

RPC Broker Version 1.1 Release Notes



RPC BROKER OVERVIEW

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

- Operates in a 32-bit environment
- Supports single signon capabilities

This version of the Broker 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 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)

The RPC Broker component (i.e., TRPCBroker) wraps the functionality of the Broker into a Delphi component resulting in a more modularized and orderly interface. All properties and methods of the Broker are in one component.



NEW FEATURES AND ENHANCEMENTS OF THE RPC BROKER

This version of the RPC Broker client/server interface provides the following *new* features and enhancements categorized by user type:

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



*Please note that several of the new features and enhancements of the RPC Broker are supported by changes made to Kernel in Kernel patch #59 (i.e., XU*8*59).*

End User Features & Enhancements

■ SUPPORT FOR SINGLE SIGNON FOR MULTIPLE USER SESSIONS

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

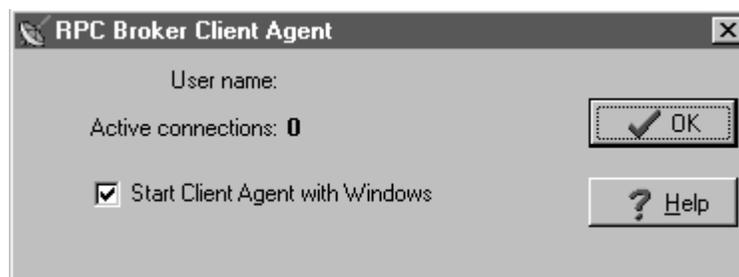
Also, if a user has opened both a **VISTA** roll-and-scroll session and a client/server application on the same workstation and the roll-and-scroll session times out, the user will still be logged on via the client/server application.



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

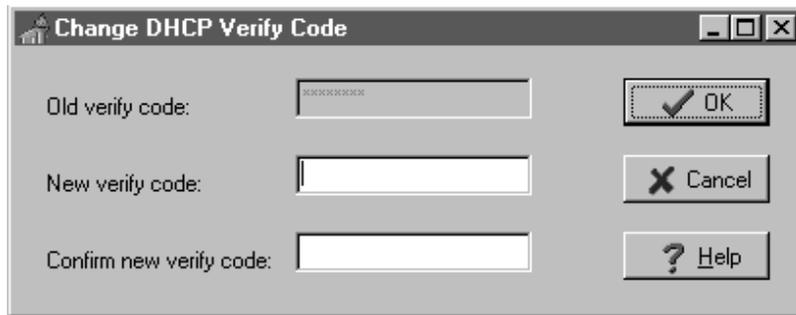
■ RPC BROKER CLIENT AGENT

The RPC Broker Client Agent program (i.e., CLAGENT.EXE) runs in support of the single signon process (a.k.a. auto signon). This program automatically and continuously runs in the background on the client workstation and normally should *not* be closed or shut down by the user. An icon (i.e., satellite dish) will be displayed in the System Tray indicating the Broker Client Agent is running. By double clicking on the Client Agent icon, you can see how many active connections are currently open. The dialogue box is displayed below:



■ **NEW CHANGE VISTA VERIFY CODE COMPONENT**

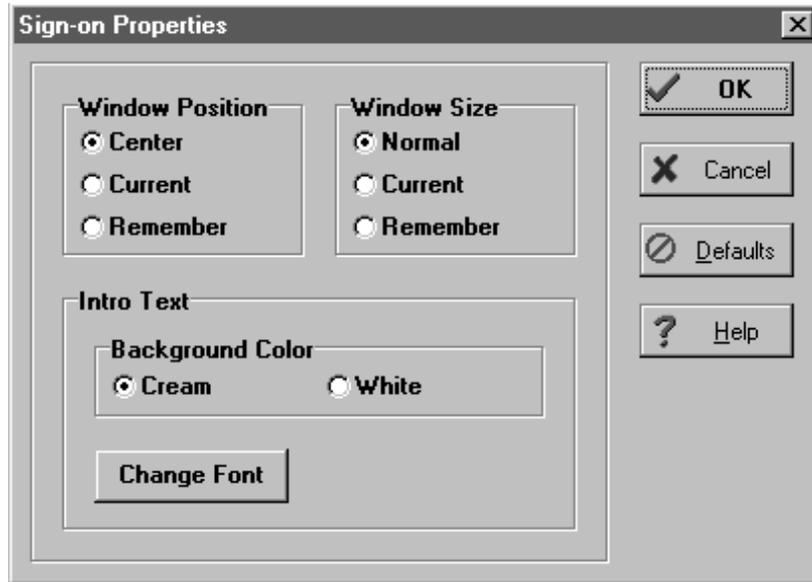
This version of the Broker includes a new Change **VISTA** Verify Code dialog for the client workstation. After a user signs on to the server, if their Verify code has expired, the user is automatically presented with the Change **VISTA** Verify Code dialog as displayed below. Users will then be able to input and confirm their new Verify code.



In the future, users will be able to invoke this dialog whenever they choose to modify their Verify code. For now, users who wish to change their Verify code prior to expiration may do so via the Edit User Characteristics option in the roll-and-scroll environment.

■ SIGNON WINDOW CAN BE CUSTOMIZED BY USERS

Users can now customize the **VISTA** signon window. When a **VISTA** program on the client connects to the server, the **VISTA** signon window is displayed for the user to identify and authenticate himself on the server. This window's System menu has a "Properties..." item. When this item is selected, the user is presented with the following configuration dialogue box:



Users can control the appearance of the signon window by modifying the following characteristics:

- **Position of the signon Window**
- **Size of the signon Window**
- **Appearance of the Introductory Text**

End User Features & Enhancements

System Managers Feature & Enhancements

ON THE SERVER

■ Management of Single Signon

Control of the single signon functionality is maintained and administered on the server for both **VISTA** client/server applications (i.e., GUI) and the roll-and-scroll environment (i.e., terminal sessions). In support of that functionality a new field, AUTO SIGN-ON, was added to both the KERNEL SYSTEM PARAMETERS file (#8989.3) and the NEW PERSON file (#200).



Single signon is *not* supported on MSM systems. All MSM sites must disable single signon (a.k.a. auto signon).

■ New RPC BROKER SITE PARAMETERS File (#8994.1)

This file is used to configure and adjust many characteristics of an RPC Broker installation at a site. This version of the Broker includes a LISTENER field and various subfields. The LISTENER field is a multiple and should contain all Listeners that you plan to run. A simple change of the STATUS subfield from STOPPED to START will start that particular Listener. Conversely, you can STOP a RUNNING Listener.



DSM sites must have TaskMan started via DCL, if they want to use the RPC BROKER SITE PARAMETERS file (#8994.1) settings to start Listeners on other nodes.

■ Load Balancing on Alpha Systems

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



For more information on load balancing, please refer to the "Load Balancing on Alpha Systems" topic in Chapter 1 in the RPC Broker V. 1.1 Systems Manual.

■ Support for InterSystems OpenM for NT V. 7

This version of the Broker runs on InterSystems OpenM for Windows NT version 7 as well as on Digital Standard M (DSM) and Micronetics Standard M (MSM) for Windows NT. Starting and stopping the Listener under OpenM is the same as it is for DSM. The RPC Broker installation and configuration procedures for OpenM are similar to DSM except where noted.



For more information on installing the Broker in the OpenM environment, please refer to the RPC Broker V. 1.1 Installation Guide.

■ Enhancements and Improvements to the Listener Process

The following enhancements to the Listener process were added to this version of the Broker:

• Support for *Multiple* Listeners

This version of the Broker allows sites to start *multiple* Broker Listener processes in *one* account. Sites can now also run *multiple* Listeners in the *same* UCI-volume, on *different* IP addresses/CPU, listening to the *same* port. If you need to run several Broker Listeners on the same IP address/CPU (e.g., one for a Production account and another for a Test account), they must all be using different ports. For example, if the Listener in the Production account is on 9200, start another Listener in the Test account on 9201. Alternatively, you may want to allocate entire ranges of ports for various accounts. If the IP addresses/CPU are different, then the port can be the same. This means that you can run a Listener on every machine listening on 9200.

• New XWB LISTENER STARTER Option

The XWB LISTENER STARTER option can be used to start one or more Broker Listeners at one time. Additionally, this option can be used to automatically start all of the Listener processes you need when TaskMan starts up (e.g., after the system is rebooted or configuration is restarted).



DSM sites must have TaskMan started via DCL, if they want to use the XWB LISTENER STARTER option to automatically start Listener processes.



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

If you're successfully using MSERVER, discontinue using the Broker Listener. That means you shouldn't use the STRT^XWBTCP and STOP^XWBTCP entry points or the XWB LISTENER STARTER option.

- **Improved Listener Startup Problem Detection**

This version of the Broker includes an enhancement that improves the detection of problems during the startup of the Listener. If a Listener cannot start listening on a given port (e.g., port may be busy), it will generate an error on the server and retry up to *five* times while continuously informing the user of its status.

- **Listener/Handler Process Names More Descriptive and Informative**

The Listener and Handler process names were changed to be more descriptive and informative. If you request a system status on a DSM system where the Broker Listener/Handler is running, the names will now appear as follows:

Listener Process: RPCB_Port:NNNN

(where NNNN is the port number being listened to)

Handler Process: ipXXX.XXX:NNNN

(where XXX.XXX are the last two octets of the client IP address and NNNN is the port number)

This should prove quite useful in finding the Listener and Handler for troubleshooting any connection problems.

- **Early Detection of Impostors and Intruders**

Before the 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 is an enhancement to the security feature of the Broker.

■ **Full Backward Compatibility with Broker V. 1.0**

Broker-based applications compiled with Broker V. 1.0 (e.g., PCMM) will continue to work with this version of the Broker server routines.

ON THE CLIENT**■ 32-Bit Processing**

This version of the Broker operates in a 32-bit Windows environment (i.e., client workstations running Microsoft Windows 95 or Windows NT operating systems). However, this version of the Broker will continue to support **VISTA** applications previously developed for the 16-bit environment (e.g., PCMM).

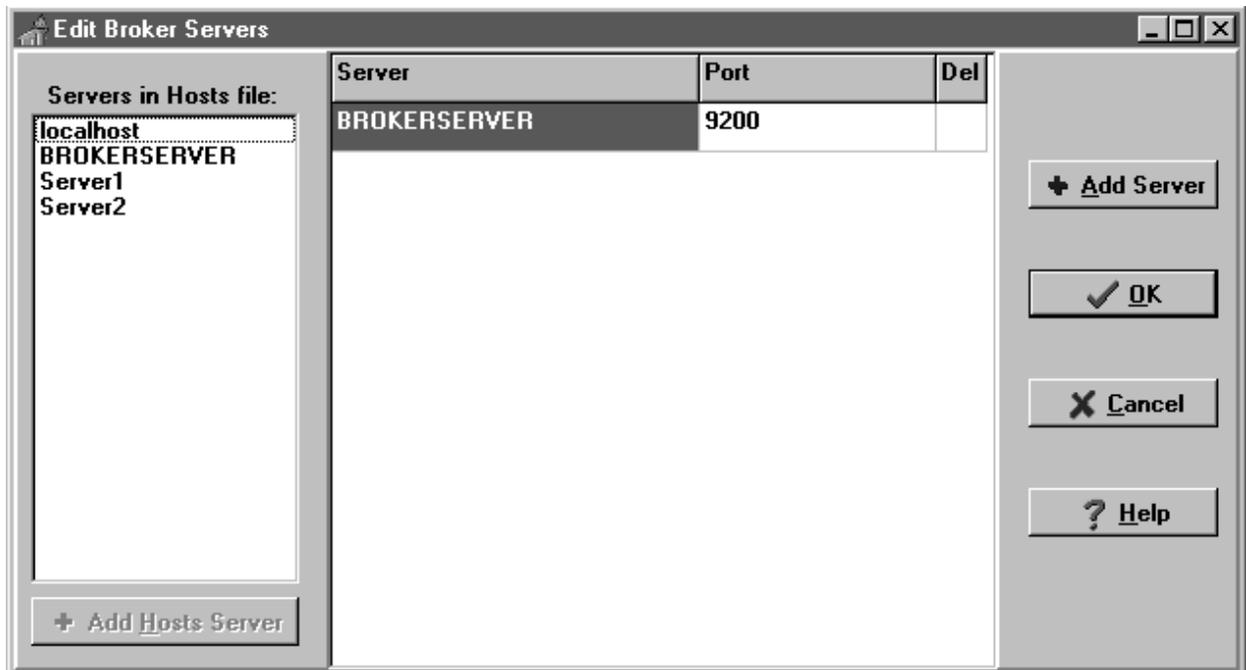


Both the RPCBI.DLL and Client Manager (i.e., CLMAN.EXE) distributed with version 1.0 of the Broker are no longer required with this version of the Broker.

However, the RPCBI.DLL and Client Manager should *not* be removed. Previous 16-bit Broker-based applications (e.g., PCMM) still require the RPCBI.DLL and Client Manager.

■ Edit Broker Servers Program

If IRM wishes to add, modify, or delete servers and ports to be used by the Broker, run the Edit Broker Servers program (i.e., ServerList.EXE) to modify or add Listeners/Ports to the Windows Registry. ServerList.EXE can be copied to any workstation for this purpose.





The ServerList.EXE program only displays HOSTS file entries; it does not edit the HOSTS file.

Adding Entries:

You are given two methods of adding new server/port entries to the Registry:

1. Add a server listed in your HOSTS file
2. Add a new server that is automatically validated via the Domain Name Service (DNS) or the HOSTS file

Modifying Entries:

In order to modify or change a server or port, simply place the cursor in the appropriate Server or Port field and make the change.

Deleting Entries:

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

■ New Broker Diagnostic Tool

This version of the Broker includes a new diagnostic/troubleshooting tool for the client workstation. This tool can be used to verify and test the Broker client/server connection and signon process. The dialogue box is displayed below:

This program (i.e., RPCTEST.EXE) also displays specific information about the client workstation that can be useful when trying to diagnose a problem with or to test the Broker.

It displays the following information:

- Default Workstation Information that includes the Name and IP Address
- Local Connection Information that includes the Name, Client IP, and Current Socket
- **VISTA** User Information that includes the Name and Last SignOn Date/Time
- Remote Connection Information that includes the Server, Port, IP Address, and Operating System (OS) Version information

- A color coded Link State indicating the status of your connection:

Link State Indicator Color	Status of Connection
RED	No link/connection
YELLOW	Attempting link/connection
GREEN	Successful link/connection



Context-sensitive help is available with the `RPCTest.EXE` program by pressing the F1 key.

Developer Features & Enhancements

The 32-bit version of the Broker Development Kit (BDK) is now available and included with the release of version 1.1 of the RPC Broker.

For those developers who received the preview BDK released in October of 1995, many of the following features are either new or enhancements to the preview BDK:

ON THE SERVER

■ **Enhanced Buffer Clearing and Timeouts on the Client**

Previously, if a remote procedure call failed to successfully complete due to a timeout on the client, the buffer on the server would contain data from the uncompleted call. A timeout on the client *doesn't* generate a network error that will automatically clear the buffer. Since there is no mechanism to "flush" or empty the buffer, the data from the *next* RPC call would be corrupted.

This version of the Broker solved the timeout/buffer problem in two steps:

1. This version of the Broker provides developers with a new `RPCTimeLimit` property on the Broker component on the client. This new property can be set by developers to help avoid the timeout issue in the first place.
2. Secondly, in the event of a cancellation of a Network I/O operation, the Broker will automatically "flush" the incoming buffer so that the next RPC will execute normally. Users may experience a delay while the corrupted RPC data is discarded. Any delay will be proportional to the amount of data in the buffer.

■ **XWB GET VARIABLE VALUE RPC**

Distributed with this version of the RPC Broker is the XWB GET VARIABLE VALUE RPC. Developers can call this RPC to retrieve the value of any M variable in the server environment.

■ **Support for New M Standard Error Trapping**

This version of the Broker uses the new Kernel error trap, which now records stack information and NEWed variables.

■ **Creation of an MSM Null Device**

An MSM null device will be used to catch all inadvertent WRITES in RPCs, specifically, those in "ID" nodes of files

Developer Features & Enhancements

ON THE CLIENT**■ 32-Bit Processing/Delphi**

This version of the Broker operates in a 32-bit Windows environment only (i.e., client workstations running Microsoft Windows 95 or Windows NT operating systems). It provides programmers with the capability to develop new **VISTA** client/server software using the Broker Delphi component (i.e., TRPCBroker) in the 32-bit environment. To develop **VISTA** applications in a 32-bit environment you must have Delphi V. 2.0 or greater. This version of the RPC Broker component will *not* allow you to develop applications in Delphi V. 1.0. However, the Broker routines on the M server will continue to support **VISTA** applications previously developed and compiled for the 16-bit environment (e.g., PCMM).



Please note that version 1.1 of the Broker supports Delphi V. 3.0 as well as Delphi V. 2.0. However, the next version of the Broker will no longer provide support for Delphi V. 2.0.

■ Delphi TRPCBroker Component

The 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".

• TRPCBroker Properties

Version 1.1 of the TRPCBroker component provides the following set of design-time and run-time properties:

ClearParameters	ListenerPort	RPCTimeLimit (<i>New, available at run-time only</i>)
ClearResults	Param (<i>available at run-time only</i>)	RPCVersion (<i>New</i>)
Connected	RemoteProcedure	Server
DebugMode	Results	



For a complete description of the RPC Broker properties, please refer to the Online RPC Broker Developer's Guide (i.e., BROKER.HLP).

- **TRPCBroker Methods**

Version 1.1 of the TRPCBroker component provides the following methods:

- Call
- CreateContext
- lstCall
- strCall



For a complete description of the RPC Broker methods, please refer to the Online RPC Broker Developer's Guide (i.e., BROKER.HLP).

- **Automatically Connects to and Disconnects From the Server**

One of the features of the Broker component is that it is *not* necessary for your **VISTA** application to manually establish a connection to the server. This version of the Broker automatically connects to and disconnects from the server. When you invoke an RPC, if a connection is not already established, one will be established for you. However, you can also manually connect and disconnect your application from the server if desired.



*The connect and disconnect functionality that was previously provided by the DLL has now been incorporated with the Broker component itself. This allows **VISTA** applications to be independent of the DLL and its restrictions.*

- **Management of Multiple TRPCBroker Components**

The first time one of the Broker components in your application connects, it will establish an actual connection with the server. The connection record will be added to the list of all active connections for your application. This list is internal to the application and is completely under the control of the Broker component and is transparent to you. If another Broker component tries to connect to the same server/port, the existing connection record will be found in the list and its socket will be shared. The new connection will also be added to this list. This process is repeated with each connection request.

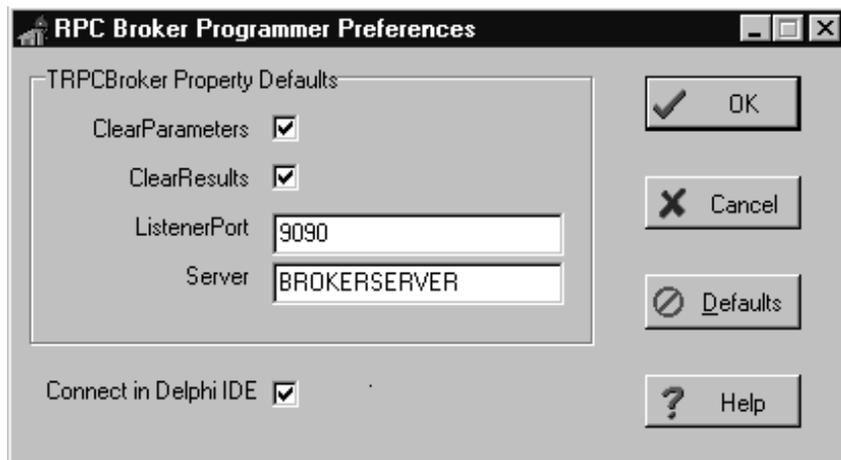
When a Broker component disconnects, its connection record is removed from the internal list of active connections. If it happens to be the last record for the particular server/port combination, the connection is actually closed. This scheme provides the illusion of multiple connections without "clogging up" the server.

- **Automatically Manages Parameters**

The Broker component completely and automatically manages parameters that are passed in the remote procedure call.

- **RPC Broker Programmer Preferences Program**

This version of the Broker includes a new developer tool for the client workstation (i.e., BrokerProgPref.EXE). The dialog box is displayed below:



This tool allows developers to initialize the default settings for the following properties of the TRPCBroker component:

- ClearParameters
- ClearResults
- ListenerPort
- Server

This program also gives developers the option of setting the "Connect in Delphi IDE". Most of the time, this setting should be checked. It will allow developers to connect to the server using the TRPCBroker component while designing the application in the Delphi Integrated Development Environment (IDE).

■ **RPC Broker Context-Sensitive Online Help**

This version of the Broker provides online context-sensitive help (i.e., BROKER.HLP) for the TRPCBroker component and its 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 the Broker developer (e.g., Tutorials, RPC information, Troubleshooting and Debugging tips, etc.).

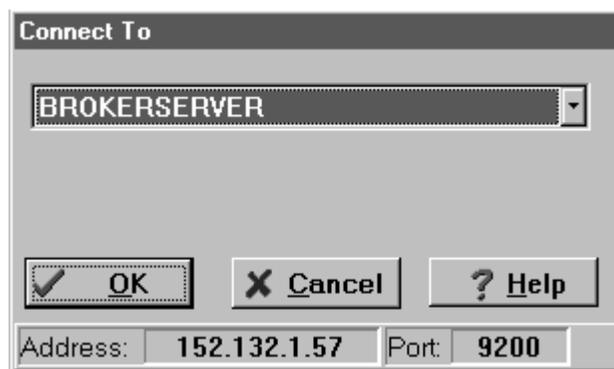
■ **Encryption and Decryption Functions**

This version of the Broker provides developers with the capability to encrypt and decrypt data as it is passed back and forth from the client to the server. Developers can use the encryption and decryption functions as implemented by Kernel and the RPC Broker. These encryption functions can be used for any communication between the client and the server where encryption is desired.

■ **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:



■ M Emulation Functions

This version of the Broker supplies the following M emulation functions:

- **Piece Function**

The Piece function is a scaled down Pascal version of M's \$PIECE function.

- **Translate Function**

The Translate function is a scaled down Pascal version of M's \$TRANSLATE function.

■ Improved Robustness and Efficiency of the Broker

We have made several improvements throughout the Broker to help reduce, if not eliminate, Broker-related errors and to improve the efficiency and error handling of the Broker:

- Using the 32-bit Windows environment should avoid the possibility of memory collisions. Each process runs in its own protected memory space.
- All of the memory pointers are initialized to nil before they are used and are reset after use.
- Careful analysis of Broker code identified and corrected potential memory leaks (i.e., memory not released after it was used).
- The **VISTA** signon form was reengineered. By simplifying its appearance and significantly reducing the number of window panels, less memory and fewer handles are required.
- The blocks of code that work with resources are now bracketed with protected blocks (i.e., Delphi Object Pascal **try...finally** constructs).

■ Debugging/Troubleshooting Your Application

Beside the normal debugging facilities provided by Delphi, you can also invoke a debug mode (i.e., TRPCBroker components DebugMode property) so that you can step through your code on the client side and your RPC code on the M server side simultaneously. In order to help isolate a problem, merely set the DebugMode property to true on the client and add any break points in the routines being called on the server. Just step through the code on your client and, simultaneously, step through the code on the server side for any RPCs that your client calls.

■ Dynamic Link Library (DLL) Interface

This version of the Broker provides a new set of Dynamic Link Library (DLL) functions that allow applications written in *any* MS Windows-based development environment (e.g., Borland's Delphi, Borland 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 Borland'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.



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



RPC BROKER DOCUMENTATION RESOURCES

A full set of documentation will be provided with the RPC Broker software package. Readers who wish to learn more about the RPC Broker should consult the following:

- *RPC Broker V. 1.1 Installation Guide*
- *RPC Broker V. 1.1 Systems Manual*
- *RPC Broker V. 1.1 Technical Manual*
- *RPC Broker V. 1.1 Getting Started with the Broker Development Kit* (written for programmers)
- *Online RPC Broker Developer's Guide* (i.e., BROKER.HLP, designed for programmers)
- RPC Broker Home Page at the following web address:

<http://www.vista.med.va.gov/softserv/infrastr.uct/broker/index.html>

This site contains additional information and documentation (e.g., Frequently Asked Questions [FAQs]) available in Hypertext Markup Language (HTML).

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

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



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

<http://www.vista.med.va.gov/softserv/infrastr.uct/acrobat/index.html>



RPC BROKER RELEASE NOTES

Version 1.1
September 1997

Department of Veterans Affairs
VISTA Software Development
OpenVISTA Product Line

Table of Contents

RPC Broker Version 1.1 Release Notes	1
 RPC Broker Overview	1
 New Features and Enhancements of the RPC Broker	1
End User Features & Enhancements.....	3
System Managers Feature & Enhancements	7
Developer Features & Enhancements.....	15
 RPC Broker Documentation Resources.....	23

Table of Contents