RPC Broker

Release Notes

Software Version 1.1

September 1997

Revised April 2014

Department of Veterans Affairs (VA)

Office of Information and Technology (OIT)

Product Development (PD)

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.

Table i: Documentation revision history

Date	Revision	Description	Author
04/10/2014	5.3	Tech Edit: • Made minor format and style corrections throughout. • Added the "Properties Modified" section. • Made minor reference updates.	• Technical Writer: T. B.
03/26/2014	5.2	Tech Edit: • Changed references from "Broker.hlp" to "Broker_1_1.chm" throughout. • Updated other help references and instructions related to updated Broker help; replacing WinHelp (Broker.hlp) to HTML help (Broker_1_1.zip with Broker_1_1.chm file and Broker_1_1-HTML_Files.zip with multiple HTML files). • Changed references from "programmer" to "developer" throughout. • Made other minor grammar and punctuation corrections throughout.	• Technical Writer: T. B.
12/04/2013	5.1	Tech Edit: • Updated document for RPC Broker Patch XWB*1.1*50 based on feedback from H. W. • 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	• Developer: H. W. • Technical Writer: T. B.

Date	Revision	Description	Author
		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 4.2.3.1. • Added Section 4.2.2. **RPC Broker 1.1**	
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**	• Developer: H. W. • Technical Writer: T. 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 4.2.3.1). • 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, CA Office of Information Field Office (OIFO): • Development Manager: J. C. • Developers: J. I. and R. M. • SQA: L. B. • Technical Writer: T. B.

Date	Revision	Description	Author
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**	Common Services (CS) Development Team Oakland, CA OIFO: • Development Manager: J. Sch. • Developer: J. I. • SQA: G. S. • 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 OIFO 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 OIFO Development Team • Technical Writer: T. B.
09/97	1.0	Initial RPC Broker Version 1.1 software release. **RPC Broker 1.1**	• Oakland and San Francisco Office of Information Field Office (OIFO) • 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, RPC Broker Patch XWB*1.1*50 is now available. This enhanced Broker software has the following functionality/features:

- **Support for Later Delphi Versions—**As of Patch XWB*1.1*50, the BDK supports Delphi XE5, XE4, XE3, and XE2.
- **Supports Secure Shell (SSH)—**As of RPC Broker Patch XWB*1.1*50, 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.
- **Supports Broker Security Enhancement (BSE)—**As of RPC Broker Patch XWB*1.1*50, the RPC Broker supports the Broker Security Enhancement (BSE). The TRPCBroker component was modified to enable visitor access to remote sites using authentication established at a home site.
- **Supports Single Sign-On/User context (SSO/UC)—**As of RPC Broker Patch XWB*1.1*40, the TCCOWRPCBroker component enabled Single Sign-On/User Context (SSO/UC) in CCOW-enabled applications.
- **Supports Non-Callback Connections—**As of RPC Broker Patch XWB*1.1*35, the RPC Broker components are built with a UCX or *non*-callback Broker connection, so that it can be used from behind firewalls, routers, etc. This functionality is controlled via the TRPCBroker component IsBackwardCompatibleConnection property.
- **Supports Silent Logon capabilities—**As of RPC Broker Patch XWB*1.1*13, the RPC Broker provides "Silent Login" capability. It provides functionality associated with the ability to make logins to a VistA M Server without the RPC Broker asking for Access and Verify code information.
- **Documented Deferred RPCs and Capability to Run RPCs on a Remote Server.**
- **Multi-instances of the RPC Broker—**As of RPC Broker Patch XWB*1.1*13, the RPC Broker code was modified to permit 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.

CAUTION: Although there should be no problems, the RPC Broker is n*ot yet guaranteed to be thread safe*.

- **Updated components, properties, methods, and types.**
- **Separate Design-time and Run-time Packages—**As of RPC Broker Patch XWB*1.1*14, the BDK contains separate run-time and design-time packages.

- **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:

- The 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 developers with the capability to develop VistA client/server software using the following RPC Broker Delphi components in the 32-bit Microsoft® Windows environment (listed alphabetically):

- TCCOWRPCBroker
- TContextorControl
- TRPCBroker (original component)
- TSharedBroker
- TSharedRPCBroker
- TXWBRichEdit

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 Features and Enhancements

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

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

2 End Users—Features and Enhancements

2.1 Support for Secure Shell (SSH) Tunneling

The Veterans Health Administration (VHA) information systems user community expressed a need for a Secure Shell (SSH) tunneling service to provide secure data transfer between the client and the VistA M Server. To address this architectural need, RPC Broker Patch XWB*1.1*50 added support for SSH tunneling.

2.2 Support for Broker Security Enhancement (BSE)

The VHA information systems user community expressed a need for support of the Broker Security Enhancement (BSE). To address this architectural need, RPC Broker Patch XWB*1.1*50 modified the TRPCBroker component to enable visitor access to remote sites using authentication established at a home site.

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

The VHA information systems user community expressed a need for a single sign-on (SSO) service with interfaces to VistA, HealtheVet VistA, and non-VistA systems. This architecture allows 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 HealtheVet clinician desktop environment. To address this architectural need, RPC Broker Patch XWB*1.1*40 provided the TCCOWRPCBroker component.

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.

When a VistA CCOW-enabled application is recompiled with the TCCOWRPCBroker component and other required code modifications are made, that application becomes 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 Non-Callback Connections

As of RPC Broker Patch XWB*1.1*35, the RPC Broker components are built with a UCX or non-callback Broker connection, so that it can be used from behind firewalls, routers, etc. This functionality is controlled via the TRPCBroker component IsBackwardCompatibleConnection property.

2.5 Support for Silent Logon

As of RPC Broker Patch XWB*1.1*13, the RPC Broker provides "Silent Login" capability. It provides functionality associated with the ability to make logins to a VistA M Server without the RPC Broker asking for Access and Verify code information.

Two types of Silent Login are provided with RPC Broker 1.1 BDK:

- **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 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 and Enhancements

3.1 On the Server

3.1.1 SSH Support

As of RPC Broker Patch XWB*1.1*50, the RPC Broker provides Secure Shell (SSH) support. Attachmate® Reflections terminal emulator software with SSH tunneling is used inside 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.

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 uses 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:

SSHPort=portnumber

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

3.1.2.1 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.

3.1.2.2 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).

3.1.3 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).

3.1.4 Full Backward Compatibility with Broker 1.0

Broker-based applications compiled with Broker 1.0 continue to work with RPC Broker 1.1 server routines.

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 XP or Windows 7 operating systems).

CAUTION: Both the RPCBI.DLL and Client Manager (i.e., CLMAN.EXE) distributed with RPC Broker 1.0 are *no longer required with RPC Broker 1.1*. However, the RPCBI.DLL and Client Manager should *not* be removed if they exist on the workstation. Previous 16-bit Broker-based applications may still require the RPCBI.DLL and Client Manager.

4 Developers—Features and Enhancements

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

- 32-Bit Processing/Delphi Support
- RPC Broker Components
  - Classes Added
  - Library Methods Added/Modified
  - Properties Added
  - Properties Modified
  - As of RPC Broker Patch XWB*1.1*40, the following property was added:
- PType—Added the following PType values:
  - global—This value is similar to list, but instead of data being placed in a local array, it is placed in a global array. Use of this value removes the potential problem of allocation errors when large quantities of data are transmitted.
  - empty—This value indicates that no parameter value is to be passed; it simply passes an empty argument.
  - stream—This value indicates that the data should be passed as a single stream of data.
  - undefined—The Broker uses this value internally. It should *not* be used by an application.
  - 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 Support

The RPC Broker 1.1 BDK operates in a 32-bit Microsoft® Windows environment only. The client workstations can be running either of the following Microsoft® operating systems:

- Windows XP
- Windows 7

It provides developers with the capability to develop VistA client/server software using the Broker Delphi components (e.g., TRPCBroker) in the 32-bit environment. RPC Broker 1.1 does *not* yet support development in a 64-bit environment.

NOTE: As of RPC Broker Patch XWB*1.1*50, RPC Broker 1.1 supports Delphi XE2, XE3, XE4, and XE5.

CAUTION: This statement defines the extent of support relative to use of Delphi. The Office of Information and Technology (OIT) only supports the Broker Development Kit (BDK) running in the currently offered version of Delphi and the immediately previous version of Delphi. This level of support became effective 06/12/2000.

Sites may continue to use outdated versions of the RPC Broker Development Kit but do so with the understanding that support is *not* available and that continued use of outdated versions do *not* afford features that can be essential to effective client/server operations in the VistA environment. An archive of old (no longer supported) Broker Development Kits is maintained in the VA Intranet Broker Archive.

4.2 RPC Broker Components

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

- TCCOWRPCBroker
- TContextorControl
- TRPCBroker (original component)
- TSharedBroker
- TSharedRPCBroker
- TXWBRichEdit

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 *RPC Broker Developer's Guide* (i.e., Broker_1_1.chm; BDK Online Help).

4.2.1 TCCOWRPCBroker

As of RPC Broker Patch XWB*1.1*40, the TCCOWRPCBroker component was added to RPC Broker 1.1. 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.

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).

4.2.2 TContextorControl

As of RPC Broker Patch XWB*1.1*40, the TContextorControl component was added to RPC Broker 1.1. The TContextorControl component communicates with the Vergence Locator service.

4.2.3 TRPCBroker

The original TRPCBroker 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 applications to connect to the VistA M Server and reference M data within Delphi's Integrated Development Environment (IDE). It makes a Delphi form and everything on it "data aware."

4.2.3.1 CCOW User Context Wrapped into the Primary TRPCBroker Component

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

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.

4.2.4 TSharedBroker

As of RPC Broker Patch XWB*1.1*26, the TSharedBroker component was added to RPC Broker 1.1. The TSharedBroker component provides applications or plugins to applications easy access to an RPC Broker component without the need for a separate M partition. Each component has its own security (i.e., option) as well. The default value of the AllowShared is **True**. If an application has RPCs that

require extensive time, it would be best to *not* share a Broker instance and the AllowShared should then be set to **False**.

4.2.5 TSharedRPCBroker

As of RPC Broker Patch XWB*1.1*26, the TSharedBroker component was added to RPC Broker 1.1. The TSharedRPCBroker component provides applications or plugins to applications easy access to an RPC Broker component without the need for a separate M partition. Each component has its own security (i.e., option) as well. The default value of the AllowShared is **True**. If an application has RPCs that require extensive time, it would be best to *not* share a Broker instance and the AllowShared should then be set to **False**.

4.2.6 TXWBRichEdit

As of RPC Broker Patch XWB*1.1*13, the TXWBRichEdit component was added to RPC Broker 1.1. 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 us to provide some requested functionality on the login form. As an XWB namespaced component we are required to put it on the Kernel tab of the component palette, however, it rightly belongs on the **Win32** tab.

4.2.7 Classes Added

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

4.2.7.1 TVistaLogin

As of RPC Broker Patch XWB*1.1*13.

4.2.7.2 TVistaUser

As of RPC Broker Patch XWB*1.1*13.

4.2.7.3 TXWBWinsock

As of RPC Broker Patch XWB*1.1* 40.

4.2.8 Library Methods Added/Modified

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

4.2.8.1 Added

As of RPC Broker Patch XWB*1.1*13, the following library methods were added to the TVCEdit Unit:

4.2.8.1.1 ChangeVerify

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

4.2.8.1.2 SilentChangeVerify

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

4.2.8.1.3 StartProgSLogin

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

As of RPC Broker Patch XWB*1.1*40, the following library methods were added with the TCCOWRPCBroker component:

4.2.8.1.4 GetCCOWtoken

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

4.2.8.1.5 IsUserCleared

```
function IsUserCleared: Boolean;
```

4.2.8.1.6 IsUserContextPending

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

4.2.8.1.7 WasUserDefined

```
function WasUserDefined: Boolean;
```

4.2.8.2 Modified

As of RPC Broker Patch XWB*1.1*13, the following library methods were modified:

4.2.8.2.1 CheckCmdLine

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

Changed from procedure to function with a Boolean return value.

4.2.8.2.2 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 also accepts and stores a valid IP address, if no name is known for the location. This permits those who have access to other Server/ListenerPort combinations, which may not be available in the list on the current workstation, to access them. However, they still need a valid Access and Verify code to log on to the added location.

4.2.8.2.3 TParams

The procedure Clear was moved from Private to Public.

4.2.8.2.4 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.

4.2.9 Properties Added

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

4.2.9.1 TCCOWRPCBroker Properties

As of RPC Broker Patch XWB*1.1*40, the following properties were added:

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

4.2.9.2 TRPCBroker Properties

As of Patches XWB*1.1*13 and 35, the following properties were added:

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

4.2.9.3 TSharedBroker and TSharedRPCBroker Properties

As of RPC Broker Patch XWB*1.1*23, the following properties were added:

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

4.2.9.4 TVistaLogin Properties

As of RPC Broker Patch XWB*1.1*40, the following properties were added:

- DomainName (Public)
- IsProductionAccount (Public)

4.2.9.5 TVistaUser Property

As of RPC Broker Patch XWB*1.1*40, the following property was added:

- Vpid (Public)

4.2.10 Properties Modified

As of RPC Broker Patch XWB*1.1*40, the following property was added:

- PType—Added the following PType values:
  - global—This value is similar to list, but instead of data being placed in a local array, it is placed in a global array. Use of this value removes the potential problem of allocation errors when large quantities of data are transmitted.
  - empty—This value indicates that no parameter value is to be passed; it simply passes an empty argument.
  - stream—This value indicates that the data should be passed as a single stream of data.
  - undefined—The Broker uses this value internally. It should *not* be used by an application.

4.2.11 Types Added/Modified

As of RPC Broker Patch XWB*1.1*13 and XWB*1.1*40, the following Types were added or modified:

- TLoginMode
- TShowErorMsgs
- TOnLoginFailure
- TOnRPCBFailure
- TParamType

4.3 Design-time and Run-time Packages

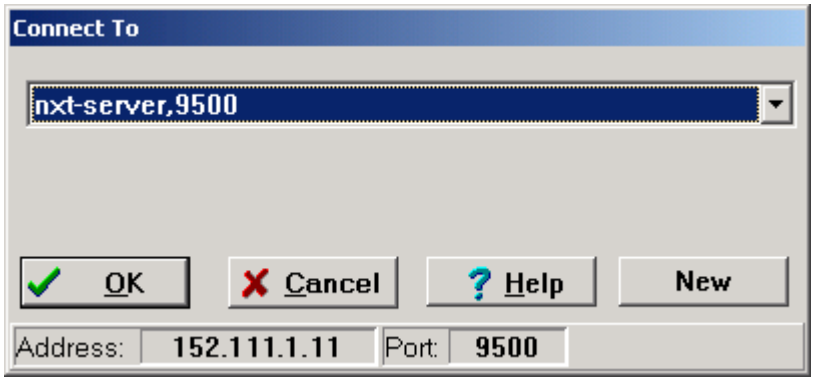
As of RPC Broker Patch XWB*1.1*14, the BDK contains separate run-time and design-time packages.

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 a set of 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 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 contains all of the 32-bit Broker DLL functions. It provides an interface to the Broker component.

4.6 Source Code Availability

As of RPC Broker Patch XWB*1.1*14, 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 Updated RPC Broker Context-sensitive Online Help

RPC Broker 1.1 provides updated online HTML help for the RPC Broker-related components and associated DLL exported procedures and functions. To open the BDK Help:

1. Extract the Help files from the Zip files.
2. Double-click on either of the following files:
    - **Broker_1_1.chm** (standalone help file)
    - **index.htm** (links to multiple HTML files)

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

5.2 Updated RPC Broker Documentation

RPC Broker 1.1 provides a full set of updated documentation. Readers who wish to learn more about the 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*—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 may want to make 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.
- 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 the following Website: 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 Health Product Support (HPS) Anonymous Directories.