broker, they can run the Edit Broker Servers program (i.e., ServerList.EXE), as shown below:

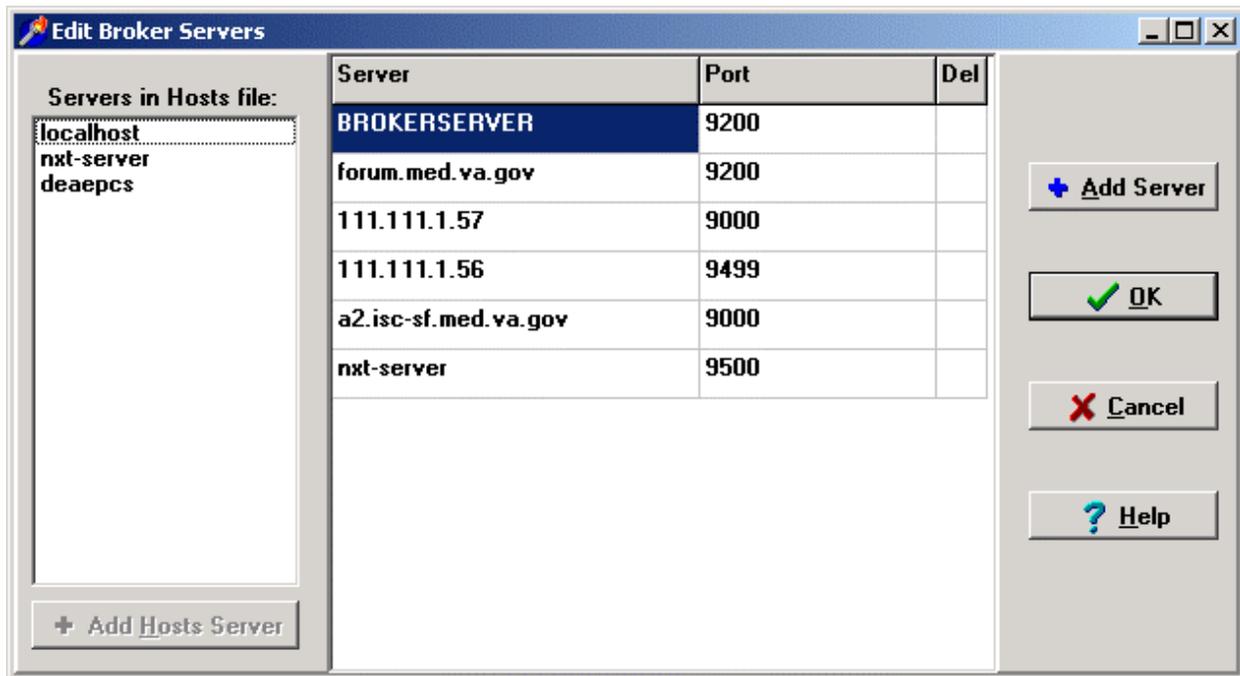


Figure 5: Edit Broker Servers dialog

Use this program to modify or add Listeners/Ports to the Microsoft Windows Registry. ServerList.EXE can be copied to any workstation for this purpose.

 This program only displays HOSTS file entries; it does not edit the HOSTS file.

Adding Entries

You are given two methods of adding new server entries to the Microsoft Windows Registry using the Edit Broker Servers program:

1. **Choose From an Existing List**—A list of servers available from your HOSTS file is displayed in the "Servers in Hosts file" list box on the left of the dialog (see Figure 5). Thus, you don't have to remember and type the server names yourself. Select one or more from the list and press the "+Add Hosts Server" button in the lower left of the dialog. This creates the new grid line(s) with the server(s) selected automatically "stuffed" into the Server cell(s). Complete each new entry by typing in the appropriate port(s). When finished, press OK.
2. **Add a New Server**—Alternatively, you can press the "+Add Server" button on the right of the dialog (see Figure 5). This creates a new grid line. Enter the name of the server you want added. Complete each new entry by typing in the appropriate port(s). When finished, press OK. The program will attempt to resolve the server name to an IP address either through the Domain Name Service (DNS) or by looking it up in the HOSTS file on the client workstation. If this is successful, the new entry will be added to the Microsoft Windows Registry. If the server name cannot be resolved, an error message will be displayed and you will have to correct your entry, as shown below:



Figure 6: Sample error message when adding a new server entry

- i** Hint: If you're running a PC Network with Microsoft Windows NT, the BROKERSEVER added to the Services list on the NT network will speed up client access times (i.e., keeps it from having to do a double lookup with first IP then service, it merely looks at the Services list).

Modifying Entries

In order to modify or change a server or port, simply place the cursor in the appropriate Server or Port field and make the change (see Figure 5). When finished, press OK.

- i** Server names must be resolvable through DNS or the HOSTS file.

Deleting Entries

In order to delete a pre-existing entry, just click in the Del column. An asterisk appears in the Del column signifying a deletion. Another click toggles the deletion off. When finished, press OK.

Standalone Programs and their Associated Help Files

Each of the following standalone Broker programs, distributed with this version of the Broker, have an associated help file that must reside in the *same* directory in order to provide online help for that particular standalone program:

Standalone Program	Associated Help File	Location
BROKERPROGPREF.EXE	BROKERPROGPREF.HLP	Programmer Workstation (BDK)
CLAGENT.EXE	CLAGENT.HLP	End-User Workstation
RPCTEST.EXE	RPCTEST.HLP	End-User Workstation
SERVERLIST.EXE	SERVERLIST.HLP	Programmer Workstation (BDK)

Table 3: Standalone RPC Broker programs and their associated help files

The installation of the Broker will automatically load these associated files into the appropriate directories. If you choose to "export" a standalone program (e.g., SERVERLIST.EXE) to another client workstation, make sure you include its associated help file and place them both in the *same* directory.



For more information on the BROKERPROGPREF.EXE, please refer to the "RPC Broker Developer's Guide" (i.e., BROKER.HLP, online help in the BDK).

For more information on the CLAGENT.EXE, please refer to the "RPC Broker Client Agent" topic previously described.

For more information on the RPCTEST.EXE, please refer to Chapter 4, "Troubleshooting" in this manual.

For more information on the SERVERLIST.EXE, please refer to the "Edit Broker Servers Program" topic previously described.

HOSTS File

The HOSTS file is an ASCII text file that contains a list of the servers and their IP addresses. However, use of the HOSTS file is *not* a requirement for the Broker. The use of the HOSTS file depends on the way the local area network (LAN) is implemented and managed at a site. Clients can bypass the HOSTS file and use DNS, DHCP (Dynamic Host Configuration Protocol), or WINS (Windows Name Service).

To modify or add servers to the HOSTS file, edit the file using a text editor (e.g., Microsoft Notepad).

The following table illustrates where you can find this file based on your client Microsoft Windows operating system (OS):

Version of Windows OS	File (Location and Name)
Windows 95	C:\WINDOWS\HOSTS
Windows 98	C:\WINDOWS\HOSTS
Windows NT 3.51	C:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS
Windows NT 4.0	C:\WINNT\SYSTEM32\DRIVERS\ETC\HOSTS
Windows 2000	C:\WINNT\SYSTEM32\DRIVERS\ETC\HOSTS

Table 4: HOSTS file location

A sample of the Microsoft Windows 2000 HOSTS file (i.e., C:\WINNT\SYSTEM32\DRIVERS\ETC\HOSTS) is displayed below (modifications/additions made to this sample file are in boldface and italicized):

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10       x.acme.com             # x client host
#
#   IP                Host
#   ADDRESS            Name                Description
#   |                  |                  |
#   |                  |                  |
#   V                  V                  V
127.0.0.1             localhost          # loopback
192.1.1.1            BROKERSERVER      # Broker Server
```

Figure 7: Sample HOSTS file

The last entry in this file (i.e., BROKERSERVER) was added to the sample HOSTS file for illustration purposes. We recommend you put in an entry that points to the main server you intend using with the Broker the majority of the time (e.g., BROKERSERVER). VISTA applications can specify any server they wish.



A DHCPSEVER entry is still required for software that uses Version 1.0 of the Broker. You may want to create an additional entry for BROKERSERVER in your HOSTS file or DNS. However, do not remove the DHCPSEVER entry already present.

Adding Entries

To add entries in the HOSTS file use a text editor (e.g., Microsoft's Notepad) to open the HOSTS file.

1. Move the cursor to the end of the last line displayed in the file.
2. Press the **Enter** key to create a new line.
3. On the new line, enter the desired IP address beginning in the first column, as described in the sample HOSTS file (see Figure 7). As recommended, add an appropriate IP address for the BROKERSERVER Host name as the next entry below "127.0.0.1".

4. After typing the IP address, type at least one space and enter the Host name that corresponds to that IP address. As recommended, type in BROKERSERVER as the next entry below "loopback".

For example, the entry for a server at your site with an IP address of 192.1.1.1 would look like the following:

```
127.0.0.1    localhost    # loopback    <---existing entry
192.1.1.1    BROKERSERVER # Broker Server <---added entry
```

5. Repeat Steps #1 - #4 until you have entered all of the IP addresses and corresponding Host names you wish to enter.
6. When your entries are complete, save the HOSTS file.



Do not save the HOSTS file with an extension!

7. Close the HOSTS file and text editor.

Modifying Entries

To modify entries in the HOSTS file use a text editor (e.g., Microsoft's Notepad) to open the HOSTS file.

1. Move the cursor to the line to be modified.
2. Modify the IP address, Host name, or both.
 - Make sure that at least one space separates the IP address from the corresponding Host name.
 - Make sure you have an entry for BROKERSERVER in this file.
3. Repeat Steps #1 - #2 until you have modified all of the IP addresses and corresponding Host names you wish to change.
4. When your entries are complete, save the HOSTS file.



Do not save the HOSTS file with an extension!

5. Close the HOSTS file and the text editor.

What Happened to the Client Manager?

The Client Manager, previously distributed with version 1.0 of the Broker, is no longer used by this version of the Broker. In version 1.0 of the Broker, the Client Manager provided two types of services:

1. It was used to invoke the RPCBI.DLL.
2. It was used by developers to set programmer preferences for using the original TRPCBroker component.

The RPCBI.DLL that was distributed with the RPC Broker V. 1.0 is no longer used, thus, the Client Manager is no longer required with this version of the Broker. Configuration of programmer preferences will now be done via the Broker Programmer Preferences dialog.



For more information on the Broker Programmer Preferences dialog, please refer to the "RPC Broker Programmer Preference Editor" topic in the "RPC Broker Developer's Guide" (i.e., BROKER.HLP, online help in the BDK).



The RPCBI.DLL and Client Manager (i.e., CLMAN.EXE) installed with Broker V. 1.0 must not be removed from the VISTA/Broker directory on the client workstation. They are still required for 16-bit Broker-based applications created using version 1.0 of the Broker (e.g., PCMM).

What Happened to the VISTA.INI File?

The VISTA.INI file is no longer used by applications built with Version 1.1 of the Broker. However, this file will continue to be used by applications built using version 1.0 of the Broker (e.g., PCMM). During the installation of the Broker, relevant data from the VISTA.INI file will be moved to the Microsoft Windows Registry. Subsequent reads and writes will be done via the Registry.



The VISTA.INI file created with Broker V. 1.0 must *not* be removed from the Microsoft Windows directory on the client workstation. It is still required for 16-bit Broker-based applications created using version 1.0 of the Broker (e.g., PCMM).

The following are a list of items from the VISTA.INI file and their disposition with Version 1.1 of the Broker:

VISTA.INI File Item	Disposition
ClientManagerPath ErrorRetry ClientManagerState	Client Manager items— <i>not</i> moved to the Microsoft Windows Registry.
IdeConnect ClearParameters ClearResults ListenerPort Server	Programmer items—moved to the Microsoft Windows Registry via a developer workstation installation (to be edited by the new Configuration form).
SignonPos SignonSiz IntroBackCol IntroTextFont	Signon items—moved to the Microsoft Windows Registry (these did <i>not</i> exist in version 1.0). These items will now be edited from the Signon form.  For more information, please refer to the "Users Can Customize Signon Dialog" topic in Chapter 2 of this manual.
HostsPath	No longer useful (i.e., Broker V. 1.1 Delphi code will <i>not</i> reference it).
[RPCBroker_Servers] section	Server/Port pairs—moved to the Microsoft Windows Registry via general workstation installations. These entries will now be edited via the Edit Broker Servers application.  For more information, please refer to the "Edit Broker Servers Program" topic in this chapter.

Table 5: VISTA.INI entries and Microsoft Windows Registry disposition table

Server Features

Menu for System Managers

Patch XWB*1.1*9 introduced a menu for system managers [XWB MENU]:

```
Select RPC Broker Management Menu Option:
```

```
RPC Listener Edit
Start All RPC Broker Listeners
Stop All RPC Broker listeners
```

Broker Listeners and Ports

You can run:

- A *single* Broker Listener, running on any available port.
- *Multiple* Broker Listeners running on the same IP address/CPU, but listening on *different* ports.
- *Multiple* Broker Listeners in the *same* UCI-volume, but on *different* IP addresses/CPUs, listening on the *same* port (or on different ports).

Thus, for example, to run one listener in a Production account and another in a Test account, on the same IP address/CPU, you must configure them to listen on different ports (e.g., 9200 for production and 9201 for Test). If, on the other hand, you are running the listeners on different IP addresses/CPUs, the ports can be the same (e.g., one Broker Listener on every system, listening on port 9200).

You need to configure your clients to connect to the appropriate listener port on your M server. While 9200 has been used as a convention for a Broker-based application service port, you can choose any available port greater than 1024 (sockets 1 to 1024 are reserved for standard, well-known services like SMTP, FTP, Telnet, etc.)

OBTAINING AN AVAILABLE LISTENER PORT (FOR ALPHA/VMS SYSTEMS ONLY)

Port selections conflict only if another process on the same system is using the same port. To list the ports currently in use on OpenVMS systems, use the DCL command:

```
$ UCX SHOW DEVICE_SOCKET
```

Device_socket	Type	Local	Port		Remote Host
			Remote	Service	
bg3	STREAM	5001	0	HL7	0.0.0.0
bg23	STREAM	9700	0	Z3ZTEST	0.0.0.0
bg24	STREAM	9600	0	ZSDPROTO	0.0.0.0

For example, if 9200 shows up in the Local Port column, some other application is already using this port number and you should choose another port.

Starting And Stopping Listeners

TO START ALL LISTENERS

To start all listeners (i.e., those configured in the RPC Broker Site Parameters file to automatically start), use the "Start All RPC Broker Listeners" option (introduced with Patch XWB*1.1*9). This option first **stops** any of these listeners that may be running, and then starts all of them up. TaskMan must be running.



DSM sites must run TaskMan in a DCL context, to start Listeners on any other node than the current ones. For more information on running TaskMan in a DCL context on DSM for OpenVMS, please refer to the "Kernel V. 8.0 Systems Manual."

TO CONFIGURE LISTENERS FOR AUTOMATIC STARTUP

To configure a given listener for startup by the "Start All RPC Broker Listeners" option, enter YES in the "Controlled By Listener Starter" field in the RPC Broker Site Parameters file for that listener.



For more information, please refer to the "RPC BROKER SITE PARAMETERS File" topic in this chapter.

TO STOP ALL RUNNING LISTENERS

To stop all running listeners (but, only those configured in the RPC Broker's site parameters to automatically start), use the "Stop All RPC Broker Listeners" option.



It is important to stop all Listeners before shutting down the system!

TO START UP A SINGLE LISTENER DIRECTLY

Enter the following at your M server's M prompt:

```
>D STRT^XWBTCP(Listener port)
```

TO STOP A SINGLE LISTENER DIRECTLY

Enter the following at your M server's M prompt:

```
>D STOP^XWBTCP(Listener port)
```



If you want to restart this listener after stopping it, and other listeners are running on your system, start the listener up directly (see above) rather than via the "Start All RPC Broker Listeners" option (which first stops all listeners before restarting them).

TO TASK THE XWB LISTENER STARTER OPTION FOR SYSTEM STARTUP

The XWB LISTENER STARTER option (which starts all configured Broker Listeners at one time). can be tasked to automatically start all of the Listener processes you need when TaskMan starts up, such as after the system is rebooted or configuration is restarted.

-  DSM sites must have TaskMan started via DCL, in order to use the XWB LISTENER STARTER option to automatically start Listener processes.

To automatically start the Listener(s) when TaskMan is restarted (i.e., in addition to the entries in the RPC BROKER SITE PARAMETERS file [#8994.1]), enter the XWB LISTENER STARTER option in the OPTION SCHEDULING file (#19.2). Schedule this option with SPECIAL QUEUING set to STARTUP. You can do this by using the TaskMan option: Schedule/Unschedule Options:

```

Select Systems Manager Menu Option: TASKMAN Management
Select Taskman Management Option: SCHedule/Unschedule Options
Select OPTION to schedule or reschedule: XWB LISTENER STARTER <Enter> Start All
RPC Broker Listeners
    ...OK? Yes// <Enter> (Yes)
    (R)

                                Edit Option Schedule
Option Name:   XWB LISTENER STARTER
Menu Text:    Start All RPC Broker Listeners      TASK ID:
-----
QUEUED TO RUN AT WHAT TIME:
DEVICE FOR QUEUED JOB OUTPUT:
QUEUED TO RUN ON VOLUME SET:
RESCHEDULING FREQUENCY:
TASK PARAMETERS:
SPECIAL QUEUEING: STARTUP
-----

```

Figure 8: Automatically starting the Listener(s) when TaskMan is restarted

-  If you are an MSM 4.3.0 site or greater and using MSERVER instead of the Broker Listener, the XWB LISTENER STARTER option is not applicable to your site. Please refer to the "MSM for NT 4.3.0 MSERVER Replaces RPC Broker Listener Process" topic below.

RPC BROKER SITE PARAMETERS File

The RPC BROKER SITE PARAMETERS file (#8994.1) contains one top-level entry, whose .01 field is a pointer to the DOMAIN file (#4.2). When the RPC Broker is installed, you create this top-level entry and assign the proper Domain Name.

The site parameters in this top-level entry pertain to listeners. For each listener that you plan to run on your system, you should make an entry for that listener in the site parameters.

EDITING THE LISTENER SITE PARAMETERS

To create or edit listener entries, use the RPC Listener Edit option.

The RPC Listener Edit option first prompts you to select a Box-Volume Pair entry. Then, within each Box-Volume Pair entry (representing the volume set and system on which the listener should run), you can configure one or more listeners:

```
Select RPC BROKER SITE PARAMETERS DOMAIN NAME: YOURSITE.VA.GOV
...OK? Yes// <Enter> (Yes)

Select BOX-VOLUME PAIR: KDE:ISC6A2// <Enter>
BOX-VOLUME PAIR: KDE:ISC6A2// <Enter>
Select PORT: 9500// <Enter>
PORT: 9500// <Enter>
STATUS: STARTING// <Enter>
CONTROLLED BY LISTENER STARTER: YES//
```

The meaning of the site parameter fields for a given listener entry is as follows:

Field	Meaning
Box-Volume Pair	Choose the Box-Volume pair representing one of the systems supporting "this" account, and on which a listener should run.
Port	The port the listener will listen on.
Status	Ordinarily should not be edited (Use the "Start All RPC Broker Listeners" and "Stop All RPC Broker Listeners" options to start and stop listeners.)
Controlled By Listener Startup	If the listener should be started by the "Start All RPC Broker Listeners" option (XWB LISTENER STARTER), set this field to YES. Otherwise, set to NO.

Table 6: Listener site parameter entries description table

Integrated Auto Signon for Multiple User Sessions

Version 1.1 of the RPC Broker supports Kernel's Auto Signon from a client workstation to the server. Users need only sign on once (i.e., enter their Access and Verify codes) when accessing both a **VISTA** roll-and-scroll (e.g., Lab, Pharmacy) and a **VISTA** client/server GUI-based application (e.g., CPRS, NOIS, PCMM) on the same workstation, regardless of which application is started first. Once logged into the server, the user will *not* be asked to re-enter their Access and Verify codes for any subsequent **VISTA** applications they may start.



Auto Signon is facilitated on the client side by the Broker Client Agent application (CLAGENT.EXE) and is only available for Telnet-based sessions in the roll-and-scroll environment.

ENABLING/DISABLING AUTO SIGNON

Control of the Auto Signon functionality is maintained and administered on the server for both **VISTA** client/server applications (i.e., GUI) and the roll-and-scroll environment (i.e., terminal sessions). In support of that functionality, the DEFAULT AUTO SIGN-ON field was added to the KERNEL SYSTEM PARAMETERS file (#8989.3) and the AUTO SIGN-ON field was added to the NEW PERSON file (#200). The valid values for these fields are Yes, No, or Disabled.

These fields, in conjunction with the other multiple signon fields, give the sites control of the implementation of Auto Signon for users in both the GUI and roll-and-scroll environments. The values in the AUTO SIGN-ON and MULTIPLE SIGN-ON fields in the NEW PERSON file (#200) take precedence over the values in the DEFAULT AUTO SIGN-ON and DEFAULT MULTIPLE SIGN-ON fields in the KERNEL SYSTEM PARAMETERS file (#8989.3). Therefore, the fields in the NEW PERSON file are checked first. If the user fields in the NEW PERSON file are null, the values in the KERNEL SYSTEM PARAMETERS file will be used.



The AUTO SIGN-ON field in the NEW PERSON file and the DEFAULT AUTO SIGN-ON field in the KERNEL SYSTEM PARAMETERS file are initially set to null.



Auto Signon is not supported on MSM systems. All MSM sites must set the DEFAULT AUTO SIGN-ON field in the KERNEL SYSTEM PARAMETERS file (#8989.3) to Disabled.



If a user is *not* allowed multiple signons, they will only be allowed to initiate a *single* session (i.e., automatically disallowing Auto Signon).

Example 1:

If a user has an active **VISTA** session and has the following characteristics:

- Allowed multiple signons (i.e., the MULTIPLE SIGN-ON field in the NEW PERSON file (#200) is set to Yes)
- Allowed Auto Signon (i.e., the AUTO SIGN-ON in the NEW PERSON file (#200) is set to Yes)

They will be allowed to start another *VISTA* session *without* having to re-enter their Access and Verify codes.

Example 2:

If a user has an active *VISTA* session and has the following characteristics:

- Allowed multiple signons (i.e., the MULTIPLE SIGN-ON field in the NEW PERSON file (#200) is set to Yes)
- Not allowed Auto Signon (i.e., the AUTO SIGN-ON field in the NEW PERSON file (#200) is set to No)

They will be allowed to start another *VISTA* session, however, they *must* re-enter their Access and Verify codes.

The following table can be used as a guide to control multiple signons and Auto Signon for some typical situations:

Description	* User Settings	** System Settings
Multiple Signon:		
Disallow <i>all</i> users from having multiple signons	No/Null	No
Allow <i>individual</i> users to have multiple signons	Yes	No
Allow <i>all</i> users to have multiple signons	Yes/Null	Yes
Auto Signon:  With the exception for disabling Auto Signon, the following settings are only affective when users are allowed multiple signons.		
Stop Auto Signon	Any Value	† Disabled
Allow <i>individual</i> users to have Auto Signon	Yes	No
Disallow <i>individual</i> users from having Auto Signon	No	Yes
Allow <i>all</i> users to have Auto Signon	Yes/Null	Yes

Table 7: Multiple and Auto Signon Settings table

* User Settings refers to the NEW PERSON file (#200) and the following fields:

- MULTIPLE SIGN-ON (#200,200.04)
- AUTO SIGN-ON (#200,200.18)



The User Settings override the **System Settings except when *disabling* Auto Signon!

** System Settings refers to the KERNEL SYSTEM PARAMETERS file (#8989.3) and the following fields:

- DEFAULT MULTIPLE SIGN-ON (#8989.3,204)
- DEFAULT AUTO SIGN-ON (#8989.3,218)

† Sites may choose to disable Auto Signon (stops calls to the Broker Client Agent) for all users in the following situations:

- Network problems
- Broker not installed
- During installation of the Broker



Auto Signon is *not* supported in MSM systems. All MSM sites must set the DEFAULT AUTO SIGN-ON field in the KERNEL SYSTEM PARAMETERS file (#8989.3) to Disabled.

RPC Broker Message Structure

The messages that are sent from a server to a client contain either several values or a single value. Presently, the RPC Broker messages are bound by the Microsoft Windows WINSOCK.DLL specifications and the size of the symbol table. The server receives a message from the client and parses out the name of the remote procedure call and its parameters. The Broker module on the server looks up the remote procedure call in the REMOTE PROCEDURE file (#8994) and executes the RPC using the passed parameters. At this point the server side of the application processes the request and returns the result of the operation. If the operation is a query, then the result is a set of records that satisfy that query. If the operation is to simply file the data on the server or it is unnecessary to return any information, then, typically, notification of the success of the operation will be returned to the client.

The basic RPC Broker message structure consists of the following:

- A header portion (which includes the name of the remote procedure call).
- The body of the message (which includes descriptors, length computations, and M parameter data).

Client/Server Timeouts

The issue of timeouts is complex in a client/server environment. Because the user may be working with applications that rely solely on the client, long periods of time may elapse that the server would traditionally have counted against the user's timeout.

Broker Patch XWB*1.1*6 was created to address timeout issues. It instituted a "keep-alive" timer that was compiled into client applications. Through monitoring this keep-alive timer, the software is able to eliminate "ghost" server Broker jobs for which there is no longer a client application, based on the keep-alive timer rather than on user activity.

"Ghost" server jobs occur when client processes are ended in a non-standard way—for example, by pressing the PC's reset button. Prior to this patch, these jobs would wait for 10 hours to receive data from the client application that no longer existed.

In order to let the server know that the client application is still active, applications compiled with the client portion of Patch XWB*1.1*6 (and beyond) initiate a periodic, background contact with the server. This "polling" of the server by the client resets the timeout so that the server job is not stopped when the client still exists. Any client application compiled with the TRPCBroker and/or TSharedRPCBroker components distributed with the latest patch automatically polls. No developer or user intervention is necessary, and this polling activity affects neither the application nor the user.

The BROKER ACTIVITY TIMEOUT field in the KERNEL SYSTEM PARAMETERS FILE controls the length of the timeout. That field was distributed by Kernel Patch XU*8.0*115 with a default value of approximately 3 minutes. By setting the timeout to a duration much shorter than 10 hours, the ghost jobs are eliminated quickly, if the client application is no longer running.



For advice regarding changing the value for this field, please refer to the help for the BROKER ACTIVITY TIMEOUT field.

The server portion of this patch is backwards compatible with client applications compiled with previous versions of the Broker. Thus, client applications do not have to be recompiled when this patch is installed on the server. The server retains a 10-hour timeout for those client applications compiled with previous Broker versions; that is, they continue to work as they did before the patch is installed.



The server side of this patch is effective only for client applications (like CPRS-GUI) that have been recompiled with the Broker Development Kit (BDK) portion of Patch XWB*1.1*6. Thus, installing the server patch alone does not eliminate the ghost jobs for client applications that have not been upgraded.

Load Balancing on Alpha Systems

The Broker, like any Telnet or IP process, can be load balanced on DSM Alpha systems, if UCX 4.1 is running. The actual steps on configuring UCX for load balancing can be acquired from the ALPHA/AXP technical support group and will not be discussed here.

Multiple Broker servers can run on the same port as long as the machine IP addresses are unique. This is *not* a Broker requirement; it is a TCP/IP requirement. This capability is necessary for UCX load balancing. The multiple servers will receive a common alias that will be the connection destination.

In UCX, you should use the BIND alias:

For example:

```
UCX> show host vista.sitename.med.va.gov

      BIND database

Server:  152.999.999.xxx      999TNG

Host address      Host name
152.999.999.yy1   VISTA.SITENAME.MED.VA.GOV
152.999.999.yy2   VISTA.SITENAME.MED.VA.GOV
152.999.999.yy3   VISTA.SITENAME.MED.VA.GOV
152.999.999.yy4   VISTA.SITENAME.MED.VA.GOV
152.999.999.yy5   VISTA.SITENAME.MED.VA.GOV
```

In order to use load balancing, your client workstation needs to have DNS enabled and pointing to the IP address of the DNS server in the list. The Broker on the client will use the PC's DNS or HOSTS file to resolve the BROKERSERVER host name. In the previous example, the first DNS server is 152.999.999.xxx.

For example, if you want CPRS GUI to be "balanced," use the Edit Broker Servers program to edit the servers in the Microsoft Windows Registry and add in the alias VISTA.*sitename*.MED.VA.GOV.



For more information on adding servers to the Registry, please refer to the "Connect To" Dialog or "Edit Broker Servers Program" topics in this chapter.

You don't want the alias in the HOSTS file because the HOSTS file is for static bindings only. If you want to put the alias in the HOSTS file, then you will have to make sure that the DNS server is first in the DNS list. Thus, when the user selects the BIND alias, the DNS will resolve it to one of the unique IP addresses and *not* to the HOSTS static assignment.



DSM sites do not need to have TaskMan started in a DCL context in order to use load balancing.

MSM for NT 4.3.0 MSERVER Replaces RPC Broker Listener Process

MSM for NT in version 4.3.0 introduced a generic TCP/IP Listener MSERVER. MSERVER is an actual routine in the Manager's UCI that runs and listens to all of the ports that you specify in SYSGEN. When a connection is established to one of these ports, MSERVER launches your code at some TAG^ROUTINE that you specify. This is similar to VMS' UCX utility. If you are running a Beta version of MSM 4.3 for NT, we encourage you to upgrade to 4.3.0. Once you upgrade, you should stop using the RPC Broker Listener and switch to the MSERVER.

Before MSERVER can be used, it must be configured using SYSGEN. The following is an example from the Oakland Office of Information Field Office (OIFO):

1. In the Manager's UCI run SYSGEN.
2. In sequence pick the following options:
 - a. 3-Edit Configuration Parameters
 - b. 15-Network Configuration
 - c. 12-User-Defined Services
3. Enter an unused index number and set it up as indicated below:
 - a. Enter Service Name <RPC BROKER>: **RPC BROKER**
 - b. Enter Routine Reference <MSM^XWBTCPC>: **MSM^XWBTCPC**
 - c. Select UCI <VAH,MNT>: **VAH,MNT**
 - d. Enter Partition Size <80>: **80**
 - e. Enter Password <>: **<Enter>**
 - f. Enter TCPIP Port Number <9200>: **9200**
 - g. Autostart? <YES>: **YES**

However, specifying YES for Autostart is not enough for MSERVER to start up when MSM is started. You must also add MSERVER to the Automatic Startup List as indicated in Steps 4-6 below.

4. To schedule MSERVER to start automatically when MSM comes up, go to the SYSGEN main menu.
5. In sequence pick the following options:
 - a. 3-Edit Configuration Parameters
 - b. 3-Autostart and Automounts
 - c. 5-Automatic Partition Startup
6. Enter an unused index number and set it up as indicated below:
 - a. Enter UCI name <MGR>: **MGR**
 - b. Enter Entry Reference <STARTUP^MSERVER("RPC BROKER")>: **STARTUP^MSERVER("RPC BROKER")**
 - c. Enter partition size <SYSTEM>: **SYSTEM**

To start MSERVER manually, in the Manager's UCI, type the following at the programmer prompt:

```
>J STARTUP^MSERVER("RPC Broker")
```

To shut down MSERVER in the Manager's UCI, type the following at the programmer prompt:

```
>J SHUTDOWN^MSERVER("RPC Broker")
```

If you're successfully using MSERVER, discontinue using the Broker Listener. That means you shouldn't use the STRT^XWBTCP and STOP^XWBTCP entry points or the RPC BROKER SITE PARAMETERS file previously discussed.

3. Security

Security Features

Security in distributed computing environments, such as in client/server systems, is much more complicated than in traditional configurations. Although it is probably impossible to protect any computer system against the most determined and sophisticated intruder, the RPC Broker implements robust security that is transparent to the end user and without additional impact on IRM.

Security with the RPC Broker is a four-part process:

1. Client workstations must have a valid connection request
2. Users must have valid Access and Verify codes
3. Users must be valid users of a *VISTA* client/server application
4. Any remote procedure call must be registered and valid for the application being executed

Validation of Connection Request

An enhancement to security has been included with this version of the Broker. Before the Broker Listener jobs off a Handler for a client, it checks the format of the incoming connection request. If the incoming message does not conform to the Broker standard, the connection is closed. This serves as an early detection of impostors and intruders.

Validation of Users

The GUI *VISTA* Sign-on dialog is integrated with the RPC Broker interface. This *VISTA* Sign-on dialog is invoked when the client application connects to the server.

VISTA Sign-on Dialog

The *VISTA* Sign-on dialog automatically prompts users for their Access and Verify codes if they are not already signed on to a *VISTA* application (see Figure 9).

VISTA Division Selection Dialog

After entering an Access and Verify code, if a user is associated with more than one Institution, the user will be presented with the following:

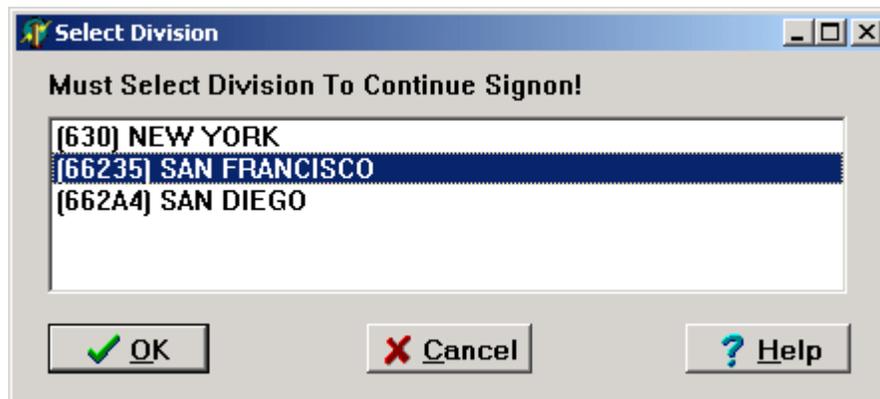


Figure 10: Sample Select Division dialog

To continue the signon process, the user must select a division from the list presented. The user's default division will initially be highlighted. To choose a different division, users should click on or use the arrow keys to highlight the appropriate division and press the OK button after making their selection. The signon process will log the user into *VISTA* with their DUZ(2) set to that division.

Client/server applications are "B"-type options (i.e., Broker options) in the OPTION file (#19). Users must have the client/server application option assigned to them like any other assigned option in *VISTA*. It can be put on their primary menu tree or as a secondary option/menu as part of their suite of permitted options. The client/server application will only run for those users who are allowed to activate it.

 The client/server application options will not be displayed in a user's menu tree.

Kernel's Menu Manager verifies that users are allowed access to a *VISTA* application or option with the following process:

1. Users start a *VISTA* program.
2. The RPC Broker in the client application invokes the *VISTA* Sign-on dialog when connecting to the server.
3. Users sign on to the server via the Kernel signon process.
4. If authorized, the user is granted access to the server, otherwise an error message is returned. This serves as an initial security check.

 For more information on Access and Verify codes or the Kernel signon process in general, please refer to the "Signon/Security" section in the "Kernel V. 8.0 Systems Manual."

Users Can Customize VISTA Sign-on Dialog

When a *VISTA* application on the client connects to the server, the *VISTA* Sign-on dialog is displayed for the user to identify and authenticate himself on the server. The *VISTA* Sign-on dialog System menu has a "Properties..." item, as shown below:

Move your mouse anywhere in the dialog window's Title bar and right click (left click if you are left handed) to display the System menu.

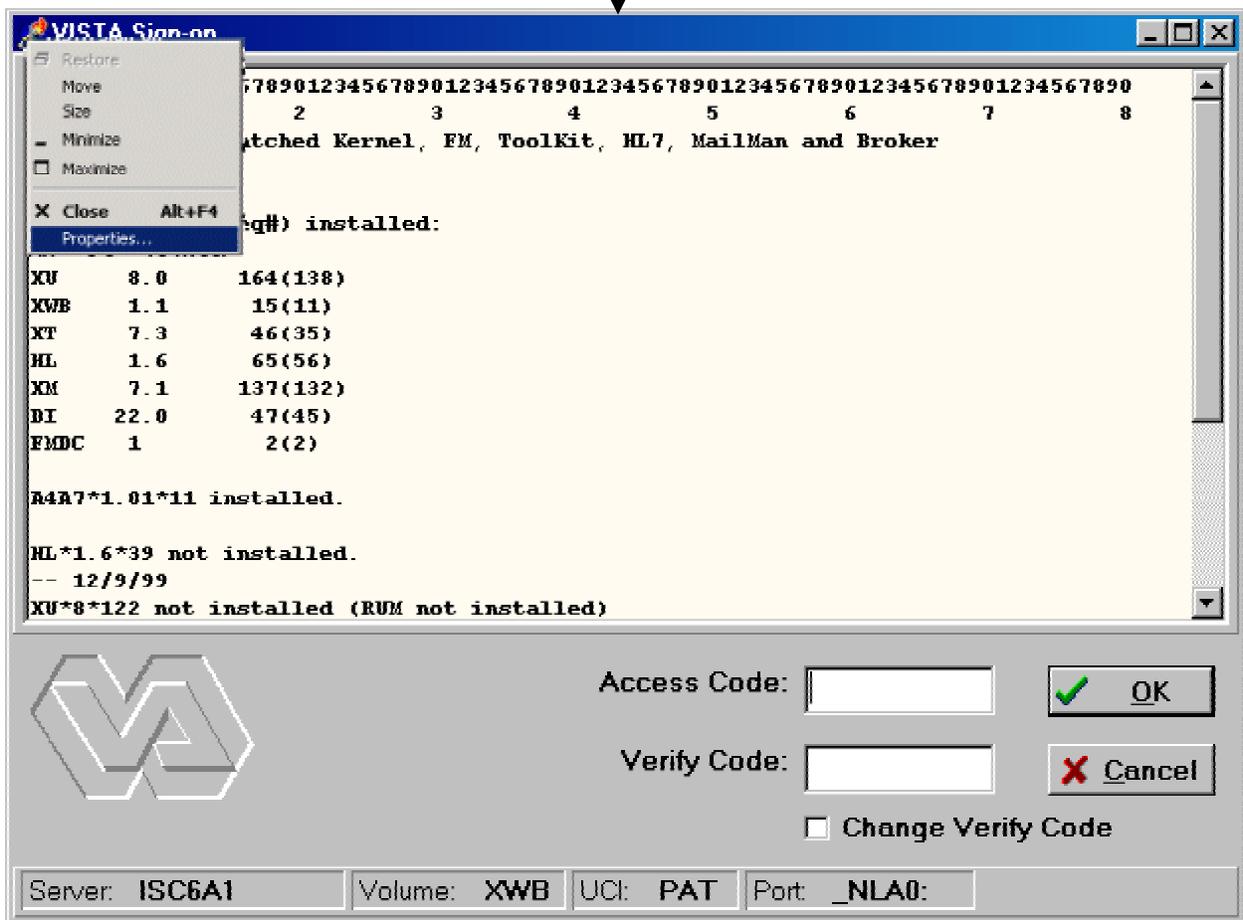


Figure 11: Sign-on Properties on the System Menu

When this item is selected, the user is presented with the Sign-on Properties dialog, as shown below:

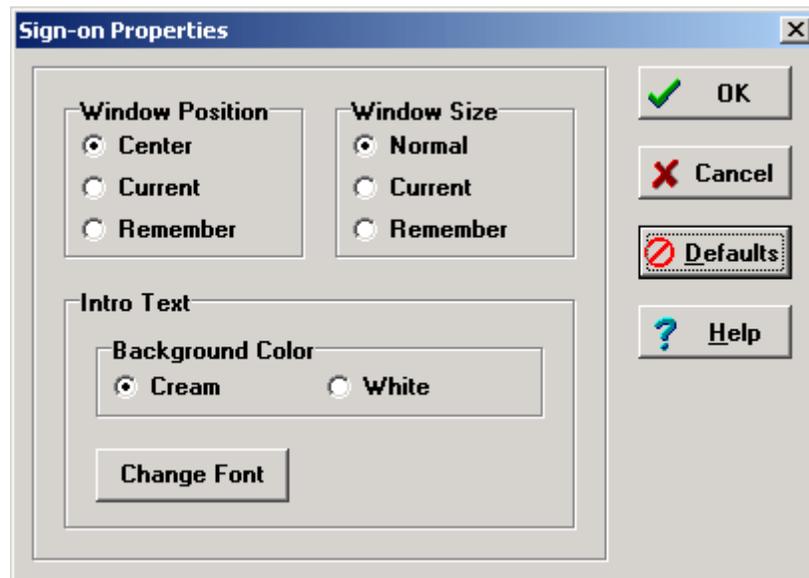


Figure 12: Sign-on Properties dialog

Using this form (see Figure 12), users can control the appearance of the *VISTA* Sign-on dialog by modifying the following characteristics:

- Window Position—The position of the *VISTA* Sign-on dialog.
- Window Size—The size of the *VISTA* Sign-on dialog.
- Introductory Text—The appearance of the introductory text in the *VISTA* Sign-on dialog.

Window Position

The *VISTA* Sign-on dialog's window position can be one of the following:

Center (default)	The <i>VISTA</i> Sign-on dialog will always appear in the center of the screen.
Current	The current position of the <i>VISTA</i> Sign-on dialog will be saved and used in the future.
Remember	Each time the <i>VISTA</i> Sign-on dialog is used and closed, it will record its position and open in that same place the next time it is used.

Window Size

The *VISTA* Sign-on dialog's window size can be one of the following:

Normal (default)	The size of the <i>VISTA</i> Sign-on dialog as it was designed. Typically, this is 500 pixels wide by 300 pixels high.
Current	The current size of the <i>VISTA</i> Sign-on dialog will be saved and used in the future.
Remember	Each time the <i>VISTA</i> Sign-on dialog is used and closed, it will record its size and open with the same size the next time it is used.

Introductory Text

The *VISTA* Sign-on dialog's introductory text has a couple of settings users can control:

Background Color:

Cream (default)	According to the VA GUI conventions, this is the background color that should be used with text that users <i>cannot</i> edit.
White	For clarity and brightness.

Font:

When users press the "Change Font" button they are presented with a Font form that can be used to change the font face, style, size, effects, and color of the introductory text of the *VISTA* Sign-on dialog, as shown below:

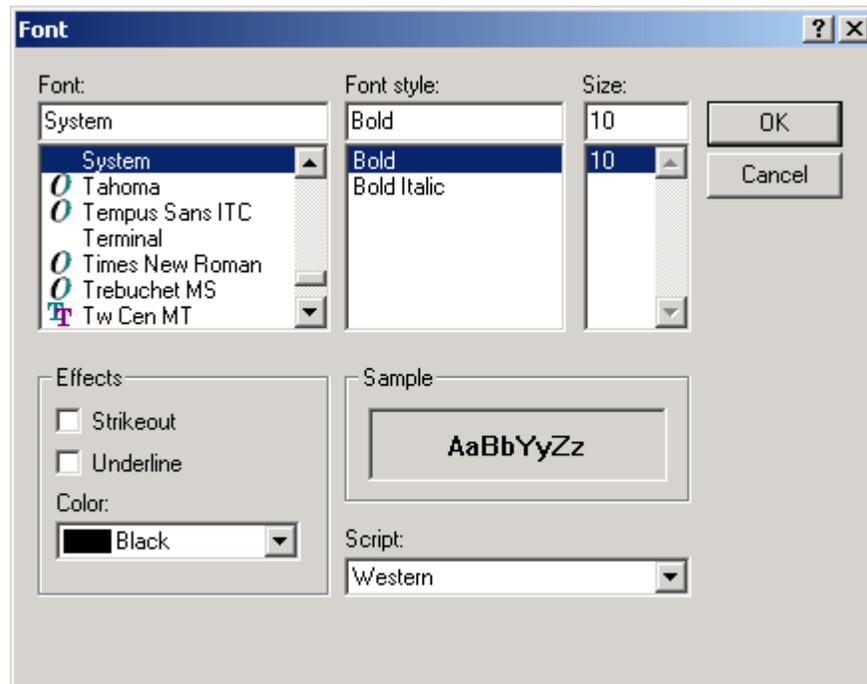


Figure 13: Sample Font dialog

Change *VISTA* Verify Code Component

Version 1.1 of the Broker includes a Change *VISTA* Verify Code dialog for the client workstation. After a user signs onto the server, if their Verify code has expired, the user is automatically prompted with the following message: "You must change your Verify code at this time." Once the user presses the OK button they are presented with the Change *VISTA* Verify Code dialog as displayed below:

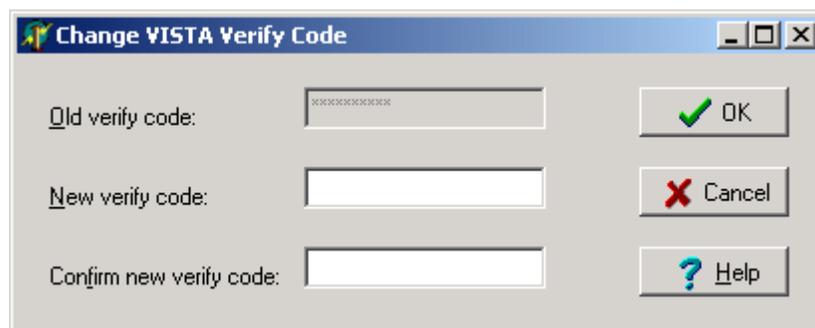


Figure 14: Change *VISTA* Verify Code dialog



The old Verify code will appear as asterisks (*) in a grayed-out box.

Users must then do the following:

- Enter their new Verify code
- Confirm their new Verify code

Users who wish to change their Verify code prior to its expiration can do so by either of the following methods:

- GUI environment (available as of Broker Patch XWB*1.1*13)—Click on the checkbox labeled "Change VC" on the Sign-on screen (see Figure 9). After signing on, it invokes the dialog described above (see Figure 14).
- Roll-and-Scroll environment (existing functionality)—Use the Edit User Characteristics option [XUSEREDITSELF] to edit your Verify code.

Validation of RPCs

The RPC Broker security allows any RPC to run when it is properly registered to the *VISTA* client/server application. The Broker on the server along with Kernel's Menu Manager determines which application a user is currently running. Menu Manager determines if a user is allowed to run this application or option by the following process:

1. A remote procedure call is sent by a client application and is received by the RPC Broker on the server.
2. The Broker verifies that the RPC is "registered" to the application that the user is currently running, *prior* to executing the remote procedure call (RPC).

The application being run is designated by a "B"-type option in the OPTION file (#19). The application must specify the option and that option *must* be in a user's menu tree.



For more information on registering an RPC to a package, please refer to the "RPC Security: How to Register An RPC" topic in the "RPC Broker Getting Started with the Broker Development Kit (BDK)" manual.

3. Menu Manager checks if the RPC is registered for this package option. If not properly registered, Menu Manager will return a message explaining why the RPC is not allowed.
4. The Broker on the server executes the RPC if it is registered, otherwise it is rejected.

Sample Security Procedures

The security steps each client user will follow and the intermediate client/server security processes are described in the following example:

- | Step | Description |
|------|--|
| 1. | The user starts a <i>VISTA</i> program on the client. For this example, the user clicks on the Computerized Patient Record System (CPRS) application icon. |
| 2. | The user must sign on to the server through the <i>VISTA</i> Sign-on dialog (see Figure 9) on the client using their Access and Verify codes invoking the Kernel signon process. |
| 3. | The Menu Manager on the server verifies the user is allowed access to the "B"-type option requested by CPRS. |
| 4. | The Menu Manager on the server verifies the option is a "client/server" type option and the requested RPC is in that option's RPC multiple. |
| 5. | If all of the previous steps complete successfully, the application RPC is launched. |

Security Features Tasks Summary

The following table summarizes required security tasks:

Security Task	Completed By
Verify valid connection request	RPC Broker
Verify valid user	Kernel Signon
Verify user is authorized to run this package	RPC Broker & Menu Manager
Verify an RPC is registered to an application	RPC Broker & Menu Manager
Application—RPC Registration	KIDS

Table 8: Security Tasks



To reiterate, an RPC is only allowed to run within the context of an application with which it is registered. Users are only able to run the server side of the application that was installed on the server by IRM.



For each release of the RPC Broker, the RPC Broker Development Team will continuously strive to implement the most complete, robust, and flexible security available at the time.

4. Troubleshooting

Test the Broker Using the RPC Broker Diagnostic Program

This version of the Broker includes a diagnostic tool for the client workstation (see Figure 15). This tool can be used to verify and test the Broker client/server connection and signon process. This program (i.e., RPCTEST.EXE) also displays specific information about the client workstation that can be useful to IRM personnel when trying to determine and/or correct any problems with or to test the Broker.

It displays the following information:

- Default workstation information that includes the Name and IP Address.
- Local connection information that includes the Name, Client IP, Current Socket, and Broker State.
- **VISTA** user information that includes the Name and Last SignOn Date/Time.
- Remote connection information that includes the Server, Port, IP Address, Operating System Version information, and Job ID.
- A color-coded Link State indicator that shows the status of your connection:
 - Red = no link/connection
 - Yellow = attempting link/connection
 - Green = successful link/connection

When you run the RPC Broker Connection Diagnostic Program (i.e., RPCTEST.EXE), the following dialog will be displayed:

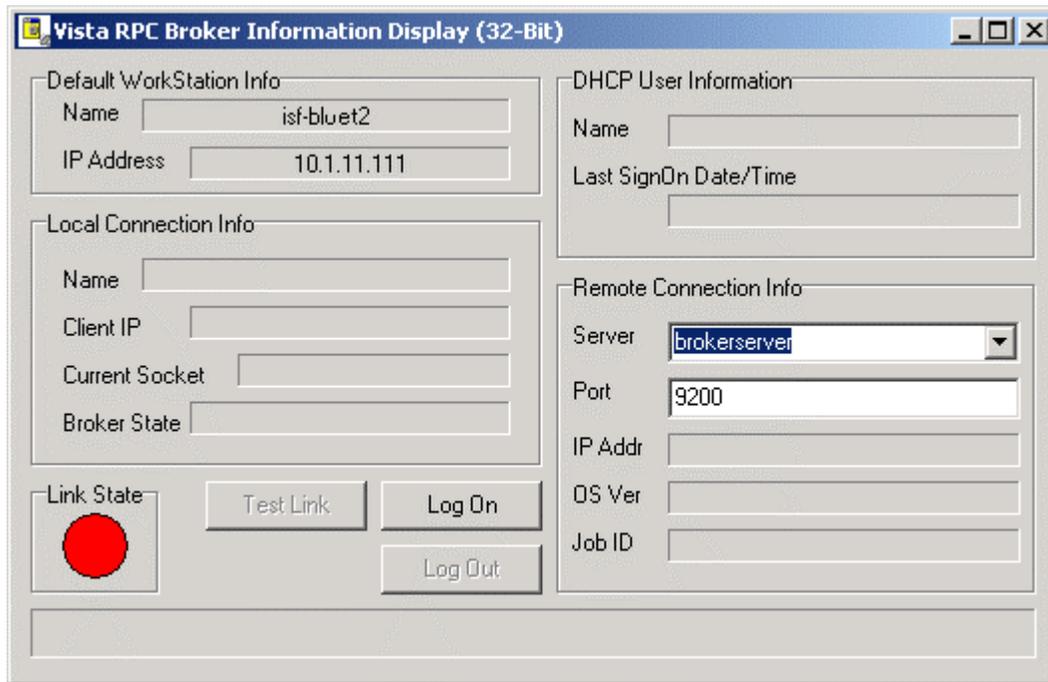


Figure 15: RPC Broker connection diagnostic program

You should verify that the connection from the client workstation to the server is functioning correctly. For example:

- Try logging on to the server by choosing a server/port combination and pressing the "Log On" button; you will be presented with the **VISTA** Sign-on dialog. The Link State indicator will change from red to yellow to green as you progress through the connection process.
- Test various connections by changing the server and port information under the "Remote Connection Info" block. To verify the connection process is working properly, try logging on to known servers and ports with Listeners running.

You can also use this tool to resolve a server address without having to log on to the server. Type in a server name in the "Server" box located in the Remote Connection Info section of the dialog and press the enter key. If the server can be found, the IP address will be displayed in the "IP Addr" box in that same section.

If you encounter an error while testing the Broker, make sure you check the following:

- Is the Broker Listener running on the specified port? If not, start the Broker Listener on the specified port.



For more information on starting the Broker Listener, please refer to the "Broker Listeners and Ports" topic in Chapter 2 of this manual.

- Have you installed all current Kernel, Kernel Toolkit, and VA FileMan patches? If not, you must install all required patches (see the "RPC Broker V.1.1 Installation Guide.>").
- Did you change the IP address for BROKERSERVER in the HOSTS file in this session? If the IP address and server name are not resolvable, you need to correct the entry.



Your site can use the HOSTS file or DNS to resolve IP addresses and server names. If the HOSTS file is not supported in your LAN, then you will need to work with the DNS database and see if the value returned by the DNS query really identifies the machine where the listener is running.

- Is the IP address resolvable for the BROKERSERVER listed under the TCP/IP Server? If not, edit the HOSTS file in your Microsoft Windows directory and correct the IP address for the BROKERSERVER or resolve the IP address with DNS.
- Does the TCP/IP address (used in the HOSTS file) correspond to the IP address that is owned by the node used to start up the Broker Listener? If you have several nodes that can service your Test/Production account, you must make sure that the one used to start up the Listener is the one being referenced in the HOSTS file.

Verify and Test the Network Connection

To detect and avoid network problems, do the following:

1. First, make sure you actually have TCP/IP running correctly on your workstation.

At the DOS/Command prompt type PING nnn.nnn.nnn.nnn to the server host to which you are trying to connect (where nnn.nnn.nnn.nnn equals the IP address of the server). For example:

```
C:\>PING 127.0.0.1
```

Alternatively, you can PING the same server name you are trying to connect to or resolve (e.g., BROKERSERVER). For example:

```
C:\>PING BROKERSERVER
```



"PINGing" is a way to test connectivity. PINGing sends an Internet Control Message Protocol (ICMP) packet to the server in question and requests a response. It verifies that the server is running and the network is properly configured.

- If the host is unreachable, there is a network problem and you should consult with your network administrator.
 - If you get a timeout, it may be your network configuration on the client workstation, proceed to Step #2.
 - If the server is reachable, proceed to Step #4.
2. Check the properties of the WINSOCK.DLL on the client workstation and make sure it's the correct version. Install the latest Service Pack.
 3. Make sure that the files on the client are in the correct directories. In Microsoft Windows 95, the WINSOCK.DLL expects the HOSTS file to be located in the WINDOWS root directory. You

should only have one copy each of the WINSOCK.DLL and the HOSTS file on the client. (However, there may be a second copy that WIN95 keeps in the WINDOWS/SYBCKUP directory). If Windows 95 detects that some of its core files have been overwritten with older versions, supposedly it will automatically update files on reboot.

4. Make sure that all of the client workstation TCP/IP settings are correct in the network properties. Typo's, etc. can be a real problem, as can gateways, DNS servers, etc. Try removing items in your WINS configuration/DNS configuration, etc.



For more information on telecommunications support, please visit the Telecommunications Support Office Home Page at the following address:

<http://vaww.va.gov/cso/>

Signon Delays

Users signing on to **VISTA** on a client workstation with the Broker Client Agent running should *not* experience any signon delays.

In order to provide users with the capability of Auto Signon in both a GUI and roll-and-scroll Telnet session, the Kernel signon process was modified.

The Kernel signon process now tries to contact the RPC Broker V. 1.1 Client Agent on the client workstation (i.e., prior to and following the Access and Verify code prompts) to allow Auto Signon to take place. A three-second (or less) delay is built into this process while attempting to connect to the Client Agent and allow for any possible network delays.

If you wish to eliminate the 3-second (or less) signon delay in a GUI/Telnet session (i.e., *not* associated with network delays), do either of the following:

1. Disable Auto Signon for *all* users by setting the DEFAULT AUTO SIGN-ON field in the Kernel System Parameters file to "Disabled"
2. Install and run the Broker Client Agent on *all* client workstations, if Auto Signon is enabled on your system.



For more information on the DEFAULT AUTO SIGN-ON field, please refer to the "Integrated Auto Signon for Multiple User Sessions" topic in Chapter 2 in this manual.

RPC BROKER FAQs

For examples of general or development-specific frequently asked questions (FAQs) about the RPC Broker, please refer to the following web site:

<http://vista.med.va.gov/broker/faqs.html>

Glossary

ACCESS CODE	A code that, along with the Verify code, allows the computer to identify you as a user authorized to gain access to the computer. Your code is greater than 6 and less than 20 characters long; can be numeric, alphabetic, or a combination of both; and is usually assigned by a site manager or application coordinator. It is used by the Kernel's Sign-on/Security system to identify the user (see Verify Code).
ALERTS	Brief online notices that are issued to users as they complete a cycle through the menu system. Alerts are designed to provide interactive notification of pending computing activities, such as the need to reorder supplies or review a patient's clinical test results. Along with the alert message is an indication that the View Alerts common option should be chosen to take further action.
ANSI MUMPS	The MUMPS programming language is a standard recognized by the American National Standard Institute (ANSI). MUMPS stands for Massachusetts Utility Multi-programming System and is abbreviated as M.
APPLICATION PACKAGE	Software and documentation that support the automation of a service, such as Laboratory or Pharmacy within VA medical centers. The Kernel application package is like an operating system relative to other VISTA applications.
CALLABLE ENTRY POINT	An authorized programmer call that may be used in any VISTA application package. The DBA maintains the list of DBIC-approved entry points.
CARET	A symbol expressed as up caret ("^"), left caret ("<"), or right caret (">"). In many M systems, a right caret is used as a system prompt and an up caret as an exiting tool from an option. Also known as the up-arrow symbol or shift-6 key.
CLIENT	A single term used interchangeably to refer to the user, the workstation, and the portion of the program that runs on the workstation. In an object-oriented environment, a client is a member of a group that uses the services of an unrelated group. If the client is on a local area network (LAN), it can share resources with another computer (server).
COMPONENT	An object-oriented term used to describe the building blocks of GUI applications. A software object that contains data and code. A component may or may not be visible. These components interact with other components on a form to create the GUI user application interface.
COTS	Commercial Off-the-Shelf. COTS refers to software packages that can be purchased by the public and used in support of VISTA.

DATA DICTIONARY	<p>The Data Dictionary is a global containing a description of the kind of data that is stored in the global corresponding to a particular file. VA FileMan uses the data internally for interpreting and processing files.</p> <p>A Data Dictionary (DD) contains the definitions of a file's elements (fields or data attributes), relationships to other files, and structure or design. Users generally review the definitions of a file's elements or data attributes; programmers review the definitions of a file's internal structure.</p>
DBIA	<p>Database Integration Agreement, a formal understanding between two or more application packages that describes how data is shared or how packages interact. The DBA maintains a list of DBIAs between package developers, allowing the use of internal entry points or other package-specific features that are not available to the general programming public.</p>
DEFAULT	<p>A response the computer considers the most probable answer to the prompt being given. In the roll-and-scroll mode of <i>VISTA</i>, the default value is identified by double forward slash marks (//) immediately following it. In a GUI-based application the default may be a highlighted button or text. This allows you the option of accepting the default answer or entering your own answer. To accept the default you simply press the enter (or return) key. To change the default answer, type in your response.</p>
DIRECT MODE UTILITY	<p>A programmer call that is made when working in direct programmer mode. A direct mode utility is entered at the M prompt (e.g., >D ^XUP). Calls that are documented as direct mode utilities <i>cannot</i> be used in application package code.</p>
DLL	<p>Dynamic Link Library. A DLL allows executable routines to be stored separately as files with a DLL extension. These routines are only loaded when a program calls for them. DLLs provide several advantages:</p> <ol style="list-style-type: none">1. DLLs help save on computer memory, since memory is only consumed when a DLL is loaded. They also save disk space. With static libraries, your application absorbs all the library code into your application so the size of your application is greater. Other applications using the same library will also carry this code around. With the DLL, you don't carry the code itself, you have a pointer to the common library. All applications using it will then share one image.2. DLLs ease maintenance tasks. Because the DLL is a separate file, any modifications made to the DLL will not affect the operation of the calling program or any other DLL.3. DLLs help avoid redundant routines. They provide generic functions that can be utilized by a variety of programs.

ERROR TRAP	A mechanism to capture system errors and record facts about the computing context such as the local symbol table, last global reference, and routine in use. Operating systems provide tools such as the %ER utility. The Kernel provides a generic error trapping mechanism with use of the ^%ZTER global and ^XTER* routines. Errors can be trapped and, when possible, the user is returned to the menu system.												
FORUM	The central e-mail system within VISTA . Developers use FORUM to communicate at a national level about programming and other issues. FORUM is located at the Washington, DC CIO Field Office (162-2).												
GUI	Graphical User Interface . A type of display format that enables users to choose commands, initiate programs, and other options by selecting pictorial representations (icons) via a mouse or a keyboard.												
ICON	A picture or symbol that graphically represents an object or a concept.												
IRM	Information Resource Management . A service at VA medical centers responsible for computer management and system security.												
KERNEL	A set of VISTA software routines that function as an intermediary between the host operating system and the VISTA application packages (e.g., Laboratory, Pharmacy, IFCAP, etc.). Kernel provides a standard and consistent user and programmer interface between application packages and the underlying M implementation. (VA FileMan and MailMan are self-contained to the extent that they can standalone as verified packages.) Some of Kernel's components are listed below along with their associated namespace assignments: <table data-bbox="565 1171 971 1369"> <tr> <td>KIDS</td> <td>XPD</td> </tr> <tr> <td>Menu Management</td> <td>XQ</td> </tr> <tr> <td>Tools</td> <td>XT</td> </tr> <tr> <td>Sign-on/Security</td> <td>XU</td> </tr> <tr> <td>Device Handling</td> <td>ZIS</td> </tr> <tr> <td>Task Management</td> <td>ZTM</td> </tr> </table>	KIDS	XPD	Menu Management	XQ	Tools	XT	Sign-on/Security	XU	Device Handling	ZIS	Task Management	ZTM
KIDS	XPD												
Menu Management	XQ												
Tools	XT												
Sign-on/Security	XU												
Device Handling	ZIS												
Task Management	ZTM												
MENU MANAGER	The Kernel module that controls the presentation of user activities such as menu choices or options. Information about each user's menu choices is stored in the Compiled Menu System, the ^XUTL global, for easy and efficient access.												
MULTIPLE	A multiple-valued field; a subfile. In many respects, a multiple is structured like a file.												
MUMPS (ANSI STANDARD)	A programming language recognized by the American National Standards Institute (ANSI). The acronym MUMPS stands for Massachusetts General Hospital Utility Multi-programming System and is abbreviated as M.												

NAMESPACING	A convention for naming VISTA package elements. The Database Administrator (DBA) assigns unique character strings for package developers to use in naming routines, options, and other package elements so that packages may coexist. The DBA also assigns a separate range of file numbers to each package.
NODE	In a tree structure, a point at which subordinate items of data originate. An M array element is characterized by a name and a unique subscript. Thus the terms: node, array element, and subscripted variable are synonymous. In a global array, each node might have specific fields or "pieces" reserved for data attributes such as name.
OPTION	As an item on a menu, an option provides an opportunity for users to select it, thereby invoking the associated computing activity. In VISTA , an entry in the OPTION file (#19). Options may also be scheduled to run in the background, non-interactively, by TaskMan.
PROMPT	The computer interacts with the user by issuing questions called <i>prompts</i> , to which the user returns a response.
REMOTE PROCEDURE CALL	A remote procedure call (RPC) is essentially M code that may take optional parameters to do some work and then return either a single value or an array back to the client application.
ROUTINE	A program or a sequence of instructions called by a program that may have some general or frequent use. M routines are groups of program lines that are saved, loaded, and called as a single unit via a specific name.
SECURITY KEY	The purpose of Security Keys is to set a layer of protection on the range of computing capabilities available with a particular software package. The availability of options is based on the level of system access granted to each user.
SERVER	The computer where the data and the Business Rules reside. It makes resources available to client workstations on the network. In VISTA , it is an entry in the OPTION file (#19). An automated mail protocol that is activated by sending a message to a server at another location with the "S.server" syntax. A server's activity is specified in the OPTION file (#19) and can be the running of a routine or the placement of data into a file.
SIGN-ON/SECURITY	The Kernel module that regulates access to the menu system. It performs a number of checks to determine whether access can be permitted at a particular time. A log of signons is maintained.
SUBSCRIPT	A symbol that is associated with the name of a set to identify a particular subset or element. In M, a numeric or string value that: is enclosed in parentheses, is appended to the name of a local or global variable, and identifies a specific node within an array.

UCI	User Class Identification , a computing area. The MGR UCI is typically the Manager's account, while VAH or ROU may be Production accounts.
USER ACCESS	<p>This term is used to refer to a limited level of access to a computer system that is sufficient for using/operating a package, but does not allow programming, modification to data dictionaries, or other operations that require programmer access. Any of <i>VISTA</i>'s options can be locked with a security key (e.g., XUPROGMODE, which means that invoking that option requires programmer access).</p> <p>The user's access level determines the degree of computer use and the types of computer programs available. The Systems Manager assigns the user an access level.</p>
USER INTERFACE	The way the package is presented to the user, such as Graphical User Interfaces that display option prompts, help messages, and menu choices. A standard user interface can be achieved by using Borland's Delphi Graphical User Interface to display the various menu option choices, commands, etc.
VERIFY CODE	The Kernel's Sign-on/Security system uses the Verify code to validate the user's identity. This is an additional security precaution used in conjunction with the Access code. Verify codes shall be at least eight characters in length and contain three of the following four kinds of characters: letters (lower- and uppercase), numbers, and, characters that are neither letters nor numbers (e.g., "#", "@" or "\$"). If entered incorrectly, the system does not allow the user to access the computer. To protect the user, both codes are invisible on the terminal screen.
VISTA	Veterans Health Information Systems and Technology Architecture. <i>VISTA</i> includes the VA's application software (i.e., Microsoft Windows-based and locally-developed applications, roll-and-scroll, and interfaces such as software links to commercial packages). In addition, it encompasses the VA's uses of new automated technology including the clinical workstations. <i>VISTA</i> encompasses the rich automated environment already present at local VA medical facilities.
WINDOW	An object on the screen (dialog) that presents information such as a document or message.

Appendix A—Patch Revision History

The following table displays the patch/version release history for the RPC Broker software. The sequence number (Seq #) is the order in which the patch was released by National VISTA Support (NVS) and installed by the site. The sequence number does not necessarily match the Patch ID number in all cases. Also, the sequence number, in some cases, can imply dependency between patches. Each table entry indicates that the documentation was reviewed and updated as needed for each patch; in some cases, a patch may not affect the content of the documentation. Regardless, the patch will still be added to the patch history in reverse patch sequence order.

Seq #	Patch ID	Brief Summary	Status
24	XWB*1.1*29	This patch provides an installation executable for advanced RPCBroker features that must be installed, or at least registered, on the client workstations.	Client-side only patch—05/19/02. This document has been reviewed and updated as needed for this patch.
23	XWB*1.1*26	This patch updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK). It supports Delphi V. 4, 5, and 6. It provides a SharedRPCBroker component. Any GUI application that uses the SharedRPCBroker will now have the ability to share a Broker connection. This patch also supports ESSO.	Client-side only patch—05/19/02. This document has been reviewed and updated as needed for this patch.
22	XWB*1.1*13	This patch updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK). It supports Delphi V. 4, 5, and 6. It provides Silent Login functionality in the Broker. Any GUI RPC Broker-based application will now have the ability to login to an M Server silently (i.e. without any user dialog). This patch also supports Enterprise Single-Sign-On (ESSO).	Client and server patch—05/19/02. This document has been reviewed and updated as needed for this patch.
21	XWB*1.1*25	This patch adds a new protected field named SUPPRESS RDV USER SETUP (#.1) to the REMOTE PROCEDURE file (#8994). It regulates the addition of Remote Users to sites' local NEW PERSON files for the RDV-based RPCs.	Server-side only patch—Patch released on 05/09/02.
20	XWB*1.1*27	This patch enables asynchronous processing, multiple jobs running at the same time. Prior to this patch, processing of requests to the HL7 package for remote data made by GCPR and CPRS, was performed synchronously - in order of time of request, each job finishing before the next job started.	Server-side only patch—Patch released on 03/15/02.

Seq #	Patch ID	Brief Summary	Status
19	XWB*1.1*16	This patch provides several bug fixes (e.g., READ/WRITE errors) initiated via NOIS.	Server-side only patch—Patch released on 02/06/02.
18	XWB*1.1*24	<p>This patch updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK). It supports only Delphi V. 4 and Delphi V. 5.</p> <p>Due to version-dependent code, a problem was recently encountered that is associated with reading the Microsoft Windows Registry in programs compiled with Delphi V. 5. Because a conditional test was specifically looking for Delphi V. 4-based applications, Delphi V. 5-based applications ended up using Broker code for Delphi V. 3. This can result in users having limited privileges, preventing their ability to read data from the registry. This has been observed when a user with limited NT privileges attempts to select a location for the RPC Broker connection, and it results in the use of the default BrokerServer/9200. However, users with higher levels of NT access do not see this problem. This version-dependent code was removed via this patch.</p>	<p>Client and server patch—Patch released on 11/09/01.</p> <p>This document was reviewed and updated as needed for this patch.</p>
17	XWB*1.1*22	<p>The calling site had a NEW PERSON file entry with a phone number containing a trailing backslash ("\"). As part of Remote Data Views (RDV), this data was then encoded and sent to the remote site.</p> <p>At the remote site, a bug caused the backslash ("\") to be appended to the end of several other strings, which then caused the reported error. This was fixed by correcting the decoding routine.</p> <p>Because the error occurred before RDV was setup to handle an error, it caused the calling site to keep sending the same message repeatedly. This has been fixed by setting an error trap at the beginning of RDV.</p> <p>If the application does not set some data into the return variable, XWB2HL7 will return a string starting with "-1^".</p> <p>The XWB EXAMPLE option, RPC's and routine (XWBEXMPL) are included to add an entry point for testing that will record the symbol table in the error trap.</p>	<p>Server-side only patch—Patch released on 10/03/01.</p> <p>This document was reviewed and updated as needed for this patch.</p>

Seq #	Patch ID	Brief Summary	Status
16	XWB*1.1*20	<p>This patch addresses the following:</p> <ul style="list-style-type: none"> • During the early testing of RDV (Remote Data View), the DUZ value was hard set to .5 just before the call to the RPC. This was done because the code to set up the user at the remote site wasn't ready. When the code was fixed to properly set the DUZ, the old code was never removed. This has been fixed in the routine XWB2HL7. • If data was left in the ^XUTL("XQ",\$J,"IO")node it could cause problems when HOME^%ZIS is called by some RPC's, so this ^XUTL node is killed off before the RPC is called. • In an e-mail message from CPRS developers: The global that may be used to pass data back to the RPC was not killed before its use. This was fixed in the routine XWBDRPC. 	<p>Server-side only patch—Patch released on 05/10/01.</p> <p>This document was reviewed and updated as needed for this patch.</p>
15	XWB*1.1*14	<p>This patch updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK). It adds no new functionality . It does the following:</p> <ul style="list-style-type: none"> • Releases the source code for the BDK. • Splits the VistaBroker package into separate design- and run-time packages. 	<p>Client and server patch—Patch released on 10/17/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>
14	XWB*1.1*18	<p>This patch fixed the following NOIS: LOM-0800-62301 and PRO-0800-11778:</p> <p>If there are problems associated with the remote site's HL7 definitions—specifically the receiving application. Then the RPC XWB REMOTE STATUS CHECK will get an UNDEF error on the variable Z.</p>	<p>Server-side only patch—Patch released on 10/17/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>
13	XWB*1.1*12	<p>This patch is in support of the CPRS Remote Data Views project. The RPC Broker is used to facilitate invocation of Remote Procedure calls on a remote server. The RPC Broker uses VISTA HL7 as the vehicle to pass RPC name and parameters from a local server to a remote server. On the return path, VISTA HL7 is also used to send results from the remote server back to the local server.</p>	<p>Server-side only patch—Patch released on 08/04/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>

Seq #	Patch ID	Brief Summary	Status
12	XWB*1.1*10	<p>This patch gives greater information about and control of RPCs. Specific new abilities are:</p> <ul style="list-style-type: none"> • Blocking an RPC either locally*, remotely*, or in both contexts by setting a value in the INACTIVE field of the Remote Procedure file. Prior to this patch, values in this field had no effect. • Assuring that an RPC is at least a specified version when it is run remotely* by setting a value in the new VERSION field of the REMOTE PROCEDURE file. • Querying a server regarding the status of RPCs by using new Remote Procedures: XWB IS RPC AVAILABLE and XWB ARE RPCS AVAILABLE. • In addition, this patch stops M errors from occurring when a client application attempts to: <ol style="list-style-type: none"> 1.) Create a context that does not exist on the server, or 2.) Run a remote procedure that does not exist on the server. 	<p>Server-side only patch—Patch released on 08/04/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>
11	XWB*1.1*15	<p>This patch should correct a problem on Cache sites with the Broker looping with COMMAND errors. This error is caused when the Broker tries to open the TCP port and the port is already open via the Broker.</p>	<p>Server-side only patch—Patch released on 04/12/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>
10	XWB*1.1*11	<p>This patch updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK)—adding support for Delphi V. 5 development.</p>	<p>Client and server patch—Patch released on 01/24/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>

Seq #	Patch ID	Brief Summary	Status
9	XWB*1.1*9	<p>This patch fixes the following:</p> <ul style="list-style-type: none"> • Intersystems License. This is the patch that works with Patch XU*8*118. The code to share licenses when GUI and Telnet users from the same workstation are connected is in place and ZU now calls it. This patch adds a similar call from XWBTCP. • This patch brings a new XWB LISTENER STOP ALL option for shutting down multiple listeners. It also brings a modified option XWB LISTENER STARTER for starting Broker listeners. 	<p>Server-side only patch—Patch released on 01/24/00.</p> <p>This document was reviewed and updated as needed for this patch.</p>
8	XWB*1.1*8	<p>This patch supports GUI Multi-Divisional signon. If a user has more than one division to choose from, the user must select one before continuing with the signon. If the user has only one division in File #200, this division will be used; otherwise, the default institution in the KERNEL SYSTEM PARAMETERS file will be used.</p>	<p>Client-side only patch—Patch released on 12/10/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>
7	XWB*1.1*6	<p>This patch does the following:</p> <ul style="list-style-type: none"> • Eliminates server Broker jobs for which there is no client application. • Changes the time that the server waits for the client to contact it. A new field in the KERNEL SYSTEM PARAMETERS file, BROKER ACTIVITY TIMEOUT (default value of approximately 3 minutes) controls the length of the timeout. 	<p>Client and server patch—Patch released on 09/09/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>
6	XWB*1.1*4	<p>This patch does the following:</p> <ol style="list-style-type: none"> 1. Introduces a shorter timeout when logging in via any GUI RPC Broker-based application. The server listener process will timeout after 90 seconds if the user has not passed in his/her Access and Verify codes. 2. Updates the Broker's Programmer Client Workstation software—also known as the Broker Development Kit (BDK)—adding support for Delphi V. 4 development. 3. Fixes a bug in which the Title bar of the Kernel Login form was being changed when a user started entering their Access code. 	<p>Client and server patch—Patch released on 06/24/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>

Seq #	Patch ID	Brief Summary	Status
5	XWB*1.1*7	<p>This patch addresses two problems:</p> <ol style="list-style-type: none"> 1. A command error is occurring at RESTART+17^XWBTCPL when the Broker tries to reopen a device that is not closed. This seems to be a problem with Cache sites only. The result of this error causes the Broker Listener to stop. The fix is in XWBTCPL. 2. The listener doesn't check for available slots before starting a new process. The listener will now check the MAX SIGNON ALLOWED field of the VOLUME SET multiple in the KERNEL SYSTEM PARAMETERS file, the same one used by Kernel logon. This fix is also in XWBTCPL. 	<p>Server-side only patch—Patch released on 06/04/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>
4	XWB*1.1*5	<p>This patch is for the support of RUM. This will allow the trapping of data for Remote Procedure Calls (RPCs) and the RPC Broker handler.</p>	<p>Server-side only patch—Patch released on 03/31/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>
3	XWB*1.1*3	<p>Under CPRS, when the DG routines call OP^XQCHK to record what option is used, it was getting back "unknown." The Broker created context needed to set the variable XQY.</p>	<p>Server-side only patch—Patch released on 01/06/99.</p> <p>This document was reviewed and updated as needed for this patch.</p>
2	XWB*1.1*2	<p>This patch addresses three problems with RPC Broker v1.1:</p> <ul style="list-style-type: none"> • Encrypted Literal—Pattern match failure in RPCs. The failure only occurs with RPCs that combine multiple literals and an array (NOIS WAS-0398-22800). • Data Collection Switch turned "Off"—Collection of data will be controlled by the use of the Capacity Management tools (NOIS BRX-0498-11768 and HUN-0498-21137). • 10 Second Network Timeout in Client Agent—A 30 second timeout is being switched to 10 for network communications with the Client Agent. 	<p>Server-side only patch—Patch released on 07/27/98.</p> <p>This document was reviewed and updated as needed for this patch.</p>

Seq #	Patch ID	Brief Summary	Status
1	XWB*1.1*1	<p>This patch fixes some small problems that were discovered after release (server-side only).</p> <ul style="list-style-type: none"> • XWBTCP—Remove the SYMBOL_TABLE from the VAX DSM JOB command. • XWBTCP—When stopping the Broker, see a failure to open a socket. • XWB BROKER EXAMPLE option—This option was missing its type field. 	<p>Server-side only patch—Patch released on 02/18/98.</p> <p>This document was reviewed and updated as needed for this patch.</p>
NA	Version 1.1	Original Version 1.1 software release.	September 1997

Table 9: RPC Broker V. 1.1 patch revision history (in reverse sequence order)

Index

A

Appendix A—Patch Revision History, 1
Assumptions About the Reader, xi
Auto Signon, 2-15
AUTO SIGN-ON, 2-15

B

BIND Services, 2-19
Broker
 Listeners and Ports, 2-11
 Message Structure, 2-17

C

Changing the VISTA Verify Code, 3-7
CLAGENT.EXE, 2-1, 2-15
Client Agent, 2-1, 2-15
Client Features, 2-1
Client Manager, What Happened to it?, 2-8
Client/Server Timeouts, 2-18
Commonly Used Terms, x

.

'Connect To' Dialog, 2-2

C

Connection Request
 Validating, 3-1
Connections
 Diagnostics, 4-2
Contents, v
Customizing the Sign-on Dialog, 3-4

D

Data Dictionary
 Data Dictionary Utilities Menu, xi
 Listings, xi
DEFAULT AUTO SIGN-ON, 2-15
DEFAULT MULTIPLE SIGN-ON, 2-15
Diagnostics
 Connection, 4-2
DNS, 2-19, 4-3
Documentation History, iii
Documentation Symbols, ix
Domain Name Service (DNS), 2-4, 2-6

E

Edit Broker Servers Program, 2-3

F

FAQs, 4-4
Figures, vii

H

Help
 At Prompts, x
 Online, x
Help files, 2-5
Home Pages
 Adobe Acrobat Quick Guide Web Address, xii
 Adobe Systems Incorporated Web Address,
 xii
 RPC Broker FAQs Home Page Web Address,
 4-4
 RPC Broker Web Address, xii
 SD&D Home Page Web Address, xi
HOSTS file, 2-4, 2-6, 2-19, 4-3, 4-4
How to
 Generate Technical Information Online, x
How To
 Use this Manual, ix

I

Integrated Auto Signon, 2-15

K

KERNEL SYSTEM PARAMETERS file, 2-15

L

List File Attributes Option, xi
LISTENERS
 STARTING, 2-12
Listeners and Ports, 2-11
Load Balancing, 2-19

M

Menu Manager, 3-3
Menus
 Data Dictionary Utilities, xi

Message Structure, 2-17
Microsoft Windows Registry, 2-2, 2-3, 2-4, 2-9,
2-10, 2-19
MSERVER Replaces RPC Broker Listener
Process, 2-20
MULTIPLE SIGN-ON, 2-15

N

Network Connection, 4-3
NEW PERSON file, 2-15

O

Obtaining the Server TCP/IP Address, 2-11
Online
Documentation, x
Help Frames, xi
Technical Information, How to Generate, x
OPTION file, 3-3
OPTION SCHEDULING file, 2-13
Options
List File Attributes, xi
Orientation, ix

P

Patch History, 1
PING, 4-3

Q

Question Mark Help, x

R

Reader, Assumptions About the, xi
Reference Materials, xii
Registry, 2-2, 2-3, 2-4, 2-9, 2-10, 2-19
REMOTE PROCEDURE file (#8994), 2-17
Revision History
Documentation, iii
Patches, 1
RPC Broker Diagnostic Program, How to test
the Broker, 4-1
RPC BROKER SITE PARAMETERS file, 2-13,
2-14, 2-21
RPCBI.DLL, 2-8
RPCs
Validating, 3-8
RPCTEST.EXE, 4-1, 4-2

S

Schedule/Unschedule Options, 2-13
Security, 3-1
Change VISTA Verify Code Component, 3-7
Features, 3-1
Sample Security Procedures, 3-9
Signon Dialog, Customizing, 3-4
Summary of Tasks, 3-9
Validating Connection Request, 3-1
Validating RPCs, 3-8
Validating Users, 3-1
Server Features, 2-11
ServerList.EXE, 2-3
Servers, using the Edit Broker Servers Program,
2-3
Sign-on Dialog, Customizing, 3-4
Sign-on Dialog, sample, 3-2
STARTING LISTENERS, 2-12
STOP^XWBTCP(Listener port), 2-12
STRT^XWBTCP(Listener port, 2-12
Symbols Found in the Documentation, ix
System
Features, 2-1
Overview, 1-4

T

Test the Broker Using the
RPC Broker Diagnostic Program, 4-1
Timeouts, 2-18
Troubleshooting, 4-1

U

UCX Load Balancing, 2-19
URLs
Adobe Acrobat Quick Guide Web Address, xii
Adobe Systems Incorporated Web Address,
xii
RPC Broker FAQs Home Page Web Address,
4-4
RPC Broker Web Address, xii
SD&D Home Page Web Address, xi
Use this Manual, How to, ix

V

Validating
Connection Request, Security, 3-1
RPCs, Security, 3-8
Users, Security, 3-1
Verify and Test the Network Connection, 4-3

Verify Code, Changing, 3-7
VISTA.INI file, What Happened to it?, 2-9

W

Web Pages

Adobe Acrobat Quick Guide Web Address, xii
Adobe Systems Incorporated Web Address,
xii
RPC Broker FAQs Home Page Web Address,
4-4

RPC Broker Web Address, xii
SD&D Home Page Web Address, xi
What Happened to the VISTA.INI File?, 2-9
Windows Registry, 2-2, 2-3, 2-4, 2-9, 2-10, 2-19
WINSOCK.DLL, 2-17, 4-3

X

XWB LISTENER STARTER option, 2-13

