RPC Broker 1.1

Release Notes

June 2015

Department of Veterans Affairs (VA)

Office of Information and Technology (OI&T)

Enterprise Program Management Office (EPMO)

Revision History

Documentation Revisions

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

Date	Revision	Description	Authors
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 Section 1.1. • Added Section 2.1. • Added Section 3.1.1. • Updated Section 3.1.2. • Deleted Section 3.1.4, "Full Backward Compatibility with Broker 1.0", since there is no means of testing this. The only Broker 1.0 application was PCMM, and the most recently released PCMM version no longer uses Broker 1.0. • Updated Section 3.2.1. • Updated Section 4. • Updated Section 4.2; deleted references to the TSharedBroker and TSharedRPCBroker components. • Deleted Section 4.2.3, "TSharedBroker" and Section 4.2.4, "TSharedRPCBroker." • Updated Section 4.2.6.2. • Updated Section 4.2.7.2. • Updated Section 4.2.7.3. • Updated Figure 1. • Added Note to Section 4.5. • Updated Sections 5.1 and 5.2 for Broker Help file references. • Updated Figure 2. • Updated references to show RPC Broker Patch XWB*1.1*60 supports Delphi XE7, XE6, XE5, and XE4 throughout. • Updated help file references from	• Developer H. W. • Technical Writer: T. B.

Date	Revision	Description	Authors
		"BROKER.HLP" to "Broker_1_1.chm" throughout. **RPC Broker 1.1**	
12/04/2013	5.1	Tech Edit: • Updated document for RPC Broker Patch XWB*1.1*50 based on feedback from H Westra. • 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: o Microsoft® XP and 7 (operating system) throughout. o Microsoft® Office Products 2010 throughout. o Changed references from "Borland" to "Embarcadero" and updated support for Delphi Versions XE5, XE4, XE3, and XE2 throughout. • Updated Section 1.1: o Supports Secure Shell (SSH). o Supports Broker Security Enhancement (BSE). o TContextorControl component. • Added Section 2.1. • Updated Section 3.1.1. • Updated Section 3.2.1. • Deleted Section 3.2.2., "Edit Broker Servers Program," because this application does not function on Windows 7 due to added security. An alternative is still being developed. • Updated Section 4.1. • Updated Section 4.2.1.2. • Added Section 4.2.4. • Updated Figure 2. **RPC Broker 1.1**	• Developer: H. W. • Technical Writer: T. B.
07/25/2013	5.0	Tech Edit: • Baselined document.	• Developer: H. W. • Technical Writer: T.

Date	Revision	Description	Authors
		• Updated all styles and formatting to follow current internal team style template. • Updated all organizational references. **RPC Broker 1.1**	B.
07/06/10	3.2	Updates for RPC Broker Patch XWB*1.1*50 (client-side only patch): • Added support for SSH for Attachmate Reflections (see Section 3.1.1). • Wrapped CCOW User Context into the primary TRPCBroker component and deleting the TCCOWRPCBroker component (see Section 4.2.1.2). • Support for Delphi 5.0, 6.0, 7.0, 2005, 2006, 2007, 2008, 2009, and 2010. • Changed references form Patch 47 to Patch 50 where appropriate. • Reformatted document to follow current *PD National Documentation Standards* and current style guidelines. **RPC Broker 1.1**	RPC Broker Development Team Oakland Office of Information and Technology Field Office (OI&TFO)
07/03/08	3.1	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**	• Oakland OI&TFO Development Team • Technical Writer: T. B.
02/22/05	3.0	Revised Version for RPC Broker Patch XWB*1.1*40 and previous undocumented patch updates. **RPC Broker 1.1**	• Oakland OI&TFO Development Team • Technical Writer: T. B.
02/19/02	2.0	Revised Version for RPC Broker Patch XWB*1.1*13. **RPC Broker 1.1**	• Oakland OI&TFO Development Team • Technical Writer: T. B.

Date	Revision	Description	Authors
09/97	1.0	Initial RPC Broker Version 1.1 software release. **RPC Broker 1.1**	• Oakland OI&TFO • Technical Writer: T. B.

Patch Revisions

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

Contents

1 RPC Broker 1.1 Release Notes

1.1 Overview

The Veterans Health Information Systems and Technology Architecture (VistA) Remote Procedure Call (RPC) Broker (also referred to as "Broker") 1.1 Patch XWB*1.1*60 is now available. This enhanced Broker software has the following functionality/features:

- Supports IPv4/IPv6 Dual-Stack Environment—TRPCBroker component uses WinSock 2.2 Application Programmer Interfaces that support network connections using Internet Protocol (IP) version 4 and/or IP version 6. IPv6 is a protocol designed to handle the growth rate of the Internet and to cope with the demanding requirements of services, mobility, and end-to-end security. (XWB*1.1*60)
- Supports Secure Shell (SSH)—TRPCBroker component enables 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)
- Supports Broker Security Enhancement (BSE)—TRPCBroker component enables visitor access to remote sites using authentication established at a home site.(XWB*1.1*50)
- Supports Single Sign-On/User context (SSO/UC)—TCCOWRPCBroker component enabled Single Sign-On/User Context (SSO/UC) in CCOW-enabled applications. (XWB*1.1*40)
- Supports *Non*-Callback Connections—By default 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. (XWB*1.1*35)
- Supports Silent Logon capabilities—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. (XWB*1.1*13)
- Documents Deferred RPCs and Capability to Run RPCs on a Remote Server.
- Supports Multi-instances of the RPC Broker—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 it is believed that there should be no problems, the RPC Broker is *not yet* guaranteed to be thread safe.
- Provides updated components, properties, methods, and types.
- Separates Design-time and Run-time Packages—BDK contains separate run-time and design-time packages. (XWB*1.1*14)
- Supports Delphi XE7, XE6, XE5, and XE4. (XWB*1.1*60)
- Operates in a 32-bit Microsoft® Windows environment.

RPC Broker 1.1 also includes the Broker Development Kit (BDK). The BDK provides VistA application developers with the following features:

- Capability to create and implement client/server technology in the 32-bit Microsoft® Windows environment using the Broker component (e.g., create Delphi-based client/server VistA applications with Graphical User Interfaces [GUI]).
- Support for Commercial Off-the-Shelf (COTS) and Hybrid Open System Technology (HOST) client/server software using the Broker Dynamic Link Library (DLL).

RPC Broker 1.1 (fully patched) provides programmers with the capability to develop VistA client/server software using the following RPC Broker Delphi components in the 32-bit environment (listed alphabetically):

- TCCOWRPCBroker
- TRPCBroker
- TXWBRichEdit
- TContextorControl

NOTE: These RPC Broker components wrap the functionality of the Broker resulting in a more modularized and orderly interface. Those components derived from the original TRPCBroker component, inherit the TRPCBroker properties and methods.

1.2 New Features and Enhancements of the RPC Broker

RPC Broker 1.1 client/server interface provides the following features and enhancements categorized by user type:

- **End Users** (e.g., clinicians)
- **System Managers** (e.g., IRM personnel)
- **Developers** (e.g., Programmers developing VistA client/server programs in the 32-bit Microsoft® Windows environment)

2 End Users—Features & Enhancements

2.1 Support for IPv4/IPv6 Dual-Stack Environment

RPC Broke upgraded Windows APIs from WinSock 1.1 IPv4 to WinSock 2.2 IPv4/IPv6 dual-stack. Applications compiled with the latest RPC Broker will be protocol independent and able to connect to both IPv4 and IPv6 VistA servers. (XWB*1.1*60)

IPv6 is a protocol designed to handle the growth rate of the Internet and to cope with the demanding requirements of services, mobility, and end-to-end security. A Federal Chief Information Office (CIO) "Transition to IPv6" memo released in September of 2010 requires agencies to continue their IPv6 transition efforts and has established specific milestones associated with enabling an IPv6 operational capability by the end of FY2014.

2.2 Support for Secure Shell (SSH) Tunneling

RPC Broker 1.1 supports Secure Shell (SSH) tunneling; a need expressed by the Veterans Health Administration (VHA) information systems user community to provide secure data transfer between the client and the VistA M Server. Support is provided for the Attachmate® Reflections terminal emulator software using SSH tunneling for clients within the VA, and support is provided for PuTTY Link (Plink) for secure channels for clients outside the VA. (XWB*1.1*50)

2.3 Support for Single Signon/User Context (SSO/UC)

RPC Broker 1.1 supports single sign-on (SSO) service with interfaces to VistA and *non*-VistA systems by using the TCCOWRPCBroker component; a need expressed by the Veterans Health Administration (VHA) information systems user community. This allows users to authenticate and sign on to multiple applications that are CCOW-enabled and Single Signon/User Context (SSO/UC)-aware using a single set of credentials, which reduces the need for multiple ID's and passwords in the VistA clinician desktop environment. (XWB*1.1*40)

The TCCOWRPCBroker component allows VistA application developers to make their applications CCOW-enabled and SSO/UC-aware with all of the client/server-related functionality in one integrated component. Using the TCCOWRPCBroker component, an application can share User Context stored in the CCOW Context Vault.

Thus, when a VistA CCOW-enabled application is recompiled with the TCCOWRPCBroker component and other required code modifications are made, that application would then become SSO/UC-aware and capable of single sign-on (SSO).

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 VHA Software Documentation Library (VDL).

2.4 Support for Silent Logon

RPC Broker 1.1 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. (XWB*1.1*13)

The BDK provides two types of Silent Login:

- **Access/Verify Code-based—**Uses Access and Verify codes provided by the application. This type of Silent Login may be necessary for an application that runs as a background task and repeatedly signs on for short periods. Another case would be for applications that are interactive with the user, but are running under conditions where they cannot provide a standard dialogue window, such as that used by the Broker to request Access and Verify codes. Examples might be applications running on handheld devices or within a browser window.
- **Token-based—**Uses a token obtained by one application that is passed along with other information as a command line argument to a second application that it is starting. The token is obtained from the VistA server and remains valid for about twenty (20) seconds. When the newly started application sends this token during login the server identifies the same user and completes the login.

Due to the various conditions under which Silent Logins might be used, it was also necessary to provide options to the applications on error handling and processing. Applications that run as system services will crash if they attempt to show a dialogue box. Similarly, applications running within Web browsers are not permitted to show a dialogue box or to accept windows messages. Properties have been provided to permit the application to handle errors in a number of ways.

As a part of the Silent Login functionality, the TVistaUser class providing basic user information was added. This class is used as a property by the TRPCBroker class and is filled with data following completion of the login process. This property and its associated data are available to all applications, whether they are using a Silent Login or not.

3 System Managers—Feature & Enhancements

3.1 On the Server

3.1.1 IPv6 Support

IPv4/IPv6 dual-stack support on the VistA server is handled by VistA Kernel software. No RPC Broker changes are needed in VistA to enable IPv6 on a Broker listener. However, VistA routines used to test a Broker listener configuration currently test only IPv4, and will be upgraded to support IPv4/IPv6 dual-stack environments in a future patch. (XWB*1.1*60)

3.1.2 SSH Support

RPC Broker 1.1 provides Secure Shell (SSH) support. Attachmate® Reflections terminal emulator software using SSH tunneling is used within the VA to provide secure data transfer between the client and the VistA M Server. SSH tunneling is also supported for PuTTY Link (Plink) for those using VistA outside of the VA. (XWB*1.1*50)

For SSH tunneling using Attachmate® Reflection, "SSH" is set as a command line option or as a property within the application (set to Attachmate® Reflection). SSH is set to true if either of the following command line parameters are set:

- SSHPort=portnumber (to specify a particular port number – if not specified, it will use the port number for the remote server)
- SSHUser=username (for the remote server, where username is of the form *xxxvista*, where the *xxx* is the station's three letter abbreviation)

For SSH tunneling using Plink.exe, "PLINK" is set as a command line option or as a property within the application (set to Plink). SSH is set to true if the following command line parameter is set:

SSHpw=password

3.1.3 CCOW—Management of Single Sign-On/User Context (SSO/UC)

3.1.3.1 Disabling SSO/UC

For sites whose policy is *not* to allow the kinds of SSO-based logins supported by SSO/UC, the User Context-based SSO can be disabled by doing either of the following:

- Mark the User subject as "unshared" in the Sentillion Vergence Context Vault so that the User subject instance is kept separate for all application instances. This is how the Sentillion Vergence Context Vaults were initially configured when VHA first procured them for Patient Context (i.e., User Context was specifically disabled).
- **Do *not* grant secure access in the Sentillion Vergence Context Vault to the application passcode used by the login components.** Without the application passcode, the login components *cannot* establish a secure binding to the User Context. This failure triggers a standard, non-SSO login process:
    1. The login component does not find a User Context.
    2. The login component prompts the user for their Access and Verify code credentials.
    3. The application logs in; and no User Context is set.

3.1.3.2 Kernel CCOW Login Token Expiration

The Kernel CCOW login token is valid from a minimum of 600 seconds to a maximum of 28,800 seconds (i.e., 10 minutes to 8 hours) from when the user first authenticated via Kernel on the VistA M Server. The default value is 5,400 seconds (i.e., 1.5 hours). This default value is a compromise between wanting to provide as rapid a Kernel CCOW login token expiration as possible for security reasons, versus the need for a SSO session to last long enough in order to be useful to the user.

To change the expiration time, IRM can change the value stored in the CCOW TOKEN TIMEOUT field (#30.1) in the KERNEL SYSTEM PARAMETERS file (#8989.3).

3.1.4 Management of Silent Logon

Control of the Silent Logon 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).

3.2 On the Client

3.2.1 32-Bit Processing

RPC Broker 1.1 operates in a 32-bit Microsoft® Windows environment (i.e., client workstations running Microsoft® Windows 7, 8.1, or 10 operating systems).

4 Developers—Features & Enhancements

This section highlights some of the major changes made to the RPC Broker 1.1 since its original release (patch references are included where applicable):

- 32-Bit Processing/Delphi
- RPC Broker Components
  - Classes Added
  - Library Methods Added/Modified
  - Properties Added
  - Types Added/Modified
- Design-time and Run-time Packages
- Modified GetServerInfo Function
- Updated Dynamic Link Library (DLL) Interface
- Source Code Availability

4.1 32-Bit Processing/Delphi

RPC Broker 1.1 Broker Development Kit (BDK) operates in a 32-bit Microsoft® Windows environment only (i.e., client workstations running Microsoft® Windows 7 or 8.1 operating systems). The current version does *not yet* support development in a 64-bit environment.

NOTE: RPC Broker 1.1 supports Delphi XE7, XE6, XE5, and XE4.

4.2 RPC Broker Components

RPC Broker 1.1 (fully patched) provides programmers with the capability to develop VistA client/server software using the following RPC Broker Delphi components in the 32-bit environment (listed alphabetically):

- TCCOWRPCBroker
- TRPCBroker
- TXWBRichEdit
- TContextorControl

NOTE: These RPC Broker components wrap the functionality of the Broker resulting in a more modularized and orderly interface. Those components derived from the original TRPCBroker component, inherit the TRPCBroker properties and methods.

REF: For a complete description of the RPC Broker components, properties, and methods, see the BDK Online Help (i.e., Broker_1_1.chm) or *RPC Broker Developer's Guide*.

4.2.1 TCCOWRPCBroker

4.2.1.1 TCCOWRPCBroker Component Added

The TCCOWRPCBroker component allows VistA application developers to make their applications CCOW-enabled and Single Sign-On/User Context (SSO/UC)-aware with all of the client/server-related functionality in one integrated component. Using the TCCOWRPCBroker component, an application can share User Context stored in the CCOW Context Vault. (XWB*1.1*40)

Thus, when a VistA CCOW-enabled application is recompiled with the TCCOWRPCBroker component and other required code modifications are made, that application would then become SSO/UC-aware and capable of single sign-on (SSO).

4.2.1.2 CCOW User Context Wrapped into the Primary TRPCBroker Component

RPC Broker 1.1 wraps CCOW User Context into the primary TRPCBroker component, so that if the Contextor property is set, then CCOW User Context will be used. This means that there is no need to have a separate TCCOWRPCBroker component. (XWB*1.1*50)

NOTE: All of the functionality used by and for the TCCOWRPCBroker component is still present, but it is now part of the regular TRPCBroker component.

4.2.2 TRPCBroker

The original TRPCBroker Delphi component provides Delphi developers with an easy, object-based access to the Broker. It is compatible with the Delphi object oriented (OO) environment. This component, when placed on a Delphi form, allows you to connect to the server and reference M data within Delphi's Integrated Development Environment (IDE). It makes a Delphi form and everything on it "data aware."

4.2.3 TXWBRichEdit

The TXWBRichEdit component replaces the Introductory Text Memo component on the Login Form. TXWBRichEdit is a version of the TRichEdit component that uses Version 2 of Microsoft's® RichEdit Control and adds the ability to detect and respond to a Uniform Resource Locator (URL) in the text. This component permits you to provide some requested functionality on the login form. As an XWB namespaced component you are required to put it on the **Kernel** tab of the component palette, however, it rightly belongs on the **Win32** tab. (XWB*1.1*13)

4.2.4 TContextorControl

The TContextorControl component communicates with the Vergence Locator service. (XWB*1.1*40)

4.2.5 Classes Added

The following Classes were added to or modified in RPC Broker 1.1:

- TVistaLogin (XWB*1.1*13)
- TVistaUser (XWB*1.1*13)
- TXWBWinsock (XWB*1.1* 40)

4.2.6 Library Methods Added/Modified

The following Library Methods were added to or modified in RPC Broker 1.1.

4.2.6.1 Added

The following library methods were added to the TVCEdit Unit (XWB*1.1*13):

- ChangeVerify:

```
function ChangeVerify(RPCBroker: TRPCBroker): Boolean;
```

- SilentChangeVerify:

```
function SilentChangeVerify(RPCBroker: TRPCBroker; OldVerify, NewVerify1,
NewVerify2: String; var Reason: String): Boolean;
```

- StartProgSLogin:

```
procedure StartProgSLogin(const ProgLine: String; ConnectedBroker:
TRPCBroker);
```

The following library methods were added to the TCCOWRPCBroker component (XWB*1.1*40):

- GetCCOWtoken:

```
function GetCCOWtoken(Contextor: TContextorControl): string;
```

- IsUserCleared:

```
function IsUserCleared: Boolean;
```

- IsUserContextPending:

```
function IsUserContextPending(aContextItemCollection:
IContextItemCollection):  Boolean;
```

- WasUserDefined:

```
function WasUserDefined: Boolean;
```

4.2.6.2 Modified

The following library methods were modified (XWB*1.1*13):

- CheckCmdLine:

```
function CheckCmdLine(SLBroker: TRPCBroker): Boolean;
```

  This was changed from a procedure to a function with a Boolean return value.

- GetServerInfo:

  The GetServerInfo library function in the RpcConf1unit, which can be used to select the desired Server name and ListenerPort, was modified to add a “new” button. This button can be used to add a new Server/ListenerPort combination to those available for selection. It will also accept and store a valid IP address, if no name is known for the location. This will permit those who have access to other Server/ListenerPort combinations that may not be available in the list on the current workstation to access them. However, they will still need a valid Access and Verify code to log on to the added location. Patch XWB*1.1*60 added a third field to store the SSHUsername for Secure Shell (SSH) connections. In other words, the Server/ListenerPort/SSHUsername combination is now stored in the Windows Registry for known VistA servers.

- TParams:

  The procedure Clear was moved from Private to Public.

- TRPCB Unit:

```
TOnLoginFailure = procedure (VistaLogin: TVistaLogin) of object;
Changed from Object: TObject, since this is what should be expected by the
procedure if it is called.
TOnRPCBFailure = procedure (RPCBroker: TRPCBroker) of object;
```

  Changed from Object: TObject, since this is what should be expected by the procedure if it is called.

4.2.7 Properties Added

The following Properties were added to or modified in RPC Broker 1.1:

4.2.7.1 TCCOWRPCBroker Properties

The following properties were added (XWB*1.1*40):

- CCOWLogonIDName (Public)
- CCOWLogonIDValue (Public)
- CCOWLogonName (Public)
- CCOWLogonNameValue (Public)
- CCOWLogonVpid (Public)
- CCOWLogonVpidValue (Public)
- Contextor (Public)

4.2.7.2 TRPCBroker Properties

The following properties were added (XWB*1.1*13 and 35):

- BrokerVersion (Public)
- CurrentContext (Public)
- KerneLogIn (Published)
- LogIn (Public)
- OnRPCBFailure (Public)
- RPCBError (Public)
- ShowErrorMsgs (Published)
- User (Public)

4.2.7.3 TSharedBroker and TSharedRPCBroker Properties

The following Shared Broker properties were removed. The Shared Broker has been deprecated. (XWB*1.1*60)

- AllowShared (Public)
- OnConnectionDropped (Public)
- OnLogout (Published)

4.2.7.4 TVistaLogin Properties

The following properties were added (XWB*1.1*40):

- DomainName (Public)
- IsProductionAccount (Public)

4.2.7.5 TVistaUser Property

The following property was added (XWB*1.1*40):

- Vpid (Public)

4.2.8 Types Added/Modified

The following Types were added to or modified in RPC Broker 1.1 (XWB*1.1*13 and XWB*1.1*40):

- TLoginMode
- TShowErorMsgs
- TOnLoginFailure
- TOnRPCBFailure
- TParamType

4.3 Design-time and Run-time Packages

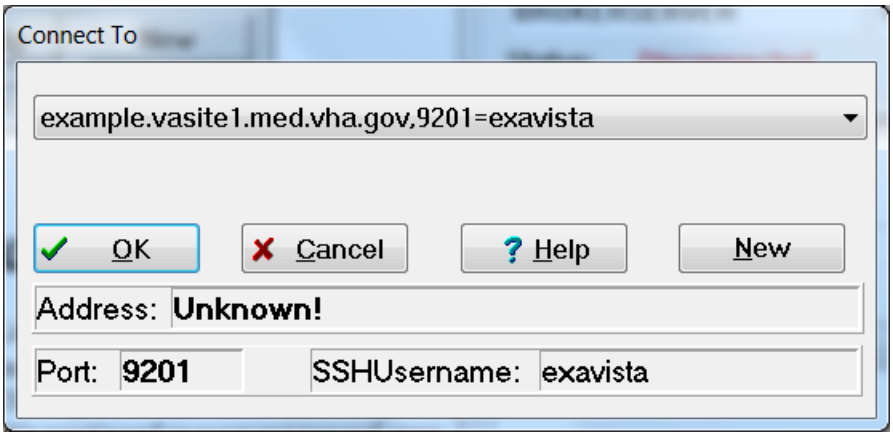
The BDK contains separate run-time and design-time packages. (XWB*1.1*14)

4.4 Modified GetServerInfo Function

The GetServerInfo function obtains the end-user's target server and port. Use this function to set the TRPCBroker component's Server and ListenerPort properties before connecting to the server.

If there is more than one server/port to choose from, GetServerInfo displays an application window that allows users to select a service to connect to:

Figure 1: "Connect To" dialogue



4.5 Updated Dynamic Link Library (DLL) Interface

RPC Broker 1.1 provides Dynamic Link Library (DLL) functions that allow applications written in *any* Microsoft® Windows-based development environment (e.g., Embarcadero's Delphi, Embarcadero C++, Microsoft® Visual Basic, and other COTS products), to take advantage of all the features offered by the RPC Broker component. This reflects VistA's continued movement toward open systems that support multiple GUI and client front-ends.

The Dynamic Link Library (DLL) functions act like a "shell" around the Delphi TRPCBroker component and provide developers with an easy function-based access to the Broker component. These functions allow GUI and client front-end applications written in Embarcadero's Delphi and other COTS products to take advantage of all the features that the Broker offers. All of the communication to the server is handled by the TRPCBroker component accessed via the DLL interface.

NOTE: 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.

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

4.6 Source Code Availability

The BDK contains the Broker source code. The source code is located in the following directory:

..\BDK32\Source

CAUTION: Modified BDK source code should *not* be used to create VistA GUI applications.

Not all methods and properties found in the source code are documented at this time. Only those documented methods and properties are guaranteed to be made backwards compatible in future versions of the BDK.

5 References

5.1 RPC Broker Context-sensitive Online Help

RPC Broker 1.1 provides online context-sensitive help (i.e., Broker_1_1.chm) for the RPC Broker-related components and associated DLL exported procedures and functions. Select the component on a form, or highlight a particular procedure or function, and press the F1 key to get help on that item.

The online help also includes other related topics for IRM and Broker developer (e.g., Tutorials, RPC information, Troubleshooting and Debugging tips, etc.).

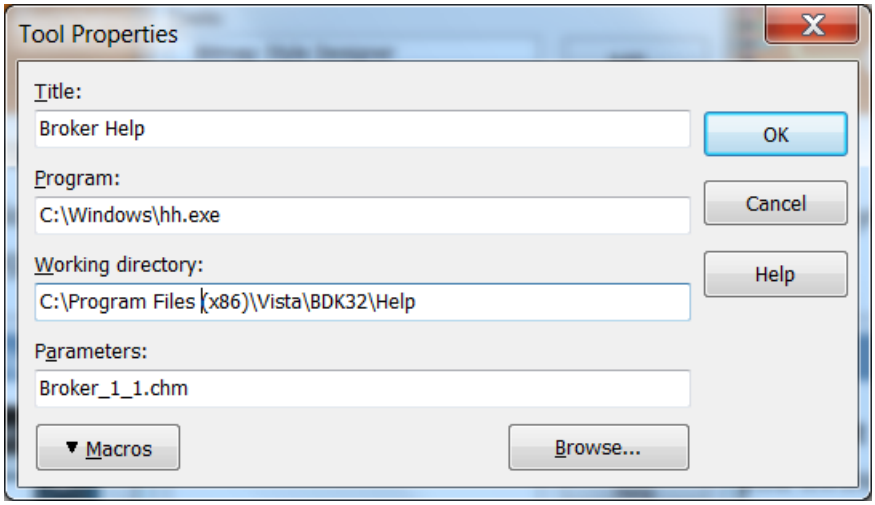
5.2 RPC Broker Documentation

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

- *RPC Broker Release Notes* (this manual)
- *RPC Broker Installation Guide*
- *RPC Broker Systems Management Guide*
- *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 2.

Figure 2: 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.