VISTALINK
INSTALLATION GUIDE

Version 1.6

July 2020

Department of Veterans Affairs
Office of Information and Technology
Product Development

Revision History

Date	Description	Author
07/08/20	XOBV*1.6*5 – VistALink Version 1.6.1 release	Health Product Support Tier 3 Sustainment team. Developer – Arsen Mikhailutsa Developer – Justin Loverme Technical Writer – Adriana Quintero
10/24/19	XOBV/S*1.6*4 N/A Changes	Dan Pendergrass
12/03/10	VistALink Version 1.6 release.	Product Development Services Security Program VistALink development team. Albany, NY OIFO: • Developer—Mike Kilmade • Developer—Liz Defibaugh Bay Pines, FL OIFO: • Development Manager—Charles Swartz • Tester—Jason Woodyard Oakland, CA OIFO: • Developer—Kyle Clarke • SQA—Gurbir Singh Technical Writer—Susan Strack
		•

Table i. Revision History

Contents

Revision History ii
Contents iii
Tables vi
Figures vi
Orientation viii
Document Overview viii
Additional Resources ix
1. Introduction 1-1
1.1. About VistALink 1-1
1.1.1. WebLogic Updates Project 1-1
1.2. VistALink Version Compatibility 1-1
1.2.1. J2EE/WebLogic Version Compatibility 1-1
1.2.2. M Listener Backwards/Forwards Version Compatibility 1-2
1.3. Known Issues and Limitations 1-2
2. Installation Overview 2-1
2.1. Restrictions 2-1
2.2. Assumptions about Installers 2-1
2.3. Separation of M and J2EE Server Installation Procedures 2-1
2.4. VistALink Distribution ZIP File <DIST FOLDER> Structure (new structure) 2-1
2.5. M Routine Checksum Information 2-2
2.6. Installation Synopsis 2-2
2.6.1. VistA/M Server 2-2
2.6.2. J2EE Application Server 2-2
3. VistA/M Server Installation Procedures 3-1
3.1. Preparation 3-1
3.1.1. Software Installation Time 3-1
3.1.2. Virgin Installations 3-1
3.1.3. System Requirements 3-1
3.1.4. System Preparation 3-2
3.1.5. HFS and Null Devices 3-3
3.1.6. Deletion of Obsolete File #18 3-3
3.2. Install VistALink KIDS Distribution 3-3

3.2.1. Preliminary Steps 3-3
3.2.2. Stop VistALink System Processes 3-4
3.2.3. Install KIDS Distribution 3-4
3.3. (Optional) Configure VistALink Listener 3-10
3.3.1. Do I Need to Configure Listeners As Part of the VistALink Installation? 3-10
3.3.2. Listener Introduction 3-10
3.3.3. Recommended VistALink Ports (all operating systems) 3-11
3.3.4. OS-Based Listener Configuration for Caché/VMS Systems 3-11
3.3.5. OS-Based Listener Configuration for Caché/Linux Systems 3-12
3.3.6. M-Based Listener Configuration for Caché/NT (Windows) Systems 3-13
3.4. (Optional) Verify Listener Connectivity 3-13
3.4.1. Telnet Test 3-14
3.4.2. VistALink J2SE SwingTester Sample Application Test (optional) 3-14
3.5. (Optional) Configure Connector Proxy User(s) for J2EE Access 3-15
3.5.1. Connector Proxy Overview 3-15
3.5.2. How to Create Connector Proxy User Kernel Accounts 3-15
3.6. Installation Back-Out/Roll-Back Procedure 3-15
3.6.1. Reinstall v1.5 3-15
3.6.2. Optional Deletions of v1.6-Only Components 3-16
4. Oracle WebLogic Application Server: Installation Procedures 4-1
4.1. Overview 4-1
4.1.1. Adapter Deployment Descriptors 4-1
4.1.2. VistALink 1.6.1 Adapter Changes 4-1
4.1.3. VistALink Adapters and Classloading 4-1
4.2. Preparation 4-2
4.2.1. Software Installation Time (Varies) 4-2
4.2.2. System Requirements 4-2
4.2.3. Deployer Requirements 4-2
4.2.4. Obtain the VistALink Distribution File 4-2
4.2.5. Obtain M Connector Proxy User and Listener Information 4-3
4.3. Upgrading a WebLogic 8.1 Domain w/Existing VistALink Adapters 4-3
4.3.1. Back Up Exploded RAR Directories and VistALink Configuration File 4-3
4.3.2. If Running the Domain Upgrade Wizard 4-3
4.4. WebLogic 10.3.6/12.1.2 Server Configuration 4-4

4.4.1. Create <HEV Configuration Folder> 4-4
4.4.2. Create/Copy VistALink Configuration File 4-4
4.4.3. Place <HEV Configuration Folder> on Server Classpath(s) 4-5
4.4.4. Create/Update Server log4j Configurations 4-6
4.4.5. Server JVM Argument: gov.va.med.environment.production 4-6
4.4.6. Server JVM Argument: gov.va.med.environment.servertype 4-7
4.5. WebLogic 10.3.6/12.1.2: Install the Standalone Console EAR (Admin Server) 4-8
4.5.1. Copy Console EAR file 4-8
4.5.2. Deploy Console EAR 4-8
4.5.3. Access Standalone VistALink Console 4-8
4.5.4. Check Configuration Editor Access to Configuration File 4-9
4.6. Deploy Shared J2EE Libraries (Production Domains Only) 4-9
4.7. Create/Deploy VistALink Adapter(s) 4-10
4.7.1. Add Connector Entry to VistALink Configuration File 4-10
4.7.2. Create New or Update Existing Adapter Folder on Admin Server 4-11
4.7.3. Back Up Deployment Descriptors 4-11
4.7.4. Copy New 1.6 Files 4-11
4.7.5. Update Deployment Descriptors 4-11
4.7.6. Deploy Adapter 4-14
4.7.7. Monitor Adapter in VistALink Console 4-14
4.8. Troubleshooting 4-14
4.9. Test with J2EE Sample Application (Development Systems Only) 4-15
4.9.1. Deploy the Sample Web Application 4-15
Appendix A: Installing and Running the J2SE Sample Apps 1
Overview 1
Installation Instructions 1
Appendix B: DSM/VMS-Specific Install Information 1
Operating System Requirements 1
Global Protection 1
Listener Management for Caché/VMS Systems 1
Glossary 1

Tables

Table i. Revision History ii
Table 3-1. VistA Software Dependencies for VistALink 1.6 installation 3-1
Table 3-2. VistALink 1.6 file and global installation 3-2
Table 3-3. Global protection 3-3
Table A-6. VistALink Sample Application Loggers ..... Appendix A-9

Figures

Figure iii. Documentation symbol descriptions ix
Figure 2-1. Directory structure of the VistALink 1.6 Distribution ZIP File 2-2
Figure 3-1. KIDS Installation option: Verify Checksums in Transport Global [XPD PRINT CHECKSUM] 3-6
Figure 3-2. KIDS Installation option: Backup a Transport Global [XPD BACKUP] 3-6
Figure 3-3. VistALink J2M Installation Example 3-9
Figure 3-4: Example XINETD Service Configuration 3-13
Figure 4-1. Admin Server: Add the classpath folder to the server classpath in the setDomainEnv script 4-5
Figure 4-3. Standalone VistALink 1.6 Console 4-9
Figure 4-4. weblogic-ra.xml sample deployment descriptor 4-13
Figure 4-5. VistALink Sample Application 4-16
Figure 4-6. VistALink Sample Application Re-authentication Page 4-17
Figure 4-7. VistALink J2EE Sample Application Results Page 4-18
Figure A-2. Test Program Access/Verify Code Entry 3
Figure A-3. SwingTester RPC List 4
Figure A-4. Test Program User Information 5
Figure A-5. log4jconfig.xml file contains extensive information on log4j configuration options 7
Figure B-1. Global protection 1

This page is left blank intentionally.

Orientation

Document Overview

This manual provides information for installing the VistALink 1.6.1 resource adapter and Mumps (M)-side listener. Its intended audience includes Java 2 Enterprise Edition (J2EE) application server administrators, Information Resource Management (IRM) Information Technology (IT) Specialists at Department of Veterans Affairs (VA) facilities, and developers of Java applications requiring communication with Veterans Health Information Systems and Technology Architecture (VistA)/M (Massachusetts General Hospital Utility Multi-Programming System) systems.

System administrators and developers should use this document in conjunction with the *VistALink 1.6 System Management Guide*, which contains detailed information on the Java 2 Platform, Enterprise Edition (J2EE) application server management, institution mapping, the VistALink console, M listener management, and VistALink security, logging, and troubleshooting.

Terminology

The term *resource adapter* is often shortened in this guide to "adapter," and is also used interchangeably with the term *connector*.

Text Conventions

File names and directory names are set off from other text using bold font (e.g., **config.xml**). Bold is also used to indicate Graphical User Interface (GUI) elements, such as tab, field, and button names (e.g., "press **Delete**").

All caps are used to indicate M routines and option names (e.g., XMINET). All caps used inside angle brackets indicate file names to be supplied by the user. Example:

<JAVA_HOME>\bin\java -Dlog4j.configuration=file:///c:/localConfigs/mylog4j.xml

Names for Java objects, methods, and variables are indicated by Courier font. Snapshots of computer displays also appear in Courier, surrounded by a border:

```
Select Installation Option: LOAD a Distribution
Enter a Host File: XOB_1_6_Bxx.KID
```

In these examples, the response that the user enters at a prompt appears in bold font:

```
Enter the Device you want to print the Install messages.
You can queue the install by enter a 'Q' at the device prompt.
Enter a '^' to abort the install.

    DEVICE: HOME// HOME;80;999 <Enter> TELNET PORT
```

Boldface text is also used in code and file samples to indicate lines of particular interest, discussed in the preceding text:

```
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bea.com/ns/weblogic/90
http://www.bea.com/ns/weblogic/90/weblogic-ra.xsd">

   <!-- For new ADAPTER-level jndi-name, recommend using value of
connection instance JNDI name, appended with "Adapter" -->
   <jndi-name>vljtestconnectorAdapter</jndi-name>
```

The following symbols appear throughout the documentation to alert the reader to special information or conditions.

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

Figure iii. Documentation symbol descriptions

Folder Conventions

The following logical folder names are used in the J2EE Installation section:

<DIST FOLDER>	The location for the unzipped VistALink distribution file.
<HEV CONFIGURATION FOLDER>	A folder placed on the classpath of WebLogic servers, containing configuration files for all HealtheVet-VistA applications.

Additional Resources

Product Web Site

The VistALink product website (http://vista.med.va.gov/vistalink) summarizes VistALink architecture and functionality and presents status updates.

VistALink Documentation Set

The following is the VistALink 1.6 end-user documentation set, which can be downloaded from the Department of Veterans Affairs (VA) Software Document Library (VDL) Web site at:

http://www.va.gov/vdl/application.asp?appid=163

- *VistALink 1.6 Installation Guide* (this manual): Provides detailed instructions for setting up, installing, and configuring the VistALink 1.6 listener on VistA/M servers and the VistALink resource adapter on J2EE application servers. Its intended audience includes server administrators, IRM IT specialists, and Java application developers.

- *VistALink 1.6 System Management Guide*: Contains detailed information on J2EE application server management, institution mapping, the VistALink console, M listener management, and VistALink security, logging, and troubleshooting.

- *VistALink 1.6 Developer Guide*: Contains detailed information about workstation setup, re-authentication, institution mapping, executing requests, VistALink exceptions, Foundations Library utilities, and other topics pertaining to writing code that uses VistALink.

- *VistALink 1.6 Release Notes*: Lists all new features included in the VistALink 1.6 release.

VistALink 1.6 end-user documentation and software can be downloaded any of the **anonymous.software** directories on the Office of Information & Technology (OI&T) File Transfer Protocol (FTP) download sites:

- Preferred Method download.vista.med.va.gov

  This method transmits the files from the first available FTP server.
- Albany OIFO ftp://ftp.fo-albany.med.va.gov/
- Hines OIFO ftp://ftp.fo-hines.med.va.gov/
- Salt Lake City OIFO ftp://ftp.fo-slc.med.va.gov/

The documentation is made available online in Microsoft Word format and Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following Web address:

http://www.adobe.com/

DISCLAIMER: The appearance of any external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA 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 VA Intranet Service.

This page is left blank intentionally.

1. Introduction

1.1. About VistALink

The VistALink resource adapter is a transport layer that provides communication between HealtheVet-Veterans Information Systems and Technology Architecture (VistA) Java applications and VistA/M servers, in both client-server and n-tier environments. It is a runtime and development tool that allows java applications to execute remote procedure calls (RPCs) on the VistA/M system and retrieve results, synchronously. VistALink is also referred to as VistALink J2M.

VistALink consists of Java-side adapter libraries and an M-side listener:

- The adapter libraries use the J2EE Connector Architecture (J2CA) 1.7 specification to integrate Java applications with legacy systems.
- The M listener process receives and processes requests from client applications.
- Java applications can call Remote Procedure Calls (RPCs) on the M server, executing RPC Broker RPCs on the M server without modification.

The previous version of VistALink, 1.5, was released in June of 2006, and provided project developers with J2EE and Java Platform, Standard Edition (J2SE) application connectivity to VistA/M servers. It was designed specifically for J2EE 1.3 application servers (e.g., WebLogic 8.1).

1.1.1. WebLogic Updates Project

In support of the Department of Veterans Affairs Information Technology application Modernization effort, the three applications Fat-client Kernel Authentication and Authorization (FatKAAT), Kernel Authentication and Authorization for the Java 2 Enterprise Edition (KAAJEE) and VistALink have been developed. Based on the direction of the Technical Review Model (TRM) and to support applications that upgrade to the new WebLogic Server versions. 10.3.6/12.1.2, this project is required. The scope of the project is to upgrade these three applications to work with the WebLogic Server 10.3.6/12.1.2.

1.2. VistALink Version Compatibility

1.2.1. J2EE/WebLogic Version Compatibility

Significant changes to the J2CA specification were made in J2EE 1.7, and additional changes in WebLogic classes (e.g., console extensions) were also made for WebLogic 10.3.6/12.1.2. As a result, some components of VistALink 1.6.1 are not compatible with WebLogic 10.3.6/12.1.2. All components of VistALink 1.6.1 are compatible with WebLogic 10.3.6/12.1.2.

VistALink version	J2EE 1.4+ WebLogic 9.x, 10.x, 11g	J2EE 1.7 WebLogic 10.3.6/12.1.2
1.6.1	**yes**	*no*

1.6.1	*No*	**Yes**

1.2.2. M Listener Backwards/Forwards Version Compatibility

The 1.5 and 1.6 M listeners are backwards and forwards compatible, as follows:

- 1.6 clients **cannot** execute requests against 1.5 M listeners
- 1.5 clients can execute requests against 1.6 M listeners
- 1.0 clients can execute requests against 1.5 and 1.6 listeners

1.3. Known Issues and Limitations

- VistALink console plug-in on WebLogic v10.0: In WebLogic v10.0, there is no navigation link for the VistALink console extension in the WebLogic console navigation tree (left hand side of the console). A possible bug has been reported to Oracle (formerly BEA).

  Workaround: An alternate route to the VistALink console is to click on the top link of the navigation tree, which is the domain name. On the right-hand page, one of the tabs is 'VistALink J2M'.

- VistALink console plug-in on WebLogic v10.3: In WebLogic v10.3, the navigation link and tab link to access the VistALink console extension may not be displayed in the WebLogic console on some systems, upon subsequent logins after initial deployment, leaving the console extension inaccessible.

  Workaround: An alternative version of the VistALink console has been provided as a standalone EAR. Use the standalone EAR version of the VistALink console for WebLogic 10.3 (and any other future version of WebLogic that has the same problem).

- Anomaly: Unexplained Production/Test Mismatch Error During Testing: One unexplained anomaly was reported during testing with CHDR 2.0. VistALink connections to VistA sites began failing on one of the 6 CHDR WebLogic servers, with the logger error being a production/test mismatch, where VistALink requests were incorrectly reporting that the CHDR server in question was not a production server. The setting used by VistALink to determine if a given WebLogic server is test or production is a server-specific Java Virtual Machine (JVM) configuration argument configured by the data center. The argument appeared to be set correctly on the server in this case.

  The anomaly has occurred once on one server, after 5 months of running in production. The impact was that the affected WebLogic server could not access production VistA servers, and that each failed connection attempt added an error to each VistA site's error log. After a number of server restarts, and examinations / possible updates to the server configuration, the problem resolved itself. Without a deeper investigation, it was not possible to isolate which system component was responsible for the observed failure.

  Workaround: None.

This page is left blank intentionally.

2. Installation Overview

2.1. Restrictions

VistALink 1.6.1 has been tested and is supported on Oracle WebLogic Server 10.3.6/12.1.2, only.

2.2. Assumptions about Installers

These instructions assume that installers will have a basic working knowledge of J2EE and M systems, including application deployments.

2.3. Separation of M and J2EE Server Installation Procedures

This guide provides VistALink installation instructions. Because VistALink consists of modules for both a Java 2 Enterprise Edition (J2EE) application server and a VistA/M server, separate sets of instructions are provided to set up, configure, and install the appropriate module(s) on each type of server.

At production facilities in particular, different administrators may be responsible for the two server types (M and J2EE); thus, separate parts of the installation process. At such sites, completing both sides of a VistALink installation will require ongoing communication and coordination between the two types of system administrators. Developers, on the other hand, may be responsible for both sides of the installation process, M *and* J2EE.

Though the VistA/M server instructions are presented first in this document, the order is arbitrary—most of the steps for the two servers are not dependent on each other.

2.4. VistALink Distribution ZIP File <DIST FOLDER> Structure (new structure)

The VistALink distribution ZIP file contains:

Directory Structure of the VistALink 1.6 Distribution ZIP File

```
/vlj-1.6.1.xxx

     /app-j2ee                Application components for J2EE installation
         /configFile-j2ee     sample gov.va.med.vistalink.connectorConfig.xml
                              configuration file
         /console-ext         Console plug-ins and standalone EAR version
         /Rar-Dev-Template    RAR for development systems
         /Rar-Prod-Template   RAR for production systems
         /sample              J2EE sample application
         /shared-lib          shared libraries for production systems

     /javadoc                 javadoc for public java-side VistALink APIs
```

```
/lib-deprecated         contains supporting jar no longer needed in most
                        cases
/log4j                  configuration file examples, VistALink logger
                        spreadsheet
/m                      KIDS distribution containing M side of VistALink
/rpc-doc                extract of RPC Broker documentation on how to write
                        RPCs
/samples-J2SE           sample J2SE rich client applications
```

Figure 2-1. Directory structure of the VistALink 1.6 Distribution ZIP File

2.5. M Routine Checksum Information

The routine name and corresponding checksum value for each M routine contained within the VistALink 1.6.1 software package is provided in the README.TXT file in the <DIST_FOLDER>'s root folder.

2.6. Installation Synopsis

2.6.1. VistA/M Server

The detailed instructions for installing VistALink on the VistA/M server are presented in chapter 3, "VistA/M Server Installation Procedures." The general steps for installing VistALink on the VistA/M server are as follows:

1. Preparation
2. Install VistALink Kernel Installation and Distribution System (KIDS) Distribution
3. (Optional) Configure VistALink Listener – **not necessary when upgrading an existing configuration**
4. (Optional) Verify Listener Connectivity
5. (Optional) Configure Connector Proxy User(s) for J2EE Access **– not necessary when upgrading an existing configuration**

2.6.2. J2EE Application Server

The detailed instructions for installing VistALink on the J2EE application server are presented in chapter 4, "Oracle WebLogic Application Server: Installation Procedures." The general steps for installing VistALink on the J2EE application are as follows:

1. Preparation
2. Upgrading a Previous Installation
3. Server Preparation
4. Install the Console Plug-In or Standalone Console (Admin Server)
5. Create/Deploy VistALink Adapters
6. Test with J2EE Sample Application (Development Systems Only)

3. VistA/M Server Installation Procedures

3.1. Preparation

3.1.1. Software Installation Time

The estimated time for the installation of the VistALink KIDS distribution is less than five minutes.

3.1.2. Virgin Installations

It is not necessary for a previous version of VistALink to be installed on your VistA/M server before you install VistALink 1.6.

3.1.3. System Requirements

3.1.3.1. Patch Requirements

Before the VistALink 1.6.1 installation, the following packages and patches must be installed:

Software	Version	Patch Information
Kernel	8.0	Fully patched.
Kernel Toolkit	7.3	Fully patched.
MailMan	8.0	Fully patched.
RPC Broker	1.1	Fully patched.
VA FileMan	22.2	Fully patched.

Table 3-1. VistA Software Dependencies for VistALink 1.6 installation

3.1.3.2. Operating System Requirements

- Caché/Linux: Caché (version 2014.1.3 or greater)

3.1.3.3. VistA/M Server Permissions

Kernel-level programmer access (DUZ(0)="@") is required for installing VistALink 1.6.

On a Virtual Memory System (VMS), the installer must have a VMS account. Installers who are also configuring Transmission Control Protocol (TCP) services for VistALink listeners must also hold sufficient VMS privileges (e.g., SYSPRV).

3.1.3.4. Namespaces

VistALink has been assigned the XOB* namespace.

3.1.3.5. File and Global Information

VistALink 1.6.1 installs the following files:

File #	File Name	Root Global	FileMan Protection
18.01	FOUNDATIONS SITE PARAMETERS	^XOB(18.01,	@
18.03	VISTALINK LISTENER CONFIGURATION	^XOB(18.03,	@
18.04	VISTALINK LISTENER STARTUP LOG	^XOB(18.04,	@
18.05	VISTALINK MESSAGE TYPE	^XOB(18.05,	@

Table 3-2. VistALink 1.6 file and global installation

3.1.4. System Preparation

3.1.4.1. Global Placement, Mapping, and Translation

VistALink utilizes one VistALink-specific global, ^XOB. For virgin installs, ^XOB should be placed in a database location appropriate for a small, static global, prior to installation.

For M configurations with multiple databases or volume sets, any necessary mapping or translation should be set up at this time as well.

3.1.4.2. Journaling

Because the ^XOB global is relatively static, journaling of this global is not required.

3.1.4.3. Global Protection

Global Name	Caché	
^XOB	Owner:	RWD
	Group:	N
	World:	N
	Network:	RWD

Table 3-3. Global protection

3.1.5. HFS and Null Devices

Verify that you have a Host File Server (HFS) device named "HFS" and a Null device named "NULL" in the DEVICE file (#3.5).

NOTE: You can have other devices with similar names, but one device is needed whose name or mnemonic is "NULL."

3.1.6. Deletion of Obsolete File #18

During the original testing of VistALink 1.0, it was discovered that some sites might still have an old Kernel file residing on their system called SYSTEM file (#18). To support virgin installs, VistALink 1.6 still includes steps to check and clean up File #18.

If your system already has VistALink 1.0 or 1.5 installed, this file has already been removed. Otherwise, if present on your system, you may wish to manually back up and delete SYSTEM file (#18). If this file is on your system at the time of installing VistALink 1.6.1, the environment check will delete the file for you.

NOTE: This file was created in the early 1980s and was a precursor to the current KERNEL SYSTEM PARAMETERS file (#8989.3). However, it is now obsolete and must be removed from your system before the VistALink package can be installed, because it shares the same number space that VistALink was assigned.

3.2. Install VistALink KIDS Distribution

Follow the steps in this section to install VistALink KIDS distribution.

3.2.1. Preliminary Steps

1. Obtain the VistALink KIDS distribution. Download either the entire VistALink ZIP distribution file (XOB_1.6.1.xx.ZIP), or just the standalone KIDS build (XOB_1_6_Bxx.KID) from the **anonymous.software** directory on any of the OI&T FTP download sites. For 2FA functionality

support refer to the XOB_1P6_3.KID build and patch description.

NOTE: If you download the entire ZIP distribution, after unzipping it, the KIDS build is located in the unzipped **m** subfolder.

2. FTP (or otherwise transfer) the KIDS build file to the intended VistA/M server.
3. Log on to your VistA/M server. Select the **Programmer Options . . .** menu from the Systems Manager Menu option (EVE).

3.2.2. Stop VistALink System Processes

- If a previous version of VistALink is running on your system, stop the VistALink Listener on the server. Follow your normal procedures to stop the VistALink Listener:
  - If your VistALink listener runs via VMS TCP Services, use VMS TCP services to disable the service (listener)
  - If your VistALink listener process runs within Caché (not via VMS TCP services), use the Foundations menu to stop the listener.
- VistALink users must be stopped.

NOTE: Check the system status for any XOBVSKT routines that are running (e.g., VistALink Handler). If you find any of these jobs running on the system, notify users to log off or FORCEX the jobs. Active users may get NOSOURCE or CLOBBER errors.

- While installing this package on the server, do not run any VistALink-based Client/Server software (e.g., Care Management).
- Roll-and-scroll and Remote Procedure Call (RPC) Broker users may remain on the system
- TaskMan does not need to be put into a wait state

CAUTION: If you accept a risk of VistALink clients getting a CLOBBER/EDITED error, VistALink/Care Management users may remain running. Otherwise stop all other VistALink/Care Management jobs on the system.

3.2.3. Install Kernel Installation and Distribution System (KIDS) Distribution

NOTE: XOB_1_6_Bxx.KID distribution exports 3 VistALink packages/transport globals: XOBU 1.6, XOBV 1.6, and XOBS 1.6. For installation, KIDS works with them as a single unit. When prompted by KIDS to enter a package for loading/installing, always enter **XOBU 1.6.** Doing this will load/install all 3 packages contained in the distribution, in the correct order. The XOB_1P6_3.KID introduces 2FA functionality support – for installation information see the corresponding patch description.

1. Load Distribution.

Use the KIDS Installation option, **Load a Distribution [XPD LOAD DISTRIBUTION]**.

Enter " XOB_1P6_3.KID" as the name of the Host file (where xx is a build number). If the KIDS file is not in the Kernel's default HFS directory on the host file system, you will need to include the directory path to the file as well.

The Load a Distribution option will load three transport globals contained within the distribution:

1. XOBU 1.6 Common files and libraries used by all the XOB* packages and menu options to manage site parameters/operations
2. XOBV 1.6 Handles system and RPC requests
3. XOBS 1.6 M-side security module

2. Verify Checksums.

Run the KIDS Installation option, **Verify Checksums in Transport Global [XPD PRINT CHECKSUM].** This option will ensure the transport global was not corrupted in transit.

At the Select **Select INSTALL NAME:** prompt, enter **XOBU 1.6**. Checksums for all 3 VistALink packages (XOBU 1.6, XOBV 1.6 and XOBS 1.6) will be displayed.

NOTE: When executing the **Verify Checksums** option, the checksums for all three packages (XOBU, XOBV, and XOBS) are displayed. However, due to page feeds, you may need to scroll back up to see the checksums for the first two packages.

Follow the example below:

```
Select Installation Option:  Verify Checksums in Transport Global
Select INSTALL NAME:    XOBU 1.6 <ENTER> Loaded from Distribution  4/3/08@09:54:49
     => Foundations, VistALink and VistALink Security v1.6  ;Created on Apr 03

This Distribution was loaded on Apr 03, 2008@09:54:49 with header of
   Foundations, VistALink and VistALink Security v1.6  ;Created on Apr 03, 2008@
09:34:33
   It consisted of the following Install(s):
       XOBU 1.6       XOBV 1.6       XOBS 1.6

Want each Routine Listed with Checksums: Yes// NO
DEVICE: HOME//   TELNET


PACKAGE: XOBU 1.6     Apr 03, 2008 9:58 am                           PAGE 1
-------------------------------------------------------------------------------

   8 Routines checked, 0 failed.

PACKAGE: XOBV 1.6     Apr 03, 2008 9:58 am                           PAGE 1
-------------------------------------------------------------------------------

   16 Routines checked, 0 failed.

PACKAGE: XOBS 1.6     Apr 03, 2008 9:58 am                           PAGE 1
-------------------------------------------------------------------------------
```

```
   7 Routines checked, 0 failed.
```

Figure 3-1. KIDS Installation option: Verify Checksums in Transport Global [XPD PRINT CHECKSUM]

3. Backup Transport Global.

   Use the KIDS Installation option, **Backup a Transport Global [XPD BACKUP**]. This option creates a MailMan message that will back-up all current routines on your VistA/M system that will be replaced by the packages in this transport global. (If you need to preserve components that are not routines, you must back them up separately.)

   At the **Select INSTALL NAME:** prompt, enter **XOBU 1.6**. All 3 VistALink packages (XOBU 1.6, XOBV 1.6 and XOBS 1.6) will be backed up.

   Follow the example below:

```
Select Installation Option: Backup a Transport Global
Select INSTALL NAME: XOBU 1.6 <ENTER> Loaded from Distribution  4/3/08@09:54:49
     => Foundations, VistALink and VistALink Security v1.6  ;Created on Apr 03

This Distribution was loaded on Apr 03, 2008@09:54:49 with header of
   Foundations, VistALink and VistALink Security v1.6  ;Created on Apr 03, 2008@
09:34:33
   It consisted of the following Install(s):
       XOBU 1.6       XOBV 1.6       XOBS 1.6
Subject: Backup of XOBU 1.6 install on Apr 03, 2008
  Replace <ENTER>
Loading Routines for XOBU 1.6.....
Routine XOBUZAP is not on the disk..
Routine XOBUZAP0 is not on the disk..
Routine XOBUZAP1 is not on the disk..
Loading Routines for XOBV 1.6................
Loading Routines for XOBS 1.6.......
Send mail to: VLUSER,ONE// <ENTER>
Select basket to send to: IN// <ENTER>
And Send to: <ENTER>
```

Figure 3-2. KIDS Installation option: Backup a Transport Global [XPD BACKUP]

4. Use the KIDS Installation option, **Install Package(s) [XPD INSTALL BUILD]** to install VistALink 1.6.

   At the **Select INSTALL NAME:** prompt, enter **XOBU 1.6**. All 3 VistALink packages (XOBU 1.6, XOBV 1.6 and XOBS 1.6) will be installed.

   Answer the following install questions as follows:

   - Although typically the answer is "No," you can answer "Yes," to the question

     `Want KIDS to Rebuild Menu Trees Upon Completion of Install?`

     Just remember that rebuilding menu trees will increase patch installation time.
   - Answer "No" to the question:

     `Want KIDS to INHIBIT LOGONs during the install?`
   - Answer "No" to the question:

```
Want to DISABLE Scheduled Options, Menu Options, and
Protocols?
```

The following is an example of a VistALink 1.6.1 installation on a VistA/M server (that has VistALink 1.5 previously installed):

```
Select Installation Option: 6 <ENTER> Install Package(s)
Select INSTALL NAME: XOBU 1.6 <ENTER>  Loaded from Distribution  4/3/08@12:00:46
     => Foundations, VistALink, and VistALink Security v1.6  ;Created on Apr 0

This Distribution was loaded on Apr 03, 2008@12:00:46 with header of
   Foundations, VistALink, and VistALink Security v1.6  ;Created on Apr 03, 2008
@11:54:01
   It consisted of the following Install(s):
       XOBU 1.6       XOBV 1.6       XOBS 1.6
Checking Install for Package XOBU 1.6
Will first run the Environment Check Routine, XOBUENV

 >>> Performing environment check...


 All running VistALink listeners should be stopped before proceeding with
 this installation. Enter ? for help on stopping VistALink listeners.

 Have all VistALink listeners been stopped? NO// YES <ENTER>  YES

 >>> VistALink environment check completed for KIDS Install Package option.


Install Questions for XOBU 1.6

Incoming Files:


   18.01     FOUNDATIONS SITE PARAMETERS
Note:  You already have the 'FOUNDATIONS SITE PARAMETERS' File.

Want KIDS to Rebuild Menu Trees Upon Completion of Install? NO// <ENTER>

Checking Install for Package XOBV 1.6

Install Questions for XOBV 1.6

Incoming Files:

   18.03     VISTALINK LISTENER CONFIGURATION
Note:  You already have the 'VISTALINK LISTENER CONFIGURATION' File.

   18.04     VISTALINK LISTENER STARTUP LOG
Note:  You already have the 'VISTALINK LISTENER STARTUP LOG' File.

   18.05     VISTALINK MESSAGE TYPE  (including data)
Note:  You already have the 'VISTALINK MESSAGE TYPE' File.
I will OVERWRITE your data with mine.

Want KIDS to Rebuild Menu Trees Upon Completion of Install? NO// <ENTER>

Checking Install for Package XOBS 1.6

Install Questions for XOBS 1.6
```

```
Want KIDS to INHIBIT LOGONs during the install? NO// <ENTER>
Want to DISABLE Scheduled Options, Menu Options, and Protocols? NO// <ENTER>

Enter the Device you want to print the Install messages.
You can queue the install by enter a 'Q' at the device prompt.
Enter a '^' to abort the install.

DEVICE: HOME// <ENTER>  TELNET

Install Started for XOBU 1.6 :
               Apr 03, 2008@12:22:04

Build Distribution Date: Apr 03, 2008

 Installing Routines:
               Apr 03, 2008@12:22:04

 Running Pre-Install Routine: EN^XOBUPRE

 Installing Data Dictionaries:
               Apr 03, 2008@12:22:04

 Installing PACKAGE COMPONENTS:

 Installing INPUT TEMPLATE

 Installing PROTOCOL

 Installing LIST TEMPLATE

 Installing OPTION
               Apr 03, 2008@12:22:04

 Running Post-Install Routine: EN^XOBUPOST

 Updating Routine file...

 Updating KIDS files...

 XOBU 1.6 Installed.
               Apr 03, 2008@12:22:04

 Install Message sent #159

 Install Started for XOBV 1.6 :
               Apr 03, 2008@12:22:04

Build Distribution Date: Apr 03, 2008

 Installing Routines:
               Apr 03, 2008@12:22:04

 Running Pre-Install Routine: EN^XOBVPRE

 Installing Data Dictionaries:
               Apr 03, 2008@12:22:04

 Installing Data:
               Apr 03, 2008@12:22:04

 Installing PACKAGE COMPONENTS:
```

```
 Installing INPUT TEMPLATE

 Installing DIALOG

 Installing PROTOCOL

 Installing REMOTE PROCEDURE

 Installing OPTION
              Apr 03, 2008@12:22:05

 Running Post-Install Routine: EN^XOBVPOST

>>> Scheduling the XOBV LISTENER STARTUP option...

>>> The XOBV LISTENER STARTUP option has previously been scheduled:

 Updating Routine file...

 Updating KIDS files...

 XOBV 1.6 Installed.
              Apr 03, 2008@12:22:05

 Install Message sent #161

 Install Started for XOBS 1.6 :
              Apr 03, 2008@12:22:05

Build Distribution Date: Apr 03, 2008

 Installing Routines:
              Apr 03, 2008@12:22:05

 Installing PACKAGE COMPONENTS:

 Installing DIALOG
              Apr 03, 2008@12:22:05

 Updating Routine file...

 Updating KIDS files...

 XOBS 1.6 Installed.
              Apr 03, 2008@12:22:05

 Install Message sent #163

_____________________________________________________________________________

  100%     |             25            50            75                |
Complete   |_________________________________________________________|

Install Completed
```

Figure 3-3. VistALink J2M Installation Example

NOTE: The option **XOBV LISTENER STARTUP** will be scheduled for Task Manager startup on Caché/NT systems only.

NOTE: The installation adds a new Kernel Application Proxy User named "XOBVTESTER,APPLICATION PROXY" to the NEW PERSON file (#200), if not already present. This application proxy user account is used in the VistALink sample Web application to demonstrate usage of the **VistaLinkAppProxyConnectionSpec** connection spec.

5. Restart listeners: If VistALink has already been set up on your server, and you want your server to resume servicing VistALink client requests, restart the VistALink Listener on the server. Follow your normal procedures to start the listener. Otherwise, configuring the listener is a follow-on task (see the section "Configure VistALink Listener"):
   - If your VistALink listener runs via VMS TCP services, use VMS TCP services to enable the service (listener).
   - If your VistALink listener is started within Caché (not via VMS TCP services), use the Foundations menu to start the listener.

3.3. (Optional) Configure VistALink Listener

3.3.1. Do I Need to Configure Listeners As Part of the VistALink Installation?

If you are upgrading an existing VistALink installation, you likely have one or more listeners already configured on your system. *You should not need to add to or change your listener configuration in any way*. Your existing listener configurations will continue to function, without reconfiguration, after upgrading VistALink.

For Caché/Linux sites only, with existing listeners, you may want to switch from the M-only listener (started from the Foundations menu) to the XINETD version of the listener (started from the OS level). You can do this switch at any time, however; it does not need to be done as part of the installation of VistALink v1.6.1.

For sites where VistALink is being installed for the first time, you will need to configure at least one new listener in order to support VistALink-based requests. You can do this as part of the installation, or at later time as is convenient.

3.3.2. Listener Introduction

A VistALink listener runs on your M system, in order for Java applications to connect to your M system using VistALink. The listener waits for and accepts incoming client connections on a specified TCP port, and spawns off handler jobs to service those connection requests.
There are two styles of listeners:

- OS-Based Service (the listener runs as an operating system process, i.e., a VMS TCP Service, or an Linux XINETD service)
- M-Based (the listener starts, stops and runs as an M process)

Recommendations for which type of listener to use are based on operating system type, and account type:

- **Production VMS systems**: Run as a VMS-based TCPIP service
- **Production Linux systems**: Run as a Linux-based XINETD service
- **Windows systems**: The M-based listener must be used, including for production.
- **Non-production VMS and Linux systems**: Either the M-based or OS service-based listener can be used

The sections below provide setup requirements for the Caché/VMS and Caché/NT operating systems, as well as general information for all operating systems.

3.3.3. Recommended VistALink Ports (all operating systems)

Though any available TCP port may be used, the recommended port for the VistALink Listener is **8000** for production systems and **8001** for test systems. This recommendation comes from the DBA's list of reserved ports, published on FORUM at **DBA Option | Port Assignments for TCP**.

NOTE: The recommended port for the VistALink listener is 8000 for production systems and 8001 for test systems.

3.3.3.1. Avoiding Port Conflicts

Within a single IP address/system, VistALink listeners can be set up as:

- A single VistALink listener, running on any available port.
- Multiple VistALink listeners running on the same IP address/system, but listening on *different* ports.

To run one listener in a production account and another in a test account on the same IP address/system, you must configure them to listen on different ports (e.g., 8000 for production and 8001 for test). If, on the other hand, you are running the listeners on different IP addresses/systems, the ports can be the same (e.g., one VistALink listener on every system listening on port 8000).

3.3.4. OS-Based Listener Configuration for Caché/VMS Systems

For production Caché/VMS systems, it is recommended to run the VistALink listener as a VMS TCP/IP service. The advantages include:

- The ability to run the TCP/IP service on multiple nodes in a cluster. This allows for an uninterrupted listening process, by redirecting the job if one of the nodes in the cluster goes down.
- Since TaskMan is not used to start the listener, it doesn't matter if the TaskMan process is running on the same node(s) as the VistALink listener(s).

REF: For further assistance with set-up of a VMS TCP/IP service for VistALink, and for the latest information on recommended configuration, we strongly recommend that you log a Remedy ticket so that the appropriate Product Support team (currently HSTS) can assist you.

The methodology for running VistALink as a TCP listener was developed and written into a cookbook by the HSTS Product Support team, to aid IRM support staff. The cookbook, as a document named VISTALINK_TCPIP_COOKBOOK.DOC, can be obtained from the HSTS team or downloaded from the standard Product Support ftp download directories.

When configuring VMS TCP Services, some issues to consider include:

- Many of the operations require elevated VMS privileges, specifically, SYSPRV. Before you begin, use the VMS SHOW PROCESS/ALL command to verify that you are logged into an account that has SYSPRV.
- If you need to create a new service, refer to VISTALINK_TCPIP_COOKBOOK.DOC for step-by-step instructions.
- To modify an already-existing VistALink service:
    - Use the TCP/IP utilities to disable the service, e.g., VLINK:

      ```
      TCPIP> DISABLE SERVICE VLINK
      ```

    - Copy any updated command file to the directory used by the service.
    - Modify the command files to match your environment. You'll need to remove the comment from the appropriate line in the 'command line:' section and then modify it to match your configuration. Refer to the comments for examples of how the line should be modified.
    - Save the file(s).
    - Enable the VistALink service, e.g., VLINK:

      ```
      TCPIP> ENABLE SERVICE VLINK
      ```

In general, use VISTALINK_TCPIP_COOKBOOK.DOC to help you:

- Set up VistALink as a TCP/IP service in VMS
- Modify the service command file templates to match your environment
- Create and update a dedicated VMS user account, e.g., VLINK with the proper authorized and default privileges (e.g., remove OPER privilege).

3.3.5. OS-Based Listener Configuration for Caché/Linux Systems

For production Caché/Linux systems, it is recommended to run the VistALink listener as a XINETD (Linux) service.

The advantages include a uniform method for starting and stopping VistALink as one of many different types of listener processes on Linux.

NOTE: For further assistance with set-up of an XINETD service for VistALink, and for the latest information on recommended configuration, we strongly recommend that you log a Remedy ticket so that the appropriate Product Support team can assist you.

An example of an XINETD configuration file for a VistALink listener is provided below.

```
#description: VA VistALink Listener for Port 8000
#
service las_vlkp
{
    type = UNLISTED
    disable = no
    flags = REUSE
    socket_type = stream
    protocol = tcp
    port = 8000
    wait = no
    user = lastcpip
    env = TZ=/usr/share/zoneinfo/US/Pacific
    server = /usr/local/cachesys/system01/bin/csession
    server_args = system01 -ci -U OEX CACHELNX^XOBVTCP
    instances = UNLIMITED
}
```

Figure 3-4: Example XINETD Service Configuration

You will need to adjust certain values to match your system environment:

- port
- user
- env
- server
- server_args

3.3.6. M-Based Listener Configuration for Caché/NT (Windows) Systems

See Appendix A, "Listener Management for Caché NT," in the *VistALink 1.6 System Management Guide*. This approach starts, manages and stops the listener entirely within M (as opposed to using VMS (TCP/IP utility) or Linux (XINETD) to start/stop the listener.

Note: You can also use the same instructions to set up an M-based (rather than OS service based) listener on Linux and/or VMS system, i.e., for non-production systems.

3.4. (Optional) Verify Listener Connectivity

The general process for testing the listener is as follows:

1. Telnet test
2. VistALink J2SE SwingTester sample application test

3.4.1. Telnet Test

Telnet from your workstation to the IP address and port of the VistALink listener. On most workstations you can do this simply by entering the telnet IP address port in a command window, e.g.:

```
c:\> telnet 10.21.1.85 8000 <Enter>
```

1. When you connect, press <Enter>. If a VistALink listener is running on that port, you should see echoed something similar to this example:

```
<?xml version="1.0" encoding="utf-8" ?><VistaLink messageType="gov.va.med.founda
tions.vistalink.system.fault" version="1.5" xmlns:xsi="http://www.w3.org/2001/XM
LSchema-instance" xsi:noNamespaceSchemaLocation="vlFault.xsd"><Fault><FaultCode>
Server</FaultCode><FaultString>System Error</FaultString><FaultActor>Request Man
ager</FaultActor><Detail><Error type="Request Manager" code="184001" ><Message>R
equest Handler Loading Error: No message type defined</Message></Error></Detail>
</Fault></VistaLink>
```

   Although there is an error message echoed in this display, the error is due to the fact that you are connecting from telnet rather than from a VistALink client. If an Extensible Markup Language (XML) message similar to the one above is echoed back, the network connection between your workstation and the VistALink listener at the requested IP address and port is valid.

If you cannot make the telnet connection, there may be a problem somewhere in the network / firewall / machine TCP configuration.

If you connect but do not see XML output similar to that in the sample in step 2 above when you press <Enter>, check the type of listener that is running in the port. (It may be a Broker, Health Level 7 [HL7], or other type of listener.)

NOTE: To disconnect the session, press and hold the CTRL key then press the right brace "]" key: CTRL +]
This will properly disconnect the telnet connection.

NOTE: Errors (at SETMSG+5^XOBVRH) will be logged in the Kernel error trap when you use telnet to test the VistALink listener. Such errors can be ignored when Telnet testing is the source.

3.4.2. VistALink J2SE SwingTester Sample Application Test (optional)

To test your M listener with the SwingTester sample application, follow the instructions provided in Appendix A of this document, "Installing and Running the J2SE Sample Applications."

The SwingTester Java 2 Platforms Standard Edition (J2SE) (client/server) sample application is supplied in the **vljSamples_1.6.1.nnn.jar** file**.**

You can use the SwingTester sample application to perform a standalone test of the M VistALink listener before proceeding with the app server installation.

3.5. (Optional) Configure Connector Proxy User(s) for J2EE Access

Follow this step only if you are setting up a brand new VistALink implementation on your VistA/M system for immediate access by one or more specific J2EE servers. This step is not necessary if you are upgrading an existing VistALink implementation.

3.5.1. Connector Proxy Overview

To allow a J2EE system to access resources on your M system via VistALink, you need to an M Kernel "connector proxy user" account for the J2EE system to connect/login to your M system. A connector proxy account represents a specific application server (not an end-user). A VistALink adapter on a J2EE system logs on to your VistA/M server using the assigned Kernel connector proxy user account, authenticating with an access/verify code pair.

3.5.2. How to Create Connector Proxy User Kernel Accounts

See the *Security* chapter, "*Creating Connector Proxy Users for J2EE Systems*" section, in the *VistALink 1.6 System Management Guide*, for complete instructions on how to create connector proxy users.

3.6. Installation Back-Out/Roll-Back Procedure

If there is an unforeseen problem with the installation of VistALink v1.6.1, it is possible to reinstall VistALink v1.5. Possible losses of functionality with a rollback to v1.5 include:

- Inability of any client applications that have upgraded to VistALink v1.6.1 (client-side) to connect to your site.

NOTE: There are no FileMan data dictionary changes between v1.5 and v1.6.1.

3.6.1. Reinstall v1.5

To re-install v1.5:

1. Obtain the v1.5 KIDS distribution from the EIE ftp server (XOB_1_5.KID).
2. Obtain the v1.5 Install Guide from the EIE ftp server and follow the installation steps in chapter 2 (VistA/M Server Installation Procedures) to reinstall VistALink v1.5. Or:
    a. Stop any running VistALink listeners (if any are running at all).
    b. Use KIDS to install the XOB_1_5.KID distribution. The install package is XOBU 1.5.
    c. Start any listeners after the installation, either from the operating system level or Mumps level, depending on how VistALink listeners have been configured at your site.
    d. Optionally verify listener connectivity – with telnet and/or with a v1.5 VistALink client application

3.6.2. Optional Deletions of v1.6-Only Components

Optionally delete the following v1.6-only components:

- Routines:
  - XOBUZAP
  - XOBUZAP0
  - XOBUZAP1

- Protocols
  - XOBU TERMINATE A JOB
  - XOBU TERMINATE ALL JOBS
  - XOBU TERMINATE CONNECTION MANAGER
  - XOBU TERMINATE JOBS REFRESH
  - XOBU TERMINATE JOBS UTILITY MENU
  - XOBU TERMINATE SYSTEM STATUS

- List Templates
  - XOBU TERMINATE JOBS UTILITY

- Dialogs
  - 182010

4. Oracle WebLogic Application Server: Installation Procedures

4.1. Overview

Goal: Install VistALink adapter(s) on application servers so that J2EE applications running on those servers can execute requests against one or more M systems.

Main installation tasks:

- Admin server:
  - Make VistALink configuration file accessible on classpath
  - Install VistALink-specific monitoring plug-in into WebLogic console
- Servers targeted for adapter(s):
  - Make a copy of VistALink configuration file accessible on classpath
  - Install supporting jars as J2EE Shared Libraries (production servers only)
  - Install VistALink adapters (one per unique M system IP address/port combination)

4.1.1. Adapter Deployment Descriptors

VistALink resource adapters have deployment descriptors that control configuration of the adapter. Text editors are the recommended tool for editing deployment descriptors. These files are located in the **META-INF** directory in each adapter archive (RAR):

- **ra.xml**: The standard J2EE deployment descriptor for J2CA resource adapters.
- **weblogic-ra.xml**: Contains WebLogic-specific extended configuration information.
- **MANIFEST.MF:** Manifest file defining information about the files packaged in the RAR.

4.1.2. VistALink 1.6.1 Adapter Changes

VistALink 1.6.1 adapters are updated to support the new J2EE 1.7 specifications for J2EE connectors, supported in WebLogic 10.3.6/12.1.2. Changes significant to the installation process are:

- Deployment Descriptors. The format of both the ra.xml and weblogic-ra.xml descriptors is different. Existing adapters' deployment descriptors need to be updated.
- Linked Adapters replaced by J2EE Shared Libraries. The primary benefit of the WebLogic 8.1 linked adapter was the re-use of one adapter's resources (jars) by other adapters. The linked adapter feature is not supported for upgraded adapters in WebLogic 10.3.6/12.1.2. However, for production servers that need to minimize resource consumption, the replacement feature for linked adapters is to deploy the adapter jars as "J2EE shared libraries".

4.1.3. VistALink Adapters and Classloading

VistALink resource adapters are intended to be deployed and run as standalone deployments in WebLogic. The adapter is then made available for use by any application on the server. To support this, the application server places java classes used in the VistALink RAR on high-level classloaders visible by all applications.

4.2. Preparation

4.2.1. Software Installation Time (Varies)

The estimated installation time installing VistALink adapters in a WebLogic domain varies, depending in part on whether it is a first-time installation, and in part on how many new or existing adapters need to be deployed or upgraded. As such, a time estimate for individual tasks is provided below, from which you can estimate on how much time is required for the installation tasks necessary on your system.

- Place VistALink configuration file on server classpath: 5 minutes per server
- Install console plug-in or standalone EAR (admin server): 5 minutes
- Update existing 10.3.6/12.1.2 RAR deployment descriptors: 5-10 minutes per adapter
- Install J2EE shared libraries (production servers only): 20 minutes
- Install new adapters: 5-15 minutes per adapter

4.2.2. System Requirements

VistALink 1.6.1 is supported only on WebLogic at the current time. This is the requirement for installation:

- Oracle WebLogic Server (WLS) 10.3.6/12.1.2

4.2.3. Deployer Requirements

The WebLogic administrator/deployer should have prior WebLogic administration experience, and be comfortable with (and have the privileges for) the following tasks:

- Modify server startup scripts
- Set "Remote Start" options for managed servers started by Node Manager
- Set JVM arguments for WebLogic servers
- Modify the classpath for WebLogic servers
- Configure log4j
- Deploy and undeploy applications
- Bounce servers

4.2.4. Obtain the VistALink Distribution File

You can obtain the VistALink distribution ZIP file from any of the **anonymous.software** directories on the Office of Information & Technology (OI&T) File Transfer Protocol (FTP) download sites. You should unzip it to a folder in a good working location for your WebLogic Server installation process,

most likely on a drive of the administration server for your WebLogic domain. This location will be referred to as the "<DIST FOLDER>" for the rest of the instructions.

4.2.5. Obtain M Connector Proxy User and Listener Information

If you are configuring a new adapter, contact the VistA/M system's Information Security Officer (ISO) and/or the VistA/M system manager to obtain the connector proxy user's credentials for the VistA/M system to which you intend to connect. This information includes:

- Access/verify codes for connector proxy user
- VistALink listener port
- IP address of the VistA/M system

See the section "Post Install: Configure Connector Proxy User(s) for J2EE Access" in this guide for more information on the connector proxy user.

4.3. Upgrading a WebLogic 8.1 Domain w/Existing VistALink Adapters

4.3.1. Back Up Exploded RAR Directories and VistALink Configuration File

You should back up (copy) all of your exploded RAR directories, and also the VistALink configuration file. You will need these to recreate your adapters in the WebLogic 10.3.6/12.1.2 domain.

4.3.2. If Running the Domain Upgrade Wizard

There are two approaches to moving from a WebLogic 9/10 domain to a WebLogic 10.3.6/12.1.2 domain (and only you can decide which is best):

- Create a new WebLogic 10.3.6/12.1.2 domain from scratch and redeploy all applications to it that you want carried forward, or
- Run Oracle's domain upgrade wizard to upgrade your WebLogic 9/10 domain to WebLogic 10.3.6/12.1.2.

If you choose to upgrade your domain by running the upgrade wizard (rather than starting from scratch with a new domain), we recommend you perform the following steps, before shutting down your WebLogic 8.1 domain and running the wizard.

4.3.2.1. Undeploy RARs

If you have any VistALink adapters deployed, delete them from the WebLogic configuration by navigating to:

```
mydomain>Deployments>Connector Modules
```

Then select each adapter, and click on the Delete button.

4.3.2.2. Undeploy VistALink Console

If you have deployed the VistALink Console, delete it from the WebLogic configuration by navigating to:

```
mydomain>Deployments>Web Application Modules
```

Then select the VistaLink console web application, and click on the Delete button.

4.3.2.3. Undeploy Sample Application

If you have deployed the VistALink sample web application, delete it from the WebLogic configuration by navigating to:

```
mydomain>Deployments>Applications
```

Then select the VistALink sample web application, and click on the Delete button.

4.4. WebLogic 10.3.6/12.1.2 Server Configuration

For the domain's admin server, and for each managed server that will run VistALink adapters, perform the following steps:

4.4.1. Create <HEV Configuration Folder>

We recommend using a single folder to hold any external configuration files for all HealtheVet (HEV) applications, including VistALink. If it is not already present, you should create this folder, on each physical WebLogic server.

- If not already present, create a secure, protected directory to place on the server classpath for each of your WebLogic servers running VistALink. This folder will be referred to as the <HEV CONFIGURATION FOLDER> in the following steps.
- Ensure that this folder is secure and protected. The **gov.va.med.vistalink.connectorConfig.xml** file it will contain holds login credentials for accessing VistA/M systems. On Linux systems, access to the folder should be restricted to the account or group under which WebLogic runs. On all J2EE systems, access to the host file system should be protected.

4.4.2. Create/Copy VistALink Configuration File

VistALink makes use of its own configuration file to load VistALink-specific connector settings. When configured for your system, it will contain one entry for each VistALink adapter.

1. Copy the *gov.va.med.vistalink.connectorConfig.xml* configuration file into the <HEV CONFIGURATION FOLDER> on each physical server that will be running VistALink adapters. Also do this on the admin server:
    - If upgrading a previous domain, copy the existing gov.va.med.vistalink.connectorConfig.xml from that domain

Obsolete Setting: primaryStationSuffix: This attribute has been eliminated. Any primary station numbers requiring an alpha suffix, should instead be entered as part of the "primaryStation" attribute, i.e., primaryStation="200M".

Note: If VA institution rules are being used, only 200-series (Austin Information Technology Center) station numbers can have alpha suffixes for the primary station number.

If any entries have primaryStationSuffix, they should remove that attribute and append the value of the suffix into the existing primaryStation attribute.

- If this is a brand new VistALink deployment, copy the example configuration file from the <DIST FOLDER>/app-j2ee/configFile-j2ee folder.

NOTE: For additional information on setting up a connector configuration file, see the section "VistALink Connector Configuration File," in the *VistALink 1.6 System Management Guide*.

4.4.3. Place <HEV Configuration Folder> on Server Classpath(s)

1. **Admin Server**. On admin servers, modify the server classpath by updating the appropriate variable in either the setDomainEnv.cmd/.sh (preferred) script, or in the startWebLogic.cmd/.sh script (both scripts are in the domain root's /bin folder). Add the <HEV Configuration Folder> classpath folder to the PRE_CLASSPATH (setDomainEnv) or CLASSPATH (startWebLogic) variable.

   The following example shows example modifications for a Windows (.cmd) setDomainEnv script:

```
. . .

@REM ADD EXTENSIONS TO CLASSPATHS

@REM for VISTALINK
set PRE_CLASSPATH=%PRE_CLASSPATH%;C:\Data\bea103-stage\admin\ClasspathFolder;

. . .
```

Figure 4-1. Admin Server: Add the classpath folder to the server classpath in the setDomainEnv script

2. **Managed Servers**. On any managed servers started by Node Manager, update the server classpath in the Configuration | Server Start tab of the console. Adding a classpath folder to the server classpath will also necessitate specifying the complete server startup classpath, which typically means, at a minimum, including the following jars:

weblogic_sp.jar	e.g., c:/bea/weblogic92/server/lib/weblogic_sp.jar
weblogic.	e.g., c:/bea/weblogic92/server/lib/weblogic.jar
webservices.jar	e.g., c:/bea/weblogic92/server/lib/webservices.jar
tools.jar	e.g., c:/bea/jdk150_04/lib/tools.jar (required only if server compilation needed, e.g., JSPs)
<HEV Configuration Folder>	(the point of this exercise)

NOTE: You can find the exact classpath used to start any given managed server by examining the log files (.out, .log) stored in the domain folder, servers/<SERVER NAME> subdirectory and looking for the value of the *java.class.path* property.

NOTE: No other classpath changes are necessary to support VistALink on WebLogic 10.3.6/12.1.2. On WebLogic 10.3.6/12.1.2, jars for adapters are loaded either as:

- J2EE shared libraries (production systems), or
- Automatically from the adapter RAR folder (development systems)

4.4.4. Create/Update Server log4j Configurations

VistALink uses log4j for logging. To enable VistALink logging, you should create (or if upgrading from a previous domain, update the existing) log4j configuration file(s) for each server that will have VistALink components installed:

- admin server (VistALink console application, and/or adapters)
- managed servers (adapters)

To help with configuring log4j, in the VistALink <DIST FOLDER>/log4j directory, VistALink-specific log4j information is provided, including:

- vistalink_1_6_loggers.xls (describes VistALink supported logger categories/levels)
- log4jSampleJ2EEConfig.xml (example log4j configuration file for VistALink for J2EE)

To enable logging:

1. Create/update a log4j configuration file on each J2EE server (admin and managed servers)
2. Configure each server to find log4j configuration file. Methods include:
    - Name the file log4j.xml and place in a folder that is on the server classpath, such as the <HEV CONFIGURATION FOLDER> (WebLogic will find automatically), or
    - Name the file anything, and put it in any location on the server file system. Then configure each server's JVM to start with the following JVM argument to explicitly provide the full filepath for the log4j configuration file:

      –Dlog4j.configurationFile=directory/filename

NOTE: Due to the fact that using deploying VistALink adapters place the log4j library on a classloader higher than all deployed applications, log4j configuration on all servers with VistALink adapters deployed must contain the logger and appender log4j configurations for ALL applications deployed to that server.

4.4.5. Server JVM Argument: gov.va.med.environment.production

The *gov.va.med.environment.production* JVM system property configures whether the WebLogic server is considered a Test or Production server, and is used in VistALink and made available to other applications through the **gov.va.med.environment.Environment** Application Program Interface (API). Optionally add the following JVM argument to your server startup(s):

JVM Argument	Value	Default Value

`-Dgov.va.med.environment.production`	false | true	false

1. **For production servers only**, set the "-Dgov.va.med.environment.production" JVM argument to true. Modify one of the following locations to set this argument:

   - Admin server: modify the setDomainEnv.cmd/.sh (preferred) or startWebLogic.cmd/.sh script (both scripts are in the domain home, /bin subdirectory). Modify the JAVA_OPTIONS variable.
   - Managed servers started by node manager: In the WebLogic console, go to the <Server Name> | Configuration | Remote Start tab, and modify the "Arguments" field.

   NOTE: The *gov.va.med.environment.production* setting marks a J2EE system as being a "production" or "test" system, and is used by VistALink adapters to prevent a test J2EE system from connecting to a production M system, and vice versa.

2. On **non-production** WebLogic servers, the argument does not need to be set, since the API using it defaults to false.

4.4.6. Server JVM Argument: gov.va.med.environment.servertype

On WebLogic servers, in most cases the argument does not need to be set (a change since VistALink 1.5), because automatic servertype detection is performed on WebLogic servers, and will succeed (except with unusual classloader configurations.) If set, however, the value of the JVM argument still overrides the automatically detected value.

The *gov.va.med.environment.servertype* JVM system property configures the value of the "current" server type returned to VistALink and other applications by **gov.va.med.environment.Environment** API. Optionally add the following JVM argument to your server startup(s):

JVM Argument	Value	Default Value
`-Dgov.va.med.environment.servertype`	weblogic, websphere, jboss, oracle, j2se, etc.	auto-detects for weblogic, otherwise defaults to "unknown"

1. If you decide to pass this argument to the server JVMs, optionally modify one of the following locations to set this argument:

   - Admin server: modify the setDomainEnv (preferred) or startWebLogic script (both are in the domain home, /bin subdirectory).
   - Managed servers started by node manager: In the WebLogic console, go to the <Server Name> | Configuration | Remote Start tab, and modify the "Arguments" field.

4.5. WebLogic 10.3.6/12.1.2: Install the Standalone Console EAR (Admin Server)

For WebLogic 10.3.6/12.1.2 we recommend installing the standalone VistALink console EAR application, rather than the console plug-in, due to difficulties integrating with the WebLogic console navigation tree and tab set.

The VistALink console is helpful to monitor and troubleshoot VistALink adapters. As such it is useful to install it prior to installing any VistALink adapters.

4.5.1. Copy Console EAR file

Copy the console EAR file from the <DIST FOLDER>/app-j2ee/console-ext folder to a staging folder on your admin server:

- VistaLinkConsole-1.6.1.xxx.ear

4.5.2. Deploy Console EAR

1. Target the deployment to the domain admin server only.
2. Finish the deployment, and activate changes. In the main "Deployments" listing, the state of the VistaLinkConsole application should be *New* or *Prepared* (depending on whether targeted servers are running or not).
3. Start the application (in the Deployment list, choose Start | Servicing all requests for the VistaLinkConsole application). The state of the application should now be *Active*.

4.5.3. Access Standalone VistALink Console

If successfully deployed, the standalone VistALink console will be reachable at the following URL:

- http://<adminserver>:<port>/vlconsole

You'll be prompted for a user name and password. Use the same credentials as you would use to login to the WebLogic administration console. From that point on, the standalone VistALink console application will look almost identical to the console extension plug-in version.

Click on the link to open the VistALink console plug-in main page. You should see a page like the following:

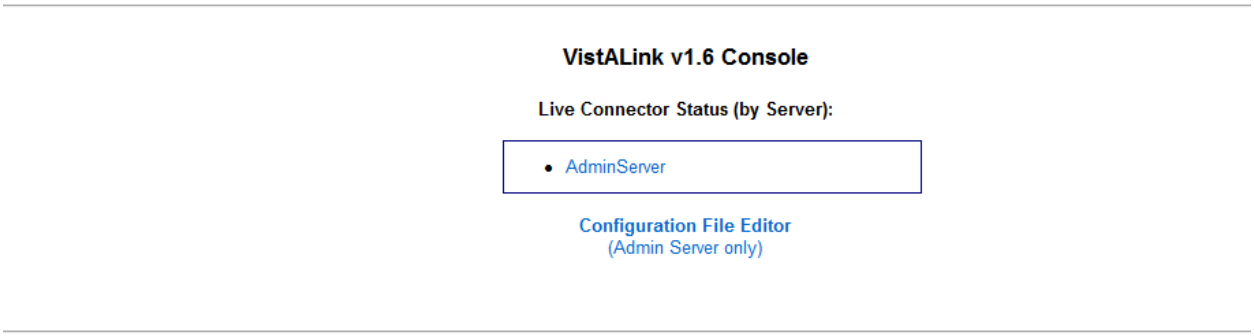

Figure 4-2. Standalone VistALink 1.6 Console

4.5.4. Check Configuration Editor Access to Configuration File

On the main page of the VistALink console, click the "Configuration File Editor" link:

- If the server classpath on the admin server file system is set up correctly, you should be presented with a list of entries from the copy of the VistALink configuration file on your admin server's file system.
- Otherwise, if there is a problem, you will see an error message, for example, "Error while retrieving configuration file: 'Missing configuration file path.'.". If you see this or similar error message, check:
    - Is the configuration file present on the host file system of the admin server?
    - Is the configuration file named "gov.va.med.vistalink.connectorConfig.xml"?
    - Is the folder containing the configuration file on the classpath specified in the setDomainEnv or startWebLogic script of the admin server?

4.6. Deploy Shared J2EE Libraries (Production Domains Only)

Copy the following jars from <DIST FOLDER>>/app-j2ee/shared-lib to your deployment staging area, and deploy each of them as shared libraries:

- log4j-api-2.10.0.jar
- log4j-core-2.10.0.jar
- vljFoundationsLib-1.6.1.xxx.jar
- vljConnector-1.6.1.xxx.jar

On production domains only, and for servers that will host adapters only, deploy these jars as J2EE shared libraries:

1. Copy each jar listed above to a file location on the admin server's file system.
2. Perform a deployment in the WebLogic console for each jar, using the same steps as you would for deploying an EAR. Accept the defaults presented by the WebLogic console.
3. Target the deployment to all servers that will be hosting VistALink adapters.
4. Activate changes, either individually or after all libraries are deployed.

NOTE: For J2CA adapters, J2EE shared libraries serve as a replacement for WebLogic 8.1's "linked adapter" feature. Linked adapters in WebLogic 8.1 allowed the sharing of jar resources across multiple adapters, reducing the amount of systems resources consumed by multiple adapters.

For development systems, deploying the jars as J2EE shared libraries is not necessary. Instead, the jars can be deployed with each adapter, inside each adapter's RAR folder.

4.7. Create/Deploy VistALink Adapter(s)

Repeat the steps in this section for each adapter you need to deploy. You would deploy one adapter for every M system that applications on your domain need to communicate with.

4.7.1. Add Connector Entry to VistALink Configuration File

1. If this is a new adapter, use the VistaLink console's configuration editor to add a new configuration entry for the new adapter. You will need to provide:
   - A unique Java Naming and Directory Interface (JNDI) name for the adapter to be deployed under, (e.g., *vlj/Salem658*) in the jndiName attribute.
   - The primary station number of the M system being connected to, in the primaryStation attribute.
   - The IP and port of the VistALink listener on the M system being connected to (ip and port attributes)
   - The access and verify code for the connector proxy user assigned by the M system administrator (access-code and verify-code attributes)
2. Be sure to set the "enabled" attribute to true.
3. Save the new entry.
4. Copy the updated configuration file to all managed servers that will be hosting the adapter (if it is a multi-server domain).

NOTE: Use of the VistaLink console's configuration editor is not mandatory. The VistALink configuration file can also be edited directly using a text editor.

4.7.2. Create New or Update Existing Adapter Folder on Admin Server

1. If this is a new adapter, on the admin server, create a new, empty folder for the adapter, with a folder name that easily identifies the adapter (e.g., "vlj/Salem658").

> **NOTE:** The folder name will become the default deployment name for the adapter when displayed in the WebLogic console. So choose folder names that will identify the adapter mnemonically to the administrators viewing them in the WebLogic console later.

2. If you are updating an existing adapter folder from a previous WebLogic 8.1 domain, delete:
   - all jar files in the root directory of the folder
   - all jar files in the /lib subdirectory

4.7.3. Back Up Deployment Descriptors

1. If you are updating an existing adapter folder from a previous WebLogic 8.1 domain,, move elsewhere, rename or otherwise back up the following files in the existing META-INF directory:
   - ra.xml
   - weblogic-ra.xml

4.7.4. Copy New 1.6 Files

1. Copy the updated 1.6 files needed for the RAR from the VistALink zip distribution to the existing or new RAR folder:
   - **Production Systems**: Copy the entire contents of the <DIST FOLDER>/app-j2ee/Rar-Prod-Template folder from the VistALink zip distribution to the new RAR folder, including the entire META-INF subfolder.
   - **Non-production systems**: <DIST FOLDER>/app-j2ee/Rar-Dev-Template folder from the VistALink zip distribution to the new RAR folder, including the entire lib and META-INF subfolders.

4.7.5. Update Deployment Descriptors

1. The new **ra.xml** deployment descriptor **no longer needs to be modified**. Leave as-is the template ra.xml descriptor copied above.
2. If creating a new adapter, determine the The Java Naming and Directory Interface (JNDI) name you want to deploy the adapter and connection-instance under. Otherwise, get the existing JNDI name from the old deployment descriptors. This value should match the value used for the adapter's entry in the VistaLink configuration file earlier (e.g., *vlj/Salem658*).
3. Edit the **weblogic-ra.xml** descriptor copied above, as follows:
   a. In the <connection-instance> section, <jndi-name> element, replace the placeholder value *"${vlj.jndi.name}"* with the chosen JNDI name.

b. In the <connection-instance> section, <connection-properties> element, <properties> element, <property> element, <value> element, replace the placeholder value *"${vlj.jndi.name}"* with the chosen JNDI name.

c. Near the top of the file, in the first, first-level <jndi-name> property, replace the placeholder value *"${vlj.jndi.name}"*: we recommend using the chosen JNDI name appended with "Adapter".

> **NOTE:** This JNDI name (for the entire adapter) must be *different* than the JNDI name of the connection instance, that was configured in previous steps a) and b).

d. If updating an existing adapter, for other any properties you changed from the defaults in the old descriptors, update the corresponding values in the new descriptors.

> **NOTE:** Linked adapters are not supported (i.e., via the WebLogic 8.1 <ra-link-ref> mechanism). Any existing linked adapters should be changed to standalone adapters before upgrading.

Example weblogic-ra.xml deployment descriptor:

```
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bea.com/ns/weblogic/90
http://www.bea.com/ns/weblogic/90/weblogic-ra.xsd">

  <!-- Warning: The order the elements appear in complex elements is usually
important.
       It is a good idea to validate and test the weblogic-ra.xml document before
committing. -->

  <!-- For new ADAPTER-level jndi-name, recommend using value of connection instance
JNDI name, appended with "Adapter" -->
  <jndi-name>vljtestconnectorAdapter</jndi-name>

  <enable-global-access-to-classes>true</enable-global-access-to-classes>

  <outbound-resource-adapter>
    <connection-definition-group>
      <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-
factory-interface>

      <default-connection-properties>
        <pool-params>
          <initial-capacity>1</initial-capacity>
          <max-capacity>5</max-capacity>
          <capacity-increment>1</capacity-increment>
          <shrinking-enabled>true</shrinking-enabled>
          <shrink-frequency-seconds>1800</shrink-frequency-seconds>
          <highest-num-waiters>2147483647</highest-num-waiters>
          <connection-creation-retry-frequency-seconds>30</connection-creation-retry-
frequency-seconds>
          <connection-reserve-timeout-seconds>0</connection-reserve-timeout-seconds>
          <test-frequency-seconds>3600</test-frequency-seconds>
          <profile-harvest-frequency-seconds>30</profile-harvest-frequency-seconds>
          <ignore-in-use-connections-enabled>false</ignore-in-use-connections-enabled>
          <match-connections-supported>true</match-connections-supported>
        </pool-params>
        <transaction-support>NoTransaction</transaction-support>
        <reauthentication-support>false</reauthentication-support>
      </default-connection-properties>

      <connection-instance>
        <description>This is the connection and JNDI name that applications will be
accessing.</description>
        <jndi-name>vljtestconnector</jndi-name>
        <connection-properties>
          <properties>
            <property>
              <!-- connectorJndiName value should be the same value as connection
instance jndi-name a few lines above -->
              <name>connectorJndiName</name>
              <value>vljtestconnector</value>
            </property>
          </properties>
        </connection-properties>
      </connection-instance>

    </connection-definition-group>
  </outbound-resource-adapter>
</weblogic-connector>
```

Figure 4-3. weblogic-ra.xml sample deployment descriptor

4.7.6. Deploy Adapter

1. Perform a deployment in the WebLogic console for the new RAR folder (i.e., an exploded RAR). Accept the defaults presented by the WebLogic console.
2. Target the deployment to all servers that will be hosting the VistALink adapter.
3. Finish the deployment, and activate the changes. In the main "Deployments" listing, the state of the deployed adapter should be *New* or *Prepared* (depending on whether targeted servers are running or not).
4. Start the server(s) the adapter is targeted to, if they aren't running. The state of the deployed adapter should now be *Prepared*.
5. Start the adapter itself (in the Deployment list, choose Start | Servicing all requests for the adapter). The state of the deployed adapter should now be *Active*.

4.7.7. Monitor Adapter in VistALink Console

With a successfully configured adapter and a successful deployment, you will be able to:

- See the adapter listed in the "Live Connector Status" section of the VistALink console for every running server it was deployed on
- On the list of connectors for any given server, under "M System Info", you should see the IP address and port for the connector. This means the adapter was able to find and load settings from an entry in the VistALink configuration file on that server.
- If you click on hyperlinked JNDI name for each connector, you should be able to access a detail page for the connector, showing additional information and performing a live query against the M system to retrieve a number of settings, including the introductory text for the M server.
- The failure counts under health monitoring should be '0'. Otherwise, an error condition exists that should be corrected.

4.8. Troubleshooting

If the adapter does not appear to be correctly configured or deployed, please refer to the "Troubleshooting VistALink" section of the *System Management Guide* for further guidance.

NOTE: Some of the first places to look when troubleshooting a non-working adapter:

- VistALink console (what error messages if any are displayed when you try to view the adapter and perform a live query?)
- WebLogic server log files (per server)
- WebLogic console "out" output
- log4j log files

4.9. Test with J2EE Sample Application (Development Systems Only)

4.9.1. Deploy the Sample Web Application

A sample J2EE application is provided to developers to demonstrate the use of VistALink in a J2EE environment. The sample application is also a way to test your basic adapter setup.

The sample applications is provided in the **<DIST FOLDER>**/app-j2ee/sample **folder**.

To deploy the sample J2EE application:

1. Copy the sample application's EAR file (VistaLinkSamples-1.6.1.xxx.ear) to the admin server's host file system.
2. Perform a deployment in the WebLogic console for the sample application's EAR. Accept the defaults presented by the WebLogic console.
3. Target the deployment to any or all servers hosting VistALink adapters.
4. Finish the deployment, and activate changes. In the main "Deployments" listing, the state of the sample application should be *New* or *Prepared* (depending on whether targeted servers are running or not).
5. Start the server(s) the application is targeted to, if they aren't running. The state of the sample application should now be *Prepared*.
6. Start the application (in the Deployment list, choose Start | Servicing all requests for the sample application). The state of the application should now be *Active*.

To run the sample J2EE application:

1. Point your browser to

   http://<yourserver>:<yourport>/VistaLinkSamples

   Example: http://localhost:7001/VistaLinkSamples.

2. If the install is successful, you should reach a page titled "VistALink Sample/Demo J2EE Application."

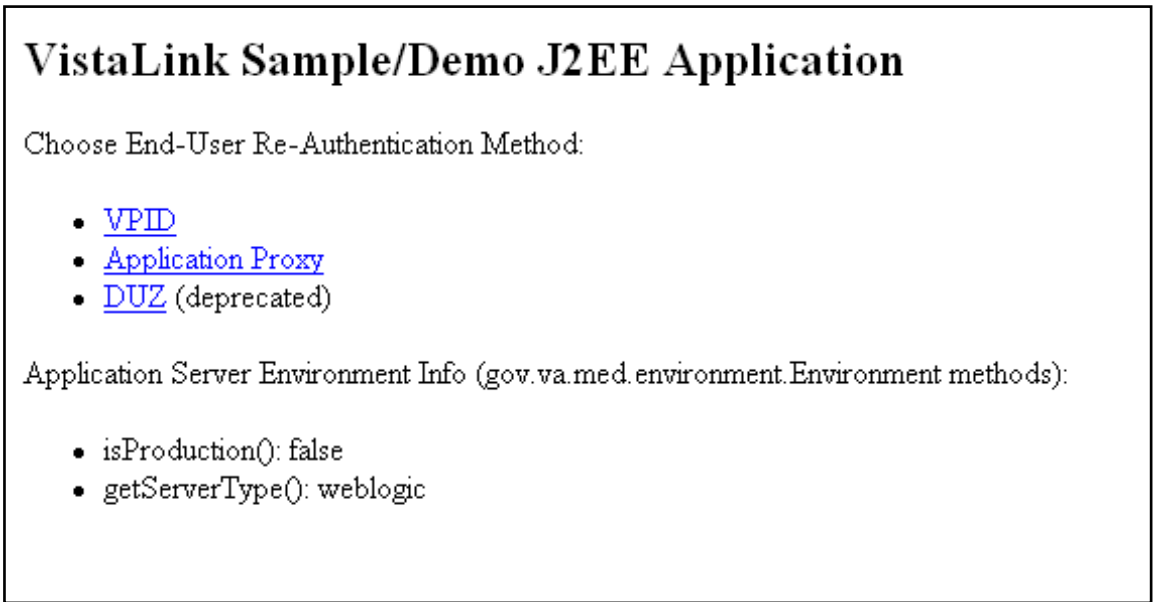

Figure 4-4. VistALink Sample Application

3. Choose a re-authentication method (VA Person ID, or VPID, Application Proxy, or User Number, also called a DUZ) that will allow you to invoke a valid user identity on the target M system to run RPCs under.

> **NOTE:** This user must hold the [XOBV VISTALINK TESTER] "B"-type option.
>
> Note also that if you select default application proxy user "XOBV VISTALINK TESTER", which is distributed/installed by VistALink, that this user is not assigned this "B"-type option by default.

4. Enter the division (for DUZ(2)) valid for both the user you selected, and the M system you're connecting to.

5. Choose the connector to use, either by using institution mapping feature, or selecting from the list of deployed connectors.

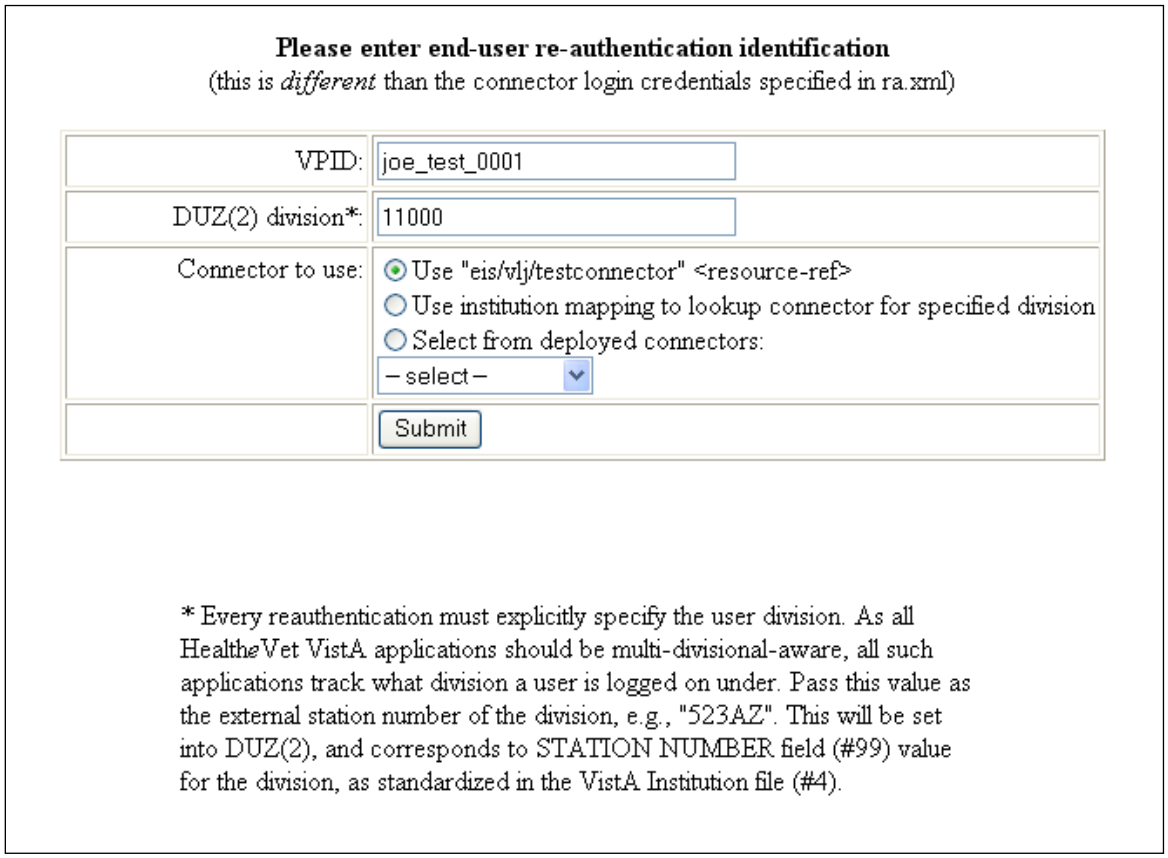

Figure 4-5. VistALink Sample Application Re-authentication Page

6. Press **Submit** to attempt to run a set of sample RPCs using the end-user and connector criteria specified.

7. The results, successful or not, are displayed on a result page:

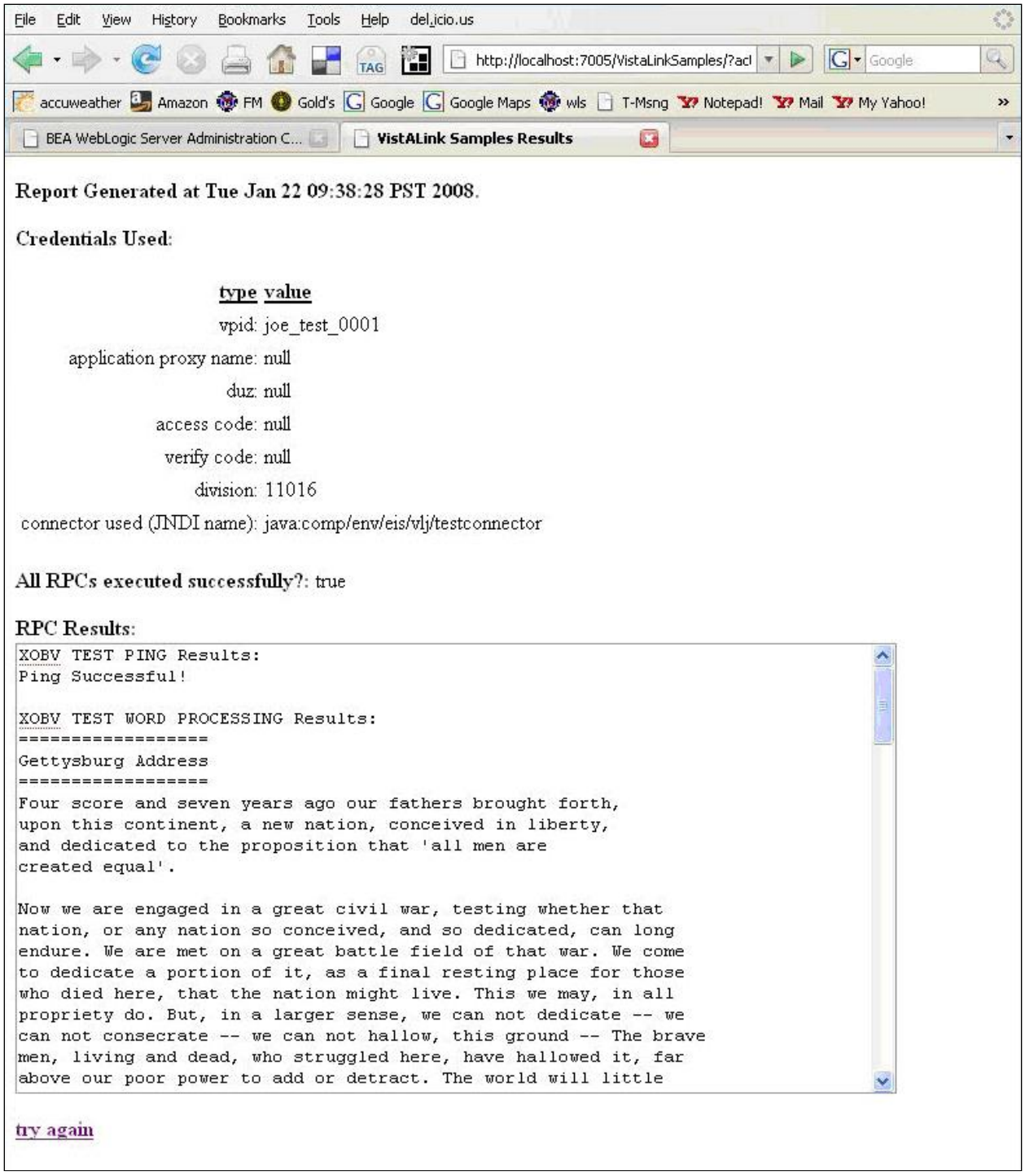

Figure 4-6. VistALink J2EE Sample Application Results Page

5. Rollback Instructions

1. Stop the new connector using the WebLogic Console.
2. Start the old connector using the WebLogic Console.

Appendix A: Installing and Running the J2SE Sample Apps

Overview

The instructions in this section for setting up the SwingTester and other sample applications assume the use of a Windows workstation. However, because VistALink is a pure Java application, it is not particularly tied to the Windows client environment.

Four batch files are supplied in the samples-J2SE folder of the distribution, one for each of the four sample applications:

- runSwingTester.bat (runs VistaLinkRpcSwingTester)
- runSwingSimple.bat (runs VistaLinkRpcSwingSimple)
- runSwingSimpleCcow.bat (runs VistaLinkRpcSwingSimpleCcow)
- runRpcConsole.bat (runs VistaLinkRpcConsole)

A fourth batch file manages the environment settings used by each of the three batch files above:

- setVistaLinkEnvironment.bat

Installation Instructions

1. Install the Java Runtime Environment (JRE)

VistALink 1.6.1 requires the J2SE Java Runtime Environment (JRE) 5.0 (or higher) or the Java Development Kit (JDK) to be installed on the client workstation.

2. Select J2SE Sample Application Location

To install the J2SE Sample Application files, you should either:

- Configure and run the samples directly in the unzipped distribution folder set, or
- Create a new folder to hold the sample application files, and copy the contents of the `\samples-J2SE` folder in the distribution file to the new folder.

3. Configure JAVA_HOME

The JAVA_HOME variable in the provided **setVistaLinkEnvironment** batch file must be modified to match the location of the Java executable to use on your workstation. You may have multiple Java Runtime Environments (JREs) or Java Development Kits (JDKs) installed on your workstation. The selected JRE for the JAVA_HOME variable must be version 1.5 or higher.

In the **setVistaLinkEnvironment.bat** file, replace default location for the `JAVA_HOME` environment variable with the location to use on your system, e.g.:

```
REM -- set directory with bin subdirectory containing java.exe
REM -- (don't include the /bin subdirectory)
REM -- Note: in general you should obtain the latest v5 JRE available
set JAVA_HOME=C:\Program Files\Java\jre1.5.0_11
```

4. Configure Jar Classpaths

If you are running the sample directly out of the unzipped distribution folder set, you can skip this step (classpaths setVistaLInkEnvironment.bat map to the correct relative folder locations.)

Otherwise, ensure the individual classpath settings in the **setVistaLinkEnvironment** batch file correctly reflect the locations of each of the following files:

- log4j-core -2.10.0.jar
- log4j-api -2.10.0.jar
- vljConnector-1.6.1.nnn.jar
- vljFoundationsLib-1.6.1.nnn.jar
- vljSecurity-1.6.1.nnn.jar

Each entry added to the CLASSPATH variable needs to be modified to match the file name and location of the corresponding library on your system, as you installed them above. For example:

```
REM -- classpath for log4j
set CLASSPATH=%CLASSPATH%;./log4j-core-2.10.0.jar;./log4j-api-2.10.0.jar
```

5. Grant Yourself Kernel Access to the Sample Application

The Kernel "B"-type option, VistALink Tester [XOBV VISTALINK TESTER] was created as part of the M-side KIDS install. To run the sample application, you will need to grant yourself access to the [XOBV VISTALINK TESTER] on the VistA/M server to which you will be connecting (unless you already have Kernel programmer access on the M server).

REF: For more information on granting yourself access to RPCs, see the *RPC Broker Systems Manual* on the VistA Documentation Library (VDL) at http://www.va.gov/vdl/.

6. Run the SwingTester Sample Application

This version of VistALink includes the SwingTester sample application, which is a diagnostic tool for the client workstation. You can use this sample application to verify and test the VistALink client/server connection and sign-on process. Use the following instructions to use this tool.

To run the SwingTester sample application:

1. Launch the batch file **runSwingTester.bat** by double-clicking on it, or run it in a command window**.** This launches the main sample application, designed to demonstrate VistALink functionality and test server connectivity.

   a. If the GUI application window opens, the JAVA_HOME and classpath locations have probably been set correctly.

   b. If the GUI application window does not open, look in the command window output for the reason for failure. Most likely the Java executable was not found at the location specified by JAVA_HOME, or one of the supporting jar files is not in its specified classpath location.

2. In the `ip` and `port` fields, enter the IP and port of the M listener your want to connect to, and press **Connect.** (Alternatively, you could select an entry in a **jaas.config** settings file to set the IP and port.)

3. Click **Connect** on the **Access/Verify Code** interface.

4. Enter the Access / Verify code pair you have been assigned. Click **OK**.

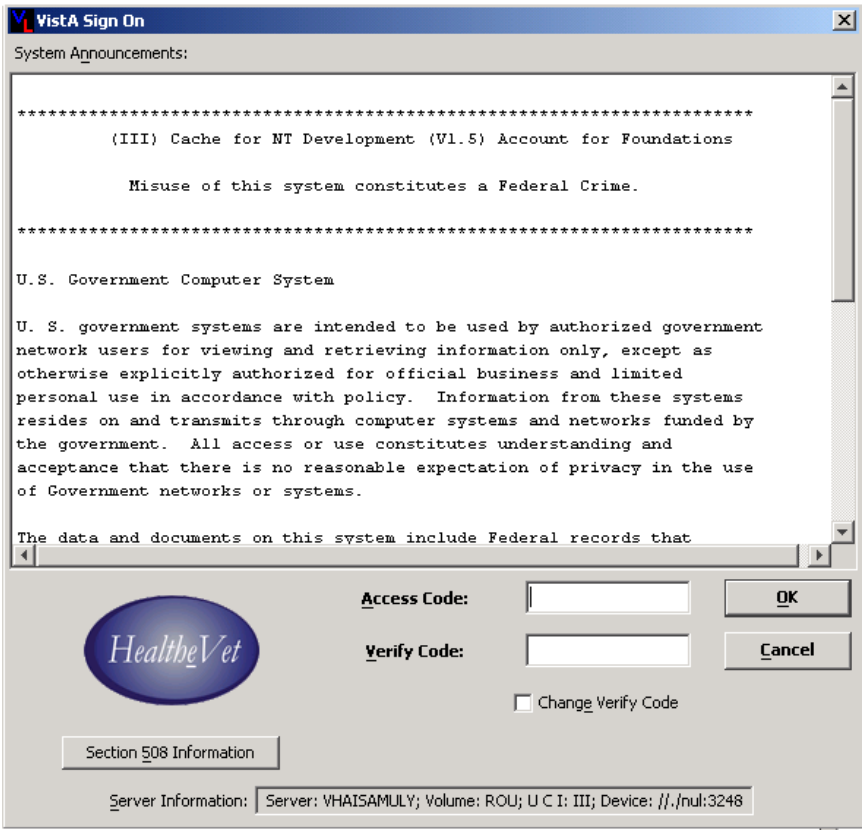

Figure A-2. Test Program Access/Verify Code Entry

5. If logon is successful, the status changes to "Connected." You can ping the M server, and also execute RPCs using the various tab options in the SwingTester application.

6. An interface with multiple tabs will display. Click on the **RPC List** tab. Type "**X**" in the **Enter namespace** box. Then click **Get RPC List** to display the information in the figure below.

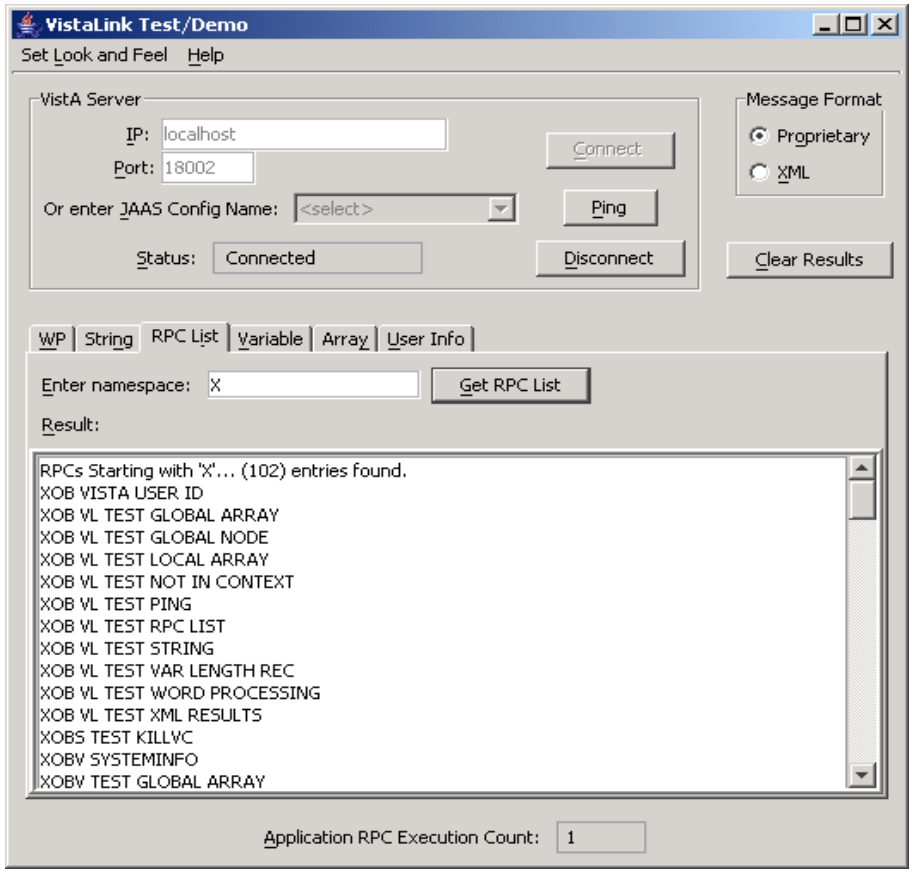

Figure A-3. SwingTester RPC List

7. To disconnect, press **Disconnect**.

Troubleshooting

If the application is unable to launch, check for errors in the command-window output. The most likely source of the problem is incorrect classpath locations set in the batch file.

When connected, you can also use the SwingTester sample app to display and verify your user information.

1. Click on the **User Info** tab in the interface shown in the figure below.

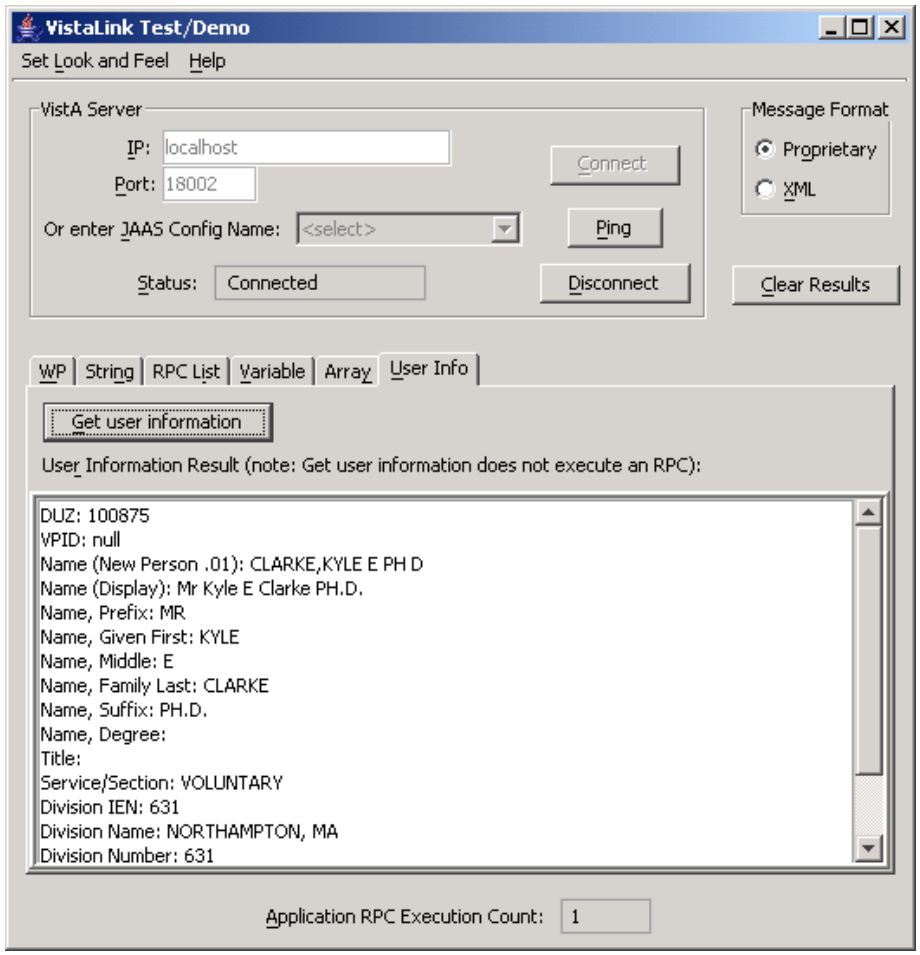

Figure A-4. Test Program User Information

2. Click **Get user information** to display your user data.

Running the Other Sample Applications

In addition to SwingTester, other sample applications are provided. Follow the steps provided in the section on the SwingTester sample application to modify **setVistaLinkEnvironment.bat** for your **JAVA_HOME** and for the locations of various libraries.

Unlike the SwingTester sample application, the remaining sample applications require the file **jaas.config** to be set up with configurations for your M server. (SwingTester allows free-form entry of M server IP and port to connect to.)

To set up jaas.config to hold the configuration for your M server's IP and port:

1. Modify the **jaas.config** file in your copied samples files, so that the settings for **ServerAddress** and **ServerPortKey** are correct for connecting to your M system.

   runRpcConsole.bat and **runSwingSimple.bat** are hard-coded to load a configuration named "DemoServer" from the **jaas.config** file. Either modify the DemoServer configuration with the settings needed for your M system, or, if you add a different configuration and configuration name, modify **runRpcConsole.bat** and **runSwingSimple.bat** to use your configuration name. (The `-s` parameter at the end of the command line that launches the application.)

   With **jaas.config** and **setVistaLinkEnvironment.bat** configured, you can then use the batch files described below to launch the other two sample applications.

runSwingSimple.bat

runSwingSimple.bat is a simpler Swing application than SwingTester. It is a better programming example program because it lacks the "bells and whistles" of SwingTester. It passes a command line parameter to specify which configuration in the **jaas.config** file should be used to connect to.

runRpcConsole.bat

runRpcConsole.bat is a console-only sample application. In addition to requiring a command-line parameter to specify the JAAS configuration to connect to, it is dependent on passing an access and verify code on the command line, unless the defaults embedded in the application work (they probably will not).

You can pass in access and verify codes with additional `"-a"` and `"-v"` command-line parameters.

Enabling Log4J Logging for Client Sample Applications (optional)

1. Assume that **c:\Program Files\vistalink\samples** is the current directory.

2. Folder **c:\Program Files\vistalink\samples\props** contains a sample **log4jconfig.xml** configuration file with various log4j configuration options.

3. Each sample application will try to load the log4j configuration from the file named "props\log4jconfig.xml," relative to the current directory. Therefore **c:\Program Files\vistalink\samples\props\log4jconfig.xml** will be loaded.

4. The **log4j2config.xml** file within the **c:\Program Files\vistalink\samples\props** folder contains extensive information on various log4j configuration options. Look at this simple example of a **log4j2config.xml** file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="gov.va.med.vistalink" level="trace" additivity="false">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

Figure A-5. log4jconfig.xml file contains extensive information on log4j configuration options

5. When you run the sample application, you should see "logger" output for debug and error information being displayed on the console window (the window in which you are starting up the application).

An example log4J properties file is provided in the <DIST FOLDER>samples-J2SE\props **folder in the distribution ZIP file.**

Sample Application Loggers

The following table lists all the loggers used by VistALink sample applications and log levels. System administrators may need to use this list when deciding which loggers to activate in the site's log4j configuration file.

This page is left blank intentionally.

Description	Environment (J2EE | J2SE)	Logger Name		Log Levels
		Package	Class	
Loggers for the sample applications that demonstrate VistALink functionality	J2SE	gov.va.med.vistalink.samples	VistaLinkRpcSwingSimple	Debug Error
	J2SE	"	VistaLinkRpcSwingSimpleCcow	Debug Error
	J2SE	"	VistaLinkRpcConsole	Debug Error
	J2SE	"	VistaLinkRpcConsole.Other	Error
	J2SE	"	VistaLinkRpcSwingTester	Debug
	J2EE	"	VistaLinkJ2EESample	Debug Error

Table A-6. VistALink Sample Application Loggers

This page is left blank intentionally.

Appendix B: DSM/VMS-Specific Install Information

NOTE: Most Office of Information and Technology (OI&T) sites have upgraded from Digital Standard Mumps (DSM)/VMS to Caché for VMS. DSM-specific installation information has been retained in this appendix.

Operating System Requirements

- DSM/VMS: DSM (version 7.2.1 for OpenVMS or greater)

  NOTE: Most DSM/VMS systems in VA OI&T have been converted to Caché/VMS.

Global Protection

Global Name	DSM for OpenVMS *	
^XOB	System:	RWP
	World:	RW
	Group:	RW
	UCI:	RW

Figure B-1. Global protection

Listener Management for Caché/VMS Systems

We recommend running VistALink on DSM/VMS systems as a TCP/IP service. See Appendix B, "Listener Management for DSM/VMS Systems," in the *VistALink 1.6 System Management Guide.*

This page is left blank intentionally.

Glossary

Access Code	A password used by the Kernel system to identify the user. It is used with the verify code.
Adapter	Another term for *resource adapter* or *connector*.
Administration Server	Each Oracle WebLogic server domain must have one server instance that acts as the administration server. This server is used to configure all other server instances in the domain.
Alias	An alternative filename.
Alpha/VMS	Alpha: Hewlett Packard computer system VMS: Virtual Memory System
Anonymous Software Directories	Directories where VHA application, documentation, and patch files are placed for distribution.
API	Application Program Interface
Application Proxy User	A Kernel user account designed for use by an application rather than an end-user.
Application Server	Software/hardware for handling complex interactions between users, business logic, and databases in transaction-based, multi-tier applications. Application servers, also known as app servers, provide increased availability and higher performance.
Authentication	Verifying the identity of the end-user.
Authorization	Granting or denying user access or permission to perform a function.
Base Adapter	Version 8.1 of WebLogic introduced a "link-ref" mechanism enabling the resources of a single "base" adapter to be shared by one or more "linked" adapters. The base adapter is a completely set up standalone adapter. Its resources (classes, jars, etc.) can be linked to and reused by other resource adapters (linked adapters), and the deployer only needs to modify a subset of linked adapters' deployment descriptor settings.
BEA WebLogic	BEA WebLogic is a J2EE Platform application server. Oracle has acquired BEA Systems, Inc. From here forward it will be referred to as Oracle.
Caché/VMS	Cache: InterSystems Caché object database that runs SQL VMS: Virtual Memory System
CCOW	The *Clinical Context Object Workgroup* is a standard defining the use of a technique called "context management," providing the clinician with a unified view on information held in separate and disparate healthcare applications that refer to the same patient, encounter or user.
Classpath	The path searched by the JVM for class definitions. The class path may be set by a command-line argument to the JVM or via an environment variable.
Client	Can refer to both the client workstation and the client portion of the program running on the workstation.

Connection Factory	A J2CA class for creating connections on request.
Connection Pool	A cached store of connection objects that can be available on demand and reused, increasing performance and scalability. VistALink 1.5 uses connection pooling.
Connector	A system-level driver that integrates J2EE application servers with Enterprise Information Systems (EIS). VistALink is a J2EE connector module designed to connect to Java applications with VistA/M systems. The term is used interchangeably with *connector module*, adapter, *adapter module*, and *resource adapter*.
Connector Proxy User	For security purposes, each instance of a J2EE connector must be granted access to the M server it connects to. This is done via a Kernel user account set up on the M system. This provides initial authentication for the app server and establishes a trusted connection. The M system manager must set up the connector user account and communicate the access code, verify code and listener IP address and port to the J2EE system manager.
COTS	Commercial, Off-The-Shelf
DBF	*Database file* format underlying many database applications (originally dBase)
DCL	*Digital Command Language*. An interactive command and scripting language for VMS.
Division	VHA sites are also called *institutions*. Each institution has a *station number* associated with it. Occasionally a single institution is made up of multiple sites, known as *divisions*. To make a connection, VistALink needs a station number from the end-user's New Person entry in the KERNEL SYSTEM PARAMETERS file (#8989.3). It looks first for a division station number and if it can't find one, uses the station number associated with default institution.
DSM	*Digital Standard MUMPS.* An M environment, a product of InterSystems Corp.
DUZ	Unknown acronym. A local variable holding a number that identifies the signed-on user. The number is the Internal Entry Number (IEN) of the user's record in the NEW PERSON file (file #200)
EAR file	*Enterprise archive* file. An enterprise application archive file that contains a J2EE application.
EIS	Enterprise Information System
FatKAAT	Fat-Client (i.e. Rich client) Kernel Authentication and Authorization
File #18	SYSTEM file #18 was the precursor to the KERNEL SYSTEM PARAMETERS file (#8989.3), and is now obsolete. It uses the same number space that is now assigned to VistALink. Therefore, file #18 must be deleted before VistALink can be installed.
FTP	File Transfer Protocol
Global	A multi-dimensional data storage structure -- the mechanism for persistent data storage in a MUMPS database.
GUI	Graphical User Interface

Health*e*Vet-VistA	The VHA is converting its MUMPS-based VistA healthcare system to a new J2EE-based platform and application suite. The new system is known as Health*e*Vet-VistA.
HL7	Health Level 7
IDE	*Integrated development environment.* A suite of software tools to support writing software.
Institution	VHA sites are also called *institutions*. Each institution has a *station number* associated with it. Occasionally a single institution is made up of multiple sites, known as *divisions*. To make a connection, VistALink needs a station number from the end-user's New Person entry in the KERNEL SYSTEM PARAMETERS file (#8989.3). It looks first for a division station number and if it can't find one, uses the station number associated with default institution.
Institution Mapping	The VistALink includes a small utility that administrators can use to associate station numbers with JNDI names, and which allows runtime code to retrieve the a VistALink connection factory based on station number.
IRM	Information Resource Management
ISO	Information Security Officer
J2CA	*J2EE Connector Architecture*. J2CA is a framework for integrating J2EE-compliant application servers with Enterprise Information Systems, such as the VHA's VistA/M systems. It is the framework for J2EE connector modules that plug into J2EE application servers, such as the VistALink adapter.
J2CA	J2EE Connector Architecture
J2CA CCI	J2EE Connector Architecture Common Client Interface
J2EE	The *Java 2 Platform, Enterprise Edition (J2EE)* is an environment for developing and deploying enterprise applications. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multi-tiered, Web-based applications. A J2EE Connector Architecture specification for building adapters to connect J2EE systems to non-J2EE enterprise information systems.
J2SE	*Java 2 Standard Edition.* Sun Microsystem's programming platform based on the Java programming language. It is the blueprint for building Java applications, and includes the Java Development Kit (JDK) and Java Runtime Environment (JRE).
JAAS	*Java Authentication and Authorization Service*. JAAS is a pluggable Java framework for user authentication and authorization, enabling services to authenticate and enforce access controls upon users.
JAR file	Java archive file. It is a file format based on the ZIP file format, used to aggregate many files into one.
Java Library	A library of Java classes usually distributed in JAR format.
Javadoc	Javadoc is a tool for generating API documentation in HTML format from doc comments in source code. Documentation produced with this tool is typically called Javadoc.
JBoss	JBoss is a free software / open source Java EE-based application server.

JDK	*Java Development Kit*. A set of programming tools for developing Java applications.
JMX	*Java Management eXtensions.* A java specification for building manageability into java applications, including J2EE-based ones.
JNDI	*Java Naming and Directory Interface*. A protocol to a set of APIs for multiple naming and directory services.
JRE	The *Java Runtime Environment* consists of the Java virtual machine, the Java platform core classes, and supporting files. JRE is bundled with the JDK but also available packaged separately.
JSP	*Java Server Pages*. A language for building web interfaces for interacting with web applications.
JVM	*Java Virtual Machine*. The JVM interprets compiled Java binary code (byte code) for specific computer hardware.
KAAJEE	Kernel Authentication and Authorization for Java 2 Enterprise Edition
Kernel	Kernel functions as an intermediary between the host M operating system and VistA M applications. It consists of a standard user and program interface and a set of utilities for performing basic VA computer system tasks, e.g., Menu Manager, Task Manager, Device Handler, and security.
KIDS	*Kernel Installation and Distribution System*. The VistA/M module for exporting new VistA software packages.
LDAP	Acronym for *Lightweight Directory Access Protocol.* LDAP is an open protocol that permits applications running on various platforms to access information from directories hosted by any type of server.
Linked Adapter	Version 8.1 of WebLogic introduced a "link-ref" mechanism enabling the resources of a single "base" adapter to be shared by one or more "linked" adapters. The base adapter is a completely set up standalone adapter. Its resources (classes, jars, etc.) can be linked to and reused by other resource adapters (linked adapters), and the deployer only needs to modify a subset of linked adapters' deployment descriptor settings.
Linux	An open-source Unix-like computer operating system that runs on various types of hardware platforms. Linux is one of the most prominent examples of free software and open source development; typically all underlying source code can be freely modified, used, and redistributed. Health*e*Vet-VistA servers use both Linux and Windows operating systems.
Listener	A socket routine that runs continuously at a specified port to field incoming requests. It sends requests to a front controller for processing. The controller returns its response to the client through the same port. The listener creates a separate thread for each request, so it can accept and forward requests from multiple clients concurrently.
log4J Utility	An open-source logging package distributed under the Apache Software license. Reviewing log files produced at runtime can be helpful in debugging and troubleshooting.
logger	In log4j, a logger is a named entry in a hierarchy of loggers. The names in the hierarchy typically follow Java package naming conventions. Application code

	can select a particular logger by name to write output to, and administrators can configure where a particular named logger's output is sent.
M (MUMPS)	*Massachusetts General Hospital Utility Multi-Programming System*, abbreviated M. M is a high-level procedural programming computer language, especially helpful for manipulating textual data.
Managed Server	A server instance in a Oracle WebLogic domain that is not an administration server, i.e., not used to configure all other server instances in the domain.
MBeans	In the Java programming language, an MBean (managed bean) is a Java object that represents a manageable resource, such as an application, a service, a component, or a device. MBeans must be concrete Java classes.
Messaging	A framework for one application to asynchronously deliver data to another application, typically using a queuing mechanism.
Multiple	A VA FileMan data type that allows more than one value for a single entry.
Namespace	A unique 2-4 character prefix for each VistA package. The DBA assigns this character string for developers to use in naming a package's routines, options, and other elements. The namespace includes a *number space*, a pre-defined range of numbers that package files must stay within.
NEW PERSON File #200	The NEW PERSON file contains information for all valid users on an M system.
NIST	National Institute for Standards and Technology
OI&T	Office of Information & Technology
Oracle WebLogic	Oracle WebLogic is a J2EE Platform application server. Oracle has acquired BEA Systems, Inc.
OS	Operating System
Patch	An update to a VistA software package that contains an enhancement or bug fix. Patches can include code updates, documentation updates, and information updates. Patches are applied to the programs on M systems by IRM services.
Plug-in	A component that can interact with or be added to an application without recompiling the application.
ra.xml	ra.xml is the standard J2EE deployment descriptor for J2CA connectors. It describes connector-related attributes and its deployment properties using a standard DTD (Document Type Definition) from Sun.
Re-authentication	When using a J2CA connector, the process of switching the security context of the connector from the original application connector "user" to the actual end-user. This is done by the calling application supplying a proper set of user credentials.
Resource Adapter	J2EE resource adapter modules are system-level drivers that integrate J2EE application servers with Enterprise Information Systems (EIS). This term is used interchangeably with *resource adapter* and *connector*.
Routine	A program or sequence of computer instructions that may have some general or frequent use. M routines are groups of program lines that are saved, loaded, and called as a single unit with a specific name.

RPC	*Remote Procedure Call*. A defined call to M code that runs on an M server. A client application, through the RPC Broker, can make a call to the M server and execute an RPC on the M server. Through this mechanism a client application can send data to an M server, execute code on an M server, or retrieve data from an M server
RPC Broker	The RPC Broker is a client/server system within VistA. It establishes a common and consistent framework for client-server applications to communicate and exchange data with VistA/M servers.
RPC Security	All RPCs are secured with an RPC context (a "B"-type option). An end-user executing an RPC must have the "B"-type option associated with the RPC in the user's menu tree. Otherwise an exception is thrown.
SAD	Software Architecture Document
SE&I	Software Engineering & Integration
Servlet	A Java program that resides on a server and executes requests from client web pages.
Socket	An operating system object that connects application requests to network protocols.
SRS	Software Requirements Specification
TCP/IP	Transmission Control Protocol (TCP) and the Internet Protocol (IP),
TXT	Text file format
VA	Department of Veterans Affairs
VACO	Veterans Affairs Central Office
Verify Code	A password used in tandem with the access code to provide secure user access. The Kernel's Sign-on/Security system uses the verify code to validate the user's identity.
VistA	*Veterans Health Information Systems and Technology Architecture*. The VHA's portfolio of M-based application software used by all VA medical centers and associated facilities.
VistALink Libraries	Classes written specifically for VistALink.
VL	*VistaLink* is a runtime and development tool providing connection and data conversion between Java and M applications in client-server and n-tier architectures, to which this document describes the architecture and design.
VMS	*Virtual Memory System*. An operating system, originally designed by DEC (now owned by Hewlett-Packard), that operates on the VAX and Alpha architectures.
VPID	*VA Person Identifier*. A new enterprise-level identifier uniquely identifying VA 'persons' across the entire VA domain.
WAR file	*Web archive* file. Contains the class files for servlets and JSPs.
WebLogic Server	A J2EE application server manufactured by Oracle WebLogic Systems.
WebSphere	WebSphere Application Server (WAS) is and IBM application server.
XLS	Microsoft Office XL worksheet and workbook file format

XML	Extensible Markup Language
XmlBeans	XMLBeans is a Java-to-XML binding framework which is part of the Apache Software Foundation XML project.
XOB Namespace	The VistALink namespace. All VistALink programs and their elements begin with the characters "XOB."

REF: For a comprehensive list of commonly used infrastructure- and security-related terms and definitions, please visit the Security and Other Common Services Glossary Web page at the following Web address:

http://vista.med.va.gov/iss/glossary.asp

For a comprehensive list of acronyms, please visit the Security and Other Common Services Acronyms Web site at the following Web address:

http://vista/med/va/gov/iss/acronyms/index.asp

This page is left blank intentionally.