



VistALink

VERSION 1.0

Technical Manual and Package Security Guide

October 2, 2003

Department of Veterans Affairs
VistA Health Systems Design & Development (HSD&D)

Revision History

Table 1, below, summarizes the *VistALink Technical Manual and Package Security Guide* revision history.

Date	Revision	Description	Author(s)
April 18, 2003	N/A	Initial draft content	Foundations team
May 23, 2003	.9	Revisions from CPRS-R, VHA Technical Writers and VHA Testing received / implemented	Foundations team / Reviewers
October 2, 2003	1.0	Revisions from VistALink Development Team received / implemented	Foundations team

Table 1: *VistALink Technical Manual and Package Security Guide Revision History*

Contents

Revision History	i
Tables	v
Using This Manual / Guide	vi
Introduction.....	ix
Chapter 1: Implementation and Maintenance	1
1.1. Installing VistALink on a Developer Workstation.....	1
1.2. Site Parameters	1
1.2.1. Performance and Scalability	1
Chapter 2: Files.....	2
2.1 M Server Files	2
2.1.1 FOUNDATIONS SITE PARAMETERS (#18.01).....	2
2.1.2 VISTALINK LISTENER CONFIGURATION (#18.03)	2
2.1.3 VISTALINK LISTENER STARTUP LOG (#18.04)	2
2.1.4 VISTALINK MESSAGE TYPE FILE (#18.05).....	3
Chapter 3: Global Translation, Journaling and Protection.....	4
3.1 Translation.....	4
3.2 Journaling	4
3.3 Protection.....	4
Chapter 4: Routines.....	5
4.1 Routine Definitions	5
4.2 Routine Mapping	6
Chapter 5: Exported Options.....	7
Chapter 6: Exported RPCs	8
Chapter 7: Other Software Elements.....	10
7.1 Input Templates	10
7.2 List Template(s).....	10
Chapter 8: Callable Routines / Entry Points / APIs.....	11

Chapter 9: External Relations.....	15
9.1 VistA Software Requirements	15
9.2 DBA Approvals and Integration Agreements	15
Chapter 10: Internal Relations.....	17
10.1 Internal Relations	17
10.2 Namespace	17
Chapter 11: Software Product Security	18
11.1 VHA Directive on Modifications to VistA and VistA Sensitive Software.....	18
11.2 Official Policies	18
11.3 Legal Requirements	18
11.4 Mail Groups and Alerts.....	18
11.5 Remote Systems.....	18
11.6 Archiving/Purging.....	19
11.7 Contingency Planning.....	19
11.8 Interfacing	19
11.9 Menus.....	19
11.10 Security Keys.....	19
11.11 File Security.....	19
11.12 References	20
Chapter 12: Java Reference	21
12.1 jar: vljFoundationsLib.jar: Foundations Projects Utility Helper Classes.....	21
12.1.1 Package: gov.va.med.foundations.net	21
12.1.2 Package: gov.va.med.foundations.utilities	21
12.1.3 Package: gov.va.med.foundations.xml.....	22
12.2 jar: vljConnector.jar: VistALink connector classes.	22
12.2.1 Package: gov.va.med.foundations.adapter.cci.....	22
12.2.2 Package: gov.va.med.foundations.adapter.heartbeat.....	22
12.2.3 Package: gov.va.med.foundations.adapter.record	23

12.2.4	Package: gov.va.med.foundations.adapter.spi	24
12.2.5	Package: gov.va.med.foundations.rpc	24
12.3	jar: vljSecurity.jar: security classes.	25
12.3.1	Package: gov.va.med.foundations.security.vistalink	25
12.4	jar: vljSamples.jar: sample application classes.....	27
12.4.1	Package: gov.va.med.foundations.samples.....	27
Chapter 13: Java Environment and Third-Party Libraries Required		28
Index.....		29

Tables

Table 1: VistALink Technical Manual and Package Security Guide Revision History.....	i
Table 2: Documentation Symbol Descriptions.....	vi
Table 3: Commonly Used VistALink terms.....	vii
Table 4: VistALink Functionality Requiring Site Parameter Review	1
Table 5: VistALink Files.....	2
Table 6: VistALink Global Information.....	4
Table 7: External Relations—VistA Software	15
Table 8: VistALink File Security	19
Table 9: Required VistALink Classpath Libraries	28

Using This Manual / Guide

This manual / guide offers advice and instructions regarding the use of VistALink and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VistA) as a whole.

To highlight different aspects of its subject matter, this guide uses several methods. Various symbols are used throughout the documentation to alert the reader to special information. **Table 2**, below, gives a description of each of these symbols:



Symbol	Description
	Used to inform the reader of general information including references to additional reading material
	Used to caution the reader to take special notice of critical information

Table 2: Documentation Symbol Descriptions

- Descriptive text is presented in a proportional font (as represented by this font).
- "Snapshots" of computer online displays (i.e., roll-and-scroll screen captures/dialogs) and computer source code are shown in a *non*-proportional font and enclosed within a box. Also included are Graphical User Interface (GUI) Microsoft Windows images (i.e., dialogs or forms).
 - User responses to online prompts will be boldface type.
 - The word "**Enter**" found within these snapshots further prompts the user to press the **Enter** or **Return** key on their keyboard.
 - Author comments are displayed in italics or as "callout" boxes.



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

- Java code uses a combination of upper- and lowercase characters. All Java-reserved words in this manual appear in boldface type.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names and security keys (e.g., the XUPROGMODE key).

Common Terms

The terms and their descriptions in **Table 3**, below, may be helpful while reading this manual / guide.

Term	Description
Adapter / Connector	Exemplified by the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (J2EE Connectors for short), this represents a uniform way to integrate J2EE application servers with Enterprise Information Systems (EIS).
Authentication	Verifying the identity of the end-user.
Authorization	Checking the permissions of a user to allow or disallow the performance of some function.
Client	A single term used interchangeably to refer the <i>client</i> workstation (i.e., PC), and the <i>client</i> portion of the program that runs on the workstation.
J2SE	Java 2 Standard Edition. The blueprint for building Java applications.
M Server	The computer where MUMPS or M data and Remote Procedure Calls (RPCs) reside.

Table 3: Commonly Used VistALink terms

How to Obtain Technical Information Online



Methods of obtaining specific technical information online will be indicated in this manual (where applicable) under the appropriate topic.

Assumptions About the Reader

This manual is written with the assumption that readers have experience with the following:

- VistA computing environment (e.g., Kernel Installation and Distribution System [KIDS])

This manual makes no attempt 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 home pages on the World Wide Web (WWW) for a general orientation to VistA. For example, go to the System Design & Development (SD&D) Home Page at the following web address:

<http://vista.med.va.gov/>

Reference Materials

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

- *VistALink Installation Guide*, available at <http://www.va.gov/vdl/>.
- *VistALink Developer / System Manager Manual*, also available at <http://www.va.gov/vdl/>.
- <http://vista.med.va.gov/migration/foundations/Foundindex.htm>

The Foundations page (available at the URL above) provides announcements, additional information (e.g., Frequently Asked Questions [FAQs] or advisories), documentation links, archives of older documentation and software downloads.

VistALink documentation is made available online, on paper and in Adobe Acrobat Portable Document Format (.PDF). A .PDF must be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following URL or Web address:

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



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



DISCLAIMER: The appearance of external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Health Administration (VHA) of this Web site or the information, products or services contained therein. The VHA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of this VHA Intranet Service.

Introduction

This guide defines one of the three (3) VistALink v1.0 documentation deliverables:

1. Installation Guide
2. Developer / System Manager Manual
3. Technical Manual and Package Security Guide

*The **VistALink Installation Guide** details all the steps required for setting up the VistALink M software components. The **VistALink Developer / System Manager Manual** gives nationwide VHA application modernization teams background information and specific instructions focused on VistALink as a tool. For questions concerning the architecture and construction of VistALink, the **VistALink Technical Manual and Package Security Guide** provides qualified answers, as well as foundational information. Together, these three publications document the current state of VistALink and anticipate upcoming development.*

Background Information. VistALink is a transport layer that allows Java to communicate with M remote procedures. VistALink is completely based on standard technologies, both on the Java and M side.

Architectural Scope. VistALink v1.0 is designed to work with a standalone J2SE application. VistALink will be used by other VistA rehosting projects as a transport layer between Java and M. VistALink implements the Java Connector Architecture (JCA) 1.0 standard – v1.0 of VistALink implements only the non-managed / J2SE version of the JCA standard.

Functionality Scope. VistALink v1.0 provides:

- a. Communication capabilities for an M application request from a client J2SE Java application
- b. An authenticated communication transport layer from Java to M
- c. End-user authentication within Java applications based on M user accounts (J2SE only)
- d. Calling RPCs in a secure environment

VistALink FAQs.

For general or frequently asked questions (FAQs) about VistALink, please refer to the following web site:

<http://vista.med.va.gov/migration/foundations/FAQ.htm>

Chapter 1: Implementation and Maintenance

1.1. Installing VistALink on a Developer Workstation

Be sure to read the VistALink Installation Guide before attempting to install VistALink. The Installation Guide provides detailed installation information. It also contains requirements and recommendations for VistALink configuration.

1.2. Site Parameters

The functional areas of VistALink detailed in **Table 4**, below, require site parameter review and configuration:

Functional Area	For Detailed Information, Please See...
VistALink Listeners	Listener Management in the <i>VistALink Developer / System Manager Manual</i>

Table 4: VistALink Functionality Requiring Site Parameter Review

1.2.1. Performance and Scalability

Current performance statistics are limited. However, results indicate that the processing time and resources consumed by VistALink itself are minimal. VistALink introduces no additional overhead to the messages sent between the client and the server.

The VistALink listener delegates incoming requests to another process and then keeps listening for another request. Only the number of partitions supported by the M configuration limits this action.

Performance should be measured at the application level to determine the amount of resources consumed by the VistA client/server applications that use VistALink. Performance and scalability, from a site point of view, are impacted by the load introduced by the application executing on the host system, as opposed to the load introduced by VistALink itself.

Chapter 2: Files

2.1 M Server Files

VistALink on the M side currently consists of a single global with four files. This chapter describes VistALink files by file number, file name, global location and details the files themselves, as summarized in **Table 5**, below.

File #	File Name	Global Location
18.01	FOUNDATIONS SITE PARAMETERS	18.01 ^XOB(18.01,
18.03	VISTALINK LISTENER CONFIGURATION	18.03 ^XOB(18.03,
18.04	VISTALINK LISTENER STARTUP LOG	18.04 ^XOB(18.04,
18.05	VISTALINK MESSAGE TYPE FILE	18.05 ^XOB(18.05,

Table 5: VistALink Files

2.1.1 FOUNDATIONS SITE PARAMETERS (#18.01)

This file holds the site parameters for this installation of Foundations Management and VistALink. The file has only one entry (DINUM=1) and the .01 field points to a DOMAIN that represents the name of the installation site. *Data is not distributed with this file.*

2.1.2 VISTALINK LISTENER CONFIGURATION (#18.03)

This file holds VistALink listener configurations. A configuration may contain multiple listener ports; each listener port may be configured to start upon system startup.

Note: The configuration detail described here applies to Cache systems ONLY.

DSM / VMS systems need to use the TCP/IP (UCX) utility to have listeners automatically started on reboot.

This file is pointed to by the Foundations Site Parameters file (#18.01) and is used when obtaining the default listener configuration for a specified BOX-VOLUME pair. *Data is not distributed with this file.*

2.1.3 VISTALINK LISTENER STARTUP LOG (#18.04)

This file contains informational status pertaining to the current state of VistALink listeners for each BOX-VOLUME pair.

Note: This file only applies to Cache systems and is not utilized for DSM / VAX systems. Cache systems must use the TCP/IP (UCX) utility to start and stop VistALink listeners.

This log file is initially populated when one or more VistALink listeners are started. This will occur when either the Start VistaLink listener Configuration option [XOBV LISTENER STARTUP] is invoked, or by using the action protocols found under the Foundations Management Menu option [XOBV SITE SETUP MENU] to either start the BOX-VOLUME default configuration or start a single VistALink listener.

When a listener is stopped, the log file is edited with an updated status. Non-active listeners are deleted from the log file when the Clean Up Log action protocol is invoked. As mentioned, this action protocol as well as the action protocols previously described above can be found under the Foundations Management Menu option [XOBV SITE SETUP MENU]. ***Data is not distributed with this file.***

2.1.4 VISTALINK MESSAGE TYPE FILE (#18.05)

This file contains the message type definitions used by the VistALink request manager. Each message type is associated with a request handler.

The request manager uses TCP/IP input stream information to lookup entries by 'message type' or 'proprietary format indicator'. Once an entry has been found, the request manager can execute the correct request handler methods for the incoming request stream. Information contained in the entries of this file allows the request manager to dynamically determine which request handler to load and execute. ***Data is distributed with this file.***

Chapter 3: Global Translation, Journaling and Protection

3.1 Translation

Translation of the ^XOB global is recommended.

3.2 Journaling

Journaling of this global on the M side is not required.

3.3 Protection

Table 6, below, details the global protection that should be set on the M side.

Global Name	Protection	
	DSM for OpenVMS	Caché
^XOB	System: RWP World: RW Group: RW User: RW	Owner: RWD Group: N World: N Network: RWD

Table 6: VistALink Global Information

Chapter 4: Routines

4.1 Routine Definitions

XOBUENV

This routine contains environment checks for the VistALink package installation including a check for proper programmer variables, an operating system check, and a check for obsolete SYSTEM file #18. The environment check provides the ability to abort the installation if file #18 is present to allow sites to manually backup / delete this file before continuing with the installation. File deletion instructions are provided as well.

XOBUM

This routine is the main entry point for the XOBU FOUNDATIONS MAIN List Manager template invoked by the XOBU SITE SETUP MENU option.

XOBUM1

This routine contains various functions and procedures used by the action protocols under the XOBU FOUNDATIONS MAIN List Manager template.

XOBUPOST

This routine contains post-initialization code to add site domain parameters to the FOUNDATIONS SITE PARAMETERS file (#18.01).

XOBUPRE

This routine contains pre-initialization code to delete old versions of VistALink files, if they exist. It will also delete the obsolete SYSTEM File #18.

XOBVLIB

This routine contains application developer supported calls as well as Foundations Developer calls (unsupported) which handle XML processing and RPC timeout processing.

XOBVLL

This routine contains functions and procedures used to control the VistALink TCP/IP listener process.

XOBVLT

This routine contains test Remote Procedure Calls (RPCs) used to test VistALink connectivity, actions and APIs.

XOBVPOST

This routine contains post-initialization code to check for the OpenM operating system. If so, it will add VistALink listener Configuration parameters to the VISTALINK LISTENER CONFIGURATION file (#18.03) and also schedule the XOBV LISTENER STARTUP option to run upon system startup.

XOBVPRE

This routine contains pre-initialization code to delete the old version of the VISTALINK MESSAGE TYPE file (#18.05), if it exists.

XOBVRH

This routine sets up VistALink request handler runtime calls based on the request type.

XOBVRM

This routine contains the VistALink request manager. The request manager parses the incoming request and calls the appropriate request handler.

XOBVRMX

This routine parses incoming XML requests using the SAX parser interface.

XOBVRPC

This routine is the RPC request handler.

XOBVRPCI

This routine implements the VistALink interface for the RPC request message type.

XOBVRPCX

This routine parses RPC requests that are in proprietary format (i.e. not in XML request format).

XOBVTLS

This routine currently contains the RPC debugger programmer support tool but will be expanded to include additional tools in the future.

XOBSCAV

This routine contains code for the access/verify code security message request handler.

XOBSCAV1

This routine contains code that continues the access/verify code security message processing.

XOBSCAV2

This routine contains code that handles the access/verify code security message parsing.

XOBSCI

This routine implements the VistALink interface for the access/verify security message type.

XOBVSKT

This routine contains code to read to/from a VistALink TCP/IP socket.

XOBVSYSI

This routine implements the VistALink interface for the system request message type.

XOBVTCP

This routine contains code to start a VistALink listener using the action protocols under the XOBU FOUNDATIONS MAIN List Manager template.

XOBVTCPL

This routine contains the VistALink listener process code for Cache.

4.2 Routine Mapping

VistALink routines have no requirement to be mapped to any account.

Chapter 5: Exported Options

Name **XOBU SITE SETUP MENU**

Menu Text Foundations Management

Type action

Description This option invokes the Foundations Management ListManager template XOBU FOUNDATIONS MAIN. This ListManager application is the main Foundations Management menu. It provides configuration and management for VistALink listeners. The Foundations Management submenu is located on the Operations Management menu.

Name **XOBV LISTENER STARTUP**

Menu Text Start VistaLink listener Configuration

Type action

Description This option will get the VistALink listener default configuration for the BOX-VOLUME pair from the VISTALINK LISTENER CONFIGURATION file (#18.03) and start all applicable listeners contained within the configuration. This option is currently used by TaskMan for OpenM-NT systems only and is scheduled to run on system startup.

Name **XOBV VISTALINK TESTER**

Menu Text VistALink Tester

Type Broker (Client/Server)

Description This option is the VistALink Tester application context. This application is used to help developers learn how to use VistALink and also for site staff to test VistALink listener ports.

***Note:** To run the VistALink Tester application, option XOBV VISTALINK TESTER must be added to the secondary menu option field in the user's record in the NEW PERSON file (#200). Please refer to the VistALink Developer/System Manager Manual for more details on using the VistALink Tester application.*

Chapter 6: Exported RPCs

VistALink distributes the following RPCs as part of its VistALink Tester application.

XOBV TEST GLOBAL ARRAY

This simple RPC takes an array as input, sets the array into a global array and returns the global array information back to the client. It is used as part of the test and learning application distributed as part of VistALink.

XOBV TEST GLOBAL NODE

This simple RPC takes a string as input, sets it as a global node and returns the global node back to the client. It is used as part of the test and learning application distributed as part of VistALink.

XOBV TEST LOCAL ARRAY

This simple RPC takes an array as input, sets the array into a local array and returns the local array information back to the client. It is used as part of the test and learning application distributed as part of VistALink.

XOBV TEST NOT IN CONTEXT

This RPC should not be used and returns the empty string (null). It is used to test the 'not in RPC context' check. For this reason, this RPC is specifically NOT in the 'XOBV VISTALINK TESTER' B-type option in the OPTION (#19) file. It is used as part of the test and learning application distributed with VistALink.

XOBV TEST PING

This simple RPC takes no input and returns a success message. It is the primary mechanism to test whether the client successfully connects to an M server using VistALink. This RPC is used as part of the test and learning application distributed with VistALink.

XOBV TEST RPC LIST

This RPC returns a list of RPC names that begin with the characters indicated in the one input parameter. It is used as part of the test and learning application distributed with VistALink.

XOBV TEST STRING

This simple RPC takes a string as input, adds some additional text to the string and returns the string back to the client. It is used as part of the test and learning application distributed with VistALink.

XOBV TEST WORD PROCESSING

This simple RPC takes no input parameters and returns a 'chunk' of text. It is used to test whether VistALink properly transports 'chunks' of text. This RPC is used as part of the test and learning application distributed with VistALink.

XOBV TEST XML RESULTS

This simple RPC takes no input parameters and returns a 'chunk' of data that is in XML format. It is used to test whether VistALink properly transports and parses application results in XML format. This RPC is used as part of the test and learning application distributed with VistALink.

Chapter 7: Other Software Elements

7.1 Input Templates

Name	XOBU SITE PARAMETERS
File	FOUNDATIONS SITE PARAMETERS (#18.01)
Description	<p>This template is used to edit the Foundations and VistALink site parameters.</p> <p>The XOBU SITE PARAMETERS protocol executes this template.</p>

Name	XOBV LISTENER CONFIG EDIT
File	VISTALINK LISTENER CONFIGURATION (#18.03)
Description	<p>This template is used to add/edit a VistALink listener configuration.</p> <p>The XOBV LISTENER CONFIG protocol executes this template.</p>

7.2 List Template(s)

Name	XOBU FOUNDATIONS MAIN
Type of List	PROTOCOL
Protocol Menu	XOBU FOUNDATIONS MAIN MENU

Chapter 8: Callable Routines / Entry Points / APIs

The following supported reference calls allow other packages to access VistALink application developer calls. These calls address XML processing and RPC timeout handling.

XML processing call tags

Tag^Routine: \$\$XMLHDR^XOBVLIB()
Description: Provides the current XML standard header.

Tag^Routine: \$\$CHARCHK^XOBVLIB(STR)
Description: Performs character resolution to replace XML character limits with XML entities.

RPC timeout handling call tags

Tag^Routine: \$\$STOP^XOBVLIB()
Description: Used by the application to determine if processing should stop. Below are the steps an application would use:

RPC Time Out Process

Step	Where	Description
1	<i>Client</i>	Set time out property at RpcRequest.setRpcClientTimeOut(int) [int :: number of seconds]
2	<i>Server</i>	Periodically check if timed out or system needs for process to stop by calling \$\$STOP^XOBVLIB() [returns :: 1 - stop processing 0 - continue processing]
3	<i>Client</i>	Catch RpcTimeOutFaultException [Related DIALOG Entry :: 182007]

Tag^Routine: \$\$GETTO()

Description: Get the timeout value if it exists, or use the default value (currently 300 seconds) if it does not exist.

Tag^Routine: \$\$SETTO(TO)

Description: Used to set the RPC timeout value. Will override the current timeout setting from the client.

The following excerpt from Supported References Integration Agreement #4090 provides further information about these calls.

```
4090      NAME: VISTALINK SUPPORTED CALLS
CUSTODIAL PACKAGE: VISTALINK
Albany
SUBSCRIBING PACKAGE:
          USAGE: Supported          ENTERED: JUN 16,2003
          STATUS: Active            EXPIRES:
          DURATION: Till Otherwise Agr VERSION:
          DESCRIPTION:              TYPE: Routine
```

Supported reference to allow other packages to access VistALink application developer calls. These calls address XML processing and RPC timeout handling.

```
XML processing call tags
=====
$$XMLHDR()
$$CHARCHK(STR)
```

```
RPC timeout handling call tags
=====
$$STOP()
$$GETTO()
$$SETTO(TO)
```

ROUTINE: XOBVLIB

COMPONENT: \$\$XMLHDR()
Provides the current XML standard header.

Example:

```
DEV> W $$XMLHDR^XOBVLIB
<?xml version="1.0" encoding="utf-8" ?>
```

VARIABLES: Output \$\$XMLHDR()
Output is the current XML header.

COMPONENT: \$\$CHARCHK(STR)
 Performs character resolution to replace XML
 character limits with XML entities.
 Example:
 call DEV>W \$\$CHARCHK^XOBVLIB("This is an example of what
 <charchk> does.")

This is an example of what call <charchk> does.
 VARIABLES: Input STR
 Input:
 STR - string of characters to input
 VARIABLES: Output \$\$CHARCHK(STR)
 Output string of characters with XML
 entities replacing character limits.

COMPONENT: \$\$STOP()
 Used by the application to determine if processing
 should stop. Below are the development steps an
 application would use:

RPC Time Out Process
 =====

Step	Where	Description
====	=====	=====
1	Client	Set time out property at RpcRequest.setRpcClientTimeOut(int) [int :: number of seconds]
2	Server	Periodically check if timed out system needs for process to stop by calling \$\$STOP^XOBVLIB() [returns ;: 1 - stop processing 0 - continue processing]
3	Client	Catch RpcTimeOutFaultException [Related DIALOG Entry :: 182007]

or

VARIABLES: Output \$\$STOP()
 Indicator to stop processing (value = 1)
 or
 continue processing (value = 0).

COMPONENT: \$\$GETTO()
 Get the current timeout value (default = 300
 seconds).

VARIABLES: Output \$\$GETTO()
 Timeout value if it exists (in seconds)
 or
 default of 300 seconds.

COMPONENT: \$\$SETTO(TO)
 Used to reset the RPC timeout value. Will override
 the
 current timeout setting received from the client via
 RpcRequest.setRpcClientTimeout(int) or the default.

Note: In processing, the new timeout value will take
 effect the next time \$\$STOP^XOBVLIB() is called.

VARIABLES: Input TO
 TO is the RPC timeout value in seconds.

VARIABLES: Output \$\$SETTO(TO)
 Function sets the RPC timeout value (in
 seconds) and returns a 1 to indicate
 value
 successfully reset or 0 if not
 successful.

KEYWORDS: VISTALINK
 XML
 RPC

Chapter 9: External Relations

9.1 VistA Software Requirements

VistALink relies on the VistA software in **Table 7**, below, to run effectively. *The VistA software is listed alphabetically.*

Software	Version	Patch Information
Kernel	8.0	Fully patched, including: XU*8*238
Kernel Toolkit	7.3	Fully patched, including: XT*7.3*67
MailMan	8.0	Fully patched.
RPC Broker	1.1	Fully patched.
VA FileMan	22.0	Fully patched.

Table 7: External Relations—VistA Software

9.2 DBA Approvals and Integration Agreements

The Database Administrator (DBA) maintains a list of Integration Agreements (IAs) or mutual agreements between software developers allowing the use of internal entry points or other software-specific features that are not available to the general programming public.

To obtain the current list of IAs, if any, to which the VistALink software (XOB) is a custodian:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].
4. Select the Custodial Package Menu option [DBA IA CUSTODIAL MENU].
5. Choose the ACTIVE by Custodial Package option [DBA IA CUSTODIAL].
6. When this option prompts you for a package, enter **VISTALINK or VISTALINK SECURITY or XOB**.
7. All current IAs to which the VistALink software is a custodian are listed.

To obtain detailed information on a specific integration agreement:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].

4. Select the Inquire option [DBA IA INQUIRY].
5. When prompted for "INTEGRATION REFERENCES," enter the specific integration agreement number of the IA you would like to display.
6. The option then lists the full text of the IA you requested.

To obtain the current list of IAs, if any, to which the VistALink software is a subscriber:

1. Sign on to the FORUM system (forum.va.gov).
2. Go to the DBA menu [DBA].
3. Select the Integration Agreements Menu option [DBA IA ISC].
4. Select the Subscriber Package Menu option [DBA IA SUBSCRIBER MENU].
5. Choose the Print ACTIVE by Subscribing Package option [DBA IA SUBSCRIBER].
6. When prompted with "START WITH SUBSCRIBING PACKAGE," enter **XOB** (in uppercase). When prompted with "GO TO SUBSCRIBING PACKAGE," enter **XOBV** (in uppercase).
7. All current IAs to which the VistALink software is a subscriber are listed.

Chapter 10: Internal Relations

10.1 Internal Relations

For complete information on VistALink management i.e., starting VistALink listener(s), stopping VistALink listener(s), and all other listener management functions, please refer to Part II / Chapter 4 of the *VistALink Developer / System Manager Manual* entitled, “Managing the Listener.”

10.2 Namespace

VistALink has been assigned the **XOB** namespace.

Chapter 11: Software Product Security

11.1 VHA Directive on Modifications to VistA and VistA Sensitive Software

The pending *VHA Directive on Modifications to VistA and VistA Sensitive Software* prohibits modification of any part of the VistALink software.

11.2 Official Policies

There are no special official policies for this package other than the pending *VHA Directive on Modifications to VistA and VistA Sensitive Software*.

Distribution of VistALink software is unrestricted.

11.3 Legal Requirements

There are no special legal requirements that pertain to the use of VistALink.

11.4 Mail Groups and Alerts

VistALink does not make use of mail groups or alerts.

11.5 Remote Systems

VistALink's M listener process receives and processes connection requests from client applications. Connection by client applications is subject to Kernel authentication, as any normal logon requires.

To authenticate incoming connections to M, VistALink wraps the current authentication mechanism of the Kernel. To authorize incoming RPC requests, VistALink wraps the current RPC authorization mechanism of the RPC Broker.

Client applications can use any RPC Broker RPC authorized to a Kernel "B"-type option, if the "B"-type option is authorized to the signed-on user. Through this mechanism, data is typically exchanged between clients and the VistALink server. Clients can be anywhere on VA TCP/IP network.

RPC Broker security consists of four elements:

1. Client workstations must send a valid connection request to the M Server.
2. Users must have valid signon credentials recognized by Kernel (typically Access and Verify codes).
3. Users must be authorized M-system users of the application whose RPC calls are being invoked by the client application.
4. Any RPC must be registered and valid for the application being executed.

For more information regarding security enforced on RPCs, please click here <http://www.va.gov/vdl/> to access the *RPC Broker Systems Manual*.

11.6 Archiving/Purging

VistALink does not have any archiving or purging requirements or features. The Kernel collects sign-on log and auditing data created by VistALink user logons. VistALink uses Kernel APIs to log users on.

11.7 Contingency Planning

It is the responsibility of the using service to develop a local contingency plan for use in the event of application problems.

11.8 Interfacing

Please see the chapters on *External Relations* (M-server side) and *Java Environment and Third-Party Libraries Required* (Java client side) in this manual for a list of external packages required by VistALink.

11.9 Menus

There are no options of special note for Information Security Officers (ISOs) to view.

11.10 Security Keys

VistALink requires *no* specific security keys to be exported.

However, to bypass M authorizations for development purposes, client/server application developers may be assigned the XUPROGMODE security key. All users assigned the XUPROGMODE security key can do the following:

- Run any VistA client/server application, regardless of whether it is in their menu tree or not, and
- Access any RPC without regard to the application context.

11.11 File Security

VistALink establishes security over its files, as detailed in **Table 8**, below.

Number	Name	DD	RD	WR	DEL	LAYGO	AUDIT
18.01	FOUNDATIONS SITE PARAMETERS	@	@	@	@	@	@
18.03	VISTALINK LISTENER CONFIGURATION	@	@	@	@	@	@
18.04	VISTALINK LISTENER STARTUP LOG	@	@	@	@	@	@
18.05	VISTALINK MESSGE TYPE	@	@	@	@	@	@

Table 8: VistALink File Security

11.12 References

For more information regarding security enforced on RPCs, please click here <http://www.va.gov/vdl/> to access the *RPC Broker Systems Manual*.

For more information regarding Kernel's authentication and authorization infrastructure used by VistALink, please click here <http://www.va.gov/vdl/> to access the *Kernel Systems Manual*.

Chapter 12: Java Reference

VistALink Version 1.0 includes the Java classes detailed below.

12.1 jar: vljFoundationsLib.jar: Foundations Projects Utility Helper Classes

12.1.1 Package: gov.va.med.foundations.net

Description: Foundations TCP socket functionality for communicating with IP endpoints used by VistALink.

Classes:

gov.va.med.foundations.net.SocketManager – a wrapper around the socket that can be used to communicate with IP endpoints. Abstracts low-level socket communication details.

gov.va.med.foundations.net.VistaSocketException – exception thrown from SocketManager.

gov.va.med.foundations.net.VistaSocketTimeOutException – timeout exception thrown from SocketManager.

12.1.2 Package: gov.va.med.foundations.utilities

Description: General Foundations utility classes.

Classes:

AuditTimer - Gives an easy way to capture performance statistics.

ExceptionUtils - Utility methods for handling exceptions.

FoundationsException – Base exception class implements exception nesting.

FoundationsExceptionInterface – Represents the interface that all VistALink exceptions implement. Declares method that can be used for working with nested exceptions.

VistaKernelHash - Implements static methods to provide the encoding algorithms used by the RPC Broker and Kernel to encode and decode data strings.

VistaKernelHashCountLimitExceededException - Represents an exception identifying that the Hash Count Limit (for a call to the VistaKernelHash **encrypt** method) has been exceeded.

12.1.3 Package: gov.va.med.foundations.xml

Description: General Foundations utility class for working with XML.

Classes:

gov.va.med.foundations.xml.XmlUtilities - Exposes utility methods to work with XML.

12.2 jar: vljConnector.jar: VistALink connector classes.

12.2.1 Package: gov.va.med.foundations.adapter.cci

Description: Implements the Common Client Interface (CCI) portion of the Java Connector Architecture (JCA) for VistALink.

Classes:

VistaLinkConnection – This class represents an application level connection handle that is used by a component to access an EIS instance.

VistaLinkConnectionFactory – This implementation class provides an interface for getting connection to an EIS instance.

VistaLinkConnectionMetaData – Provides information about an EIS instance connected through a Connection instance.

VistaLinkConnectionSpec – This implementation class is used by an application component to pass connection-specific info/properties to the getConnection() method in VistaLinkConnectionFactory class.

VistaLinkResourceAdapterMetaData – Implementation class provides info about the capabilities of a resource adapter.

VistaLinkResourceException – Represents a ResourceException thrown by the VistALink adapter.

VistaLinkServerInfo – Represents M VistA connection information, like address and port.

12.2.2 Package: gov.va.med.foundations.adapter.heartbeat

Description: Implements the heartbeat keep-alive timer process for VistALink.

Classes:

HeartBeatFailedException – This exception class is used to notify the managed connection and its event listeners that a scheduled heartbeat has failed.

HeartBeatInteractionFailedException - This exception class is thrown when the heartbeat fails to make its first interaction to retrieve the heartbeat rate from M.

VistaHeartBeatTimerManager - This class manages the VistaHeartBeatTimerTask to synchronize all access to the Timer and the TimerTask.

VistaHeartBeatTimerRequest – This class represents a heart beat request.

VistaHeartBeatTimerResponse – This class is a value object that represents a response from the M heartbeat.

VistaHeartBeatTimerResponseFactory – This class represents the Response Factory used to create the VistaHeartBeatTimerResponse value object.

VistaHeartbeatTimerTask – This class represents the TimerTask used to run the heart beat tasks on a schedule.

12.2.3 Package: gov.va.med.foundations.adapter.record

Description: Implements basic request- and response-related classes for VistALink.

Classes:

LoginsDisabledFaultException – This exception represents the case where the M side has logins disabled - that is when the site sets the parameter to not allow any logins.

NoJobSlotsAvailableFaultException - This exception represents the case where on the M side there are no license slots available to start another process.

VistaLinkFaultException – Exception encapsulates Fault information coming from M side.

VistaLinkRequestFactory – Request factory interface to be used to construct request objects.

VistaLinkRequestVO – Base request interface.

VistaLinkRequestVOImpl – Base request implementation.

VistaLinkResponseFactory - Response factory interface used by the connection to parse response string into the response object.

VistaLinkResponseFactoryImpl – Base response factory implementation.

VistaLinkResponseVO – Base interface for response objects.

VistaLinkResponseVOImpl – Base ResponseVO interface implementation.

12.2.4 Package: gov.va.med.foundations.adapter.spi

Description: Implements the Service Provider Interface (spi) portion of the Java Connector Architecture (JCA) for VistaLink.

Classes:

ConnectionHandlesExceededException – This exception class is thrown when a VistaLinkManagedConnection object has exceeded its maximum allowable connection handles.

VistaLinkConnectionEventNotifier – This class manages a list of event listeners and notifies each of them when an event has been fired by a managed connection.

VistaLinkConnectionManager – The default ConnectionManager implementation for the non-managed(J2SE) scenario.

VistaLinkManagedConnection – This class represents a managed connection to M.

VistaLinkManagedConnectionEventListener – This class represents the default(J2SE) ConnectionEventListener that listens for the connectionClosed() and connectionErrorOccurred events fired by VistaLinkManagedConnection.

VistaLinkManagedConnectionFactory –An object of this class is a factory of both ManagedConnection and connection factory instances.

VistaLinkManagedConnectionMetaData – This class gives info about a VistaLinkManagedConnection.

VistaLinkSocketClosedException – This exception class is thrown when an attempt is made to access the VistaLinkManagedConnection's underlying VistaSocketConnection after its has been closed or invalidated.

VistaSocketConnection – Helper class that deals with lower level socket connectivity / reading / writing issues.

12.2.5 Package: gov.va.med.foundations.rpc

Description: Implements RPC requests and responses over VistaLink.

Classes:

NoRpcContextFaultException – This exception represents the case where the request RPC context does not exist or the current user does not have access to the B-option representing the context.

RpcFaultException – This fault exception class is used for all RPC-related errors returned from the M system.

RpcNotInContextFaultException – This exception represents the case where the requested RPC is not contained in the current RPC context.

RpcReferenceType – Represents a reference type object for an RPC parameter.

RpcRequest – Represents a RPC request to an M Vista server.

RpcRequestFactory – This class represents a factory that creates instances of RpcRequest.

RpcRequestParams – Represents the collection of parameters associated with an RPC.

RpcResponse – Represents a data structure that holds the response value(s).

RpcResponseFactory – Implements RPC specific response parsing logic.

RpcResponseTypeIsNotXmlException – Represents an exception indicating the RpcResponse type if not XML.

RpcTimeOutFaultException – This exception represents the case where an RPC execution takes too long on the server and the application gracefully stops the RPC's processing.

12.3 jar: vljSecurity.jar: security classes.

12.3.1 Package: gov.va.med.foundations.security.vistalink

Description: Security module for VistaLink; contains classes supporting a JAAS login to a Vista M system

Classes:

CallbackChangeVc – VistaLoginModule callback for marshalling user input for a "change verify code" event.

CallbackCommit - VistaLoginModule callback for system interactions during the commit phase of login.

CallbackConfirm – VistaLoginModule callback for marshalling user input for a "display error" or "display information" to user event.

CallbackHandlerSwing – Callback Handler for interactive login. Implements the JAAS CallbackHandler interface.

CallbackHandlerUnitTest – Callback Handler for unit testing non-interactive login. Implements the JAAS CallbackHandler interface.

CallbackLogon – VistaLoginModule callback for marshalling user input for a "enter access and verify code" event.

CallbackSelectDivision – VistaLoginModule callback for marshalling user input for a "select division" event.

DialogChangeVc – Swing Dialog to collect user input for a "change verify code" event.

DialogConfirm – Swing Dialog to collect user input for a "display error" or "display information" to user event.

DialogLogon – Swing Dialog to collect user input for a "enter access and verify code" event.

DialogSelectDivision – Swing Dialog to collect user input for a "select division" event.

SecurityDataChangeVcResponse – Implements response-specific fields for an AV.UpdateVC security message.

SecurityDataLogonResponse – Implements response-specific fields for an AV.Logon security message.

SecurityDataLogoutResponse – Implements response-specific fields for an AV.Logout security message.

SecurityDataSelectDivisionResponse – Implements response-specific fields for an AV.SelectDivision security message.

SecurityDataSetupAndIntroTextResponse – Implements response-specific fields for an AV.SetupAndIntroText security message.

SecurityDataUserDemographicsResponse – Implements response-specific fields for an AV.GetUserDemographics security message.

SecurityFaultException – This fault exception class is used for all security-related errors returned from the M system.

SecurityRequest – Represents a security request, to be sent to an M system for processing and response.

SecurityRequestFactory – Creates Security module message requests, to be sent to an M Vista server for processing and response.

SecurityResponse – Base class for response objects to receive and handle AVSecurity module responses from an M Vista server.

SecurityResponseFactory – Performs XML parsing on a security message returned from M, creating response objects of the correct class to match the type of response received, and parsing information from the message into the appropriate fields on the appropriate response object type.

SecurityTooManyInvalidLoginAttemptsFaultException – This exception fault is returned from M, and signifies that the user's login credentials were invalid too many times, and the M system is rejecting further login attempts as a result.

VistaInstitution – Represents a Vista Institution, including IEN, Station Name and Station Number.

VistaKernelPrincipal – Provides an interface to marks a principal that represents a logged on Kernel user on an M system.

VistaKernelPrincipalImpl – A JAAS principal representing a logged on Kernel user on an M system.

VistaLoginModule – A JAAS-compliant LoginModule to log users on to a Vista system.

VistaLoginModuleException – Represents a LoginException thrown by the LoginModule.

VistaLoginModuleLoginsDisabledException – If thrown, logins are disabled on the M server. When attempting a login, you can trap for this specific exception, in addition to the more general VistaLoginModuleException and LoginException exceptions.

VistaLoginModuleNoJobSlotsAvailableException – If thrown, job slot maximum has been exceeded on M server.

VistaLoginModuleNoPathToListenerException – If thrown, no reachable listener was found on the path represented by the specified IP address and Port.

VistaLoginModuleTooManyInvalidAttemptsException – If thrown, the user tried to login too many times with invalid credentials.

VistaLoginModuleUserCancelledException – Represents a user cancellation of Login.

VistaLoginModuleUserTimedOutException – Represents a user cancellation of Login.

VistaSetupAndIntroTextInfo – An object to hold information gathered from a SetupAndGetIntroText call, to then pass to a visual callback to present to the user when prompting for a/v codes.

12.4 jar: vljSamples.jar: sample application classes.

12.4.1 Package: gov.va.med.foundations.samples

Description: VistALink sample programs showing how to use VistALink to login and execute RPCs on an M system.

Classes:

DialogConfirm - Swing Dialog to collect user input for a "display error" or "display information" to user event.

VistaLinkRpcConsole – Console application demonstration of VistALink's Vista connectivity and RPC functionality.

VistaLinkRpcSwingSimple – simplified Swing-based developer example application.

VistaLinkRpcSwingTester – Full-featured Swing-based demonstration of VistALink's Vista connectivity and RPC functionality.

Chapter 13: Java Environment and Third-Party Libraries Required

The Java client portion of VistALink requires the following non-VA products to be installed on the client workstation for applications using VistALink:

J2SE Java Runtime Environment (JRE) v. 1.4.1:

The complete Java 2 Standard Edition (J2SE) environment, v. 1.4.1 or higher, must be installed on the client workstation. Obtain from <http://java.sun.com/>.

Other Required Supporting Libraries: The following libraries, detailed in **Table 9**, below, are distributed in Java Archive (JAR) files and must be installed / made available on the classpath of applications using VistALink:

Library	Minimum Version	Jar File Name and Description	Obtain From
J2EE core library	1.3.1	j2ee.jar Part of the Java 2 Enterprise Edition (J2EE) SDK	http://java.sun.com
Jaxen	1.0FCS	jaxen-full.jar Java XPath engine	http://sourceforge.net/projects/jaxen/
SAXPath	1.0FCS	saxpath.jar Simple API for XPath	Included in Jaxen distribution
JAXP - Compatible XML Parser	JAXP 1.1	(various) Any XML parser that implements the JAXP interface. For example: <ul style="list-style-type: none"> • Xerces • Crimson • Oracle XDK 	Xerces: Included in Jaxen (xerces.jar) Crimson: http://xml.apache.org/crimson/ Oracle XDK: http://technet.oracle.com/tech/xml/
JAXP - Compatible XSLT Processor	JAXP 1.1	(various) Any XSLT processor that implements the JAXP interface. For example: <ul style="list-style-type: none"> • Xalan-Java • Saxon 	Xalan-Java: included in J2EE.jar (v. 1.3.1), or http://xml.apache.org/xalan-j/ Saxon: http://saxon.sourceforge.net/
Log4J	1.2.7	log4j-x.x.x.jar (optional) Java Logging Utility. Note: Log4J is not required by VistALink, but will be used if present.	http://jakarta.apache.org/log4j/docs/

Table 9: Required VistALink Classpath Libraries

Index

- ACTIVE by Custodial Package Option, 15
- Assumptions About the Reader, vii
- Commonly Used Terms, vii
- Custodial Package Menu, 15
- DBA Approvals and Integration Agreements, 15
- DBA IA CUSTODIAL MENU, 15
- DBA IA CUSTODIAL Option, 15
- DBA IA INQUIRY Option, 15
- DBA IA ISC Menu, 15, 16
- DBA IA SUBSCRIBER MENU, 16
- DBA IA SUBSCRIBER Option, 16
- DBA Menu, 15, 16
- Documentation Symbols, vi
- Global Journaling, 4
- Home Pages
 - Adobe Acrobat Quick Guide Web Address, viii
 - Adobe Systems Incorporated Web Address, viii
 - RPC Broker FAQs Home Page Web Address, ix
 - SD&D Home Page Web Address, vii
- Inquire Option, 15
- Integration Agreements, 15
 - Menu Option, 15
- Integration Agreements Menu Option, 15, 16
- Journaling, 4
- Kernel Patches
 - XU*8.0*186, 15
- Mapping (Routine), 6
- Menus
 - Custodial Package Menu, 15
 - DBA, 15, 16
 - DBA IA CUSTODIAL MENU, 15
 - DBA IA ISC, 15, 16
 - DBA IA SUBSCRIBER MENU, 16
 - DBA Option, 15, 16
 - Integration Agreements Menu, 15, 16
 - Subscriber Package Menu, 16
- Options
 - ACTIVE by Custodial Package, 15
 - Custodial Package Menu, 15
 - DBA, 15, 16
 - DBA IA CUSTODIAL, 15
 - DBA IA CUSTODIAL MENU, 15
 - DBA IA INQUIRY, 15
 - DBA IA ISC, 15, 16
 - DBA IA SUBSCRIBER MENU, 16
 - DBA IA SUBSCRIBER Option, 16
 - DBA Option, 15, 16
 - Inquire, 15
 - Integration Agreements Menu, 15, 16
 - Print ACTIVE by Subscribing Package, 16
 - Subscriber Package Menu, 16
- Patches
 - XU*8.0*186, 15
- Performance, 1
- Print ACTIVE by Subscribing Package Option, 16
- Reader, Assumptions About the, vii
- Reference Materials, viii
- Routine Mapping, 6
- Scalability, 1
- Site Parameters, 1
- Subscriber Package Menu Option, 16
- Symbols Found in the Documentation, vi
- Terms, Commonly Used, vii
- URLs
 - Adobe Acrobat Quick Guide Web Address, viii
 - Adobe Systems Incorporated Web Address, viii
 - RPC Broker FAQs Home Page Web Address, ix
 - SD&D Home Page Web Address, vii
- VistA Software Requirements, 15
- Web Pages
 - Adobe Acrobat Quick Guide Web Address, viii
 - Adobe Systems Incorporated Web Address, viii
 - RPC Broker FAQs Home Page Web Address, ix
 - SD&D Home Page Web Address, vii