

RPC Broker 1.1
Systems Management Guide



June 2015

Department of Veterans Affairs (VA)
Office of Information and Technology (OI&T)
Enterprise Program Management Office (EPMO)

Revision History

Documentation Revisions

Date	Revision	Description	Authors
04/27/2016	6.0	<p>Tech Edits based on release of RPC Broker Patch XWB*1.1*60 (released 06/11/2015):</p> <ul style="list-style-type: none"> • Reformatted document to follow current documentation standards and style formatting requirements. • Updated the “Orientation” section. • Updated Section 1.1. • Updated Section 2.1.1. • Updated Figure 3. • Updated Section 2.1.2. • Updated Figure 4. • Updated Section 2.1.3. • Added Figure 5 and Figure 6. • Updated Section 2.1.4. • Updated Table 4. • Updated Section 2.2.1.4. • Updated Section 2.2.1.5. • Updated Section 2.2.1.6. • Added Section 2.2.2.2 for a Linux example. • Updated Figure 13. • Updated Section 4.1. • Deleted references to TSharedRPCBroker and TSharedBroker components throughout, since they were removed from the software. • Updated help file references from “BROKER.HLP” to “Broker_1_1.chm” throughout. • Updated references to show RPC Broker Patch XWB*1.1*60 supports Delphi XE7, XE6, XE5, and XE4 throughout. <p>RPC Broker 1.1</p>	<ul style="list-style-type: none"> • Developer H. W. • Technical Writer: T. B.
12/04/2013	5.1	<p>Tech Edit:</p> <ul style="list-style-type: none"> • Updated document for RPC Broker Patch XWB*1.1*50 based on feedback from H Westra. • Removed references related to 	<ul style="list-style-type: none"> • Developer: H. W. • Technical Writer: T. B.

Date	Revision	Description	Authors
		<p>Virgin Installations throughout.</p> <ul style="list-style-type: none"> • Updated file name references throughout. • Removed distribution files that are obsolete or no longer distributed throughout. • Updated RPC Broker support on the following software: <ul style="list-style-type: none"> ○ Microsoft® XP and 7.0 (operating system) throughout. ○ Microsoft® Office Products 2010 throughout. ○ Changed references from “Borland” to “Embarcadero” and updated support for Delphi Versions XE5, XE4, XE3, and XE2 throughout. • Updated Section 1.1. • Updated Section 1.2. • Updated Figure 3 and note underneath the figure regarding admin privileges. • Updated Section 2.1.2 and Figure 4. • Updated Figure 8. • Updated Section 2.1.3. • Updated Section 2.1.4 and Table 3. • Updated Section 2.1.5 and Table 4. • Updated Table 5. • Updated Figure 7 copyright reference. • Updated all images for prior Microsoft® Windows operating systems to Windows 7 dialogues. • Redacted document for the following information: <ul style="list-style-type: none"> ○ Names (replaced with role and initials). ○ Production IP addresses and ports. ○ Intranet websites. <p>RPC Broker 1.1</p>	
07/25/2013	5.0	<p>Tech Edit:</p> <ul style="list-style-type: none"> • Baselined document. • Updated all styles and formatting to follow current internal team style template. 	<ul style="list-style-type: none"> • Developer: H. W. • Technical Writer: T. B.

Date	Revision	Description	Authors
		<ul style="list-style-type: none"> Updated all organizational references. <p>RPC Broker 1.1</p>	
08/26/2008	4.2	<p>Updates for RPC Broker Patch XWB*1.1*50:</p> <ul style="list-style-type: none"> Added new properties. Support for Delphi 5, 6, 7, 2005, 2006, and 2007. Changed references from Patch 47 to Patch 50 where appropriate. <p>RPC Broker 1.1</p>	<ul style="list-style-type: none"> Project Manager: J. Sch. Developer: J. I. SQA: G. S. Technical Writer: T. B.
07/03/2008	4.1	<p>Updates for RPC Broker Patch XWB*1.1*47:</p> <ul style="list-style-type: none"> No content changes required; no new public classes, methods, or properties added to those available in XWB*1.1*40. Bug fixes to the ValidAppHandle function and fixed memory leaks. Support added for Delphi 2005, 2006, and 2007. Reformatted document. Changed references from Patch 40 to Patch 47 where appropriate. <p>RPC Broker 1.1</p>	<ul style="list-style-type: none"> Project Manager: J. Sch. Developer: J. I. SQA: G. S. Technical Writer: T. B.
02/23/2005	4.0	<p>Revised Version for RPC Broker Patches XWB*1.1*35 and 40.</p> <p>Also, reviewed document and edited for the “Data Scrubbing” and the “PDF 508 Compliance” projects.</p> <p>Data Scrubbing—Changed all patient/user TEST data to conform to HSD&D standards and conventions as indicated below:</p> <ul style="list-style-type: none"> The first three digits (prefix) of any Social Security Numbers (SSN) start with “000” or “666.” Patient or user names are formatted as follows: XWBPATIENT,[N] or XWBUSER,[N] respectively, where the N is a number written out and incremented with each new entry (e.g., XWBPATIENT, ONE, XWBPATIENT, TWO, etc.). Other personal demographic-related data (e.g., addresses, phones, IP addresses, etc.) were also changed to be generic. <p>PDF 508 Compliance—The final PDF document was recreated and now supports</p>	<ul style="list-style-type: none"> Developer: J. I. Technical Writer: T. B.

Date	Revision	Description	Authors
		the minimum requirements to be 508 compliant (i.e., accessibility tags, language selection, alternate text for all images/icons, fully functional Web links, successfully passed Adobe Acrobat Quick Check). RPC Broker 1.1	
05/08/2002	3.0	Revised Version for RPC Broker Patch XWB*1.1*26. RPC Broker 1.1	<ul style="list-style-type: none"> • Developer: J. I. • Technical Writer: T. B.
04/08/2002	2.0	Revised Version for RPC Broker Patch XWB*1.1*13. RPC Broker 1.1	<ul style="list-style-type: none"> • Developer: J. I. • Technical Writer: T. B.
09/--/1997	1.0	Initial RPC Broker Version 1.1 software release. RPC Broker 1.1	<ul style="list-style-type: none"> • Developer: J. I. • Technical Writer: T. B.

Patch Revisions

For the current patch history related to this software, see the Patch Module on FORUM.

Contents

Revision History	ii
List of Figures	viii
List of Tables	ix
Orientation	x
1 Introduction.....	1
1.1 Overview	1
1.2 How Does It All Work?	3
1.3 System Overview	5
2 System Features.....	6
2.1 Client Features	6
2.1.1 RPC Broker Client Agent	6
2.1.2 “Connect To” Dialogue	7
2.1.3 Edit Broker Servers Application.....	7
2.1.4 Standalone Applications and their Associated Help Files	9
2.1.5 HOSTS File	10
2.1.5.1 Adding Entries.....	12
2.1.5.2 Modifying Entries.....	12
2.1.6 What Happened to the Client Manager?.....	13
2.1.7 What Happened to the VISTA.INI File?	13
2.2 Server Features.....	14
2.2.1 RPC Broker Management Menu.....	14
2.2.1.1 RPC Listener Edit Option.....	14
2.2.1.2 Start All RPC Broker Listeners Option	14
2.2.1.3 Stop All RPC Broker Listeners Option	15
2.2.1.4 Clear XWB Log Files Option.....	15
2.2.1.5 Debug Parameter Edit Option.....	15
2.2.1.6 View XWB Log Option.....	15
2.2.2 Broker Listeners and Ports.....	15
2.2.2.1 Obtaining an Available Listener Port—Alpha/VMS Systems	16
2.2.2.2 Obtaining an Available Listener Port—Linux Systems	16
2.2.3 Starting and Stopping Listeners	16

2.2.3.1	To Start All Listeners	16
2.2.3.2	To Configure Listeners for Automatic Startup.....	17
2.2.3.3	To Stop All Running Listeners.....	17
2.2.3.4	To Start Up a Single Listener Directly	17
2.2.3.5	To Stop a Single Listener Directly	17
2.2.3.6	To Task the XWB LISTENER STARTER Option for System Startup	17
2.2.4	RPC BROKER SITE PARAMETERS File.....	18
2.2.4.1	Editing the Listener Site Parameters	18
2.2.5	Integrated Auto Signon for Multiple User Sessions	19
2.2.5.1	Enabling/Disabling Auto Signon.....	19
2.2.5.2	Examples	20
2.2.6	RPC Broker Message Structure	21
2.2.7	Client/Server Timeouts	22
3	Security	23
3.1	Security Features.....	23
3.2	Validation of Connection Request	23
3.3	Validation of Users	23
3.3.1	VistA Signon Dialogue.....	24
3.3.2	VistA Division Selection Dialogue.....	25
3.3.3	Users Can Customize VistA Signon Dialogue	26
3.3.3.1	Sign-on Properties	27
3.3.4	Change VistA Verify Code Component	29
3.4	Validation of RPCs	30
3.5	Sample Security Procedures.....	31
3.6	Security Features Tasks Summary	31
4	Troubleshooting.....	32
4.1	Test the Broker Using the RPC Broker Diagnostic Program.....	32
4.2	Verify and Test the Network Connection	34
4.3	Signon Delays	35
4.4	RPC Broker FAQs	35
	Glossary	36
	Index	38

List of Figures

Figure 1: Delphi's Tool Properties dialogue—Broker_1_1.chm entry.....	xv
Figure 2: VistA RPC Broker system overview diagram.....	5
Figure 3: RPC Broker Client Agent dialogue.....	6
Figure 4: Server and port configuration selection dialogue.....	7
Figure 5: Edit Broker Servers Application—Opened normally.....	8
Figure 6: Edit Broker Servers Application—Opened with Administrator privileges.....	9
Figure 7: Sample HOSTS file.....	11
Figure 8: RPC Broker Management Menu option [XWB MENU].....	14
Figure 9: Obtaining an Available Listener Port—Alpha/VMS Systems.....	16
Figure 10: Obtaining an Available Listener Port—Linux Systems.....	16
Figure 11: Automatically starting the Listener(s) when TaskMan is restarted.....	18
Figure 12: RPC Listener Edit option—Sample user dialogue.....	18
Figure 13: Sample VistA Signon security dialogue.....	24
Figure 14: Select Division dialogue—Sample entries.....	25
Figure 15: Sign-on Properties on the System Menu.....	26
Figure 16: Sign-on Properties dialogue.....	27
Figure 17: Sample Font dialogue.....	29
Figure 18: Change VistA Verify Code dialogue.....	29
Figure 19: RPC Broker connection diagnostic program.....	33

List of Tables

Table 1: Documentation symbol descriptions.....	xi
Table 2: Commonly used RPC Broker terms.....	xiii
Table 3: Standalone RPC Broker applications and their associated help files.....	9
Table 4: HOSTS file location in Windows	10
Table 5: VISTA.INI entries and Microsoft® Windows Registry disposition table	13
Table 6: Listener site parameter entries description table.....	19
Table 7: Multiple and Auto Signon Settings table	20
Table 8: Window position.....	27
Table 9: Window size	28
Table 10: Introductory text background color	28
Table 11: Sample security procedures	31
Table 12: Security tasks summary	31

Orientation

How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of the Remote Procedure Call (RPC) Broker 1.1 Development Kit (BDK) and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VistA).

Intended Audience

The intended audience of this manual is the following stakeholders:

- Enterprise Program Management Office (EPMO)—VistA legacy development teams.
- Information Resource Management (IRM)—System administrators at Department of Veterans Affairs (VA) sites who are responsible for computer management and system security on the VistA M Servers.
- Information Security Officers (ISOs)—Personnel at VA sites responsible for system security.
- Product Support (PS).

Disclaimers

Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is *not* subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed and/or modified freely provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.



CAUTION: To protect the security of VistA systems, distribution of this software for use on any other computer system by VistA sites is prohibited. All requests for copies of this software for *non-VistA* use should be referred to the VistA site's local Office of Information and Technology Field Office (OI&TFO).

Documentation Disclaimer

This manual provides an overall explanation of RPC Broker and the functionality contained in RPC Broker 1.1; however, 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 Internet and Intranet Websites for a general orientation to VistA. For example, visit the Office of Information and Technology (OI&T) VistA Development Intranet website.



DISCLAIMER: The appearance of any external hyperlink references in this manual does *not* constitute endorsement by the Department of Veterans Affairs (VA) of this Website or the information, products, or services contained therein. The VA does *not* exercise



any editorial control over the information you find at these locations. Such links are provided and are consistent with the stated purpose of this VA Intranet Service.

Documentation Conventions

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. [Table 1](#) gives a description of each of these symbols:

Table 1: Documentation symbol descriptions

Symbol	Description
	NOTE/REF: Used to inform the reader of general information including references to additional reading material.
	CAUTION / RECOMMENDATION / DISCLAIMER: Used to caution the reader to take special notice of critical information.

- Descriptive text is presented in a proportional font (as represented by this font).
- Conventions for displaying TEST data in this document are as follows:
 - The first three digits (prefix) of any Social Security Numbers (SSN) begin with either “000” or “666.”
 - Patient and user names are formatted as follows:
 - [Application Name]PATIENT,[N]
 - [Application Name]USER,[N]

Where “Application Name” is defined in the Approved Application Abbreviations document and “N” represents the first name as a number spelled out and incremented with each new entry.

For example, in RPC Broker (XWB) test patient names would be documented as follows:

XWBPATIENT,ONE; XWBPATIENT,TWO; XWBPATIENT,14, etc.

For example, in RPC Broker (XWB) test user names would be documented as follows:

XWBUSER,ONE; XWBUSER,TWO; XWBUSER,14, etc.

- “Snapshots” of computer online displays (i.e., screen captures/dialogues) and computer source code are shown in a *non*-proportional font and may be enclosed within a box.
- User’s responses to online prompts are in **boldface** and highlighted in yellow (e.g., **<Enter>**).
- Emphasis within a dialogue box is in **boldface** and highlighted in blue (e.g., **STANDARD LISTENER: RUNNING**).
- Some software code reserved/key words are in **boldface** with alternate color font.
- References to “<Enter>” within these snapshots indicate that the user should press the <Enter> key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing <PF1>.

- Author's comments are displayed in italics or as "callout" boxes.



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

- This manual refers to the M programming language. Under the 1995 American National Standards Institute (ANSI) standard, M is the primary name of the MUMPS programming language, and MUMPS will be considered an alternate name. This manual uses the name M.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field/file names, and security keys (e.g., the XUPROGMODE security key).



NOTE: Other software code (e.g., Delphi/Pascal and Java) variable names and file/folder names can be written in lower or mixed case.

Documentation Navigation

This document uses Microsoft® Word's built-in navigation for internal hyperlinks. To add **Back** and **Forward** navigation buttons to your toolbar, do the following:

1. Right-click anywhere on the customizable Toolbar in Word 2010 (*not* the Ribbon section).
2. Select **Customize Quick Access Toolbar** from the secondary menu.
3. Press the drop-down arrow in the "Choose commands from:" box.
4. Select **All Commands** from the displayed list.
5. Scroll through the command list in the left column until you see the **Back** command (circle with arrow pointing left).
6. Click/Highlight the **Back** command and press **Add** to add it to your customized toolbar.
7. Scroll through the command list in the left column until you see the **Forward** command (circle with arrow pointing right).
8. Click/Highlight the **Forward** command and press **Add** to add it to your customized toolbar.
9. Press **OK**.

You can now use these **Back** and **Forward** command buttons in your Toolbar to navigate back and forth in your Word document when clicking on hyperlinks within the document.




NOTE: This is a one-time setup and is automatically available in any other Word document once you install it on the Toolbar.

Commonly Used Terms

[Table 2](#) lists terms and their descriptions that can be helpful while reading the RPC Broker documentation:

Table 2: Commonly used RPC Broker terms

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.  REF: For a more detailed description, see the <i>Embarcadero Delphi for Windows User Guide</i> .
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.



REF: For additional terms and definitions, see the “[Glossary](#).”

How to Obtain Technical Information Online

Exported VistA M Server-based software file, routine, and global documentation can be generated using Kernel, MailMan, and VA FileMan utilities.



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

REF: See the *RPC Broker Technical Manual* for further information.

Help at Prompts

VistA M Server-based software provides online help and commonly used system default prompts. Users are 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 M Server-based software.

Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the List File Attributes option on the Data Dictionary Utilities submenu in VA FileMan to print formatted data dictionaries.



REF: For details about obtaining data dictionaries and about the formats available, see the “List File Attributes” chapter in the “File Management” section of the *VA FileMan Advanced User Manual*.

Assumptions

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

- VistA computing environment:
 - Kernel—VistA M Server software
 - Remote Procedure Call (RPC) Broker—VistA Client/Server software
 - VA FileMan data structures and terminology—VistA M Server software
- Microsoft® Windows environment
- M programming language
- Object Pascal programming language
- Object Pascal programming language/Embarcadero Delphi Integrated Development Environment (IDE)—RPC Broker

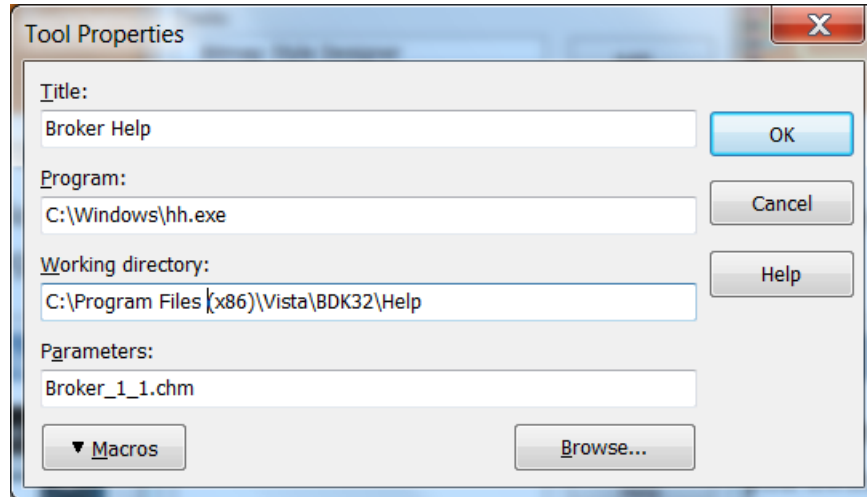
References

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

- *RPC Broker Release Notes*
- *RPC Broker Installation Guide*
- *RPC Broker Systems Management Guide* (this manual)
- *RPC Broker Technical Manual*
- *RPC Broker User Guide*
- *RPC Broker Developer’s Guide*—Document and BDK Online Help, which provides an overview of development with the RPC Broker. The help is distributed in two zip files:
 - *Broker_1_1.zip* (i.e., *Broker_1_1.chm*)—This zip file contains the standalone online HTML help file. Unzip the contents and double-click on the **Broker_1_1.chm** file to open the help.
 - *Broker_1_1-HTML_Files.zip*—This zip file contains the associated HTML help files. Unzip the contents in the same directory and double-click on the **index.htm** file to open the help.

You can create an entry for **Broker_1_1.chm** in Delphi's Tools Menu, to make it easily accessible from within Delphi. To do this, use Delphi's **Tools | Configure Tools** option and create a new menu entry as shown in [Figure 1](#).

Figure 1: Delphi's Tool Properties dialogue—Broker_1_1.chm entry



- RPC Broker VA Intranet website.

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

VistA documentation is made available online in Microsoft® Word format and in Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader, which is freely distributed by Adobe Systems Incorporated at: <http://www.adobe.com/>

VistA documentation can be downloaded from the VA Software Document Library (VDL) Website: <http://www.va.gov/vdl/>

VistA documentation and software can also be downloaded from the Product Support (PS) Anonymous Directories.

1 Introduction

1.1 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 Embarcadero’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 1.1 provides 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.
- Broker Development Kit (BDK). The BDK provides VistA application programmers with the following features:
 - The capability to create GUI client/server VistA applications using Embarcadero’s Delphi software. The BDK provides the TRPCBroker, 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).

- Capability to operate in a 32-bit environment. The client workstation can be running any of the following Microsoft® operating systems:
 - Windows 7
 - Windows 8.1
 - Windows 10
- Support for IPv4/IPv6 Dual-stack Environments. The TRPCBroker component enabled Internet Protocol (IP) version 4 or version 6 to be used for Vista connections. This functionality is transparent to the user and is available to any application compiled with RPC Broker 1.1 Development Kit (BDK). (XWB*1.1*60)
- Support for Secure Shell (SSH). The TRPCBroker component enabled Secure Shell (SSH) Tunnels to be used for secure connections. This functionality is controlled by setting an internal property value (mandatory SSH) or command line option at run time. (XWB*1.1*50)
- Support for Broker Security Enhancement (BSE). The TRPCBroker component enables visitor access to remote sites using authentication established at a home site. (XWB*1.1*50)



REF: For more information on BSE, see the *Broker Security Enhancement (BSE) Sup: XWB*1.1*45 XU*8.0*404* document located on the VDL at: <http://www.va.gov/vdl/application.asp?appid=23>

- Support for Single Sign-On/User Context. As of Patch XWB*1.1*40, the TCCOWRPCBroker component enabled Single Sign-On/User Context (SSO/UC) in CCOW-enabled applications. This allow users to authenticate and sign on to multiple applications that are CCOW-enabled and SSO/UC-aware using a single set of credentials, which reduces the need for multiple ID's and passwords in the HealthVet clinician desktop environment.



REF: For more information on SSO/UC, see the *Single Sign-On/User Context (SSO/UC) Installation Guide* and *Single Signon/User Context (SSO/UC) Deployment Guide* located on the VDL at: <http://www.va.gov/vdl/application.asp?appid=162>

- Support for Non-Callback Connections. As of Patch XWB*1.1*35, the RPC Broker components are built with a UCX or non-callback Broker connection, so that it can be used from behind firewalls, routers, etc. This functionality is controlled via the new TRPCBroker component IsBackwardCompatibleConnection property.
- Support for Silent Logons. As of Patch XWB*1.1*13, the RPC Broker provides “Silent Login” capability. It provides functionality associated with the ability to make logins to a Vista M Server without the RPC Broker asking for Access and Verify code information.
- 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.



REF: For more information on Auto Signon, see the “[Integrated Auto Signon for Multiple User Sessions](#)” section.

- Support for multi-instances of the RPC Broker. The RPC Broker code permits an application to open two separate Broker instances with the same Server/ListenerPort combination, resulting in two separate partitions on the server. Previously, an attempt to open a second Broker instance ended up using the same partition. For this capability to be useful for concurrent processing, an application would have to use threads to handle the separate Broker sessions. (XWB*1.1*13)



CAUTION: Although there should be no problems, the RPC Broker is *not yet guaranteed to be thread safe*.

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



REF: For more information on troubleshooting the Broker, see the “[Troubleshooting](#)” section.

1.2 How Does It All Work?

The process begins on a user’s workstation (i.e., PC), running Microsoft® Windows, which is connected 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).



REF: For more specific environment requirements, see the *RPC Broker Installation Guide*.

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.



REF: For more information on Auto Signon, see the “[Integrated Auto Signon for Multiple User Sessions](#)” section.

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-in 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.



NOTE: RPC Broker supports messaging for *non*-Delphi client applications (e.g., C++, Microsoft[®] Visual Basic, or other COTS Microsoft[®] Windows-based products). Earlier versions of RPC Broker 1.1 supplied 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.

REF: For more specific information about the Broker DLLs, see the BDK Online Help (i.e., Broker_1_1.chm) or *RPC Broker Developer's Guide*.

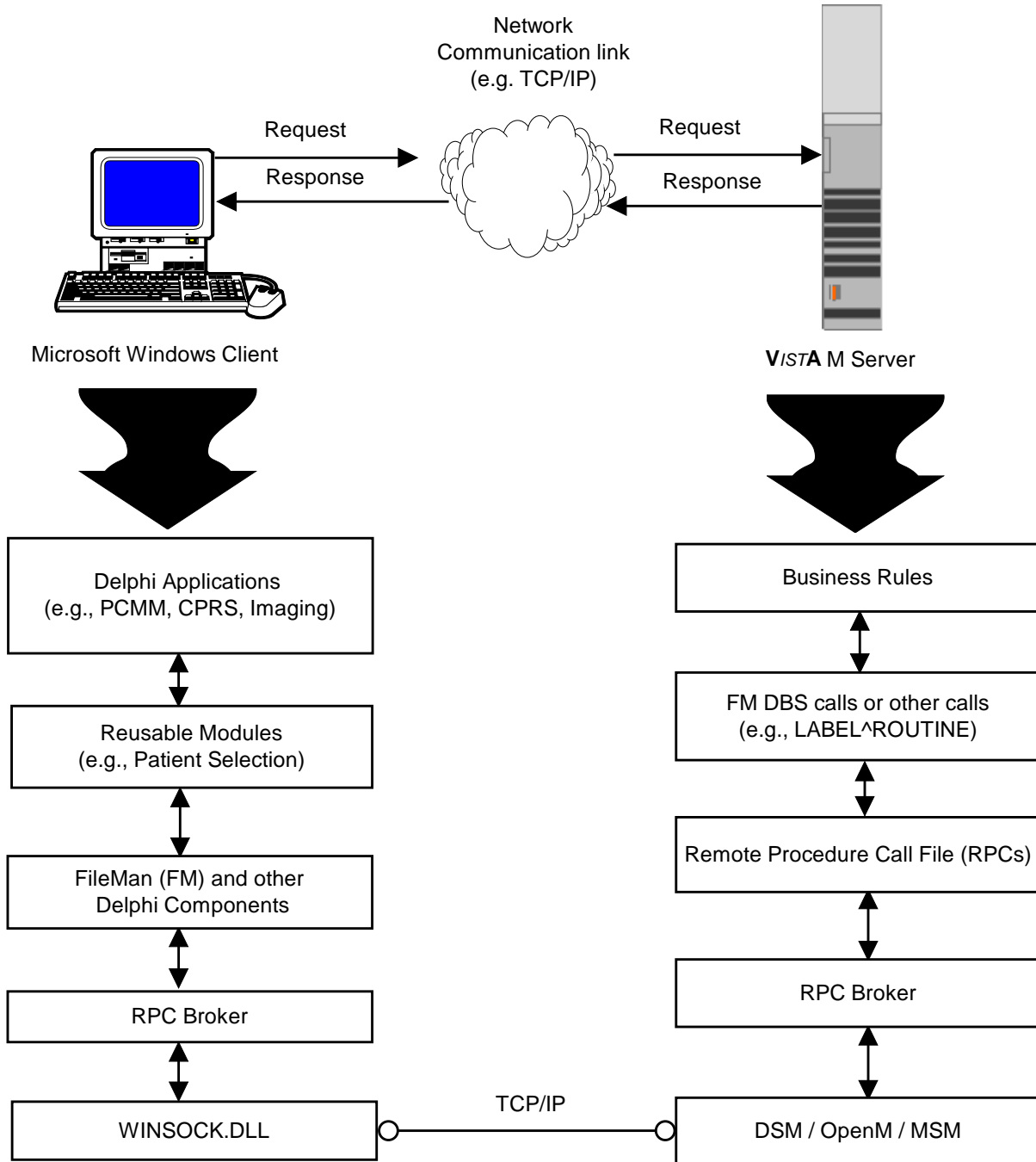


NOTE: The BAPI32.DLL contains all of the 32-bit Broker DLL functions. It provides an interface to the Broker component. The BAPI32.DLL has *not* been updated to support Secure Shell (SSH) or IPv4/IPv6 dual-stack environments. It is expected that future application development will use the VA Enterprise Services Bus (ESB) environment and discontinue the use of the RPC Broker.

1.3 System Overview

Figure 2 gives an overview of the VistA/RPC Broker environment:

Figure 2: VistA RPC Broker system overview diagram



2 System Features

2.1 Client Features

2.1.1 RPC Broker Client Agent

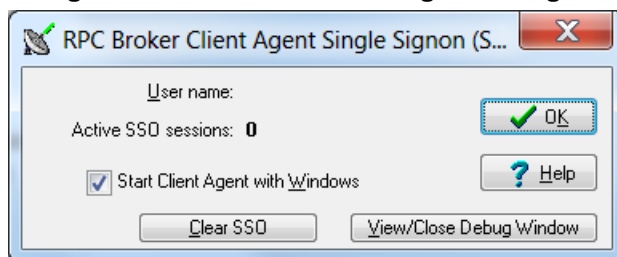
The RPC Broker Client Agent program (i.e., CLAGENT.EXE) runs in support of the Auto Signon process. This program automatically and continuously runs in the background on the client workstation and normally should *not* be closed or shut down by the user. A satellite dish icon will be displayed in the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



NOTE: A “beta” CAgent.exe application is included in XWB*1.1*60 that supports IPv4/IPv6 dual-stack environments. It has *not* been field tested or Section 508 certified, and is made available to developers to assist in application testing. It is expected that this application will be released to the field in a future patch.

By double clicking on the Client Agent icon, you can see how many active connections are currently open, as shown in [Figure 3](#). However, the “Active connections” count may include “orphan” connections that are no longer active. Use this count as an approximate count only.

Figure 3: RPC Broker Client Agent dialogue



NOTE: 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 can prevent the Client Agent from starting automatically. Administrators can always reset it to start automatically by starting the Client Agent manually with administrative privileges first and re-checking this checkbox. Changing the default settings requires administrative privileges on the workstation.



REF: For more information on Auto Signon, see the “[Integrated Auto Signon for Multiple User Sessions](#)” section.

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).

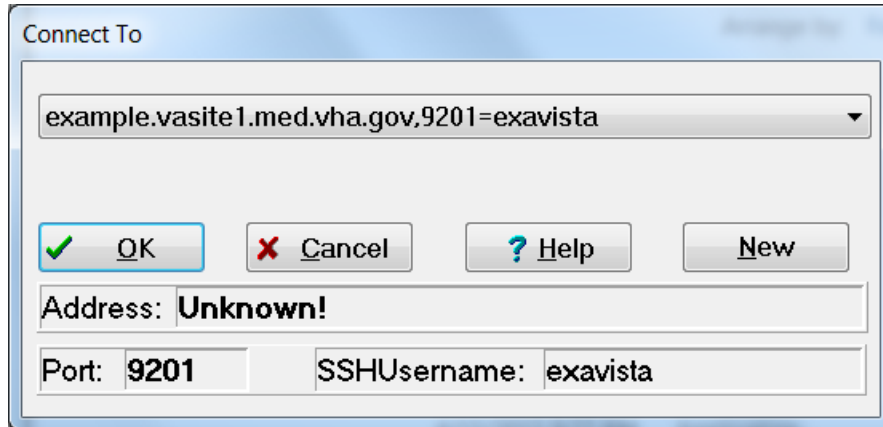


REF: For more information on the End-User Client Workstation and Client Agent installation, see the *RPC Broker Installation Guide*.

2.1.2 “Connect To” Dialogue

Upon logging in to a Vista client/server application, users may be presented with the “Connect To” dialogue, as shown in [Figure 4](#):

Figure 4: Server and port configuration selection dialogue



This server and port configuration selection dialogue 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. If a Secure Shell (SSH) connection is desired, also enter the SSHUsername associated with the server.

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 associate port number to the Microsoft® Windows Registry requires administrator privileges on the workstation. You can add, remove, or modify as many registry key values under the following location, as you want there to be server entries available:

- Key Name: HKEY_LOCAL_MACHINE\Software\Wow6432Node\Vista\Broker\Servers
- Value: mnemonic name (e.g., BROKERSERVER1)
- Data: IP Address;Port (e.g., abcde.site.med.va.gov;####0)

2.1.3 Edit Broker Servers Application

The Edit Broker Servers application (i.e., SERVERLIST.EXE), previously distributed with earlier versions of the Broker, has been replaced by a “beta” ServerList.exe in RPC Broker 1.1. The Edit Broker Servers application provides the means to add, edit, or delete server and port number entries in the Microsoft® Windows Registry.

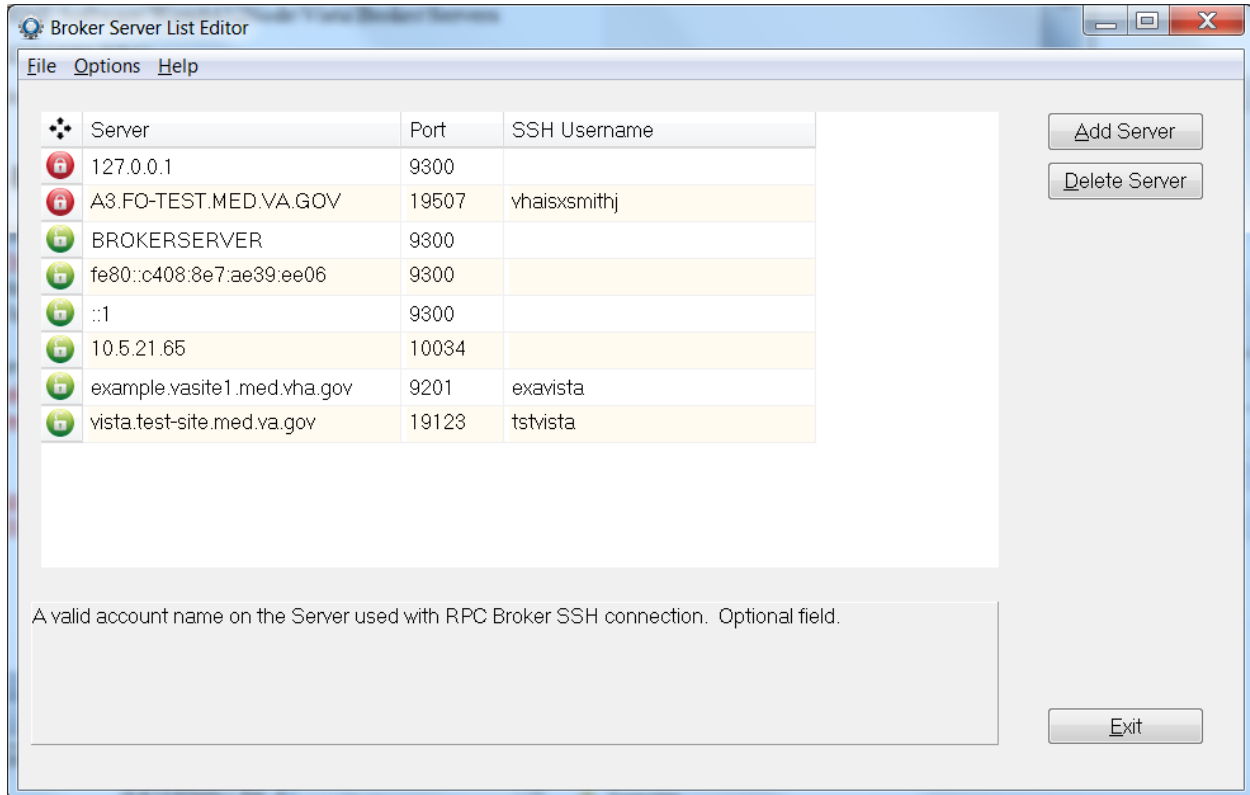


NOTE: Microsoft® Windows 7 includes additional levels of security that prevented earlier versions of this application from working. The ServerList.exe application included in XWB*1.1*60 has *not* been field tested or Section 508 certified, and is made available to developers to assist in application testing.

When opened normally (i.e., double-click), the application allows users to edit entries for their Windows environment only. The following entries are displayed in [Figure 5](#):

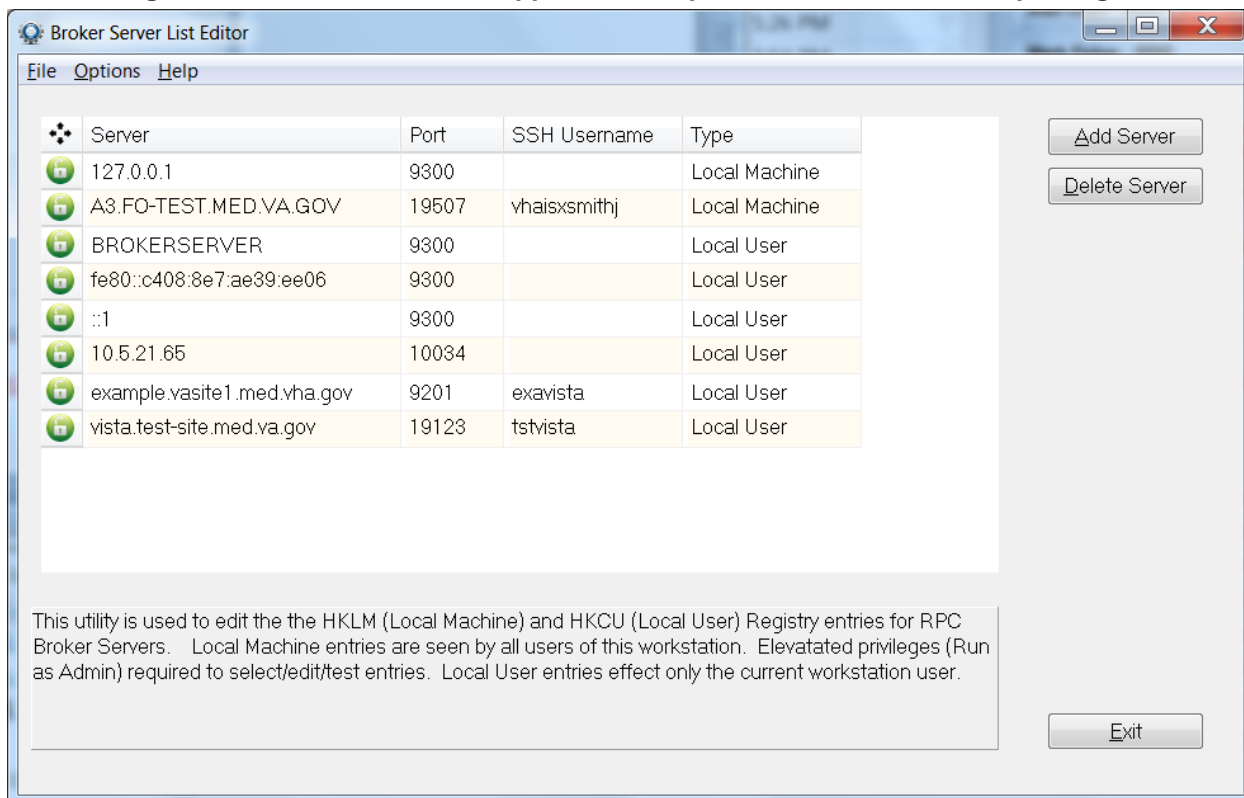
- Entries with a green “open lock” icon are *not* available to other users of the same computer.
- Entries with a red “closed lock” icon are shared and *cannot* be edited.

Figure 5: Edit Broker Servers Application—Opened normally



When opened with Administrator privileges (i.e., right-click and Run as administrator), all entries can be edited by a user with Administrator privileges on the computer.

Figure 6: Edit Broker Servers Application—Opened with Administrator privileges



2.1.4 Standalone Applications and their Associated Help Files

Each of the standalone Broker applications listed in [Table 3](#), which were distributed with earlier versions of the RPC Broker, have an associated help file that *must* reside in the *same* directory in order to provide online help for that particular standalone program:

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

Standalone Program	Associated Help File	Location
CLAGENT.EXE	CLAGENT.HLP	End-User Workstation
RPCTEST.EXE	RPCTEST.HLP	End-User Workstation

The installation of the Broker automatically loads these associated files into the appropriate directories. If you choose to “export” a standalone application to another client workstation, make sure you include its associated help file and place them both in the *same* directory.



REF: For more information on the CLAGENT.EXE, see the “[RPC Broker Client Agent](#)” section.

REF: For more information on the RPCTEST.EXE, see the “[Troubleshooting](#)” section.

2.1.5 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 any of the following:

- DNS
- DHCP (Dynamic Host Configuration Protocol)
- WINS (Windows Name Service).

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

Windows 7 systems include additional levels of security that require additional steps to edit the HOSTS file:

1. Folder options *must* be set to display hidden folders.
2. The text editor needs to “Run as Administrator” or the HOSTS file will not be permitted to be saved in the proper directory.

[Table 4](#) illustrates where you can find this file based on your client Microsoft® Windows operating system (OS):

Table 4: HOSTS file location in Windows

Version of Windows OS	File (Location and Name)
Windows 7	C:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS
Windows 8.1	C:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS
Windows 10	C:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS

A sample of the Microsoft® Windows 7 HOSTS file (i.e., C:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS) is displayed in [Figure 7](#) (modifications/additions made to this sample file are in boldface and italicized):

Figure 7: Sample HOSTS file

```
# Copyright (c) 1993-2009 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
#
# localhost name resolution is handled within DNS itself.
#   127.0.0.1         localhost
#   ::1               localhost
#
#   IP                Host
#   ADDRESS           Name                Description
#   |                 |                 |
#   V                 V                 V
#
192.1.1.1         BROKERSERVER    # Broker Server
```

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.



CAUTION: A DHCPSEVER entry is still required for software that uses RPC Broker 1.0. You can create an additional entry for BROKERSERVER in your HOSTS file or DNS. However, do *not* remove the DHCPSEVER entry if it is already present.

2.1.5.1 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 **Enter** 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 ([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.



NOTE: Do not save the HOSTS file with an extension!

7. Close the HOSTS file and text editor.

2.1.5.2 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.



NOTE: Do not save the HOSTS file with an extension!

5. Close the HOSTS file and the text editor.

2.1.6 What Happened to the Client Manager?

The Client Manager, previously distributed with version 1.0 of the Broker, is no longer used by RPC Broker 1.1. 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 1.0 is no longer used, thus, the Client Manager is no longer required with RPC Broker 1.1. Configuration of programmer preferences will now be done via the Broker Programmer Preferences dialogue.



REF: For more information on the Broker Programmer Preferences dialogue, see the “RPC Broker Programmer Preference Editor” section in the BDK Online Help (i.e., BROKER.HLP; *RPC Broker Developer’s Guide*).



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

2.1.7 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. 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.




CAUTION: The VISTA.INI file created with Broker 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.

[Table 5](#) lists items from the VISTA.INI file and their disposition with RPC Broker 1.1:

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

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	Sign-on 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 Sign-on form.

VISTA.INI File Item	Disposition
IntroTextFont	 REF: For more information, see the “ Users Can Customize Vista Signon Dialogue ” section.
HostsPath	No longer useful (i.e., Broker 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.

2.2 Server Features

2.2.1 RPC Broker Management Menu

The RPC Broker Management Menu [XWB MENU] is for system managers. It contains the options shown in [Figure 8](#):

Figure 8: RPC Broker Management Menu option [XWB MENU]

```
Select RPC Broker Management Menu Option:

    RPC Listener Edit
    Start All RPC Broker Listeners
    Stop All RPC Broker Listeners
    Clear XWB Log Files
    Debug Parameter Edit
    View XWB Log
```



NOTE: This menu was introduced with RPC Broker Patch XWB*1.1*9 and updated with subsequent RPC Broker patches.

2.2.1.1 RPC Listener Edit Option

Use the RPC Listener Edit option. [XWB LISTENER EDIT] to create or edit listener entries.



REF: For more information on the RPC Listener Edit option, see the “[Editing the Listener Site Parameters](#)” section.

2.2.1.2 Start All RPC Broker Listeners Option

Use the Start All RPC Broker Listeners option [XWB LISTENER STARTER] to automatically start all listeners configured in the RPC BROKER SITE PARAMETERS file (#8994.1).



REF: For more information on the Start All RPC Broker Listeners option, see the “[To Start All Listeners](#)” section.

2.2.1.3 Stop All RPC Broker Listeners Option

Use the Stop All RPC Broker Listeners option [XWB LISTENER STOP ALL] to stop all running listeners configured in the RPC BROKER SITE PARAMETERS file (#8994.1) set to automatically start.



REF: For more information on the Stop All RPC Broker Listeners option, see the [“To Stop All Running Listeners”](#) section.

2.2.1.4 Clear XWB Log Files Option

Use the Clear XWB Log Files option [XWB LOG CLEAR] to clear (KILL) the XWB log files, which are stored in a temporary global under ^TMP(“XWBDEBUG”, \$J).

2.2.1.5 Debug Parameter Edit Option

Use the Debug Parameter Edit option [XWB DEBUG EDIT] to edit the “XWBDEBUG” parameter defined in the PARAMETER DEFINITION file (#8989.51) and stored in the PARAMETERS file (#8989.5) when set.

2.2.1.6 View XWB Log Option

Use the View XWB Log option [XWB LOG VIEW] to view the temporary debug log files that the Broker can set. The XWBDEBUG parameter *must* be set for log files to be recorded in temporary global ^TMP(“XWBDEBUG”, \$J).

2.2.2 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., ###0 for production and ###1 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 ###0).

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.).

2.2.2.1 Obtaining an Available Listener Port—Alpha/VMS Systems

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:

Figure 9: Obtaining an Available Listener Port—Alpha/VMS Systems

```
$ UCX SHOW DEVICE_SOCKET
```

Port		Remote			
Device_socket	Type	Local	Remote	Service	Host
bg3	STREAM	###1	0	HL7	0.0.0.0
bg23	STREAM	###2	0	Z3ZTEST	0.0.0.0
bg24	STREAM	###3	0	ZSDPROTO	0.0.0.0

For example, if port *###1* shows up in the Local Port column as shown in the first entry above, some other application is already using port *###1* and you should choose another port.

2.2.2.2 Obtaining an Available Listener Port—Linux Systems

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

Figure 10: Obtaining an Available Listener Port—Linux Systems

```
$ netstat -lntu
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	Sta
tcp	0	0	0.0.0.0:###1	0.0.0.0:*	LIS
tcp	0	0	99.999.999.999:###6	0.0.0.0:*	LIS
tcp	0	0	127.0.0.1:199	0.0.0.0:*	LIS
tcp	0	0	0.0.0.0:###2	0.0.0.0:*	LIS
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LIS
tcp	0	0	0.0.0.0:###3	0.0.0.0:*	LIS
tcp	0	0	127.0.0.1:###7	0.0.0.0:*	LIS
tcp	0	0	0.0.0.0:###4	0.0.0.0:*	LIS
tcp	0	0	127.0.0.1:###5	0.0.0.0:*	LIS

For example, if port *###1* shows up in the Local Address column as shown in the first entry above, some other application is already using port number *###1* and you should choose another port.

2.2.3 Starting and Stopping Listeners

2.2.3.1 To Start All Listeners

To automatically start all listeners configured in the RPC BROKER SITE PARAMETERS file (#8994.1), use the Start All RPC Broker Listeners option [XWB LISTENER STARTER]. This option first **stops** any of these listeners that may be running, and then starts all of them up.



NOTE: TaskMan *must* be running to use this option.



NOTE: This option was introduced with patch XWB*1.1*9.

2.2.3.2 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 (#8994.1) for that listener.



REF: For more information about the RPC BROKER SITE PARAMETERS file (#8994.1), see the “[RPC BROKER SITE PARAMETERS File](#)” section.

2.2.3.3 To Stop All Running Listeners

To stop all running listeners configured in the RPC BROKER SITE PARAMETERS file (#8994.1) set to automatically start, use the Stop All RPC Broker Listeners option [XWB LISTENER STOP ALL].



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

2.2.3.4 To Start Up a Single Listener Directly

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

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

2.2.3.5 To Stop a Single Listener Directly

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

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



NOTE: If you want to restart this listener after stopping it, and other listeners are running on the system, start the listener up directly (see Section [2.2.3.4](#)) rather than via the Start All RPC Broker Listeners option (which first **stops** all listeners *before* restarting them).

2.2.3.6 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.

To automatically start the Listeners 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:

Figure 11: Automatically starting the Listener(s) when TaskMan is restarted

```
Select Systems Manager Menu Option: TASKMAN <Enter> Management
Select Taskman Management Option: SCH <Enter> edule/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
```

2.2.4 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.

2.2.4.1 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:

Figure 12: RPC Listener Edit option—Sample user dialogue

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

Select BOX-VOLUME PAIR: ABC:DEF1213// <Enter>
BOX-VOLUME PAIR: ABC:DEF1213// <Enter>
Select PORT: ##0// <Enter>
PORT: ###0// <Enter>
STATUS: STARTING// <Enter>
CONTROLLED BY LISTENER STARTER: YES//
```

The meaning of the site parameter field for a given listener entry is shown in [Table 6](#):

Table 6: Listener site parameter entries description table

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 upon which the listener will listen.
STATUS	Ordinarily, this field 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.

2.2.5 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, 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.



NOTE: 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.

2.2.5.1 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 (#200) are checked first. If the user fields in the NEW PERSON file (#200) are null, the values in the KERNEL SYSTEM PARAMETERS file (#8989.3) will be used.



NOTE: The AUTO SIGN-ON field in the NEW PERSON file (#200) and the DEFAULT AUTO SIGN-ON field in the KERNEL SYSTEM PARAMETERS file (#8989.3) are initially set to null.



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

2.2.5.2 Examples

2.2.5.2.1 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.

2.2.5.2.2 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.

[Table 7](#) can be used as a guide to control multiple signons and Auto Signon for some typical situations:

Table 7: Multiple and Auto Signon Settings table

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:		
 NOTE: 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

* 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)



CAUTION: 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.

2.2.6 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-in 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).

2.2.7 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 (#8989.3) 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.



REF: For advice regarding changing the value for this field, see 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.



NOTE: 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.

3 Security

3.1 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.

3.2 Validation of Connection Request

An enhancement to security has been included with RPC Broker 1.1. 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.

3.3 Validation of Users

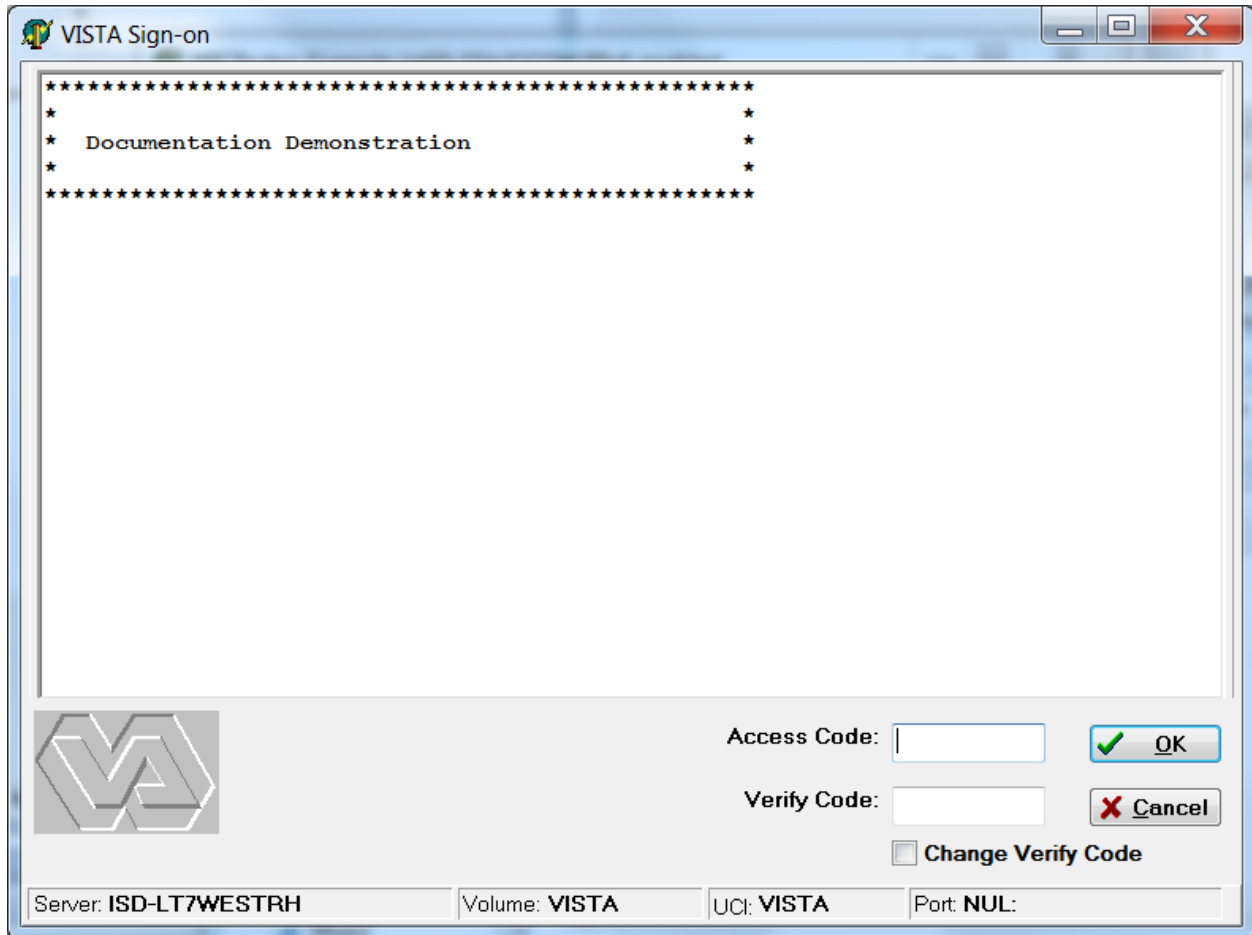
The GUI VistA Sign-on dialogue is integrated with the RPC Broker interface. This VistA Sign-on dialogue is invoked when the client application connects to the server.

3.3.1 VistA Signon Dialogue

The VistA Signon dialogue automatically prompts users for their Access and Verify codes if they are not already signed on to a VistA application ([Figure 13](#)).

A sample of the VistA Signon dialogue integrated with the RPC Broker is illustrated [Figure 13](#):

Figure 13: Sample VistA Signon security dialogue



NOTE: RPC Broker 1.1 supports both Auto Signon and Single Sign-On/User Context (SSO/UC).

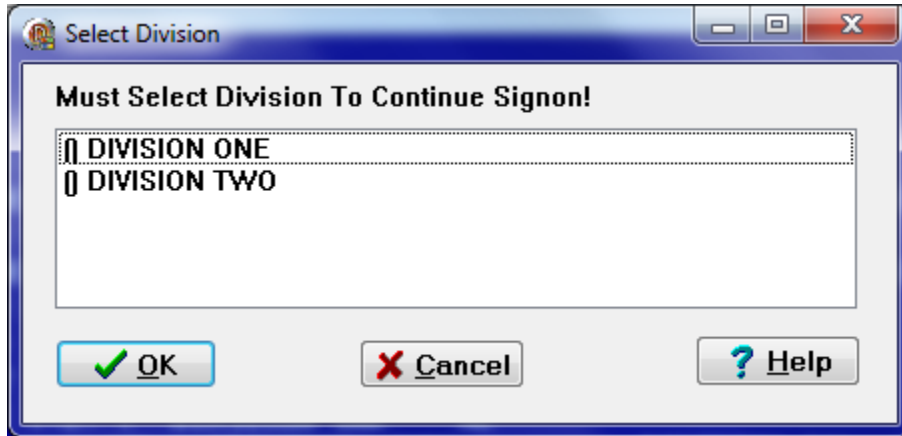
REF: For more information regarding Auto Signon, see the [“Integrated Auto Signon for Multiple User Sessions”](#) section or the description in Kernel Patch XU*8.0*59.

REF: For more information on SSO/UC, please consult the *Single Sign-On/User Context (SSO/UC) Installation Guide* and *Single Sign-On/User Context (SSO/UC) Deployment Guide* on the VHA Software Documentation Library (VDL).

3.3.2 VistA Division Selection Dialogue

After entering an Access and Verify code, if a user is associated with more than one Institution, the user will be presented with a dialogue similar to [Figure 14](#):

Figure 14: Select Division dialogue—Sample entries



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 **OK** 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.



NOTE: 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 Signon dialogue ([Figure 13](#)) 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.



REF: For more information on Access and Verify codes or the Kernel signon process in general, see the “Signon/Security” section in the *Kernel Systems Management Guide*.

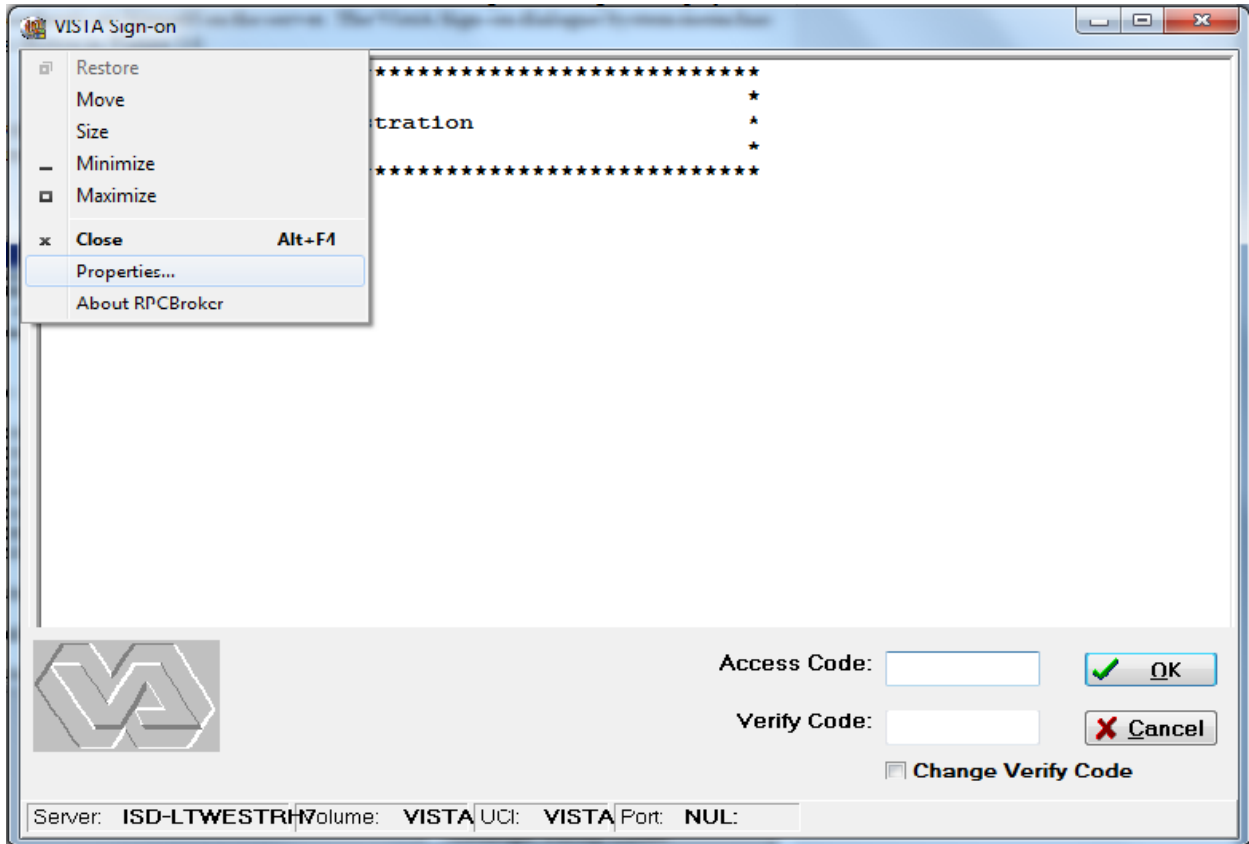
3.3.3 Users Can Customize VistA Signon Dialogue

When a VistA application on the client connects to the server, the VistA Sign-on dialogue is displayed for the user to identify and authenticate himself on the server. The VistA Sign-on dialogue System menu has a “Properties...” item, as shown in [Figure 15](#):



NOTE: Move your mouse anywhere in the dialogue’s Title bar and right click to display the System menu.

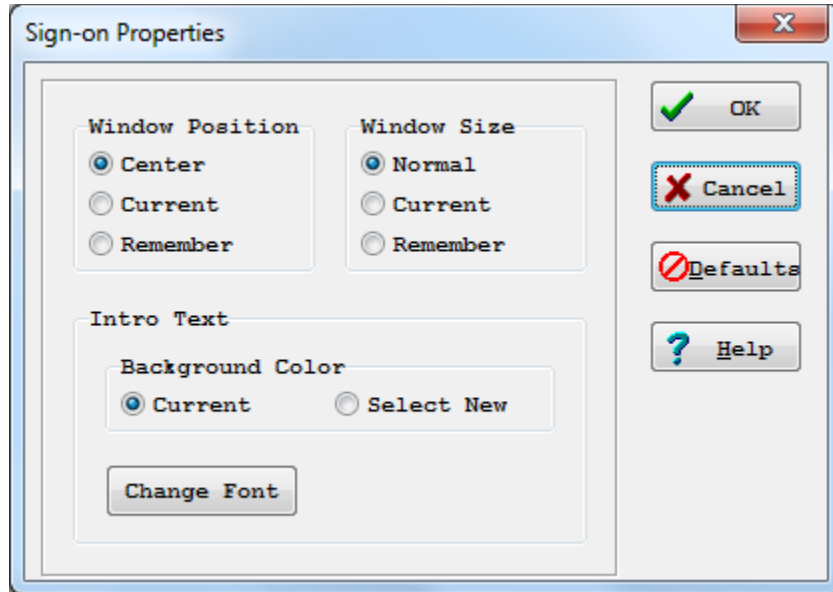
Figure 15: Sign-on Properties on the System Menu



3.3.3.1 Sign-on Properties

When this item is selected, the user is presented with the Sign-on Properties dialogue, as shown in [Figure 16](#):

Figure 16: Sign-on Properties dialogue



Using this dialogue ([Figure 16](#)), users can control the appearance of the VistA Sign-on dialogue by modifying the following characteristics:

- Window Position—Position of the VistA Sign-on dialogue.
- Window Size—Size of the VistA Sign-on dialogue.
- Introductory Text—Appearance of the introductory text in the VistA Sign-on dialogue.

3.3.3.1.1 Window Position

The VistA Sign-on dialogue's window position can be one of the entries in [Table 8](#):

Table 8: Window position

Position	Description
Center (default)	The VistA Sign-on dialogue will always appear in the center of the screen.
Current	The current position of the VistA Sign-on dialogue will be saved and used in the future.
Remember	Each time the VistA Sign-on dialogue is used and closed, it will record its position and open in that same place the next time it is used.

3.3.3.1.2 Window Size

The VistA Sign-on dialogue's window size can be one of the entries in [Table 9](#):

Table 9: Window size

Size	Description
Normal (default)	The size of the VistA Sign-on dialogue as it was designed. Typically, this is 500 pixels wide by 300 pixels high.
Current	The current size of the VistA Sign-on dialogue will be saved and used in the future.
Remember	Each time the VistA Sign-on dialogue is used and closed, it records its size and open with the same size the next time it is used.

3.3.3.1.3 Introductory Text

The VistA Sign-on dialogue's introductory text has a couple of settings users can control:

- **Background Color:**

Table 10: Introductory text background color

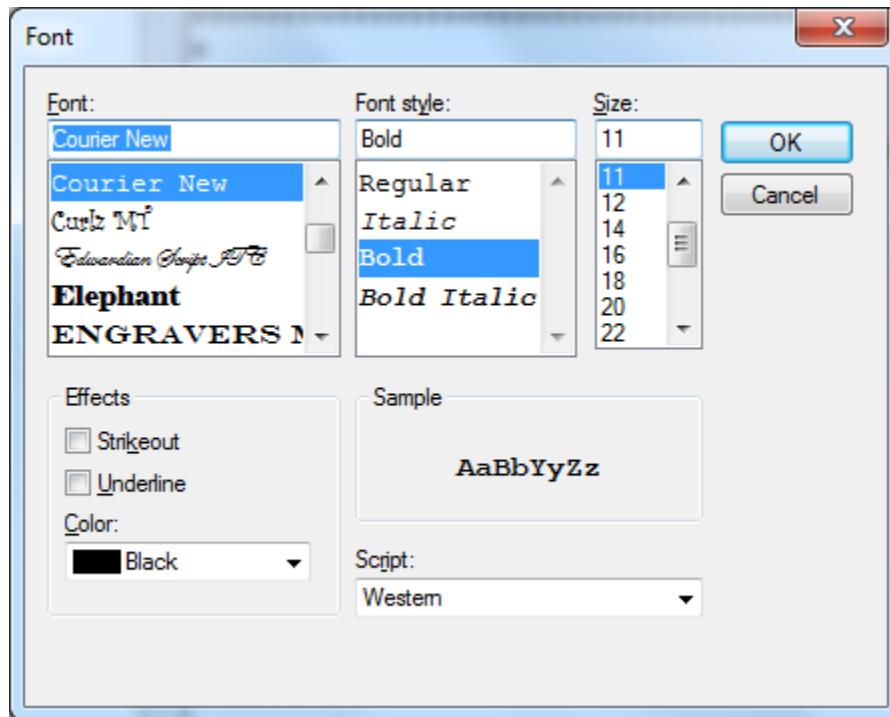
Color	Description
Cream (default)	According to the VA GUI conventions, this is the background color that should be used with text that users <i>cannot</i> edit.
White	For clarity and brightness.

- **Font:**

When users press **Change Font** they are presented with a Font dialogue ([Figure 17](#)) that can be used to change the following font attributes of the introductory text of the VistA Sign-on dialogue:

- Face
- Style
- Size
- Effects
- color

Figure 17: Sample Font dialogue



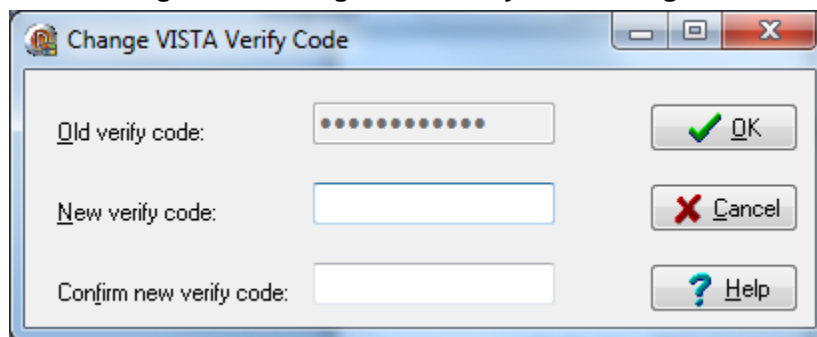
3.3.4 Change Vista Verify Code Component

RPC Broker 1.1 includes a Change Vista Verify Code dialogue 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 **OK** they are presented with the Change Vista Verify Code dialogue as displayed in [Figure 18](#):

Figure 18: Change Vista Verify Code dialogue



NOTE: The old Verify code appears 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 Verify Code” on the Sign-on screen ([Figure 13](#)). After signing on, it invokes the dialogue described in [Figure 18](#).
- Roll-and-Scroll environment (existing functionality)—Use the Edit User Characteristics option [XUSEREDITSELF] to edit your Verify code.

3.4 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.



REF: For more information on registering an RPC to a VistA software application, see the “RPC Security: How to Register an RPC” section in the *RPC Broker Getting Started with the Broker Development Kit (BDK)* manual.

3. Menu Manager checks if the RPC is registered for this software 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.

3.5 Sample Security Procedures

The security steps each client user will follow and the intermediate client/server security processes are described in the example in [Table 11](#):

Table 11: Sample security procedures

Step	Description
1.	The user starts a VistA 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 VistA Sign-on dialogue (Figure 13) 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.

3.6 Security Features Tasks Summary

[Table 12](#) summarizes required security tasks:

Table 12: Security tasks summary

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



NOTE: 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.



CAUTION: 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

4.1 Test the Broker Using the RPC Broker Diagnostic Program

RPC Broker Patch XWB*1.1*47 included a diagnostic tool for the client workstation ([Figure 19](#)). 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.



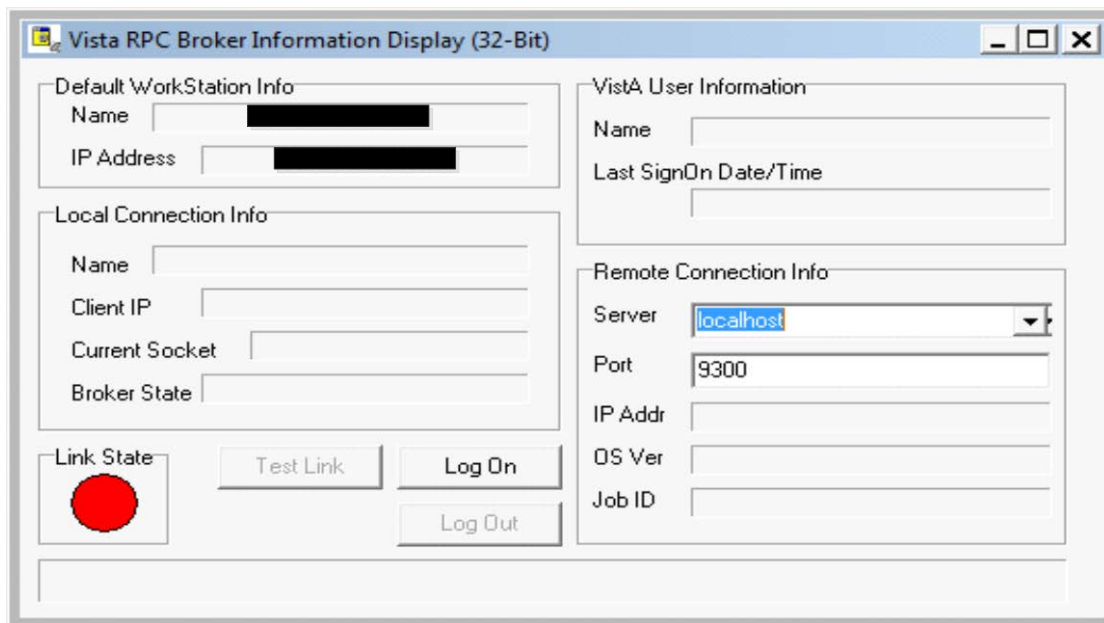
NOTE: This utility has *not* yet been updated to support IPv4/IPv6 dual-stack environment testing and has *not* been reviewed for Section 508 conformance.

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 dialogue in [Figure 19](#) is displayed:

Figure 19: 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 **Log On**; you will be presented with the VistA Sign-on dialogue. 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 dialogue 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.



REF: For more information on starting the Broker Listener, see the “[Broker Listeners and Ports](#)” section.

- Have you installed all current Kernel, Kernel Toolkit, and VA FileMan patches? If not, you must install all required patches (see the *RPC Broker 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.



NOTE: 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.

4.2 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 ###.###.###.### to the server host to which you are trying to connect (where ###.###.###.### 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
```



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



REF: For more information on telecommunications support, please visit the Telecommunications Support Office Home Page on the VA Intranet.

4.3 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 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:

- Disable Auto Signon for *all* users by setting the DEFAULT AUTO SIGN-ON field in the KERNEL SYSTEM PARAMETERS file (#8989.3) to “DISABLED”
- Install and run the Broker Client Agent on *all* client workstations, if Auto Signon is enabled on your system.



REF: For more information on the DEFAULT AUTO SIGN-ON field, see the “[Integrated Auto Signon for Multiple User Sessions](#)” section.

4.4 RPC Broker FAQs

For examples of general or development-specific frequently asked questions (FAQs) about the RPC Broker, see VA Intranet website.

Glossary

Term	Description
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.
DHCP	D ynamic H ost C onfiguration P rotocol.
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> <ul style="list-style-type: none"> • 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 do not carry the code itself; you have a pointer to the common library. All applications using it will then share one image. • 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. • Help avoid redundant routines. They provide generic functions that can be used by a variety of programs.
GUI	G raphical U ser I nterface. 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.
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.
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.

Term	Description
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 software, but does not allow programming, modification to data dictionaries, or other operations that require programmer access. Any of VistA'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	<p>The way the software 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 Embarcadero's Delphi Graphical User Interface to display the various menu option choices, commands, etc.</p>
WINDOW	<p>An object on the screen (dialogue) that presents information such as a document or message.</p>



REF: For a list of commonly used terms and definitions, see the [OI&T Master Glossary VA Intranet Website](#).

For a list of commonly used acronyms, see the [VA Acronym Lookup Intranet Website](#).

Index

A

- Access Code, 2, 3, 19, 20, 23, 24, 25, 31, 35
- Acronyms
 - Intranet Website, 37
- Adding Entries
 - HOSTS File, 12
- APIs
 - STOP^XWBTCP, 17
 - STRT^XWBTCP, 17
- Assumptions, xiv
- Auto Signon, 2, 3, 6, 19, 20, 21, 24, 35
 - Enabling/Disabling, 19
- AUTO SIGN-ON Field, 19, 21

B

- Background Color, 28
- BOX-VOLUME PAIR Field, 19
- Broker
 - FAQs, 35
 - Listeners and Ports, 15
 - Message Structure, 21
- BROKER ACTIVITY TIMEOUT Field, 22
- BROKERSERVER, 11, 12, 34
- B-type Options, 25

C

- Callout Boxes, xii
- Changing the VistA Verify Code, 29
- CLAGENT.EXE, 6, 9, 19
- Clear XWB Log Files Option, 15
- Client Agent, 6, 19, 21, 35
- Client Manager
 - What Happened to it?, 13
- Client/Server Timeouts, 22
- Codes
 - Access, 2, 3, 19, 20, 23, 24, 25, 31, 35
 - Verify, 2, 3, 19, 20, 23, 24, 25, 29, 30, 31, 35
- Color
 - Background, 28
- Commands
 - DCL, 16
- Commonly Used Terms, xiii
- Configuring Listeners, 17
- Connect To Dialogue, 7

- Connection Request
 - Validating, 23
- Connections
 - Diagnostics, 33
- Contents, vi
- CONTROLLED BY LISTENER STARTER
 - Field, 17
- CONTROLLED BY LISTENER STARTUP
 - Field, 19
- Customizing the Signon Dialogue, 26

D

- Data Dictionary
 - Data Dictionary Utilities Menu, xiv
 - Listings, xiv
- DCL Command, 16
- Debug Parameter Edit Option, 15
- DEFAULT AUTO SIGN-ON Field, 19, 21, 35
- DEFAULT MULTIPLE SIGN-ON Field, 19, 21
- DHCP, 10
- Diagnostics
 - Connection, 33
- Disclaimers, x
 - Software, x
- Divisions
 - VistA Division Selection Dialogue, 25
- DLL, 1, 4
 - RPCBI.DLL, 13
 - WINSOCK.DLL, 21, 34
- DNS, 34
- Documentation
 - Revisions, ii
 - Symbols, xi
- Documentation Conventions, xi
- Documentation Navigation, xii
- DOMAIN File (#4.2), 18
- Domain Name Service (DNS), 10

E

- Edit Broker Servers Application, 7
- Edit User Characteristics Option, 30
- Editing the Listener Site Parameters, 18
- Enabling/Disabling Auto Signon, 19

F

FAQs, 35

Features

- Client, 6
- Security, 23
- Server, 14
- System, 6

Fields

- AUTO SIGN-ON, 19, 21
- BOX-VOLUME PAIR, 19
- BROKER ACTIVITY TIMEOUT, 22
- CONTROLLED BY LISTENER STARTER, 17
- CONTROLLED BY LISTENER STARTUP, 19
- DEFAULT AUTO SIGN-ON, 19, 21, 35
- DEFAULT MULTIPLE SIGN-ON, 19, 21
- MULTIPLE SIGN-ON, 19, 21
- PORT, 19
- SPECIAL QUEUING, 17
- STATUS, 19

Figures, viii

Files

- DOMAIN (#4.2), 18
- Help, 9
- HOSTS, 10, 11, 12, 34
 - Adding Entries, 12
 - Modifying Entries, 12
- KERNEL SYSTEM PARAMETERS (#8989.3), 19, 21, 22, 35
- NEW PERSON (#200), 19, 21
- OPTION (#19), 25, 30
- OPTION SCHEDULING (#19.2), 17
- REMOTE PROCEDURE (#8994), 1, 3, 4, 21
- RPC BROKER SITE PARAMETERS (#8994.1), 3, 14, 15, 16, 17, 18
- VISTA.INI, 13

Font, 28

Frequently Asked Questions, 35

G

Glossary, 36

- Intranet Website, 37

H

Help

- At Prompts, xiii
- Online, xiii

Question Marks, xiii

Help Files, 9

History

- Revisions, ii

Home Pages

- Acronyms Intranet Website, 37
- Adobe Website, xv
- Glossary Intranet Website, 37
- RPC Broker Website, xv
- VA Software Document Library (VDL) Website, xv

HOSTS File, 10, 11, 12, 34

- Adding Entries, 12
- Modifying Entries, 12

How Does It All Work?, 3

How to

- Obtain Technical Information Online, xiii
- Use this Manual, x

I

Integrated Auto Signon, 19

Intended Audience, x

Introduction, 1

Introductory Text, 28

K

KERNEL SYSTEM PARAMETERS File (#8989.3), 19, 21, 22, 35

L

LAN, 3, 10, 34, 36

List File Attributes Option, xiv

Listeners

- Configuring, 17
- Starting, 16
 - Single, 17
- Stopping, 17
 - Single, 17
- Tasking, 17

Listeners and Ports, 15

M

Menu for System Managers, 14

Menu Manager, 25

Menus

- Data Dictionary Utilities, xiv

- RPC Broker Management Menu, 14
- XWB MENU, 14
- Message Structure, 21
- Microsoft Windows Registry, 7, 13, 14
- Modifying Entries
 - HOSTS File, 12
- MULTIPLE SIGN-ON Field, 19, 21

N

- Network Connection, 34
- NEW PERSON File (#200), 19, 21

O

- Obtaining
 - Available Listener Port
 - Alpha/VMS Systems, 16
 - Linux Systems, 16
 - Data Dictionary Listings, xiv
- Online
 - Documentation, xiii
 - Technical Information, How to Obtain, xiii
- OPTION File (#19), 25, 30
- OPTION SCHEDULING File (#19.2), 17
- Options
 - B-type, 25
 - Clear XWB Log Files, 15
 - Data Dictionary Utilities, xiv
 - Debug Parameter Edit, 15
 - Edit User Characteristics, 30
 - List File Attributes, xiv
 - RPC Broker Management Menu, 14
 - RPC Listener Edit, 14, 18
 - Schedule/Unschedule Options, 17
 - Start All RPC Broker Listeners, 14, 16, 17, 19
 - Stop All RPC Broker Listeners, 15, 17, 19
 - View XWB Log, 15
 - XUSEREDITSELF, 30
 - XWB DEBUG EDIT, 15
 - XWB LISTENER EDIT, 14
 - XWB LISTENER STARTER, 14, 16, 17, 19
 - XWB LISTENER STOP ALL, 15, 17
 - XWB LOG CLEAR, 15
 - XWB LOG VIEW, 15
 - XWB MENU, 14
- Orientation, x
- Overview, 1
 - System Diagram, 5

P

- Patches
 - Revisions, v
- PING, 34
- PORT Field, 19
- Ports and Listeners, 15
- Position
 - Window, 27
- Product Support (PS)
 - Anonymous Directories, xv
- Programs
 - CLAGENT.EXE, 6, 9, 19
 - RPCTEST.EXE, 9, 32, 33
- PS
 - Anonymous Directories, xv

Q

- Question Mark Help, xiii

R

- Registry, 7, 13, 14
- REMOTE PROCEDURE File (#8994), 1, 3, 4, 21
- Revision History, ii
 - Documentation, ii
 - Patches, v
- RPC Broker
 - Diagnostic Program
 - How to test the Broker, 32
 - FAQs, 35
 - Website, xv
- RPC Broker Management Menu, 14
- RPC BROKER SITE PARAMETERS File (#8994.1), 3, 14, 15, 16, 17, 18
- RPC Listener Edit Option, 14, 18
- RPCBI.DLL, 13
- RPCs
 - Validating, 30
- RPCTEST.EXE, 9, 32, 33

S

- Schedule/Unschedule Options Option, 17
- Security, 23
 - Change VistA Verify Code Component, 29
 - Features, 23
 - Sample Security Procedures, 31

- Signon Dialogue
 - Customizing, 26
 - Summary of Tasks, 31
 - Validating Connection Request, 23
 - Validating RPCs, 30
 - Validating Users, 23
- Server
 - Features, 14
- Signon
 - Delays, 35
 - Dialogue
 - Customizing, 26
 - Sample, 24
- Size
 - Window, 28
- Software Disclaimer, x
- SPECIAL QUEUING Field, 17
- Standalone Applications and their Associated Help Files, 9
- Start All RPC Broker Listeners Option, 14, 16, 17, 19
- Starting Listeners, 16
 - Single, 17
- STATUS Field, 19
- Stop All RPC Broker Listeners Option, 15, 17, 19
- STOP^XWBTCPI API, 17
- Stopping Listeners, 17
 - Single, 17
- STRT^XWBTCPI API, 17
- Support
 - Anonymous Directories, xv
- Symbols
 - Found in the Documentation, xi
- System
 - Features, 6
 - Overview Diagram, 5

T

- Table of Contents, vi
- Tables, ix
- Tasking Listeners, 17
- TCP/IP, 3, 34
- Test the Broker Using the RPC Broker Diagnostic Program, 32
- Text
 - Introductory, 28
- Timeouts, 22
- Troubleshooting, 32
 - Network Connection, 34

- RPC Broker Diagnostic Program, 32
- Signon Delays, 35

U

- URLs
 - Acronyms Intranet Website, 37
 - Adobe Website, xv
 - Glossary Intranet Website, 37
 - RPC Broker Website, xv
 - VA Software Document Library (VDL) Website, xv
- Users Can Customize VistA Signon Dialogue, 26

V

- VA Software Document Library (VDL) Website, xv
- Validating
 - Connection Request, Security, 23
 - RPCs
 - Security, 30
 - Users, Security, 23
- Verify and Test the Network Connection, 34
- Verify Code, 2, 3, 19, 20, 23, 24, 25, 29, 30, 31, 35
 - Changing, 29
- View XWB Log Option, 15
- VistA Division Selection Dialogue, 25
- VistA Signon Dialogue, 24
- VISTA.INI File, 13
 - What Happened to it?, 13

W

- Websites
 - Acronyms Intranet Website, 37
 - Adobe Website, xv
 - Glossary Intranet Website, 37
 - RPC Broker, xv
 - VA Software Document Library (VDL) Website, xv
- What Happened to the Client Manager?, 13
- What Happened to the VISTA.INI File?, 13
- Window
 - Position, 27
 - Size, 28
- Windows Registry, 7, 13, 14
- WINSOCK.DLL, 21, 34

X

XUSEREDITSELF Option, 30

XWB DEBUG EDIT Option, 15

XWB LISTENER EDIT Option, 14

XWB LISTENER STARTER Option, 14, 16,
17, 19

XWB LISTENER STOP ALL Option, 15, 17

XWB LOG CLEAR Option, 15

XWB LOG VIEW Option, 15

XWB MENU, 14