RPC Broker 1.1

Systems Management Guide

May 2020

Department of Veterans Affairs (VA)

Office of Information and Technology (OIT)

Enterprise Program Management Office (EPMO)

Revision History

Documentation Revisions

Date	Revision	Description	Authors
05/06/2020	8.0	Tech Edits based on the Broker Development Kit (BDK) release with RPC Broker Patch XWB*1.1*71. • Updated Section 1.1. functionality added with XWB*1.1*71. • Changed all references throughout to “Patch XWB*1.1*71” as the latest BDK release. • Updated references to show RPC Broker Patch XWB*1.1*71 supports Delphi 10.3, 10.2, 10.1, 10.0, and XE8 throughout. • This was a bug fix and small enhancement patch; however, there are no new options, routines, files, fields, security keys, APIs, or RPCs. • Reformatted all references to file and field name numbers throughout. • Updated all styles and formatting to match current documentation standards and style guidelines. **RPC Broker 1.1; XWB*1.1*71 BDK**	Developer: RG Technical Writer: TB
04/25/2017	7.1	Removed the following sections, since they are obsolete with the release of the latest Broker Development Kit (BDK); released with RPC Broker Patch XWB*1.1*65: • Removed Section 2.2.3.4, “To Start Up a Single Listener Directly.” • Removed Section 2.2.3.5, “To Stop a Single Listener Directly.” **RPC Broker 1.1; XWB*1.1*65 BDK**	Developer: HW Technical Writer: TB

Date	Revision	Description	Authors
01/31/2017	7.0	Tech Edits based on release of RPC Broker Patch XWB*1.1*65: • Reformatted document to follow current documentation standards and style formatting requirements. • Added support for 2-factor authentication (2FA) and Windows Server 2012 R2 as a supported operating system to Section 1.1. • Removed references to the Broker Security Enhancement (BSE) supplemental documentation throughout, since that documentation is being incorporated into the RPC Broker documentation. • Removed references to the IsBackwardCompatibleConnection property in Section 1.1, because it was removed with this patch. • Removed references to support for Auto Signon throughout, since it is in direct conflict with the mandate for 2-factor authentication (2FA); it also breaks with IPv6. The developer has disabled it in the XWB*1.1*65 client code (BDK) but has *not* removed it from Kernel code yet (that will happen in a future patch). • Added reference to 2-factor authentication (2FA) and removed reference to Auto Signon in Section 1.2. Also, removed the “Integrated Auto Signon for Multiple User Sessions” section and sub-sections. • Updated Note references to DLL and **BAPI32.DLL** in Section 1.2. • Updated Figure 2.	Developer: HW Technical Writer: TB

Date	Revision	Description	Authors
		• Removed reference to the RPC Broker Client Agent throughout, since it is used only for Auto Signon and will no longer have any value after May 2020; once all the new applications have rolled out for 2-factor authentication (2FA). Removed (prior) the "RPC Broker Client Agent" section. • Added/Updated Windows registry information, including registry format and example (Figure 4) to Section 2.1.1. • Updated Section 2.1.3 and Table 3. • Removed Caution note referring to RPC Broker 1.0 from Section 2.1.4. • Removed the "What Happened to the Client Manager?" and "What Happened to the VISTA.INI File?" sections, since there are no longer any 16-bit RPC Broker 1.0 applications in the VA. • Added Caution note to Section 2.2.1. • Changed references to "**WINSOCK.DLL**" to "WinSock Application Programming Interface (API)" throughout. • Removed/Deprecated references to the HOSTS file (Section 2.1.4), BROKERSERVER, and localhost throughout, since Windows APIs no longer reference the HOSTS file but are strictly dependent upon DNS. • Removed references to TSharedRPCBroker component and backward compatibility prior to patch	

Date	Revision	Description	Authors
		XWB*1.1*6 in Section 2.2.6. • Added reference to 2-factor authentication (2FA) in Step 2 in Section 3.1. • Added Section 3.3.1 for 2-factor authentication (2FA). • Renamed Section 3.3.2. • Added reference to 2-factor authentication (2FA) in Section 3.3.3, 3.3.5, 4.1, and Table 8 (Step 2). • Updated Section 4.3 to remove reference to the Client Agent and Auto signon. Also, added a reference to 2-factor authentication (2FA). **RPC Broker 1.1; XWB*1.1*65 BDK**	
04/27/2016	6.0	Tech Edits based on release of RPC Broker Patch XWB*1.1*60 (released 06/11/2015): • 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. • Added Figure 5 and Figure 6. • Updated Section 2.1.3. • 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 16. • Updated Section 4.1. • Deleted references to TSharedRPCBroker and TSharedBroker components	Developer: HW Technical Writer: TB

Date	Revision	Description	Authors
		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. **RPC Broker 1.1; XWB*1.1*60 BDK**	

12/04/2013	5.1	Tech Edit:
		• Updated document for RPC Broker Patch XWB*1.1*50 based on feedback from HW.
		• Removed references related to Virgin Installations throughout.
		• Updated file name references throughout.
		• Removed distribution files that are obsolete or no longer distributed throughout.
		• Updated RPC Broker support on the following software:
		  ○ 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 2.1.1 and Figure 3.
		• Updated Figure 7.
			Developer: HW
			Technical Writer: TB

Date	Revision	Description	Authors

		• Updated Section 2.1.2.
		• Updated Section 2.1.3 and Table 3.
		• Updated Section 2.1.4.
		• Updated Table 5.
		• Updated copyright reference.
		• Updated all images for prior Microsoft® Windows operating systems to Windows 7 dialogues.
		• Redacted document for the following information:
		  ○ Names (replaced with role and initials).
		  ○ Production IP addresses and ports.
		  ○ Intranet websites.
		RPC Broker 1.1; XWB*1.1*50 BDK	

07/25/2013	5.0	Tech Edit: • Baselined document. • Updated all styles and formatting to follow current internal team style template. • Updated all organizational references. **RPC Broker 1.1; XWB*1.1*50 BDK**	Developer: HW Technical Writer: TB
08/26/2008	4.2	Updates for RPC Broker Patch XWB*1.1*50: • Added new properties. • Support for Delphi 5, 6, 7, 2005, 2006, and 2007. • Changed references form Patch 47 to Patch 50 where appropriate. **RPC Broker 1.1; XWB*1.1*50 BDK**	Project Manager: JSch Developer: JI SQA: GS Technical Writer: TB

Date	Revision	Description	Authors
07/03/2008	4.1	Updates for RPC Broker Patch XWB*1.1*47: • 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 form Patch 40 to Patch 47 where appropriate. **RPC Broker 1.1; XWB*1.1*47 BDK**	Project Manager: JSch Developer: JI SQA: GS Technical Writer: TB
02/23/2005	4.0	Revised Version for RPC Broker Patches XWB*1.1*35 and 40. Also, reviewed document and edited for the “Data Scrubbing” and the “PDF 508 Compliance” projects. **Data Scrubbing—**Changed all patient/user TEST data to conform to HSD&D standards and conventions as indicated below: • 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. **PDF 508 Compliance—**The final PDF document was recreated and now	Developer: JI Technical Writer: TB

Date	Revision	Description	Authors
		supports 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; XWB*1.1*35 & 40 BDK**	
05/08/2002	3.0	Revised Version for RPC Broker Patch XWB*1.1*26. **RPC Broker 1.1; XWB*1.1*26 BDK**	Developer: JI Technical Writer: TB
04/08/2002	2.0	Revised Version for RPC Broker Patch XWB*1.1*13. **RPC Broker 1.1; XWB*1.1*13 BDK**	Developer: JI Technical Writer: TB
09/--/1997	1.0	Initial RPC Broker Version 1.1 software release. **RPC Broker 1.1**	Developer: JI Technical Writer: TB

Patch Revisions

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

Table of Contents

Revision History ii
List of Figures xi
List of Tables xii
Orientation xiii
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 “Connect To” Dialogue 6
2.1.2 Edit Broker Servers Application 7
2.1.3 Standalone Applications and their Associated Help Files 9
2.1.4 HOSTS File 10
2.2 Server Features 10
2.2.1 RPC Broker Management Menu 10
2.2.1.1 RPC Listener Edit Option 10
2.2.1.2 Start All RPC Broker Listeners Option 11
2.2.1.3 Stop All RPC Broker Listeners Option 11
2.2.1.4 Clear XWB Log Files Option 11
2.2.1.5 Debug Parameter Edit Option 11
2.2.1.6 View XWB Log Option 11
2.2.2 Broker Listeners and Ports 11
2.2.2.1 Obtaining an Available Listener Port—Alpha/VMS Systems 12
2.2.2.2 Obtaining an Available Listener Port—Linux Systems 12
2.2.3 Starting and Stopping Listeners 13
2.2.3.1 To Start All Listeners 13
2.2.3.2 To Configure Listeners for Automatic Startup 13
2.2.3.3 To Stop All Running Listeners 13
2.2.3.4 To Task the XWB LISTENER STARTER Option for System Startup 13
2.2.4 RPC BROKER SITE PARAMETERS File 14
2.2.4.1 Editing the Listener Site Parameters 14
2.2.5 RPC Broker Message Structure 15
2.2.6 Client/Server Timeouts 15
3 Security 17
3.1 Security Features 17
3.2 Validation of Connection Request 17

3.3 Validation of Users **17**
3.3.1 VistA 2-Factor Authentication Dialogue 17
3.3.2 VistA Access/Verify Code Sign-on Dialogue 21
3.3.3 VistA Division Selection Dialogue 22
3.3.4 Users Can Customize VistA Sign-on Dialogue 23
3.3.4.1 Sign-on Properties 24
3.3.5 Change VistA Verify Code Component 26
3.4 Validation of RPCs **27**
3.5 Sample Security Procedures **27**
3.6 Security Features Tasks Summary **28**
4 Troubleshooting **29**
4.1 Test the Broker Using the RPC Broker Diagnostic Program **29**
4.2 Verify and Test the Network Connection **31**
4.3 Signon Delays **32**
4.4 RPC Broker FAQs **32**
Glossary 33
Index 35

List of Figures

Figure 1: Delphi's "Tool Properties" Dialogue—Broker_1_1.chm Entry xviii
Figure 2: VistA RPC Broker System Overview Diagram 5
Figure 3: "Connect To" Dialogue: Server and Port Configuration Selection 6
Figure 4: Sample Registry Information 7
Figure 5: Edit Broker Servers Application—Opened Normally 8
Figure 6: Edit Broker Servers Application—Opened with Administrator Privileges 9
Figure 7: RPC Broker Management Menu Option [XWB MENU] 10
Figure 8: Obtaining an Available Listener Port—Alpha/VMS Systems 12
Figure 9: Obtaining an Available Listener Port—Linux Systems 12
Figure 10: Automatically Starting Listeners when TaskMan is Restarted 14
Figure 11: RPC Listener Edit Option—Sample User Dialogue 14
Figure 12: Sample VistA Application "Signon" Splash Screen 17
Figure 13: Microsoft "Windows Security" Dialogue—Certificate Selection 18
Figure 14: "ActivClient Login" Dialogue—PIN Entry 18
Figure 15: Sample "System Use Notification" Dialogue 20
Figure 16: Sample "VistA Sign-on" Security Dialogue 21
Figure 17: "Select Division" Dialogue—Sample Entries 22
Figure 18: "VistA Sign-on" Dialogue—Properties System Menu 23
Figure 19: "Sign-on Properties" Dialogue 24

Figure 20: Sample “Font” Dialogue 26
Figure 21: “Change VistA Verify Code” Dialogue 26
Figure 22: RPC Broker Connection Diagnostic Application 30

List of Tables

Table 1: Documentation Symbol Descriptions xiv
Table 2: Commonly used RPC Broker Terms xvi
Table 3: Standalone RPC Broker Applications and Associated Help Files 9
Table 4: Listener Site Parameter Entries Descriptions 15
Table 5: Window Position 24
Table 6: Window Size 25
Table 7: Introductory Text Background Color 25
Table 8: Sample Security Procedures 27
Table 9: Security Tasks Summary 28
Table 10: Glossary of Terms and Acronyms 33

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.
- System Administrators—Personnel responsible for regional and local computer management and system security on VistA M Servers.
- Information Security Officers (ISOs)—Personnel 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 (OITFO).

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 (OIT) 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 *Embarcadero Delphi for Windows User Guide*.
GUI	The Graphical User Interface application that is developed for the client workstation.
Host	The term Host is used interchangeably with the term Server.

Term	Description
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** [DILIST] option on the **Data Dictionary Utilities** [DI DDU] menu 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

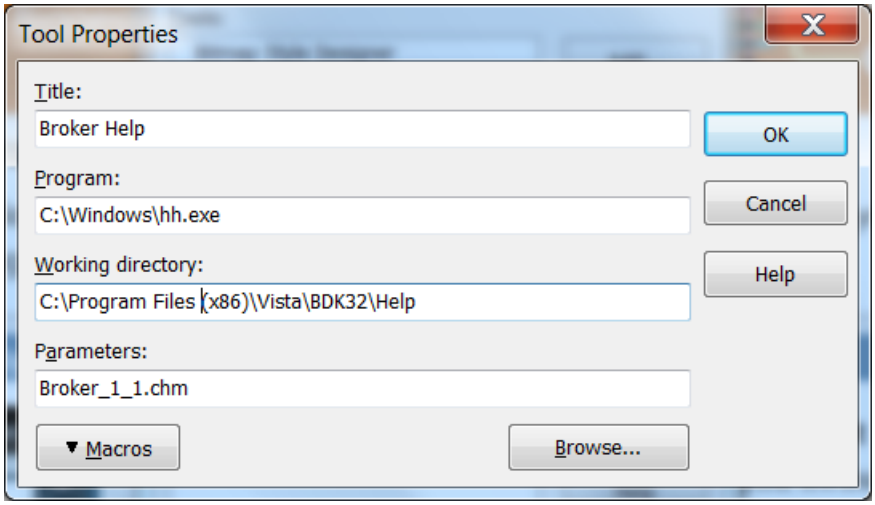
References

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

- *RPC Broker Release Notes*
- *RPC Broker Deployment, Installation, Back-Out, and Rollback (DIBR) 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 Properties" dialogue, to make it easily accessible from within Delphi. To do this, use Delphi's **Tools | Configure Tools** option and create a new 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/

The RPC Broker documentation is located on the VDL at:
https://www.va.gov/vdl/application.asp?appid=23

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, system administrators, 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.

RPC Broker 1.1 provides the following functionality:

- A common communications driver interface that handles the device-specific characteristics of the supported communications protocol.
- An interface component that is 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 [#8994] file).
- Architecture that supports multiple GUI and client front-ends.
- Broker Development Kit (BDK). The BDK provides VistA application developers 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 Server 2012 R2
  - Windows 10
  - Windows 8.1
  - Windows 7

- Supports Active Directory (AD) Credentials—When a user is unable to log onto a workstation with their Personal Identity Verification (PIV) card, the user contacts the Enterprise Service Desk (ESD) to receive a PIV exemption to allow them to log on with their Active Directory (AD) credentials (username and password). This enhanced BDK detects this condition and allows the user to use their AD credentials to secure a SAML token from IAM for logging onto VistA via applications compiled with this version of the BDK. (XWB*1.1*71)
- Supports 2-Factor Authentication (2FA)—TRPCBroker component enables 2-factor authentication (2FA) with Identity and Access Management (IAM) by making a call to the IAM Secure Token Service (STS). The user's Active Directory credentials are exchanged for a Security Assertion Markup Language (SAML) token, which is digitally signed by IAM and contains the authenticated user's identity. The SAML token is passed to the VistA M Server, which validates the digital signature and integrity of the token and identifies the VistA user for server access. (XWB*1.1*65)
- 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)
- 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 Health*e*Vet 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.
- 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 2-factor authentication (2FA) or Access and Verify code information.
- 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* guaranteed to be thread safe.

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

  REF: For more information on troubleshooting the Broker, see the "Troubleshooting" section.

- Supports Delphi versions: 10.3, 10.2, 10.1, 10.0, and XE8.

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 Deployment, Installation, Back-Out, and Rollback 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 2-factor authentication (2FA) or Access and Verify codes. After a successful login, the application is active on both the server and the client.

As you manipulate the interface, your client process is reading and writing data to the server. The reading and writing are 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 [#8994] file). 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 (#8994) file, 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). RPC Broker 1.1 supplies a set of functions providing a Dynamic Link Library (DLL) interface that allows *non*-Delphi applications to conform to the client-side interface of the Broker.

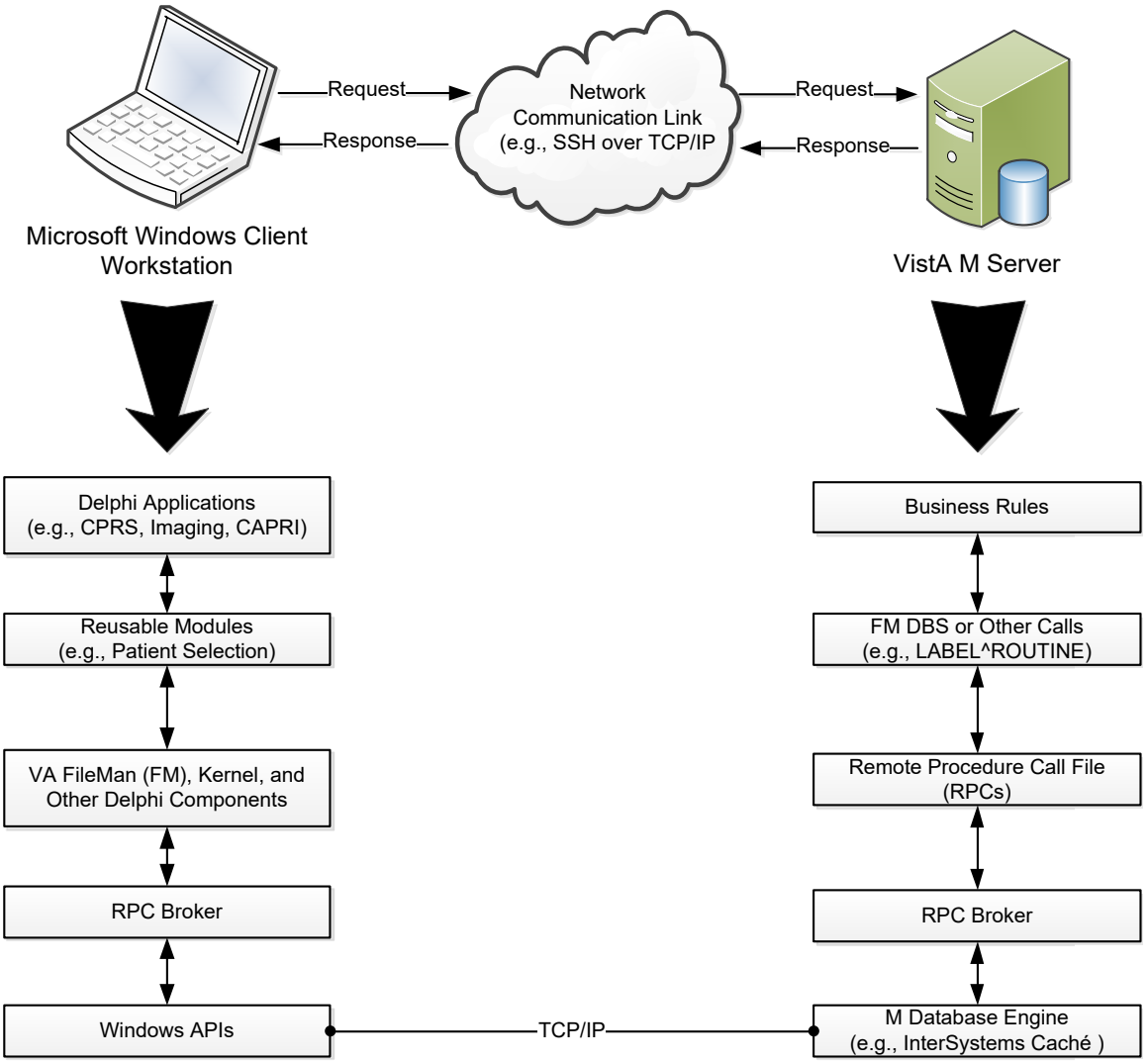
REF: For more specific information about the Broker DLL, see the BDK Online Help (i.e., **Broker_1_1.chm**) or *RPC Broker Developer's Guide*.

NOTE: The **BAPI32.DLL** contains all of the 32-bit Broker DLL functions. It provides an interface to the Broker component.

1.3 System Overview

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

Figure 2: VistA RPC Broker System Overview Diagram



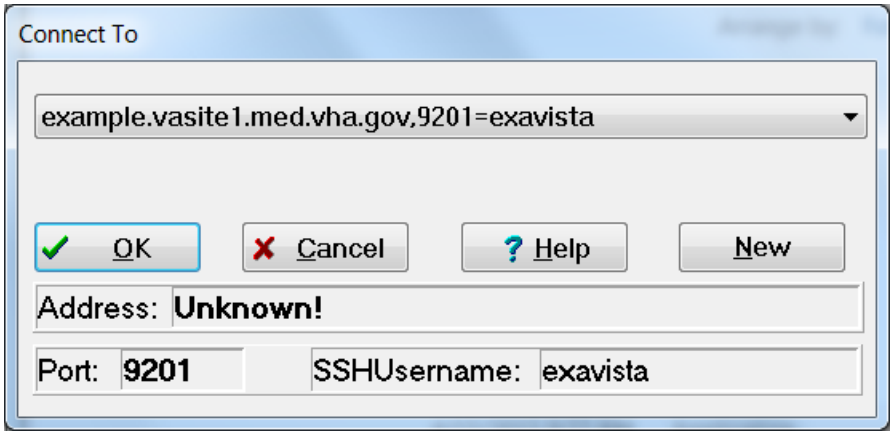
2 System Features

2.1 Client Features

2.1.1 "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 3:

Figure 3: "Connect To" Dialogue: Server and Port Configuration Selection



Delphi VistA client/server applications can use this "Connect To" dialogue with server and port configuration selection options to allow users to do the following:

- 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.

If exactly one server and port entry is defined in the Microsoft® Windows Registry, then the dialogue in Figure 3 is *not* displayed and no user interaction is required. If more than one server and port entry exists, then the dialogue in Figure 3 is displayed and the user chooses to which server they want to connect.

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. The values are stored in either of the following registries:

- Key Name: HKEY_LOCAL_MACHINE\Software\Wow6432Node\Vista\Broker\Servers
- Key Name: HKEY_CURRENT_USER\Software\Vista\Broker\Servers

Entries are of the format:

- Name: Server,ListenerPort
- Type: REG_SZ
- Date: SSHUsername

For example, a connection to a server with the following information would look like Figure 4:

- Address: **r08dhcp017.vha.med.va.gov**
- Port: **19001**
- SSHUsername: **xkgvista**

Figure 4: Sample Registry Information

2.1.2 Edit Broker Servers Application

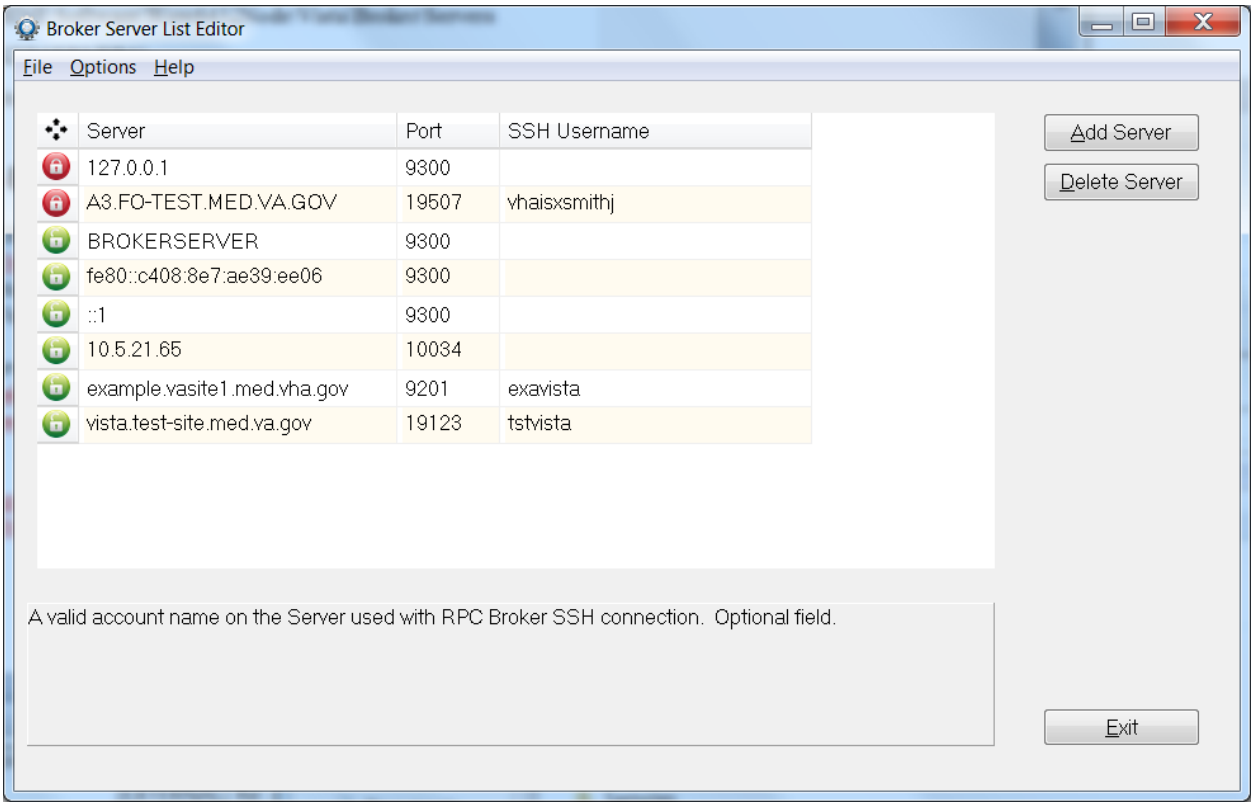
The Edit Broker Servers application (i.e., **ServerList.exe**), previously distributed with earlier versions of the Broker, was replaced by a "beta" **ServerList.exe** in RPC Broker 1.1 (patch XWB*1.1*60). The Edit Broker Servers application provides the means to add, edit, or delete server and port 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:

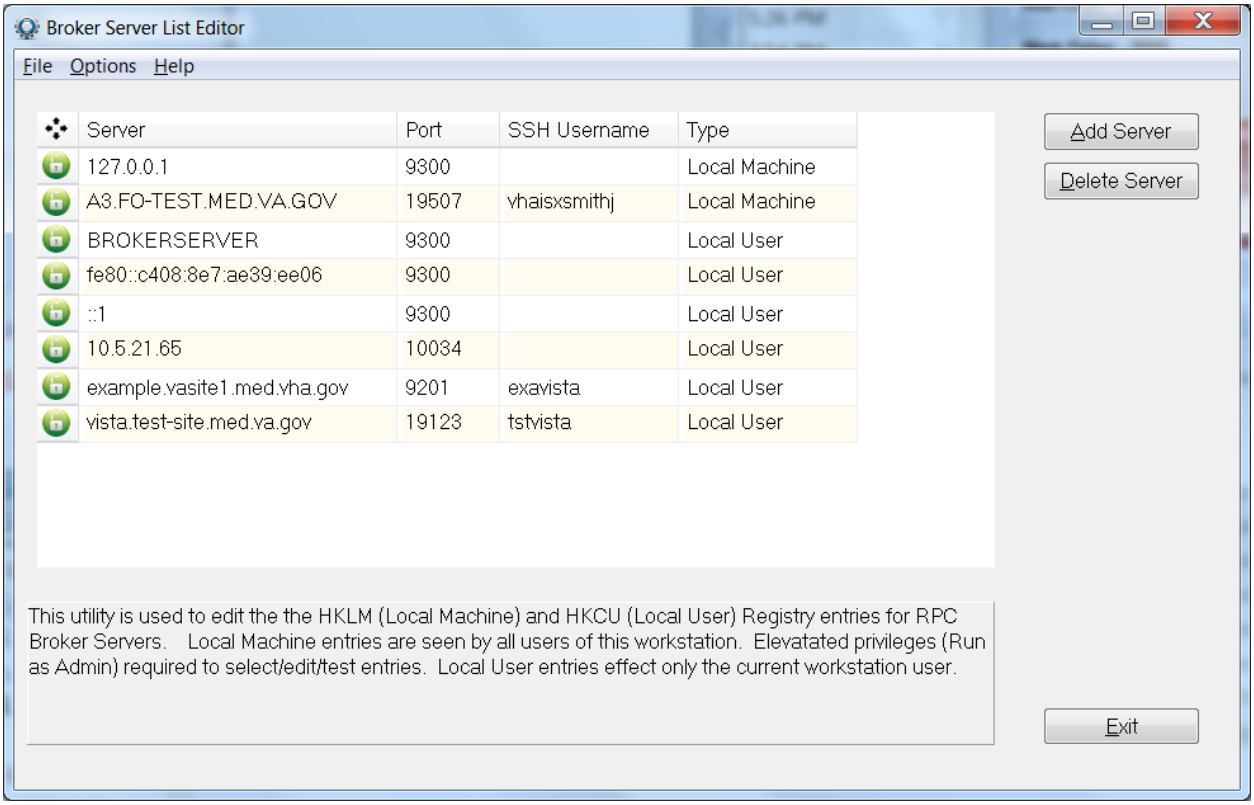
- Entries with a green "open lock" icon are *not* available to other users of the same computer.
- Entries with a red "closed lock" icon are shared and *cannot* be edited.

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



2.1.3 Standalone Applications and their Associated Help Files

The standalone Broker application listed in Table 3, which was distributed with earlier versions of the RPC Broker, has 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 Associated Help Files

Standalone Program	Associated Help File	Location
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 **RPCTEST.EXE** application, see the "Troubleshooting" section.

2.1.4 HOSTS File

The HOSTS file is an ASCII text file that contains a list of the servers and their IP addresses. Microsoft has deprecated the use of the HOSTS file for resolution of server names and IP addresses. The current Microsoft Windows APIs are strictly dependent upon Domain Name Service (DNS) for host name to IP address resolution. RPC Broker code uses the newer Microsoft APIs and no longer supports the use of the HOSTS file.

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 7:

Figure 7: 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
```

CAUTION: On many servers, listeners are configured as an operating system process and *cannot* be edited, started, or stopped using VistA menu options.

For more information regarding the setup of listeners, see the "Setup for XWB LISTENER STARTER Option" section in the *RPC Broker Deployment, Installation, Back-Out, and Rollback Guide*.

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** [XWB LISTENER EDIT] option 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** [XWB LISTENER STARTER] option to automatically start all listeners configured in the RPC BROKER SITE PARAMETERS (#8994.1) file.

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** [XWB LISTENER STOP ALL] option to stop all running listeners configured in the RPC BROKER SITE PARAMETERS (#8994.1) file 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** [XWB LOG CLEAR] option 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** [XWB DEBUG EDIT] option to edit the **XWBDEBUG** parameter defined in the PARAMETER DEFINITION (#8989.51) file and stored in the PARAMETERS (#8989.5) file when set.

2.2.1.6 View XWB Log Option

Use the **View XWB Log** [XWB LOG VIEW] option 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 8: 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 in Figure 8, 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, as shown in Figure 9:

Figure 9: 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 in Figure 9, 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 (#8994.1) file, use the **Start All RPC Broker Listeners** [XWB LISTENER STARTER] option. 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, which 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** [XWB LISTENER STARTER] option, enter **YES** in the CONTROLLED BY LISTENER STARTER field in the RPC BROKER SITE PARAMETERS (#8994.1) file for that listener.

REF: For more information about the RPC BROKER SITE PARAMETERS (#8994.1) file, 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 (#8994.1) file set to automatically start, use the **Stop All RPC Broker Listeners** [XWB LISTENER STOP ALL] option.

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

2.2.3.4 To Task the XWB LISTENER STARTER Option for System Startup

The **Start All RPC Broker Listeners** [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 [#8994.1] file), enter the XWB LISTENER STARTER option in the OPTION SCHEDULING (#19.2) file. Schedule this option with SPECIAL QUEUING set to **STARTUP**. You can do this by using TaskMan's **Schedule/Unschedule Options** [XUTM SCHEDULE], as shown in Figure 10.

Figure 10: Automatically Starting Listeners 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 (#8994.1) file contains one top-level entry, whose **.01** field is a pointer to the DOMAIN (#4.2) file. 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** [XWB LISTENER EDIT] option.

The **RPC Listener Edit** [XWB LISTENER EDIT] 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 11: 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 4:

Table 4: Listener Site Parameter Entries Descriptions

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** [XWB LISTENER STARTER] and **Stop All RPC Broker Listeners** [XWB LISTENER STOP ALL] options to start and stop listeners.)
CONTROLLED BY LISTENER STARTUP	If the listener should be started by the **Start All RPC Broker Listeners** [XWB LISTENER STARTER] option, set this field to **YES**. Otherwise, set it to **NO**.

2.2.5 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 Application Programming Interface (API) 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 (#8994) file 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).

2.2.6 Client/Server Timeouts

The issue of timeouts is complex in a client/server environment. Because the user can be working with applications that rely solely on the client, long periods of time can 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 (e.g., by pressing the PC’s **Reset** button). Prior to patch XWB*1.1*6, 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 component distributed with the latest patch automatically polls. No developer or user intervention is necessary, and this polling activity does *not* affect the application or the user.

The BROKER ACTIVITY TIMEOUT field in the KERNEL SYSTEM PARAMETERS (#8989.3) file controls the length of the timeout. That field was distributed by Kernel Patch XU*8.0*115 with a default value of approximately 3 minutes. By setting the timeout to a duration much shorter than 10 hours, the ghost jobs are eliminated quickly, if the client application is no longer running.

REF: For advice regarding changing the value for this field, see the help for the BROKER ACTIVITY TIMEOUT field.

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 system administrators.

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

1. Client workstations *must* have a valid connection request.
2. Users *must* have valid 2-factor authentication (2FA) or 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 2-Factor Authentication Dialogue

After starting the application, many applications display a splash screen. An example of a VistA application splash screen is shown in Figure 12:

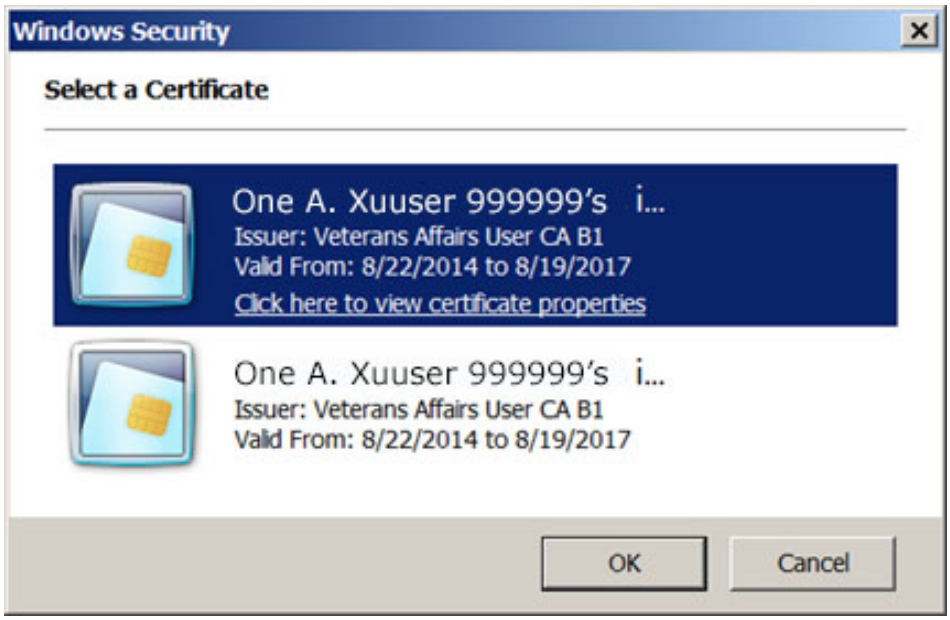
Figure 12: Sample VistA Application "Signon" Splash Screen

The application then opens, and the user is prompted for 2-factor authentication (2FA). The form of authentication can vary depending on the network security implementation at that time.

An example of 2-factor authentication (2FA) follows:

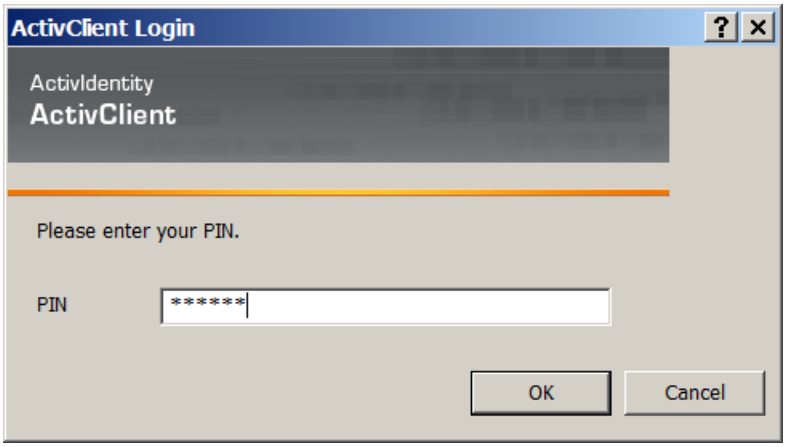
1. After inserting a Smart Card (e.g., Personal Identity Verification [PIV] card), the system displays the available Public Key Infrastructure (PKI) certificates from which to choose, as shown in Figure 13:

Figure 13: Microsoft "Windows Security" Dialogue—Certificate Selection



2. After selecting a valid certificate, the user is prompted to enter a Personal Identification Number (PIN):

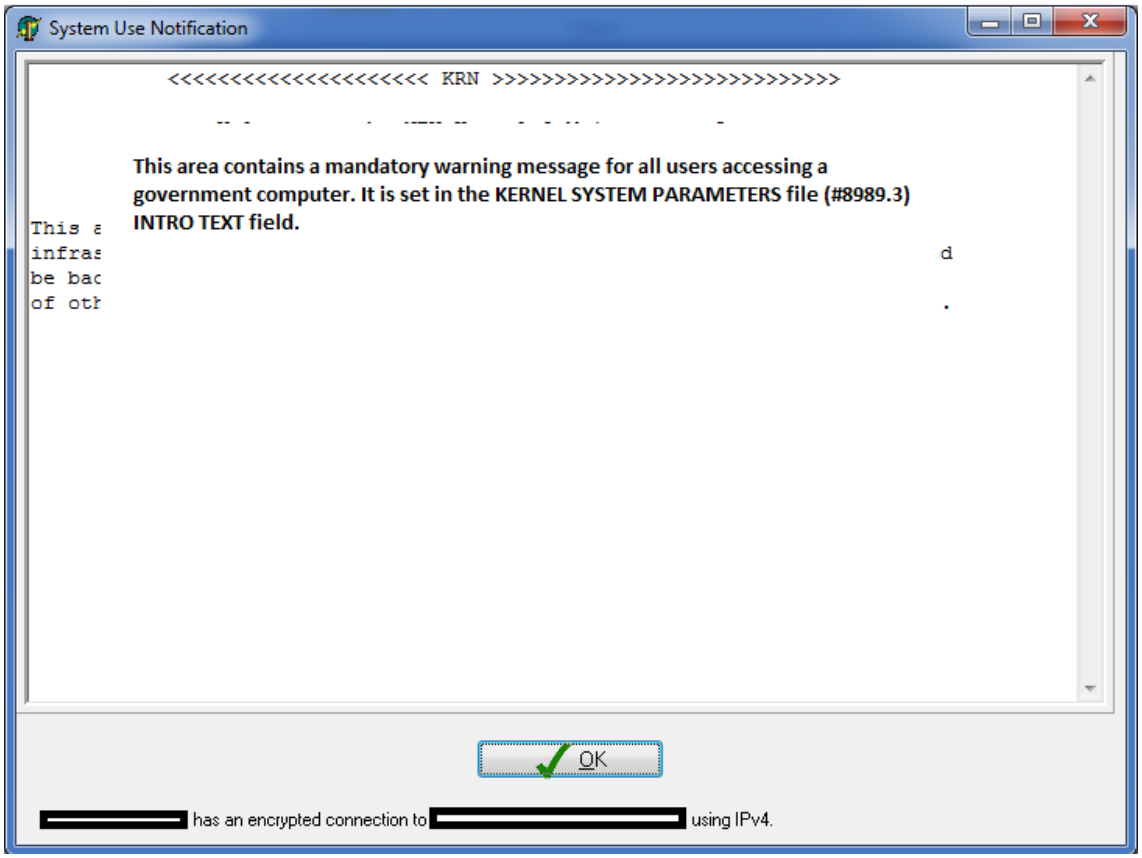
Figure 14: "ActivClient Login" Dialogue—PIN Entry



3. After entering a PIN, there will be a short system delay as the following occurs in the background:
   a. Identity and Access Management (IAM) validates the user credentials with Microsoft Active Directory.
   b. The user's Active Directory credentials are exchanged for a digitally signed Secure Token Service (STS) Security Assertion Markup Language (SAML) token containing several user attributes, including a unique Security ID (SecID).
   c. VistA validates the SAML token for the following information:
      - Digital Signature
      - Integrity
      - Expiration

   If the token is good, it identifies the user based on the SecID and other attributes.
4. A mandatory warning message is then displayed to the user as shown in Figure 15. The message box shows the following information taken from the SAML token), whether or not the connection is encrypted:
   - User Name
   - Server Name
   - Connection Protocol: IPv4 or IPv6

Figure 15: Sample "System Use Notification" Dialogue



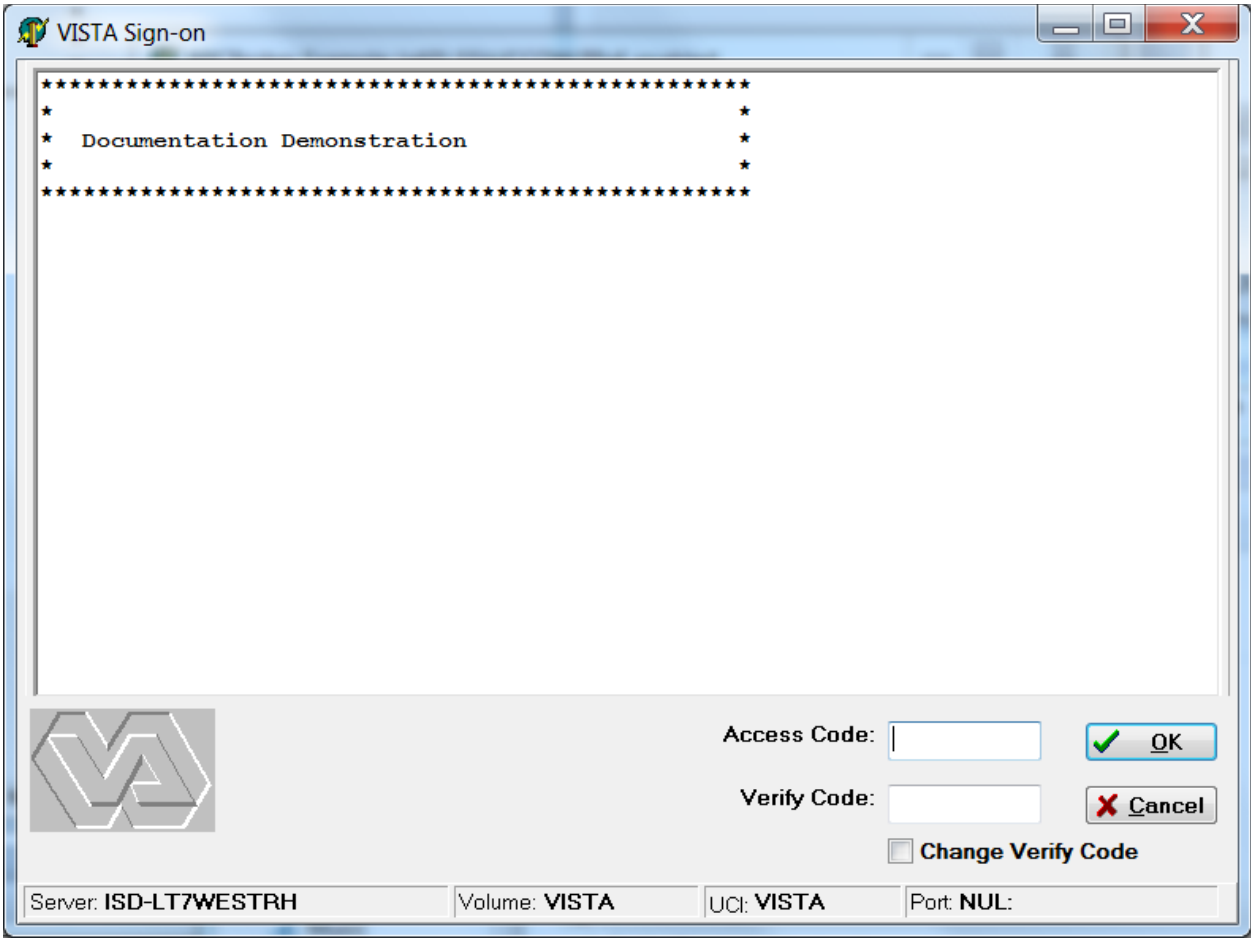
NOTE: When a user is unable to log onto a workstation with their Personal Identity Verification (PIV) card, the user contacts the Enterprise Service Desk (ESD) to receive a PIV exemption to allow them to log on with their Active Directory (AD) credentials (username and password). This enhanced BDK detects this condition and allows the user to use their AD credentials to secure a SAML token from IAM for logging onto VistA via applications compiled with this version of the BDK. (XWB*1.1*71)

3.3.2 VistA Access/Verify Code Sign-on Dialogue

The "VistA Sign-on" dialogue automatically prompts users for their Access and Verify codes if they are *not* already signed on using 2-factor authentication (2FA; Figure 16).

Figure 16 illustrates a sample of the "VistA Sign-on" dialogue integrated with the RPC Broker:

Figure 16: Sample "VistA Sign-on" Security Dialogue



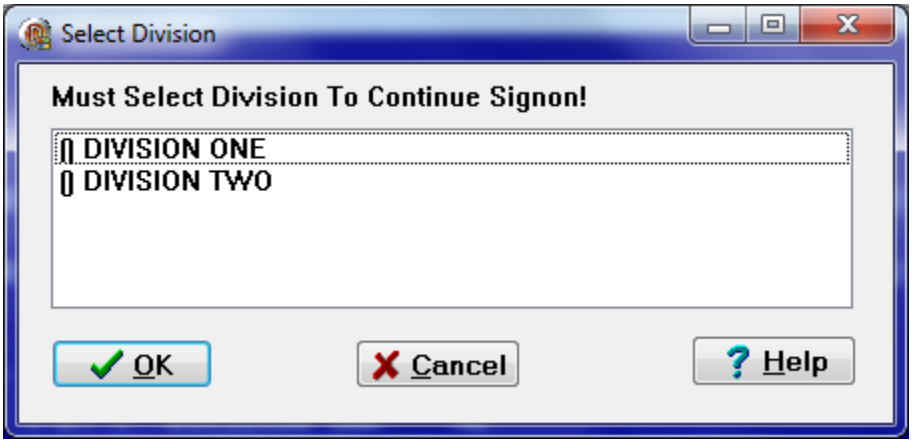
NOTE: RPC Broker 1.1 supports Single Sign-On/User Context (SSO/UC).

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

3.3.3 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 17:

Figure 17: "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 (#19) file. 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 application.
2. The RPC Broker in the client application invokes the "VistA Sign-on" dialogue (Figure 16) 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.

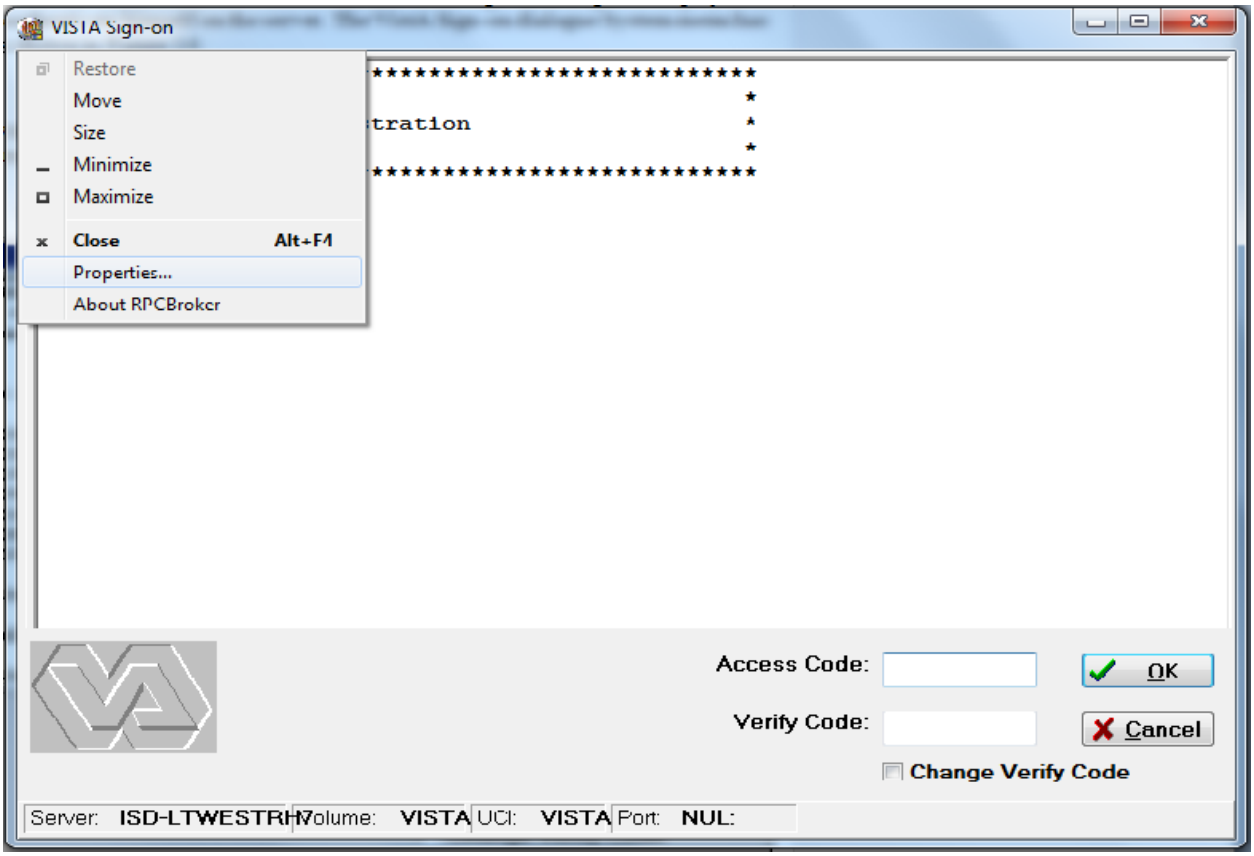
REF: For more information on 2-factor authentication (2FA), Access and Verify codes, or the Kernel signon process in general, see the "Signon/Security" section in the *Kernel 8.0 & Kernel Toolkit 7.3 Systems Management Guide*.

3.3.4 Users Can Customize VistA Sign-on 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 18:

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

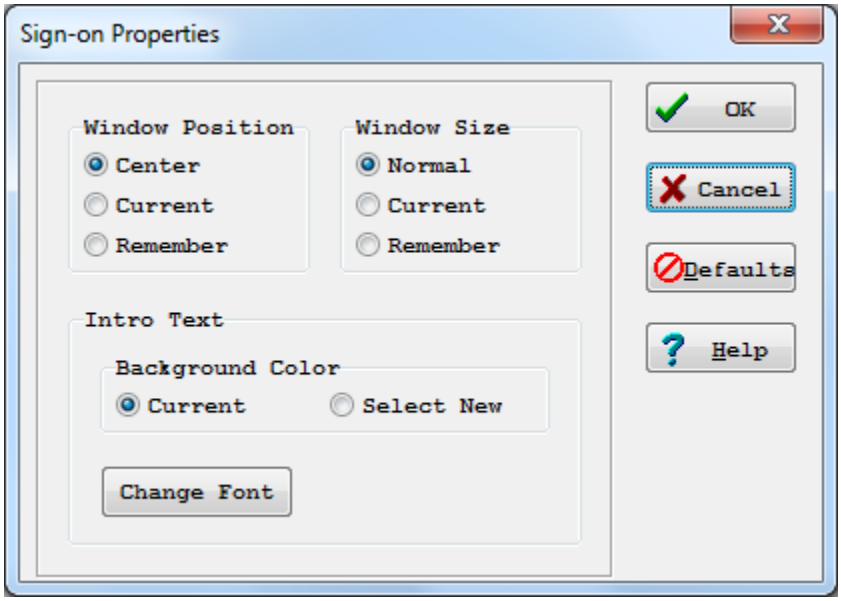
Figure 18: "VistA Sign-on" Dialogue—Properties System Menu



3.3.4.1 Sign-on Properties

When this item is selected, the user is presented with the "Sign-on Properties" dialogue, as shown in Figure 19:

Figure 19: "Sign-on Properties" Dialogue



Using this dialogue (Figure 19), 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.

3.3.4.1.1 Window Position

The "VistA Sign-on" dialogue's window position can be one of the entries in Table 5:

Table 5: 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.

3.3.4.1.2 Window Size

The "VistA Sign-on" dialogue's window size can be one of the entries in Table 6:

Table 6: 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.

3.3.4.1.3 Introductory Text

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

- Background Color:

Table 7: Introductory Text Background Color

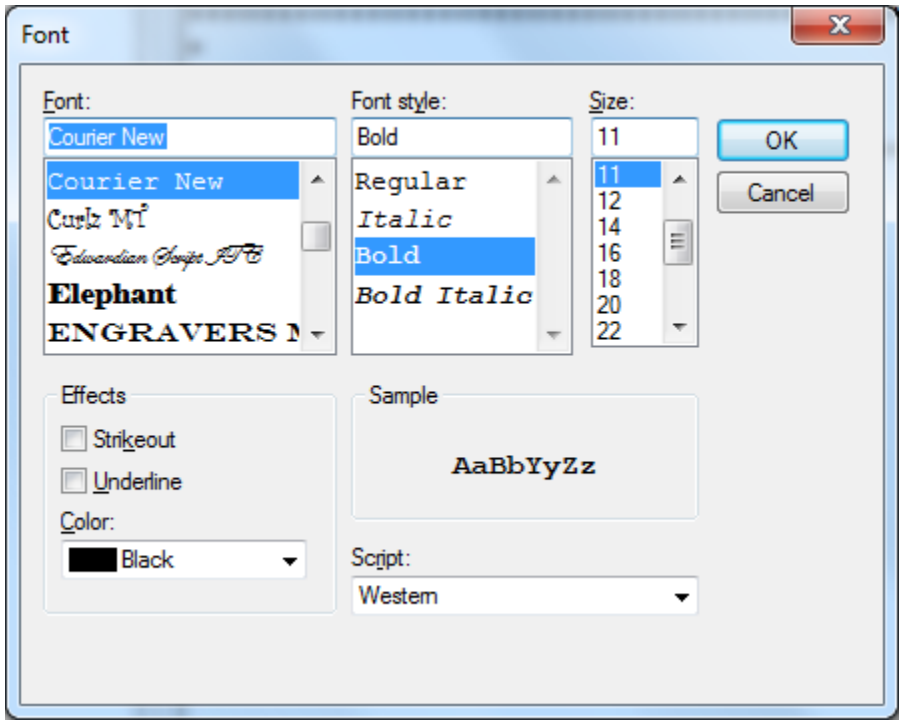
Color	Description
Cream (default)	According to the VA GUI conventions, this is the background color that should be used with text that users *cannot* edit.
White	For clarity and brightness.

- **Font:**

  When users press **Change Font** they are presented with a "Font" dialogue (Figure 20) 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 20: Sample "Font" Dialogue



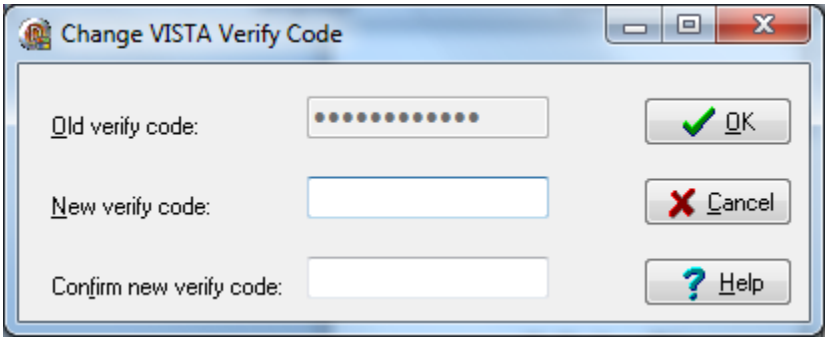
3.3.5 Change VistA Verify Code Component

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 21:

Figure 21: "Change VistA Verify Code" Dialogue



NOTE: The old Verify code appears as asterisks (*) in a grayed-out box if it is available. If a user authenticated with 2-factor authentication (2FA) and a Verify code is expired, then the old Verify code *must* also be provided.

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 16). After signing on, it invokes the dialogue described in Figure 21.
- Roll-and-Scroll environment (existing functionality)—Use the **Edit User Characteristics** [XUSEREDITSELF] option to edit your Verify code.

3.4 Validation of RPCs

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 8:

Table 8: 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 16) on the client using 2-factor authentication (2FA) or 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 9 summarizes required security tasks:

Table 9: 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 system administrators.

CAUTION: For each release of the RPC Broker, the RPC Broker Development Team continuously strives 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 1.1 includes a diagnostic tool for the client workstation (Figure 22). 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 system administrators when trying to determine and/or correct any problems with or to test the Broker.

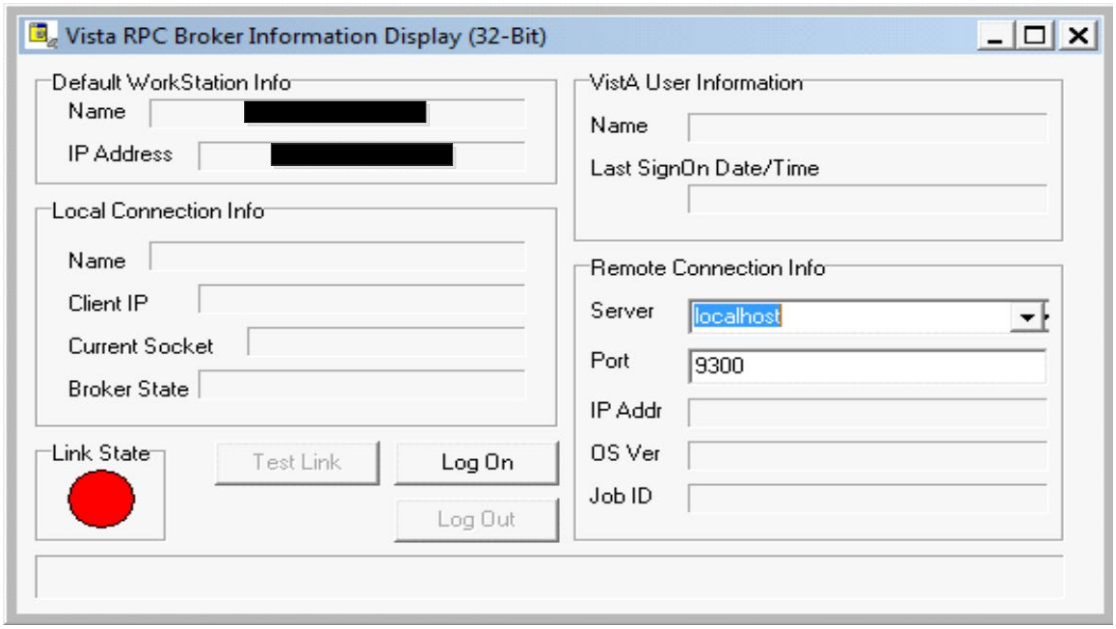
NOTE: This utility has *not yet* been updated to support 2-factor authentication (2FA) or IPv4/IPv6 dual-stack environment testing and has *not* been reviewed for Section 508 conformance.

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 (OS) Version (Ver) 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 application (i.e., **RPCTEST.EXE**), the dialogue in Figure 22 is displayed:

Figure 22: RPC Broker Connection Diagnostic Application



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 Deployment, Installation, Back-Out, and Rollback Guide*).
- Is the server name resolvable using DNS? Current Microsoft® Windows APIs no longer look at the HOSTS file for name resolution but are strictly dependent upon DNS.

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
   ```

   NOTE: "PING" is a way to test connectivity. PING 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. Make sure that Microsoft® Windows is patched to the current version. Install the latest Service Pack or patches.
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 should *not* experience any significant signon delays.

If there are network problems preventing 2-factor authentication (2FA), there will be a delay until the client application times out waiting for an authentication token. After the delay, the user will fail over to Access and Verify code signon.

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

Table 10: Glossary of Terms and Acronyms

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	**D**ynamic **L**ink **L**ibrary. 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: • 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 Interface. A type of display format that enables users to choose commands, initiate programs, and other options by selecting pictorial representations (icons) via a mouse or a keyboard.
ICON	A picture or symbol that graphically represents an object or a concept.
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.

Term	Description
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 (#19) file. 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 (#19) file and can be the running of a routine or the placement of data into a file.
USER ACCESS	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). 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.
USER INTERFACE	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.
WINDOW	An object on the screen (dialogue) that presents information such as a document or message.

REF: For a list of commonly used terms and definitions, see the OIT Master Glossary VA Intranet Website.

For a list of commonly used acronyms, see the VA Acronym Lookup Intranet Website.

Index

2

2-Factor Authentication (2FA), 2, 3, 4, 17, 18, 21, 22, 27, 29, 32

A

Access Code, 3, 4, 17, 21, 22, 27
Acronyms
  Intranet Website, 34
Active Directory, 2, 19
Assumptions, xvii
Authentication
  2-Factor (2FA), 2, 3, 4, 17, 18, 21, 22, 27, 29, 32

B

Background Color, 25
BOX-VOLUME PAIR Field, 15
Broker
  FAQs, 32
  Listeners and Ports, 11
  Message Structure, 15
  Non-Callback Connections, 3
BROKER ACTIVITY TIMEOUT Field, 16
Broker Security Enhancement (BSE), 2
B-type Options, 22

C

Callout Boxes, xv
Changing the VistA Verify Code, 26
Clear XWB Log Files Option, 11
Client/Server Timeouts, 15
Codes
  Access, 3, 4, 17, 21, 22, 27
  Verify, 3, 4, 17, 21, 22, 26, 27
Color
  Background, 25
Commands
  DCL, 12
Commonly Used Terms, xvi
Configuring Listeners, 13
Connect To Dialogue, 6
Connection Request
  Validating, 17
Connections
  Diagnostics, 30
Contents, x
CONTROLLED BY LISTENER STARTER Field, 13
CONTROLLED BY LISTENER STARTUP Field, 15
Customizing the Signon Dialogue, 23

D

Data Dictionary
  Data Dictionary Utilities Menu, xvii
  Listings, xvii
DCL Command, 12
Debug Parameter Edit Option, 11
DI DDU Menu, xvii
Diagnostics
  Connection, 30
DILIST Option, xvii
Disclaimers, xiv
  Software, xiii
Divisions
  VistA Division Selection Dialogue, 22
DLL, 1, 4
  WinSock Application Programming Interface (API), 15
Documentation
  Revisions, ii
  Symbols, xiv
Documentation Conventions, xiv
Documentation Navigation, xvi
DOMAIN (#4.2) File, 14

E

Edit Broker Servers Application, 7
Edit User Characteristics Option, 27
Editing the Listener Site Parameters, 14

F

FAQs, 32
Features
  Security, 17
  Server, 10
Fields
  BOX-VOLUME PAIR, 15
  BROKER ACTIVITY TIMEOUT, 16
  CONTROLLED BY LISTENER STARTER, 13
  CONTROLLED BY LISTENER STARTUP, 15
  PORT, 15
  SPECIAL QUEUING, 13
  STATUS, 15
Figures, xi
Files
  DOMAIN (#4.2), 14
  Help, 9
  HOSTS, 10
  KERNEL SYSTEM PARAMETERS (#8989.3), 16
  OPTION (#19), 22
  OPTION SCHEDULING (#19.2), 13
  PARAMETER DEFINITION (#8989.51), 11
  PARAMETERS (#8989.5), 11
  REMOTE PROCEDURE (#8994), 1, 4, 15
  RPC BROKER SITE PARAMETERS (#8994.1), 3, 11, 13, 14
Font, 25
Frequently Asked Questions, 32

G

Glossary, 33
  Intranet Website, 34

H

Help
  At Prompts, xvii
  Online, xvii
  Question Marks, xvii
Help Files, 9
History
  Revisions, ii
HKEY_CURRENT_USER\Software\Vista\Broker\Servers Registry, 6
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Vista\Broker\Servers Registry, 6
Home Pages
  Acronyms Intranet Website, 34
  Adobe Website, xix
  Glossary Intranet Website, 34
  RPC Broker Website, xviii
  VA Software Document Library (VDL) Website, xix
    RPC Broker, xix
HOSTS File, 10
How Does It All Work?, 3
How to
  Obtain Technical Information Online, xvii
  Use this Manual, xiii

I

IAM, 2
Identity and Access Management, 2
Identity and Access Management (IAM), 19
Intended Audience, xiii
Introduction, 1
Introductory Text, 25

K

KERNEL SYSTEM PARAMETERS (#8989.3) File, 16

L

LAN, 3, 33
List File Attributes Option, xvii
Listeners
  Configuring, 13
  Starting, 13
  Stopping, 13
  Tasking, 13
Listeners and Ports, 11

M

Menu for System Managers, 10
Menu Manager, 22
Menus
  Data Dictionary Utilities, xvii
  DI DDU, xvii
  RPC Broker Management Menu, 10
  XWB MENU, 10
Message Structure, 15
Microsoft Windows Registry, 6
Multi-Instances Support, 3

N

Network Connection, 31
Non-Callback Connections, 3

O

Obtaining
  Available Listener Port
    Alpha/VMS Systems, 12
    Linux Systems, 12
  Data Dictionary Listings, xvii
Online
  Documentation, xvii
  Technical Information, How to Obtain, xvii
OPTION (#19) File, 22
OPTION SCHEDULING (#19.2) File, 13
Options
  B-type, 22
  Clear XWB Log Files, 11
  Data Dictionary Utilities, xvii
  Debug Parameter Edit, 11
  DI DDU, xvii
  DILIST, xvii
  Edit User Characteristics, 27
  List File Attributes, xvii
  RPC Broker Management Menu, 10
  RPC Listener Edit, 10, 14
  Schedule/Unschedule Options, 13
  Start All RPC Broker Listeners, 11, 13, 15
  Stop All RPC Broker Listeners, 11, 13, 15
  View XWB Log, 11
  XUSEREDITSELF, 27
  XUTM SCHEDULE, 13
  XWB DEBUG EDIT, 11
  XWB LISTENER EDIT, 10, 14
  XWB LISTENER STARTER, 11, 13, 15
  XWB LISTENER STOP ALL, 11, 13, 15
  XWB LOG CLEAR, 11
  XWB LOG VIEW, 11
  XWB MENU, 10
Orientation, xiii
Overview, 1
  System Diagram, 5

P

PARAMETER DEFINITION (#8989.51) File, 11
PARAMETERS (#8989.5) File, 11
Patches
  Revisions, ix
PING, 31
PORT Field, 15
Ports and Listeners, 11
Position
  Window, 24
Product Support (PS)
  Anonymous Directories, xix
Programs
  RPCTEST.EXE, 9, 29, 30
PS
  Anonymous Directories, xix

Q

Question Mark Help, xvii

R

Registries
  HKEY_CURRENT_USER\Software\Vista\Broker\Servers, 6
  HKEY_LOCAL_MACHINE\Software\Wow6432Node\Vista\Broker\Servers, 6
Registry, 6
REMOTE PROCEDURE (#8994) File, 1, 4, 15
Revision History, ii
  Documentation, ii

Patches, ix
RPC Broker
Diagnostic Program
How to test the Broker, 29
FAQs, 32
Website, xviii
RPC Broker Management Menu, 10
RPC BROKER SITE PARAMETERS (#8994.1) File, 3, 11, 13, 14
RPC Listener Edit Option, 10, 14
RPCTEST.EXE, 9, 29, 30

S

SAML
Token, 2
SAML Token, 19
Schedule/Unschedule Options Option, 13
Secure Token Service (STS), 2
Security, 17
Change VistA Verify Code Component, 26
Features, 17
Sample Security Procedures, 27
Signon Dialogue
Customizing, 23
Summary of Tasks, 28
Validating Connection Request, 17
Validating Users, 17
Security ID (SecID), 19
Server
Features, 10
Signon
Delays, 32
Dialogue
Customizing, 23
Sample, 21
Silent Logons, 3
Single Sign-On/User Context (SSO/UC), 2
Size
Window, 25
Software Disclaimer, xiii
SPECIAL QUEUING Field, 13
Standalone Applications and their Associated Help Files, 9
Start All RPC Broker Listeners Option, 11, 13, 15
Starting Listeners, 13
STATUS Field, 15
Stop All RPC Broker Listeners Option, 11, 13, 15
Stopping Listeners, 13
Support
Anonymous Directories, xix
Support for Secure Shell (SSH), 2, 6, 7
Symbols
Found in the Documentation, xiv
System
Overview Diagram, 5

T

Table of Contents, x
Tables, xii
Tasking Listeners, 13
TCP/IP, 3, 4, 31
Test the Broker Using the RPC Broker Diagnostic Program, 29
Text
Introductory, 25
Timeouts, 15
Tokens
SAML, 2, 19
Secure Token Service (STS), 2
Troubleshooting, 29
Network Connection, 31
RPC Broker Diagnostic Program, 29
Signon Delays, 32

U

URLs
Acronyms Intranet Website, 34
Adobe Website, xix
Glossary Intranet Website, 34
RPC Broker Website, xviii
VA Software Document Library (VDL)
Website, xix
RPC Broker, xix
Users Can Customize VistA Sign-on Dialogue, 23

V

VA Software Document Library (VDL)
  Website, xix
    RPC Broker, xix
Validating
  Connection Request, Security, 17
  Users, Security, 17
Verify and Test the Network Connection, 31
Verify Code, 3, 4, 17, 21, 22, 26, 27
  Changing, 26
View XWB Log Option, 11
VistA 2-Factor Authentication Dialogue, 17
VistA Division Selection Dialogue, 22
VistA Sign-on Dialogue
  Access/Verify Codes, 21

W

Websites
  Acronyms Intranet Website, 34
  Adobe Website, xix
  Glossary Intranet Website, 34
  RPC Broker, xviii
  VA Software Document Library (VDL)
    Website, xix
    RPC Broker, xix
Window
  Position, 24
  Size, 25
Windows Registry, 6
WinSock Application Programming
  Interface (API), 15

X

XUSEREDITSELF Option, 27
XUTM SCHEDULE Option, 13
XWB DEBUG EDIT Option, 11
XWB LISTENER EDIT Option, 10, 14
XWB LISTENER STARTER Option, 11,
  13, 15
XWB LISTENER STOP ALL Option, 11,
  13, 15
XWB LOG CLEAR Option, 11
XWB LOG VIEW Option, 11
XWB MENU, 10