VistALink 1.5

Installation Guide

May 2006

Revision History

Date	Revision	Description	Contacts
5/11/06	1.5	Initial VistALink 1.5 release.	Jim Alexander, technical writer Dawn Clark, project manager

Revision History

Table of Contents

List of Figures

List of Tables

Introduction

VistALink 1.5 Overview

The VistALink 1.5 resource adapter is a transport layer that provides communication between Health*e*Vet-VistA Java applications and VistA/M servers, in both client-server and n-tier environments. It allows java applications to execute remote procedure calls (RPCs) on the VistA/M system and retrieve results, synchronously. VistALink 1.5 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 (J2C) 1.0 specification to integrate Java applications with legacy systems.

- The M listener process receives and processes requests from client applications.

VistALink 1.5 can be installed on a VistA/M system with or without previous installation of VistALink 1.0. If version 1.0 is already present, only the new features of VistALink 1.5 will be installed.

Document Overview

This manual provides information for installing the VistALink 1.5 resource adapter and M-side listener. Its intended audience includes J2EE application server administrators, IRM IT Specialists at VHA facilities, and developers of Java applications requiring communication with VistA/M systems.

Developers and administrators will need to use this document in conjunction with the *VistALink 1.5 System Management Guide,* which contains detailed information on J2EE application server management, institution mapping, the VistALink console, M listener management, and VistALink security, logging, and troubleshooting.

Generally, the installation instructions presented here assume the use of Windows as the client operating system. Where appropriate, separate steps are displayed for Linux, in the following fashion:

	Special instructions for Linux systems are set off and indicated with the Linux "Tux" penguin icon.

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 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_5.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//   TELNET PORT
```

Bold font is also used in code samples to indicate lines of particular interest, discussed in the preceding text:

```
<!DOCTYPE weblogic-connection-factory-dd PUBLIC '-//BEA Systems, Inc.//DTD
WebLogic 8.1.0 Connector//EN'
'http://www.bea.com/servers/wls810/dtd/weblogic810-ra.dtd'>
<weblogic-connection-factory-dd>
  <connection-factory-name>VistaLinkAdapter</connection-factory-name>
  <jndi-name>vlj/testconnector</jndi-name>
  <pool-params>
    <initial-capacity>1</initial-capacity>
    <max-capacity>1</max-capacity>
```

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

Folder Conventions

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

<DIST FOLDER>	The location for the unzipped VistALink file.
<APPLICATION STAGING FOLDER>	A folder where EAR, WAR and RAR distributions are placed on your application server prior to deployment
<HEV CONFIGURATION FOLDER>	A folder placed on the classpath of WebLogic servers, containing configuration files for all Health*e*Vet-VistA applications.

Additional Resources

VistALink Web Site

The VistALink website (http://vista.med.va.gov/migration/foundations/vl/index.htm) summarizes VistALink architecture and functionality and presents status updates.

VistALink Documentation Set

The following documents are provided in the VistALink 1.5 documentation set:

- *VistALink 1.5 Installation Guide*: Provides detailed instructions for setting up, installing, and configuring the VistALink 1.5 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 1.5 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 1.5 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 1.5 Release Notes*: Lists all new features included in the VistALink 1.5 release.

- *Getting Started With the BDK, Chapter 3*: *RPC Overview*. A short guide on writing RPCs from the *RPC Broker* manual.

BEA Systems

VistALink 1.5 has been tested and is supported on BEA WebLogic Server 8.1 (Service Pack 4) only. WebLogic product documentation can be found at the following website: http://edocs.bea.com/.

1 Installation Overview

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, different administrators may be responsible for the two server types, and 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.

This chapter presents an overview of the steps required to install the VistALink 1.5 resource adapter (connector) on each server. The general process of installing the adapter on both server types is as follows:

1. Obtain the VistALink 1.5 distribution file
2. Install the KIDS file and update the VLINK command files
3. Deploy the adapter
4. Test the connection between the J2EE application server and the intended VistA/M system by running a sample program included in the distribution file.

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.

1.1 Restrictions

Version 1.5.0 is the target version for the final release. **Pre-release iterations (versions 1.5.0.devxx) should not be used in a production environment**.

VistALink 1.5 has been tested and is supported on BEA WebLogic Server 8.1 (Service Pack 4 or higher) only.

1.2 System Administrators

It is assumed that the administrators installing VistALink 1.5 will have basic working knowledge of the systems they are administering and deploying applications to. For VistA/M installations, the installer should have working knowledge of VistA/M system administration. Likewise, it is assumed that a J2EE installer has working knowledge of J2EE system administration. **It is strongly recommend that both types of administrators obtain training necessary to administer both system types.**

1.3   VistALink Distribution Zip File

The person deploying the resource adapter can obtain the VistALink distribution zip file from one of the **anonymous.software** directories. The distribution zip file contains:

(root)	Readme.txt, ReleaseNotes.rtf
console\	VistALink console application (packaged and exploded)
jars\	VistALink jar files
javadoc\	API javadoc
log4j\	logger spreadsheet, and sample log4j config files
m\	KIDS build for VistA/M server
rar\	VistALink connector
rar\configExamples\	example configuration files
rar\ExplodedVistaLinkRar\	exploded VistALink connector
samples\	
samples\J2EE\	Sample J2EE application (packaged and exploded)
samples\J2SE\	client/server sample applications

1.4   Installation Summary

1.4.1 VistA/M Server

The detailed instructions for installing VistALink on the VistA/M server are presented in chapter 2, "M Server Installation Procedures." The general steps for installing VistALink on the VistA/M server and links to the appropriate sections in this manual are as follows:

1. Check installation prerequisites ("Preparation").
2. Install the KIDS build ("Installing VistAlink 1.5 KIDS Build").
3. Set up the VistALink listener ("Setting up the Listener").
4. Test the listener ("Verifying Listener Connectivity").
5. Create the connector proxy user for a specific J2EE server (or data center). This step creates a VistA/M user account for initial authentication for the application server ("Post-Install: Configuring Connector Proxy User(s) for J2EE Access").

1.4.2 J2EE Application Server

The detailed instructions for installing VistALink on the J2EE application server are presented in Chapter 3, "WebLogic Application Server Installation Procedures." The general steps for installing VistALink on the J2EE application server and links to the appropriate sections in this guide are as follows:

1. Preparation: review system requirements, request connector proxy user credentials, and obtain VistALink 1.5 distribution file ("Preparation").

2. If upgrading from a previous installation, remove jars and undeploy the VistALink Console and Sample Applications before the installation ("Upgrading a Previous Installation").
3. Install the base resource adapter/connector ("Installing the VistALink 1.5 Adapter").
4. Verify that the installation is successful ("Verifying Successful Adapter Installation or Upgrade").
5. Deploy the VistALink console ("Deploying the VistALink Console").
6. Deploy the sample application ("Deploying the Sample J2EE Application").
7. (Optional) Re-configure the adapter to connect to your M system. ("Testing the Sample App with Your Own M Server.")

2 VistA/M Server Installation Procedures

2.1 Overview

The figure below summarizes the flow of steps for installing and testing VistALink 1.5 on the VistA/M server. Each step references the section of this document where it is described in detail.
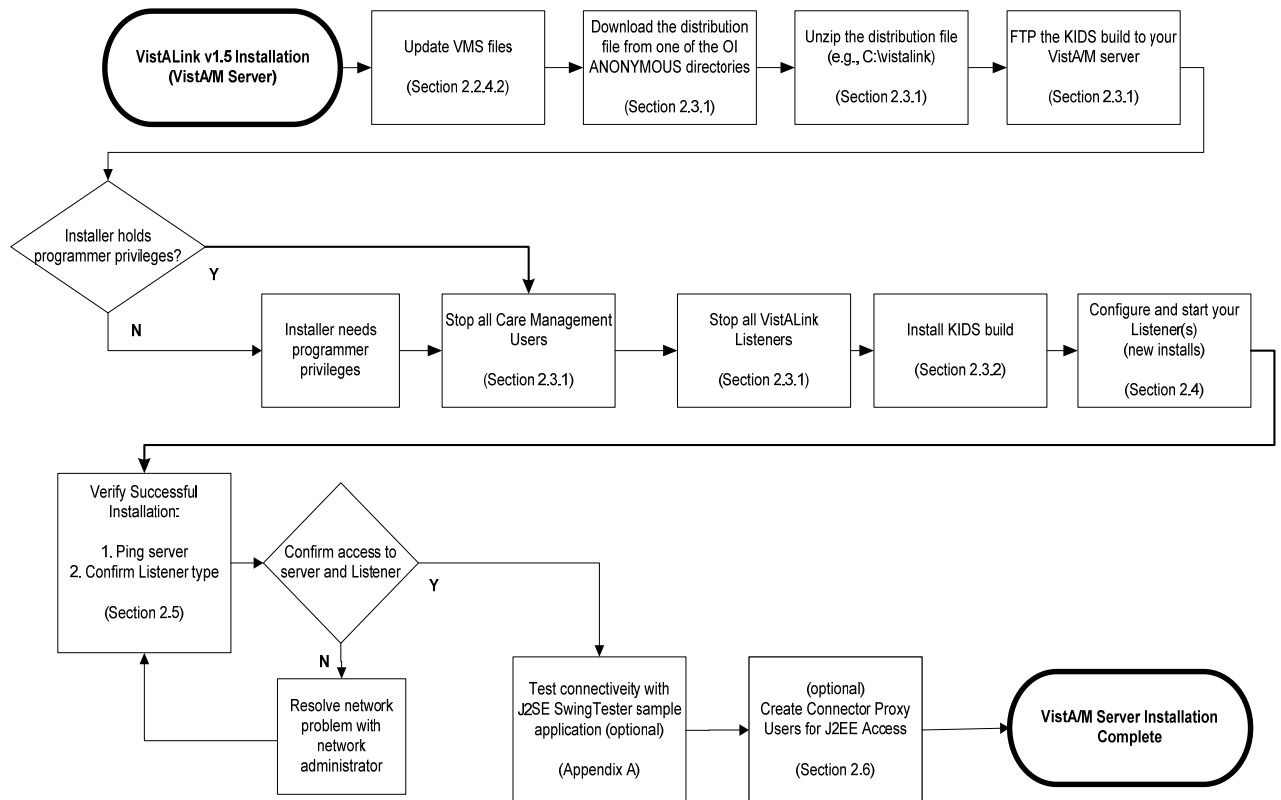
Figure 1. Flowchart for VistALink 1.5 VistA/M Server Install

2.2 Preparation

2.2.1 Software Installation Time

The estimated installation time for the installation of VistALink 1.5 is less than five minutes.

2.2.2 Virgin Installations

It is not necessary for VistALink 1.0 to be installed on your VistA/M server before you install VistALink 1.5.

2.2.3 **System Processes**

- VistALink users must be stopped
- The VistALink TCP/IP service (VLINK) must be disabled
- Roll-and-scroll and RPC Broker users may remain on the system
- TaskMan does not need to be put into a wait state

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

2.2.4 **System Requirements**

2.2.4.1 **Patch Requirements**

Before the VistALink 1.5 installation, the following packages and patches must be installed:

Software	Version	Patch Information
Kernel	8.0	Fully patched, including: XU*8*265 XU*8*337 XU*8*361
Kernel Toolkit	7.3	Fully patched, including: XT*7.3*89
MailMan	8.0	Fully patched.
RPC Broker	1.1	Fully patched.
VA FileMan	22.0	Fully patched.

2.2.4.2 **Operating System Requirements**

- Caché/VMS: Caché (version 4.1 or greater)
- Caché/NT: Caché (version 3.2.31.1 or greater)

  Note: "NT" means Windows NT, 2000, XP, etc.

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

  Note: DSM/VMS systems have been converted to Caché/VMS.

2.2.4.3 **VistA/M Server Permissions**

Programmer access (DUZ(0)="@") is required for installing VistALink 1.5. Installers who are also creating/modifying the VLINK .COM files, must also hold sufficient VMS

privileges (e.g., SYSPRV). On a DSM/VMS system, the installer must have a VMS account.

2.2.4.4 <u>Namespaces</u>

VistALink has been assigned the XOB* namespace.

2.2.4.5 <u>File and Global Information</u>

VistALink 1.5 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,	@

2.2.5 System Preparation

2.2.5.1 <u>Global Placement, Mapping, and Translation</u>

VistALink utilizes one VistALink-specific global, ^XOB. For virgin installs, ^XOB should be placed in a location appropriate for a small, static global, prior to installation: database in Caché, volume set in DSM. For M configurations with multiple databases or volume sets, any necessary mapping or translation should be set up at this time as well.

2.2.5.2 <u>Journaling</u>

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

2.2.5.3 <u>Protection</u>

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

* Most DSM/VMS systems have been converted to Caché/VMS.

2.2.6 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).

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

2.2.7 Deleting File #18

During 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 1.5 includes steps to check and clean up file #18.

This file was created in the early 1980s and was a precursor to the current Kernel System Parameters file. 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.

You may wish to manually back up and delete System file #18. If this file is on your system, the VistALink environment check will ask you a series of questions during the installation phase to either abort the installation or allow the VistALink installation to delete the file for you.

2.3 Installing VistALink 1.5 KIDS Build

Follow the steps in this section to install VistALink 1.5. Section 2.3.3 contains an example of a complete VistALink 1.5 installation on a VistA/M server.

2.3.1 Preliminary Steps

1. Find the VistALink 1.5 KIDS build (XOB_1_5.KID) in the **m** folder of the VistALink distribution zip file. You can download the distribution file from the **anonymous.software** directory on any of the OIFO FTP download sites.

> **i** The VistALink 1.5 KIDS distribution is contained in the **m** folder of the VistALink distribution zip file. It is also available as a standalone file on the anonymous.software directories.

2. FTP 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).

4. While installing this package on the server, do not run any VistALink-based Client/Server software (e.g., Care Management).

> 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.

5. 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.

6. Stop all VistALink users.

2.3.2 Build Installation

1. Use the KIDS Installation option, **Load a Distribution [XPD LOAD DISTRIBUTION]**. Enter "XOB_1_5.KID" as the name of the Host File. This will load three transport globals contained within the distribution:

- XOBU 1.5   Common files and libraries used by all the XOB* packages and menu options to manage site parameters/operations
- XOBV 1.5   Handles system and RPC requests
- XOBS 1.5   M-side security module

2. You can 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. Use "XOBU 1.5" as the response to the **Select INSTALL NAME:** prompt.

   Follow the example below:

```
Select Installation Option:  Verify Checksums in Transport Global
Select INSTALL NAME: XOBU 1.5       Loaded from Distribution  12/17/05@11:46:46
    => Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]

This Distribution was loaded on Dec 17, 2005@11:46:46 with header of
   Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
 ;Created on Sep 09, 2005@15:09:32
   It consisted of the following Install(s):
      XOBU 1.5       XOBV 1.5       XOBS 1.5
DEVICE: HOME//   TELNET PORT



PACKAGE: XOBU 1.5    Dec 17, 2005 11:49 am                              PAGE 1
--------------------------------------------------------------------------------


   5 Routine checked, 0 failed.
```

```
PACKAGE: XOBV 1.5     Dec 17, 2005 11:49 am                           PAGE 1
-------------------------------------------------------------------------------


   17 Routine checked, 0 failed.



PACKAGE: XOBS 1.5     Dec 17, 2005 11:49 am                           PAGE 1
-------------------------------------------------------------------------------


   7 Routine checked, 0 failed.
```

> **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.

3. Use the KIDS Installation option, **Backup a Transport Global [XPD BACKUP]**. This option creates a MailMan message that will backup 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.)

   Follow the example below:

```
Select Installation Option: BACKUP a Transport Global
Select INSTALL NAME: XOBU 1.5      Loaded from Distribution  12/17/04@11:46:46
     => Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0]

This Distribution was loaded on Dec 17, 2005@11:46:46 with header of
  Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
 ;Created on Sep 09, 2005@15:09:35
   It consisted of the following Install(s):
      XOBU 1.5       XOBV 1.5       XOBS 1.5
Subject: Backup of XOBU 1.5 install on Dec 17, 2005
  Replace
Loading Routines for XOBU 1.5.....
Loading Routines for XOBV 1.5.
Routine XOBVLJU is not on the disk................
Loading Routines for XOBS 1.5....
Routine XOBSRA is not on the disk..
Routine XOBSRA1 is not on the disk..
Routine XOBSRAKJ is not on the disk..
Send mail to: CLARK,DAWN//   CLARK,DAWN
Select basket to send to: IN// J2M
```

4. Use the KIDS Installation option, **Install Package(s) [XPD INSTALL BUILD]** to install VistALink 1.5.

   Enter "XOBU 1.5" at the **Select Install Name:** prompt and answer the 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?
    ```

5. If VistALink has already been set up on your server, 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 "Setting up the 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.

2.3.3 Sample VistA/M Installation

The following is an example of a VistALink 1.5 installation on a VistA/M server:

```
Installation of XOBU 1.5 on a Caché/VMS system already running XOBU 1.0


Select Kernel Installation & Distribution System Option: INSTallation


   1        Load a Distribution
   2        Verify Checksums in Transport Global
   3        Print Transport Global
   4        Compare Transport Global to Current System
   5        Backup a Transport Global
   6        Install Package(s)
            Restart Install of Package(s)
            Unload a Distribution
            Play a script

Select Installation Option: 1  Load a Distribution
Enter a Host File: USER$:[CLARK]XOB_1_5.KID

KIDS Distribution saved on Sep 09, 2005@15:09:32
Comment: Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
```

```
This Distribution contains Transport Globals for the following Package(s):
   XOBU 1.5
   XOBV 1.5
   XOBS 1.5
Distribution OK!

Want to Continue with Load? YES// y  YES
Loading Distribution...
Build XOBU 1.5 has an Environmental Check Routine
Want to RUN the Environment Check Routine? YES// YES
   XOBU 1.5
Will first run the Environment Check Routine, XOBUENV

 >>> Checking environment...


 >>> VistALink environment check completed for KIDS Load a Distribution option.

   XOBV 1.5
   XOBS 1.5


Use INSTALL NAME: XOBU 1.5 to install this Distribution.



   1        Load a Distribution
   2        Verify Checksums in Transport Global
   3        Print Transport Global
   4        Compare Transport Global to Current System
   5        Backup a Transport Global
   6        Install Package(s)
            Restart Install of Package(s)
            Unload a Distribution
            Play a script

Select Installation Option: 2  Verify Checksums in Transport Global
Select INSTALL NAME: XOBU 1.5       Loaded from Distribution  12/28/05@09:04:06
     => Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]

This Distribution was loaded on Dec 28, 2005@09:04:06 with header of
   Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
 ;Created on Sep 09, 2005@15:09:32
   It consisted of the following Install(s):
      XOBU 1.5       XOBV 1.5        XOBS 1.5
DEVICE: HOME//   IP network




PACKAGE: XOBU 1.5    Dec 28, 2005 9:04 am                             PAGE 1
-------------------------------------------------------------------------------



   5 Routine checked, 0 failed.




PACKAGE: XOBV 1.5    Dec 28, 2005 9:04 am                             PAGE 1
-------------------------------------------------------------------------------
```

```
   17 Routine checked, 0 failed.




PACKAGE: XOBS 1.5    Dec 28, 2005 9:04 am                               PAGE 1
-------------------------------------------------------------------------------



   7 Routine checked, 0 failed.



   1        Load a Distribution
   2        Verify Checksums in Transport Global
   3        Print Transport Global
   4        Compare Transport Global to Current System
   5        Backup a Transport Global
   6        Install Package(s)
            Restart Install of Package(s)
            Unload a Distribution
            Play a script

Select Installation Option: Backup a Transport Global
Select INSTALL NAME: XOBU 1.5       Loaded from Distribution  12/28/05@09:04:06
     => Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]

This Distribution was loaded on Dec 28, 2005@09:04:06 with header of
   Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
 ;Created on Sep 09, 2005@15:09:32
   It consisted of the following Install(s):
      XOBU 1.5        XOBV 1.5        XOBS 1.5
Subject: Backup of XOBU 1.5 install on Dec 28, 2005
  Replace
Loading Routines for XOBU 1.5.....
Loading Routines for XOBV 1.5.
Routine XOBVLJU is not on the disk.................
Loading Routines for XOBS 1.5....
Routine XOBSRA is not on the disk..
Routine XOBSRA1 is not on the disk..
Routine XOBSRAKJ is not on the disk..
Send mail to: CLARK,DAWN//   CLARK,DAWN
Select basket to send to: IN// J2M

   1        Load a Distribution
   2        Verify Checksums in Transport Global
   3        Print Transport Global
   4        Compare Transport Global to Current System
   5        Backup a Transport Global
   6        Install Package(s)
            Restart Install of Package(s)
            Unload a Distribution
            Play a script

Select Installation Option: 6  Install Package(s)
Select INSTALL NAME:   XOBU 1.5    Loaded from Distribution  12/28/05@09:04:06
     => Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]

This Distribution was loaded on Dec 28, 2005@09:04:06 with header of
```

```
   Foundations, VistALink, and VistALink Security v1.5 [Build: 1.5.0.026]
 ;Created on Sep 09, 2005@15:09:32
   It consisted of the following Install(s):
       XOBU 1.5        XOBV 1.5        XOBS 1.5
Checking Install for Package XOBU 1.5
Will first run the Environment Check Routine, XOBUENV

 >>> Checking environment...


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


Install Questions for XOBU 1.5

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? YES//

Checking Install for Package XOBV 1.5

Install Questions for XOBV 1.5

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? YES//

Checking Install for Package XOBS 1.5


Install Questions for XOBS 1.5



Want KIDS to INHIBIT LOGONs during the install? YES//
Want to DISABLE Scheduled Options, Menu Options, and Protocols? YES// NO

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//   IP network
```

```
 Install Started for XOBU 1.5 :
               Dec 28, 2005@09:05:38

Build Distribution Date: Sep 09, 2005

 Installing Routines:
               Dec 28, 2005@09:05:38

 Running Pre-Install Routine: EN^XOBUPRE

 Installing Data Dictionaries:
               Dec 28, 2005@09:05:38

 Installing PACKAGE COMPONENTS:

 Installing INPUT TEMPLATE

 Installing PROTOCOL
  Located in the XOBV (VISTALINK) namespace.

 Installing LIST TEMPLATE

 Installing OPTION
               Dec 28, 2005@09:05:39

 Running Post-Install Routine: EN^XOBUPOST


 Updating Routine file...

 Updating KIDS files...

 XOBU 1.5 Installed.
               Dec 28, 2005@09:05:39

 Install Message sent #2074

 Install Started for XOBV 1.5 :
               Dec 28, 2005@09:05:39

Build Distribution Date: Sep 09, 2005

 Installing Routines:
               Dec 28, 2005@09:05:39

 Running Pre-Install Routine: EN^XOBVPRE

 Installing Data Dictionaries:
               Dec 28, 2005@09:05:41

 Installing Data:
               Dec 28, 2005@09:05:41

 Installing PACKAGE COMPONENTS:

 Installing INPUT TEMPLATE

 Installing DIALOG

 Installing PROTOCOL

 Installing REMOTE PROCEDURE
```

```
 Installing OPTION
               Dec 28, 2005@09:05:41

 Running Post-Install Routine: EN^XOBVPOST

Added new Kernel Application Proxy User 'XOBVTESTER,APPLICATION PROXY'.
  ::This application proxy user account is used in the VistALink sample web
  ::application, to demonstrate usage of the VistaLinkAppProxyConnectionSpec
  ::connection spec.

 Updating Routine file...

 Updating KIDS files...

 XOBV 1.5 Installed.
               Dec 28, 2005@09:05:41

 Install Message sent #2075

 Install Started for XOBS 1.5 :
               Dec 28, 2005@09:05:41

Build Distribution Date: Sep 09, 2005

 Installing Routines:
               Dec 28, 2005@09:05:42

 Installing PACKAGE COMPONENTS:

 Installing DIALOG
               Dec 28, 2005@09:05:42

 Updating Routine file...

 Updating KIDS files...

 XOBS 1.5 Installed.
               Dec 28, 2005@09:05:42

 Install Message sent #2076

 Call MENU rebuild

Starting Menu Rebuild:  Dec 28, 2005@09:05:44

Collecting primary menus in the New Person file...

                   Primary menus found in the New Person file
                   ------------------------------------------

OPTION NAME          MENU TEXT                      # OF       LAST     LAST
                                                    USERS      USED     BUILT


DIUSER               VA FileMan                     1          10/01/03 07/28/05
XMUSER               MailMan Menu                   17         05/17/05 07/28/05
EVE.MGT01            Main Menu for Clinical Staff   1                   07/28/05
ZZUSER.MCCR          MCCR Menu                      6          10/28/03 07/28/05
ZZUSER.FRM01         Fileroom Menu                  1          10/28/03 07/28/05
EVE.MGT02            Main Menu for Management       1          10/28/03 07/28/05
ZZ PHARMACIST        Anchorage Pharmacy             15         10/28/03 07/28/05
EVE.FIS01            Fiscal Service Package Co...   4          10/28/03 07/28/05
ZZEVE.MAS            MAS Main Menu                  1          10/28/03 07/28/05
ZZEVE.SWS            Social Work Service Coord...   2          10/28/03 07/28/05
```

```
ZZUSER.FIS01          Fiscal - Operations Menu    6          10/28/03   07/28/05
----snip-----



Building secondary menu trees....

Merging.... done.

Menu Rebuild Complete:  Dec 28, 2005@09:06:14
_____

  100%              25              50              75
Complete




Install Completed
```

Figure 2. VistALink J2M Installation Example

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

> ⓘ 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.

2.4 Setting up the Listener

2.4.1 Introduction

For Java applications to connect to your VistA/M system using VistALink, the VistALink
listener(s) must be configured to start running on your M system (although not necessarily
in M). It waits for and accepts incoming client connections on a specified TCP port, and
spawns off handler jobs to service those connection requests.

Configuration of the listener(s) will vary depending on the operating system in use. The
sections below provide setup requirements for the Caché/VMS, Caché/NT, and
DSM/VMS operating systems, as well as general information for all operating systems.

2.4.2 VistALink Listeners and 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**.

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

2.4.2.1 Listener Topography

VistALink offers the following listener/Port/IP address possibilities:

- A single VistALink listener, running on any available port.

- Multiple VistALink listeners running on the same IP address/CPU, but listening on *different* ports.

To run one listener in a production account and another in a test account on the same IP address/CPU, 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/CPUs, the ports can be the same (e.g., one VistALink listener on every system listening on port 8000).

Clients accessing your listener will need to be configured with the appropriate listener IP and port.

2.4.3 Listener Management for Caché/VMS Systems

We recommend running VistALink on Caché/VMS and DSM/VMS systems as a 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. The TaskMan process does not need to be running on the same node as the node the VistALink listener(s) reside on.

The method for starting the TCP listener was written in collaboration with HSITES to aid IRM support staff in running VistALink listener(s) on an M server as a TCP/IP service.

A TCP/IP cookbook and associated VLINK command files to enable and manage VistALink TCP/IP services have been written by HSITES, and can be downloaded from the [ANONYMOUS.SOFTWARE] directory at the following FTP sites:

OIFO	FTP ADDRESS	DIRECTORY
Download Site	download.vista.med.va.gov	anonymous.software
Albany	ftp.fo-albany.med.va.gov	anonymous.software
Hines	ftp.fo-hines.med.va.gov	anonymous.software
Salt Lake City	ftp.fo-slc.med.va.gov	anonymous.software

The following files are available:

FILE NAME	DESCRIPTION
VISTALINK_TCPIP_COOKBOOK.DOC	VistALink TCP/IP service cookbook
VLINK_CREATE_UAF.COM	Used to create OpenVMS user account
VLINK_CREATE_SERVICE.COM	Used to create the TCP/IP service for VistALink

VLINK.COM	Used by the VistALink service

These files are provided to assist you in creating or modifying VistALink's VMS user account and command files for both test and production environments. Note that the VLINK files for VistALink 1.5 have changed (e.g., no PIPE commands) from those you might have set up for the VistALink 1.0 service.

- 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 the VLINK service, refer to the HSITES cookbook for step-by-step instructions.
- If you have created the VLINK service:
  - Use the TCP/IP utilities to disable the service

    ```
    TCPIP> DISABLE SERVICE VLINK
    ```

  - FTP the new VLINK.COM file from the ANONYMOUS directory (remember to use ASCII mode when you get the file).
  - Copy the new VLINK.com file to the directory used by the VLINK service.
  - Modify the file to match the 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.
  - Enable the VLINK service

    ```
    TCPIP> ENABLE SERVICE VLINK
    ```

In general, use the VistALink TCP/IP cookbook and VLINK files to help you:

- Set up VistALink as a TCP/IP service in VMS
- Modify the new VLINK files to match your environment
- Modify the VLINK VMS user account (and the .COM file to create the account) with the proper authorized and default privileges (e.g., remove OPER from both). Here are the steps:

  - Enter the VMS authorize utility and SHOW the account to get a 'before' picture

    ```
    LASHLEYA_3A1$ MCR AUTHORIZE


    UAF> SHOW VLINK
    ```

```
Username: VLINK                                    Owner:  VLINK
Account:  NETWORK                                  UIC:    [50,173]
([VLINK])
CLI:      DCL                                      Tables: DCLTABLES
Default:  USER$:[VLINK]
LGICMD:   NL:
Flags:  DisCtlY Restricted DisWelcome DisNewMail DisMail
DisReport
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                       Sat Sun
Primary   00000000001111111111222   Secondary
00000000001111111111222
Day Hours 012345678901234567890123  Day Hours
012345678901234567890123
Network:  ##### Full access ######           ##### Full access
######
Batch:    -----  No access  ------            -----  No access
------
Local:    -----  No access  ------            -----  No access
------
Dialup:   -----  No access  ------            -----  No access
------
Remote:   -----  No access  ------            -----  No access
------
Expiration:            (none)    Pwdminimum:  6   Login Fails:
0
Pwdlifetime:         90 00:00    Pwdchange:       (pre-expired)
Last Login:            (none) (interactive),  9-FEB-2006 08:04
(non-interactive)
Maxjobs:         0  Fillm:         300  Bytlm:         120000
Maxacctjobs:     0  Shrfillm:        0  Pbytlm:             0
Maxdetach:       0  BIOlm:        1024  JTquota:         4096
Prclm:          32  DIOlm:        2048  WSdef:          13000
Prio:           10  ASTlm:        2098  WSquo:          20000
Queprio:         4  TQElm:          10  WSextent:       65536
CPU:        (none)  Enqlm:        3005  Pgflquo:       120000
Authorized Privileges:
  NETMBX        OPER          TMPMBX
Default Privileges:
  NETMBX        OPER          TMPMBX
UAF>
```

o Now, use the MODIFY command to remove the OPER privilege.

```
UAF> MOD VLINK/DEFPRIVILEGES=NOOPER/PRIVILEGES=NOOPER
%UAF-I-MDFYMSG, user record(s) updated
```

o SHOW the VLINK account again to verify that the privilege has been
   removed from the account as both an Authorized and a Default privilege.

```
UAF> SHOW VLINK

Username: VLINK                                    Owner:  VLINK
```

```
Account:  NETWORK                              UIC:    [50,173]
([VLINK])
CLI:      DCL                                  Tables: DCLTABLES
Default:  USER$:[VLINK]
LGICMD:   NL:
Flags:  DisCtlY Restricted DisWelcome DisNewMail DisMail
DisReport
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
Primary   000000000011111111112222  Secondary
000000000011111111112222
Day Hours 012345678901234567890123  Day Hours
012345678901234567890123
Network:  ##### Full access ######       ##### Full access
######
Batch:    -----  No access  ------         -----  No access
------
Local:    -----  No access  ------         -----  No access
------
Dialup:   -----  No access  ------         -----  No access
------
Remote:   -----  No access  ------         -----  No access
------
Expiration:             (none)   Pwdminimum: 6   Login Fails:
0
Pwdlifetime:         90 00:00   Pwdchange:      (pre-expired)
Last Login:          (none) (interactive),  9-FEB-2006 08:04
(non-interactive)
Maxjobs:        0  Fillm:        300  Bytlm:       120000
Maxacctjobs:    0  Shrfillm:       0  Pbytlm:           0
Maxdetach:      0  BIOlm:       1024  JTquota:       4096
Prclm:         32  DIOlm:       2048  WSdef:        13000
Prio:          10  ASTlm:       2098  WSquo:        20000
Queprio:        4  TQElm:         10  WSextent:     65536
CPU:        (none)  Enqlm:       3005  Pgflquo:      120000
Authorized Privileges:
  NETMBX       TMPMBX
Default Privileges:
  NETMBX       TMPMBX
```

o Exit the VMS authorize utility

```
UAF> EXIT
%UAF-I-DONEMSG, system authorization file modified
%UAF-I-RDBNOMODS, no modifications made to rights database
```

i   For further assistance with set-up of a TCP/IP service, log a Remedy ticket so that the appropriate HSITES infrastructure support team can assist you.

> If you have implemented enhanced Caché Cube security as described in AXP
> INFO #27, *Enhanced Caché Cube and DCL Access Security*, you will need to
> grant the new VLINK account access to Caché from the VMS command prompt.
> Information about enhanced Caché Cube security and instructions for granting
> access are described in AXP INFO #27 which can be found on the
> HealtheSystems Technical Support Team (HSTS) web page at:
> http://vaww.va.gov/custsvc/cssupp/axp/default.asp. The information can also be
> obtained from FORUM in the SHARED MAIL basket labeled AXP INFO
> MESSAGES.

2.4.4 Listener Management for Caché/NT Systems

See Appendix A, "Listener Management for Caché NT," in the *VistALink 1.5 System
Management Guide*.

2.4.5 Listener Management for DSM/VMS Systems

See Appendix B, "Listener Management for DSM/VMS Systems," in the *VistALink 1.5
System Management Guide.*

2.5 Verifying Listener Connectivity

The general process for testing the listener is as follows:

1. Ping the server
2. Confirm the Listener type via Telnet
3. Test connectivity with the VistALink J2SE SwingTester sample application

2.5.1 Ping the Server

To detect and avoid network problems, try the following:

1. Make sure you can reach the VistA/M server you are trying to connect to through TCP.

2. At the DOS/Command prompt type "PING nnn.nnn.nnn.nnn" for the VistA/M server to which you are trying to connect (where nnn.nnn.nnn.nnn equals the IP address of the server). For example:

```
C:\> PING 127.0.0.1 <RET>
```

> **i** PING is a way to test connectivity. It sends an Internet Control Message Protocol (ICMP) packet to the server in question and requests a response. It verifies that the server is running and the network is properly configured.

> **i** If the VistA/M server is unreachable, there is a network problem, and you should consult with your network administrator.

2.5.2 Connect to Listener via Telnet

1. 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 <RET>
```

2. When you connect, press <RET>. 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.foundations.vistalink.system.fault"
version="1.5" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"xsi:noNamespaceSchemaLocation="vlFault.xsd"><Fault><FaultCo
de>
Server</FaultCode><FaultString>System
Error</FaultString><FaultActor></FaultActor><Detail><Error
type="system" code="181001" ><Message><![CDATA[A system error
occurred in M:
<SUBSCRIPT>SETMSG+5^XOBVRH]]></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 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 <RETURN>, check the type of listener that is running in the port. (It may be a Broker, HL7, or other type of listener.)

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

	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.

2.5.3 Test Listener with SwingTester J2SE Sample Application (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 J2SE (client/server) sample application is supplied in the **vljSamples_1.5.0.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. Or you can wait to test the entire setup with J2EE sample apps at the conclusion of the app server installation. (See "Testing the Sample Application with Your Own M Server.")

2.6 Post-Install: Configuring Connector Proxy User(s) for J2EE Access

Follow this step only if you are setting up VistALink on your VistA/M system for immediate access by one or more specific J2EE servers.

2.6.1 Security Caution

	By setting up connector proxy users, you are granting access on your VistA/M server to execute a <u>wide variety of RPCs</u> on your system. Therefore you need to do the following:

	• Create connector proxy users only for J2EE systems needing access to your M system. • Give the access/verify codes (credentials) of the connector proxy users to approved server administrators only. • Create a different connector proxy user (with different access/verify code credentials) for each J2EE cluster (or data center) that will be connecting to your VistA/M system. • Prevent dissemination of the access/verify codes for a connector proxy user outside of secure communication channels.

2.6.2 Connector Proxy Overview

To allow VistALink access from a specific J2EE system (app server), you need an M Kernel "connector proxy user" account. Each connector deployed on the app server uses this account to establish initial authentication and a trusted connection. Creating this account is not part of the M-side VistALink installation per se, but needs to be performed in M before app server installation can be completed.

A connector proxy account represents a specific application server (not an end-user). A VistALink adapter logs on to the VistA/M server using the assigned Kernel connector proxy user account, authenticating with an access/verify code pair. The connector proxy user account is used by the VistALink connection pool each time it creates a new connection to your VistA/M system.

2.6.3 Creating the Connector Proxy User Kernel Account

The Vista/M system manager is the only one who can grant access to incoming VistALink connections from a J2EE system. Using the Foundations Management menu, the system manager must create a distinct Kernel account to allow VistALink access from any specific J2EE system. The Kernel account must be a "connector proxy user" account, which is created using the CONT^XUSAP entry point (provided by Kernel as part of patch XU*8.0*361).

The VistA/M system manager should do the following:

- Create a Kernel "connector proxy" user account for each distinct J2EE system connecting to the M server through VistALink.
- Securely communicate the access code, verify code, and listener IP address and port to (each) J2EE system manager configuring an adapter to access the VistA/M system.

ℹ	The Kernel XUMGR key is required to create the connector proxy user account.

To create a "Connector Proxy User" account for a J2EE resource adapter, or "connector" user, follow these steps:

1. You must hold the Kernel XUMGR key.

2. Add a new connector proxy user by using the **Foundations Management Menu [XOBU SITE SETUP MENU]** on your VistA/M system, and choosing the **Enter/Edit Connector Proxy User** action.

    The account requires no more information than what is prompted for by the option.

3. Leave the connector proxy user's **Primary Menu** empty.

4. Securely communicate the access code and verify code you enter for the connector proxy user (in addition to the IP and port of your VistALink listener) to the J2EE system manager setting up access from J2EE to your system.

i	You should observe the following points when creating or editing connector proxy users: • Do not enter divisions for a connector proxy user • Do not enter a primary menu • Do not also use the connector proxy user as a test "end-user" • Utilize the user *only* as a connector proxy user

i	To generate a list of existing proxy users on the VistA/M system, use the **Operations Management…|User Management Menu…|Proxy User List [XUSAP PROXY LIST]** option.

i	The Office of Cyber and Information Security (OCIS) has provided draft guidance documents on the required means and process for securely communicating the connector proxy credentials to the J2EE application server administrator. Three draft documents are available for review: *Briefing Note*, *Memorandum*, and *Memorandum of Understanding*. Please contact Ms. Gail Belles for the status of these documents and for directions to obtain the official versions.

The steps to create a connector proxy user account are detailed in the following example. Here, the site system manager (or designee) is creating a connector proxy user for the Falling Waters data center. Note that you can name the connector proxy user anything you wish.
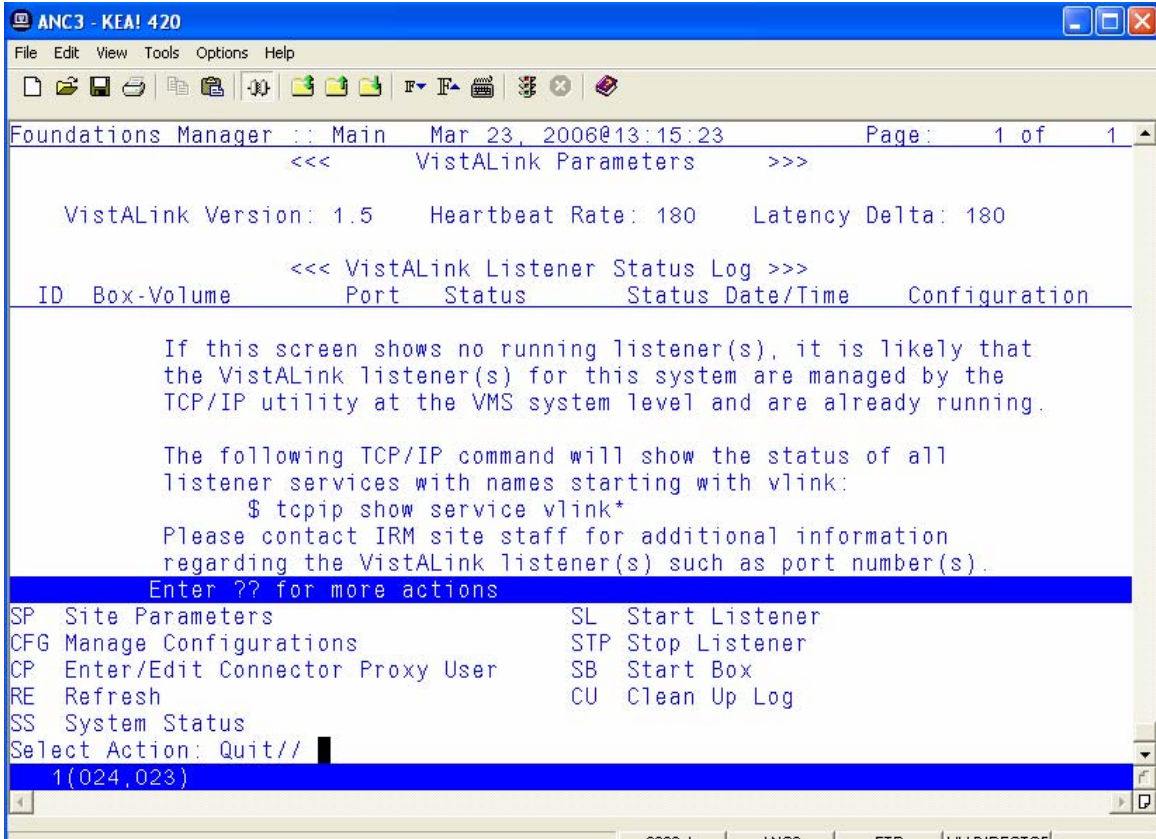
Figure 3. Creating a Connector Proxy User Account

```
SP  Site Parameters                      SL  Start Listener
CFG Manage Configurations                STP Stop Listener
CP  Enter/Edit Connector Proxy User      SB  Start Box
RE  Refresh                              CU  Clean Up Log
SS  System Status
Select Action: Quit// CP   Enter/Edit Connector Proxy User
Enter NPF CONNECTOR PROXY name : CONNECTOR,FALLING WATERS
  Are you adding 'CONNECTOR,FALLING WATERS' as
    a new NEW PERSON (the 14227TH)? No// Y  (Yes)
Checking SOUNDEX for matches.
     CONNECTOR,TEST PROXY
     CONNECTOR,AAC CHDR
     CONNECTOR,HINES EMC
Do you still want to add this entry: NO//Y
Want to edit ACCESS CODE (Y/N): Y
Enter a new ACCESS CODE <Hidden>: ***********
Please re-type the new code to show that I have it right: ***********
OK, Access code has been changed!
The VERIFY CODE has been deleted as a security measure.
The user will have to enter a new one the next time they sign-on.


Want to edit VERIFY CODE (Y/N): Y


Enter a new VERIFY CODE: **********
```

```
Please re-type the new code to show that I have it right: **********

OK, Verify code has been changed!
```

To list existing proxy user accounts (connector and application proxy users) on the
VistA/M system, use the **Proxy User List** option, [XUSAP PROXY LIST].

```
   FIND   Find a user
          List users
          Print Sign-on Log
          Proxy User List
          Release user
          User Inquiry
          User Status Report

Select User Management Menu Option: PROXY User List
DEVICE:   IP network
PROXY USER LIST                                    MAR 23,2006  13:28
PAGE 1
NAME                                  User Class               ISPRIMARY
------------------------------------------------------------------------
XOBVTESTER,APPLICATION PROXY          APPLICATION PROXY
CONNECTOR,AAC CHDR                     CONNECTOR PROXY               Yes
CONNECTOR,HINES EMC                    CONNECTOR PROXY               Yes
CONNECTOR,FALLING WATERS               CONNECTOR PROXY               Yes
```

⚠	The connector proxy names shown above are examples to illustrate the Proxy User List option. You may choose to name the connector proxy user account(s) differently. Sites do not create and should not modify the application proxy user account names.

2.6.4 VistA/M Server Installation Summary

This completes the VistALink 1.5 VistA/M system installation activities. You have
successfully:

- Installed the VistALink 1.5 KIDS build
- Created/modified the VLINK TCP/IP service VMS user account and command files
- Confirmed the new VLINK service is enabled
- Created a connector proxy user account (if necessary)

3 BEA WebLogic Application Server Installation Procedures

3.1 Overview

The figure below summarizes the flow of steps for installing and testing VistALink 1.5 on a J2EE application server. The numbers refer to the section where the step is described in detail.
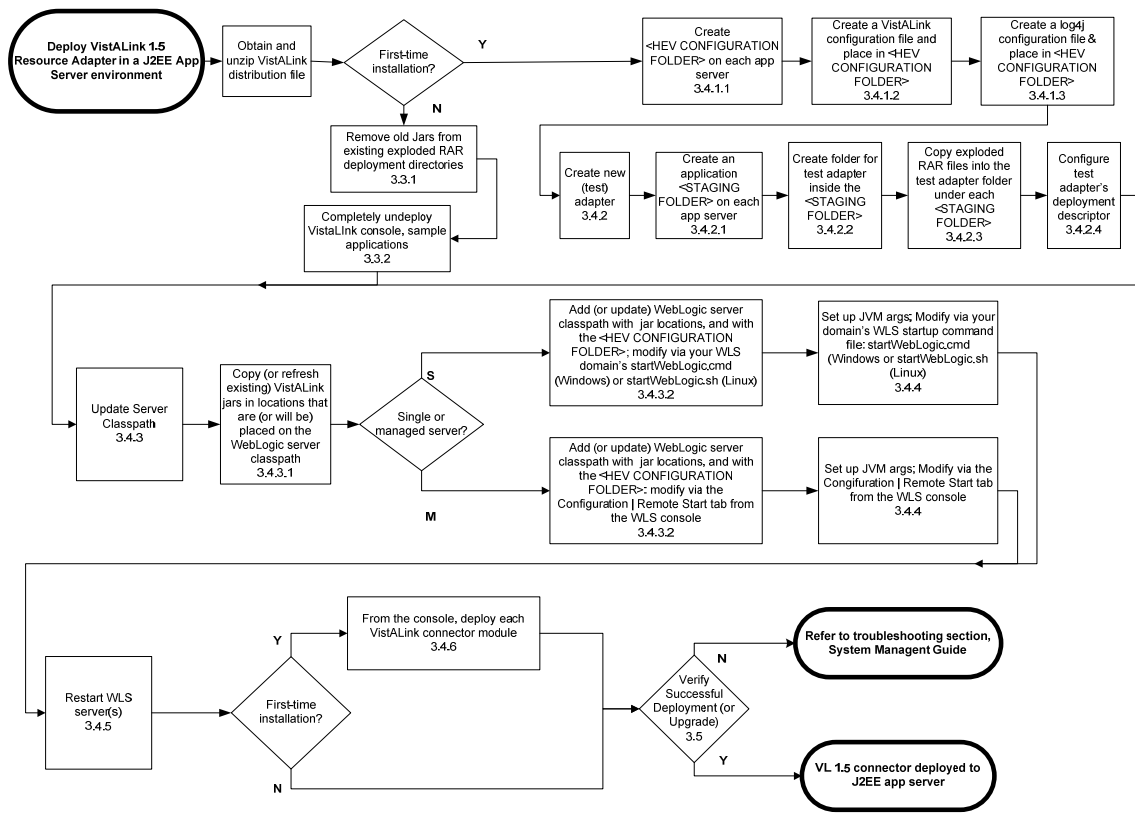
Figure 4. Flowchart for VistALink Application Server Installation

3.2 Preparation

3.2.1 System Requirements

VistALink is supported only on WebLogic at the current time. The following are requirements for installation:

- BEA WebLogic Server (WLS) 8.1, service pack 4, is up and running

- The platform operating system for WebLogic Server is either Windows or Linux

- The WLS deployer should have Java, J2EE, and WebLogic administration experience.

3.2.2 J2CA Deployment Descriptor Overview

The structure of a resource adapter and its runtime behavior are defined in deployment descriptors. The deployment descriptors are created by programmers during the packaging process and become part of the application deployment when the application is compiled.

Resource adapters have two deployment descriptors that affect configuration of the adapter targeted at WebLogic servers. Both files are located in the **META-INF** directory for each VistALink RAR (packaged adapter):

- **ra.xml**: The standard J2EE deployment descriptor for J2EE resource adapters (connectors) such as VistALink. This file describes VistALink's connector-related attributes and its deployment properties using a standard Document Type Definition (DTD) from Sun.
- **weblogic-ra.xml**: Contains WebLogic-specific extended configuration information.

There are various tools available for editing these files. For example, you can use:

- *WebLogic Builder* application (packaged RARs). This tool allows you to edit the deployment descriptor files inside a packaged RAR without needing to un-jar and re-jar the RAR.

- The WLS console configuration tabs to view and modify a subset of the deployment descriptor elements (exploded RAR deployments only. Some of the descriptor element changes take effect dynamically at run-time without redeploying the resource adapter. Other descriptor elements will require redeployment.

- An XML editor such as XMLSpy (exploded RAR only)

- A text editor such as Notepad (exploded RAR only).

3.2.3 Overview of Base and Linked Adapters

Version 8.1 of WebLogic introduced a "link-ref" mechanism, enabling resources of a single "base" adapter to be shared by one or more "linked" adapters. The base adapter is merely a completely set up standalone adapter. Its resources (classes, jars, etc.), however, can be linked to and reused by other resource adapters (linked adapters). Each linked adapter needs only a subset of files and deployment descriptor settings.

When setting up multiple VistALink adapters, for connections to multiple VistA/M systems, we recommend setting up one adapter as a base adapter, and any additional adapters as linked adapters. You must always have at least one base adapter set up to

some VistA/M system. Each linked adapter refers back to the base adapter via the **weblogic-ra.xml** "`<ra-link-ref>`" property.

For more information related to configuring base and linked adapters, see "Adapter Configuration," in the *VistALink 1.5 System Management Guide*.

3.2.4 Obtain 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.

3.2.5 Obtain the VistALink Distribution File

You can obtain the VistALink distribution zip file from any of the **anonymous.software** directories on the OIFO 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.

3.3 *Upgrading a Previous Installation*

3.3.1 Remove Jars in Exploded RAR Directories

To clean up existing adapters: remove or delete all jars from the exploded RAR directory of each existing adapter.

3.3.2 Undeploy VistALink Console and Sample Applications

If you have an existing VistALink installation, we are currently recommending that you undeploy previous versions of the VistALink console and sample applications. Follow the steps below:

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

        mydomain>Deployments>Web Application Modules

    and clicking on the trashcan icon (🗑).

2. If you have deployed the VistALink sample web applications, delete them from the WebLogic configuration by navigating to

```
mydomain>Deployments>Applications
```

and clicking on the trashcan icon (🗑).

3.4 Installing the VistALink 1.5 Adapter(s)

The resource adapter is the central piece of the J2CA 1.0 Connector Architecture. It serves as the connector between the Java client application and a VistA/M system. Each VistALink resource adapter deployed in a J2EE application server environment allows Health*e*Vet applications to access a specific VistA/M system.

The next few steps are for first-time installations only. If upgrading existing adapters, skip ahead to section "Update the WebLogic Server Classpath." These steps assume, for the sake of simplicity, that you are deploying a single VistALink adapter.

	In the case of first time installations, these instructions assume, for the sake of simplicity, that you are deploying a single VistALink adapter.

	The next few steps are for first-time installations only. If upgrading existing adapters, skip ahead to section "Update the WebLogic Server Classpath."

3.4.1 Set up Configuration Files

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

3.4.1.1 Create <HEV Configuration Folder>

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

We recommend using a single folder for configuration files for all HEV applications, including VistALink. If it is not already present, you should create this folder on each separate physical WebLogic server.

1. Create a folder 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.

3.4.1.2 Create VistALink Configuration File

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

VistALink makes use of its own configuration file to load VistALink-specific connector settings. You will need to include one entry for each VistALink adapter. The rules for this file are as follows:

- It must be named "gov.va.med.vistalink.connectorConfig.xml"

- It must be placed in a folder on the Java classpath of the Java Virtual machine (JVM) of each WebLogic server instance on which you are deploying VistALink adapters.

 The following are also recommended:

- Using this folder to hold configuration files for all Health*e*Vet-VistA applications

- Creating this folder for each physical server

- Ensuring that this folder is secure and protected. The **gov.va.med.vistalink.connectorConfig.xml** file 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.

To create the VistALink configuration file:

1. Locate the example configuration file provided in the VistALink distribution zip file:

   ```
   <DIST FOLDER>/RAR/configExamples/gov.va.med.vistalink.connectorConfig.xml
   ```

   This example configuration file contains a single entry, identified by the **jndiName** attribute **vlj/testconnector**. This entry is pre-configured to connect to a VistALink demo VistA/M server in Albany, NY, that runs the latest VistALink M listener.

2. Copy the provided configuration file into the <HEV CONFIGURATION FOLDER> on each physical server that will be running VistALink adapters. You may want to include the administration server as well.

3. Later in this installation procedure you will add the <HEV CONFIGURATION FOLDER> folder to the Java classpath of your WebLogic server JVM(s).

For additional information on setting up a connector configuration file, see the section "VistALink Connector Configuration File," in the *VistALink 1.5 System Management Guide*.

3.4.1.3 Create log4j Configuration File

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

To turn on logging for VistALink, you need to set up a log4j configuration file for each WebLogic server running VistALink. You also need to pass the name and location to each JVM by one of the following methods:

- Name the file "log4j.xml" and place it in the <HEV CONFIGURATION FOLDER>, on your server classpath. (This is probably the easier approach.)

- Give the file a name of your choice, place it anywhere on your physical file system for each physical server, and pass the location of the file to the JVM at WebLogic startup via the **-Dlog4j.configuration** JVM argument. (See the section below, "Set JVM Arguments," for more information on JVM arguments.)

You may already have a **log4j** configuration file active for your server, possibly containing loggers and appenders for applications other than VistALink. If so, you may want to add to the existing file the logger entries for VistALink. (For HEV configurations, a single JVM-wide log4j configuration is expected to be used.)

If you need a log4j configuration file and do not already have one set up, follow these steps:

1. Copy one of the sample log4j configuration files provided in the VistALink distribution zip file.

   These sample files may be located in the **<DIST FOLDER>/log4j/configExamples** folder. They are named:

   - **log4jVLJConfig.xml** (minimal VistALink logging configuration)

   - **log4jVLJConfigDebug.xml** (debug-level VistALink logging configuration).

     Note: Turning on the "debug" level can adversely affect system performance.

2. Name the file "log4j.xml" and place it in your <HEV CONFIGURATION FOLDER>.

3.4.2 Create VistALink Adapter(s)

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

3.4.2.1   Create an Application Staging Folder

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

A folder is needed to hold the EARs, WARs, and adapter RAR folders that you create prior to deployment. The instructions and examples in this chapter refer to this folder as the "staging folder." The name "/bea-stage" is suggested for this folder, though you can name it something else. You can use the same staging folder for other application deployments if you wish (not just VistALink).

1. If you don't already have a staging folder, create one on each separate physical server.

   In the instructions in the rest of this document, this folder will be referred to as the <APPLICATION STAGING FOLDER>.

3.4.2.2   Create an Adapter Folder

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

You must create a folder under your <APPLICATION STAGING FOLDER> for each adapter you are deploying (e.g., "\bea-stage\vljSalem658", "\bea-stage\vljBoston523", etc.). 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 each adapter mnemonically to the administrators viewing them in the WebLogic console.

1. Create a single folder in the <APPLICATION STAGING FOLDER> (e.g., "\bea-stage\testConnector"). Use a folder name that will readily identify the adapter.

3.4.2.3   Copy the VistALink RAR Adapter

(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

For each adapter you are deploying, copy the RAR files from the <DIST FOLDER>RAR folder to the new adapter staging folder(s).

The adapter is provided in the <DIST FOLDER>RAR folder of the VistALink distribution zip file in two formats:

- **Exploded RAR:** the contents of the <DIST FOLDER>RAR /ExplodedVistaLinkRAR folder

- **Packaged RAR:** one file, <DIST FOLDER>RAR /vljConnector-1.5.0.rar

> **i** At the current time, we recommend deploying the exploded form of the adapter, to allow for easier editing of deployment descriptors.

1. Copy the entire file structure from inside the **<DIST FOLDER>RAR /ExplodedVistaLinkRAR** folder to your single adapter folder (e.g., "bea-stage/testConnector"). The contents of your adapter folder will then be preconfigured to create a "base" adapter configured to connect to the VistALink demo server in Albany.

3.4.2.4 Edit the Adapter Deployment Descriptor Files
(**For first-time installations only. If upgrading adapters, skip to the section** "Update the WebLogic Server Classpath.")

To configure the adapter, edit the **ra.xml** and **weblogic-ra.xml** files in each adapter folder. These files are in the META-INF subfolder of each adapter folder.

If you are deploying adapters for the first time, we recommend leaving the settings in **weblogic-ra.xml** as they are, and making sure that **ra.xml** is configured to first connect to the VistALink demo server in Albany.

1. Verify that the **connectorJndiName** config-property is set to "vlj/testconnector" in the **ra.xml** file. This way, on deployment, the adapter will retrieve the pre-configured entry of the same name from the distributed version of **gov.va.med.vistalink.connectorConfig.xml**.

```
<config-property>
  <config-property-name>connectorJndiName</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>vlj/testconnector</config-property-value>
</config-property>
```

2. Proceed to the section "Update the WebLogic Server Classpath."

3.4.3 Update the WebLogic Server Classpath
You now need to add or update several libraries on the server classpath for each server you are deploying adapters to. You may want to create a separate folder in the <APPLICATION STAGING FOLDER> to hold these library jar files (e.g., **bea-stage\ClasspathLibraries**).

You must also set several properties when launching any JVMs for WebLogic servers that VistALink adapters are deployed to.

3.4.3.1 Copy Jars to the WebLogic Server Classpath Locations

For both upgrades and new installations, the following jar files need to be copied from the VistALink distribution zip file to locations on the WebLogic server classpath:

- vljConnector-1.5.0.nnn.jar
- vljFoundationsLib-1.5.0.nnn.jar
- jaxen-core.jar
- jaxen-dom.jar
- log4j-1.2.8.jar
- saxpath.jar
- xbean.jar

If upgrading a previous installation, delete or archive any older versions of these files before copying the new ones. This set of jars is provided inside the **rar\ExplodedVistaLinkRAR** directory of the distribution zip file, in the root of that directory (**vlj*.jar**) and in the **lib** subdirectory of the remaining jars. Place them in a location that will be easy to add to the server classpath, possibly as a single directory.

3.4.3.1.1 Single Server (One-Server Domain)

Copy the jar files to (preferably) a single directory on the single server.

3.4.3.1.2 Managed Server (Multi-Server Domain)

Copy the jar files to (preferably) a single directory on each target server.

3.4.3.1.3 Admin Server (Multi-Server Domain)

Ordinarily, the jar files do not need to be placed on the admin server. (In a multi-server environment, VistALink connectors will usually not be run on an admin server.)

3.4.3.2 Update the Server Classpath.

Seven jar files and one directory need to be added to the server classpath:

- vljConnector-1.5.0.nnn.jar
- vljFoundationsLib-1.5.0.nnn.jar
- jaxen-core.jar
- jaxen-dom.jar
- log4j-1.2.8.jar
- saxpath.jar
- xbean.jar
- the <HEV CONFIGURATION FOLDER>, containing your **gov.va.med.vistalink.connectorConfig.xml** and (possibly) the **log4j.xml** files you created earlier in this installation.

For upgrades to existing installations, the classpath needs to be updated to reflect the new jar versions.

The method to update the server classpath depends on how you start your listener. The sections below explain how to modify the classpath for the following configurations:

- Single server (one-server domain)
- Managed server (multi-server domain)
- Administration server (multi-server domain)

3.4.3.2.1 Single Server (One-Server Domain)

You must edit the WebLogic domain's **startWebLogic.cmd** (Windows) or **startWebLogic.sh** script (Linux), used to start WebLogic for your domain. Add or update the jar file names you copied in the previous step and their directory to the classpath that is passed to the JVM that starts up the application server.

1. After the line "set "JAVA_VENDOR=Sun" in the **startWebLogic.cmd** script, set up the **VLJ_CP** variable to match the following example:

```
------------------<snip>--------------------
set JAVA_VENDOR=Sun
@rem setup VistALink classpath variable VLJ_CP
@rem you need to set VLJ_STAGE for your configuration
set VLJ_STAGE=c:\bea\bea-stage\ClasspathLibraries
set VLJ_CP=%VLJ_STAGE%\vljConnector-1.5.0.nnn.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\vljFoundationsLib-1.5.0.nnn.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\jaxen-core.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\jaxen-dom.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\log4j-1.2.8.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\saxpath.jar
set VLJ_CP=%VLJ_CP%;%VLJ_STAGE%\xbean.jar
set VLJ_CP=%VLJ_CP%;c:\myCommonConfigFolder
------------------<snip>--------------------
```

The name of the VistALink version must be exact in lines 5 and 6 in the example above (shown in bold). If the <APPLICATION STAGING FOLDER> name in line 4 and the <HEV CONFIGURATION FILE> folder name in the last line are different than those in the example, they must be changed accordingly.

> ℹ️ If you are using a variable such as VLJ_STAGE in the startup script as a shortcut reference to a single folder holding all your libraries, make sure you update it to the location where you have placed all the libraries in this install.

2. Add VLJ_CP variable to the WebLogic Classpath (**startWebLogic.cmd**). Change:

```
set
CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\
rt.jar;%WL_HOME%\server\lib\webservices.jar;%CLASSPATH%
```

to:

```
set
CLASSPATH=%WEBLOGIC_CLASSPATH%;%VLJ_CP%;%POINTBASE_CLASSPATH%;%JAVA_HOME%
\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar;%CLASSPATH
```

Linux startWebLogic.sh Example

The **<USER_DOMAIN_HOME>/startWebLogic.sh** file needs to be modified for the classes contained in the VistALink jar files and third party jar files to be found at run-time.

1. Add the following lines after the line "`JAVA_VENDOR=`" line:

```
------------------<snip>--------------------

JAVA_VENDOR="Sun"
#rem setup VistALink classpath variable VLJ_CP
#rem you need to set VLJ_STAGE for your configuration
VLJ_STAGE="/opt/bea-stage"
VLJ_CP="${VLJ_STAGE}/vljConnector-1.5.0.nnn.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/vljFoundationsLib-1.5.0.nnn.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/jaxen-core.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/jaxen-dom.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/log4j-1.2.8.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/saxpath.jar"
VLJ_CP="${VLJ_CP}:${VLJ_STAGE}/xbean.jar"
VLJ_CP="${VLJ_CP}:/opt/myCommonConfigFolder"

------------------<snip>--------------------
```

2. Modify the line

```
CLASSPATH="${WEBLOGIC_CLASSPATH}:${POINTBASE_CLASSPATH}:${JAVA_
HOME}/jre/lib/rt.jar:${WL_HOME}/server/lib/webservices.jar:${CL
ASSPATH}"
```

to

```
CLASSPATH="${WEBLOGIC_CLASSPATH}:${VLJ_CP}:${POINTBASE_CLASSPAT
H}:${JAVA_HOME}/jre/lib/rt.jar:${WL_HOME}/server/lib/webservice
s.jar:${CLASSPATH}"
```

3.4.3.2.2 Managed Server (Multi-Server Domain)

Managed servers are started from the WebLogic console. You must modify the **Configuration | Remote Start** "classpath" setting in the WebLogic console for each managed server that will have VistALink adapters deployed to it. Then you must add or update the jar file names at the locations you copied them to (see previous step) and directory to the classpath.

On Windows systems, if you set any value in the Remote Start "classpath," you must specify all the jars needed by WebLogic to start the managed server – not just the VistALink-related jars listed at the beginning of this section.

On Linux systems, you may be able to use the string `${CLASSPATH}` to pick up the existing non-VistALink classpath needed by WebLogic in the Remote Start "classpath," depending on your Node Manager setup. In that case, you would only need to specify `${CLASSPATH}` in addition to the jars needed for VistALink.

"Remote Start" Classpath Example

```
C:\bea\jdk141_05\lib\tools.jar;C:\bea\WEBLOG~1\server\lib\weblogic
_sp.jar;C:\bea\WEBLOG~1\server\lib\weblogic.jar;C:\bea\WEBLOG~1\se
rver\lib\ojdbc14.jar;C:\bea\WEBLOG~1\common\eval\pointbase\lib\pbs
erver44.jar;C:\bea\WEBLOG~1\common\eval\pointbase\lib\pbclient44.j
ar;C:\bea\jdk141_05\jre\lib\rt.jar;C:\bea\WEBLOG~1\server\lib\webs
ervices.jar;c:/bea-stage/ClasspathLibs/jaxen-core.jar;c:/bea-
stage/ClasspathLibs/jaxen-dom.jar;c:/bea-
stage/ClasspathLibs/log4j-1.2.8.jar;c:/bea-
stage/ClasspathLibs/saxpath.jar;c:/bea-
stage/ClasspathLibs/vljConnector-1.5.0.nnn.jar;c:/bea-
stage/ClasspathLibs/vljFoundationsLib-1.5.0.nnn. jar;c:/bea-
stage/ClasspathLibs/xbean.jar;c:/myCommonConfigFolder;
```

<table>
<tr><td>ⓘ</td><td>The values needed vary with the server configuration. One way of obtaining the classpath libraries needed for a WebLogic managed server is to use the startManagedWebLogic startup script to capture the classpath echoed to the console, and then use that classpath to fill in the Remote Start <code>classpath</code> value.</td></tr>
</table>

Linux Example

<table>
<tr><td>🐧</td><td>Note: On Linux systems, you may be able to use the value <code>${CLASSPATH}</code> to include the existing non-VistALink classpath needed by WebLogic, depending on your Node Manager setup. If not, follow the same technique as to obtain the jars needed by WebLogic as you would on a Windows system.

<code>${CLASSPATH}:/u01/app/staged/vl/vljConnector-1.5.0.nnn. jar:
/u01/app/staged/vl/vljFoundationsLib-
1.5.0.nnn.jar:/u01/app/staged/vl/jaxen-dom.jar:
/u01/app/staged/vl/jaxen-core.jar:/u01/app/staged/vl/log4j-
1.2.8.jar:
/u01/app/staged/vl/saxpath.jar:/u01/app/staged/vl/xbean.jar:
/opt/myCommonConfigFolder:</code></td></tr>
</table>

3.4.3.2.3 <u>Admin Server (Multi-Server Domain)</u>

In order to make it easy to use the VistALink Configuration Editor, you should consider placing a folder containing the VistALink configuration file on the classpath of the admin server, in multi-server domain.

In a production setting, VistALink adapters will probably not be deployed on admin servers. Therefore, there is no reason to put VistALink (and supporting) libraries on the server classpath.

Likewise, there is no requirement to put a folder containing the VistALink configuration file on the admin server's classpath. However, doing so makes it easy to edit the configuration file using the Configuration Editor. If such a folder is on the admin server classpath, the Configuration Editor can load the VistALink configuration file without prompting and save it on the admin server.

Note: After editing on the admin server, you can propagate/copy the configuration file out to the managed servers.

The Configuration Editor is deployed as part of the VistALink console, which runs on the admin server. For more information, see the section "Configuration Editor" in the "VistALink Console" section of the *VistALink 1.5 System Management Guide*.

3.4.4 Update WebLogic Server JVM Arguments

(**For first-time installations only. If upgrading adapters, skip this step, or simply verify that the JVM arguments are set.**)

The following JVM system properties are used to store environment information that is used in VistALink and made available to other applications through the **gov.va.med.environment.Environment** API:

- `-Dgov.va.med.environment.servertype=` (weblogic | websphere | jboss | oracle. Defaults to unknown if not present).

  If you are setting up a WebLogic server, for example, set `servertype` to "weblogic".

- `-Dgov.va.med.environment.production=` (true | false. Defaults to "false" if not present).

  This setting marks a J2EE system as being a "production" or "test" system, and will be used by VistALink in the future to prevent a test J2EE system from connecting to a production M system, and vice versa.

The following JVM property is used for log4j configuration:

`-Dlog4j.configuration=` (full path/filename of a log4j configuration file).

For example:

```
-Dlog4j.configuration=file:/c:/bea-stage/myLog4JConfig.xml
```

This log4j JVM argument is required only if your log4j configuration file is both:

- not named "log4j.xml"
- not placed in a folder on the server classpath (e.g., not in the <HEV CONFIGURATION FOLDER>).

However, it is recommended that you do name the log4j config file "log4j.xml" and place it in a folder on the server classpath. Then you will not need the log4j JVM argument.

You must set all of these properties listed above when launching any JVMs for WebLogic servers on which VistALink adapters are going to be installed. Depending on your WebLogic domain configuration, the set of servers may include managed servers, admin servers, or both.

3.4.4.1 <u>Single Server (One-Server Domain)</u>

1. In the **startup cmd** files generated by the WebLogic configuration wizard, use the **JAVA_OPTIONS** variable to set these JVM arguments. For example:

```
@rem setup the VLJ-specific Java command-line options for running the server
set JAVA_OPTIONS=-Dgov.va.med.environment.servertype=weblogic
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dgov.va.med.environment.production=false
```

3.4.4.2 <u>Managed Server (Multi-Server Domain)</u>

1. If you launch a given WebLogic server from a command file, modify the command file to pass the JVM argument. If you launch a server from the WebLogic 8.1 console, use the **Remote Start** tab of the server configuration to specify these arguments. For example:

```
-Dgov.va.med.environment.servertype=weblogic
-Dgov.va.med.environment.production=false
```

⚠	If you have not already performed the KIDS install on the Vista/M server, you cannot establish a connection as described in the following steps.

3.4.5 Stop/Restart WebLogic Server(s)

Restart your WebLogic server(s) to activate the new classpath settings in the running JVM(s).

3.4.6 Deploy Adapter(s)

(For first-time installations only. If upgrading adapters, your adapters are already deployed, so you can skip this step.)

Follow the steps below to deploy each of your VistALink adapters.

> **i** The exact prompts may vary on different WebLogic domain configurations, and between different versions of WebLogic server.

1. Under your domain in the navigation tree of the WebLogic console, select "Deployments | Connector Modules."

2. Select "Deploy a New Connector Module."

3. Navigate to your <APPLICATION STAGING FOLDER>.

4. A radio button should appear next to your "exploded" adapter staging folder where you copied the exploded RAR files. Select the radio button and choose "Target Module" or "Deploy." If you are asked to target servers, select the server(s) on which you will deploy the adapter.

5. If prompted for "Source Accessibility," it's recommended to select "Copy this Connector Module onto every target for me."

6. When you press **Deploy**, WebLogic should deploy the adapter.
   Wait for a module status of "Active." When you see this, the adapter successfully deployed (from WebLogic's point of view).

   You should also see under the Connector Modules node a new node for the adapter in the WebLogic console navigation tree.

3.5 Verifying Successful Adapter Installation or Upgrade

There are several areas to check to verify that an adapter installation or upgrade is successful:

1. Select the node for the new adapter in the WebLogic console navigation tree. Then look on the **Monitoring** tab of the adapter for the number of connections listed. This number should match the Initial Capacity set in the adapter's **weblogic-ra.xml**.

If the initial capacity is non-zero, and the number of connections shown matches, WebLogic was able to create connections to the M systems.

If the numbers do not match (e.g., initial capacity is non-zero but the number of connections is zero), WebLogic may be having difficulty creating connections, most likely due to a configuration or installation issue.

2. Look for the deployed adapter to be displayed in the VistALink console for all server(s) you deployed it to. Check if the console is able to contact the VistA/M server and return VistALink M/VistA Server Information for the adapter. This is usually a good indicator of a successful deployment. See the section below, "Deploying the VistALink Console."

3. (Optional) If your adapter is configured to connect to your M system (as opposed to the Foundations VistA/M server at the Albany OI Field Office), and if your adapter's initial capacity is non-zero, look for XOBVSKT jobs on your VistA/M system.

For each XOBVSKT job, check that each IO ("IP") variable matches your WLS IP address. (In Caché, use the Caché control panel to choose the Detail view to get this information.)

4. (Optional) You can exercise the adapter by using it with the VistALink sample J2EE application. See the steps in the below, "Deploying the J2EE Sample Application."

3.6 Deploying the VistALink Console

The VistALink console is an optional tool for managing VistALink adapters. Currently, this console extends the WebLogic console. It is provided in the console folder of the VistALink distribution zip file in two forms:

- A packaged WAR:

      `<DIST FOLDER>/console/VistaLinkConsole-1.5.0.nnn.war`

- An exploded WAR folder:

      `<DIST FOLDER>/console/exploded/VistaLinkConsole-1.5.0.nnn.war`

The VistALink console can be deployed in either packaged or exploded format – we do not recommend one format over the other at this time. The console should be deployed only on admin servers. If you are using a one-server domain, deploy it on your single server.

The figure below shows the flow of steps for deploying the VistALink console:
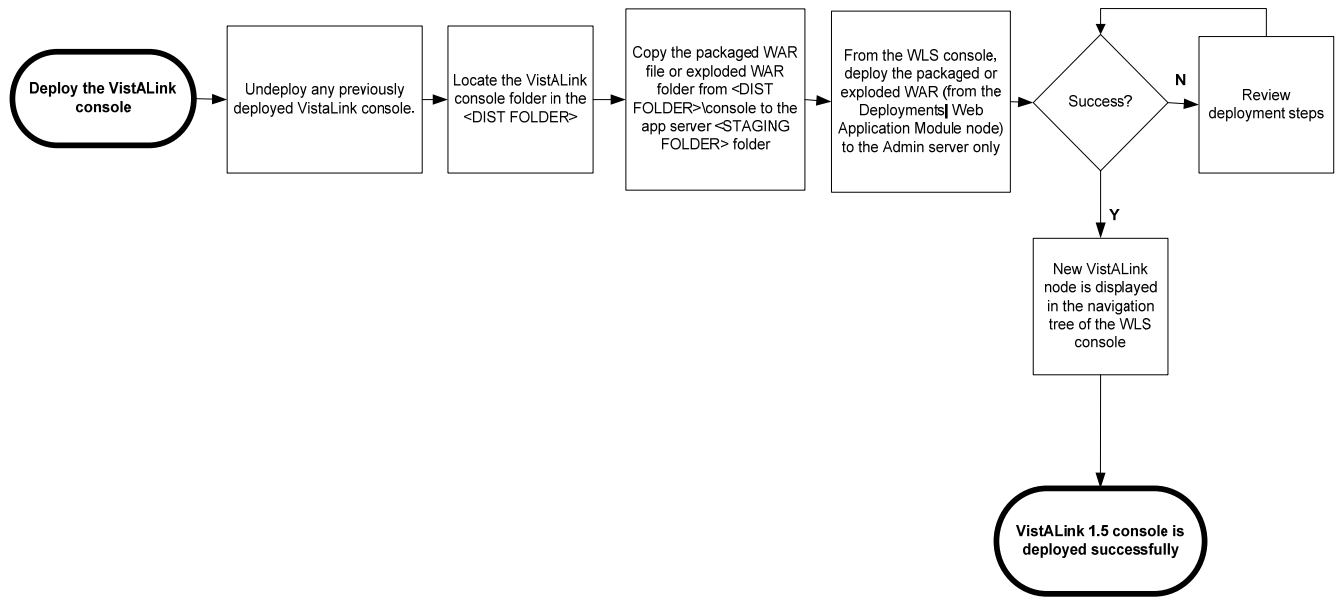
Figure 5. Flowchart for VistALink Console Deployment on WLS 8.1

To deploy the VistALink console:

1. You must completely undeploy any previous version of the VistALink console first.

2. Copy either the packaged WAR or the exploded WAR folder from **<DIST FOLDER>/console** to your **<APPLICATION STAGING FOLDER>**.

3. Using the WebLogic console, deploy the packaged or exploded WAR (via **<domain name> | Deployments | Web Application Module** node) :

   - Navigate to where you copied the packaged or exploded WAR (e.g., <APPLICATION STAGING FOLDER>)
   - Select the packaged or exploded WAR file for the VistALink console
   - Target your admin (or single) server.

4. If successful, the navigation tree on the left-hand side of your WebLogic console should, after a few seconds, display a new node named "VistALink" at the bottom.
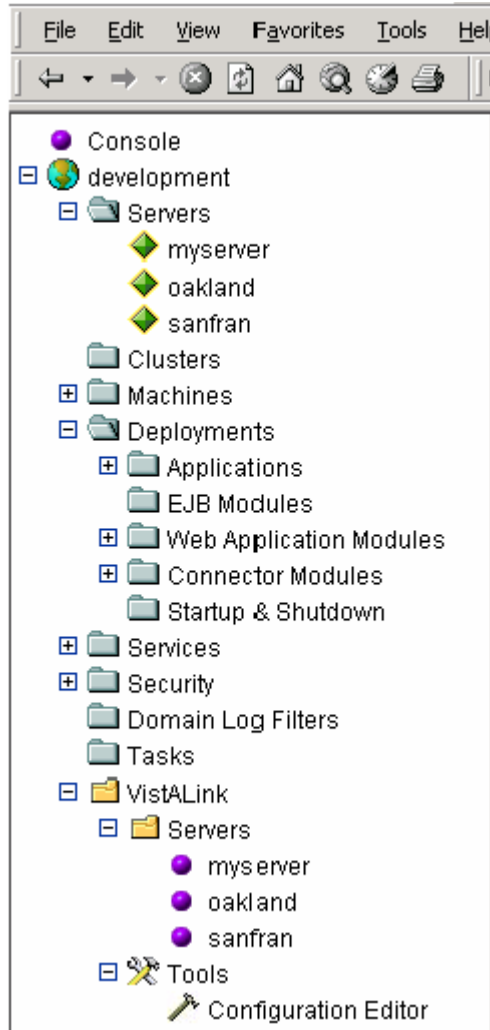
Figure 6. VistALink Console in the WebLogic Console

3.6.1 Multi-Server Domains

For multi-server domains, you may want to put the folder containing the VistALink configuration file on the admin server's classpath – even if adapters are not deployed on the admin server. This makes it easy for the Configuration Editor to edit the admin server's copy of the file. The Configuration Editor can load the VistALink configuration file without prompting, and save it on the admin server. Then you can propagate the changed file out to the other managed servers.

> **i** The Configuration Editor is deployed as part of the VistALink console, which runs on the admin server.

3.7   Deploying the Sample J2EE Application

A sample J2EE application is provided to demonstrate the use of VistALink in a J2EE environment. The sample application is also a way to test your basic adapter setup. The figure below shows the flow of steps for deploying the sample application:
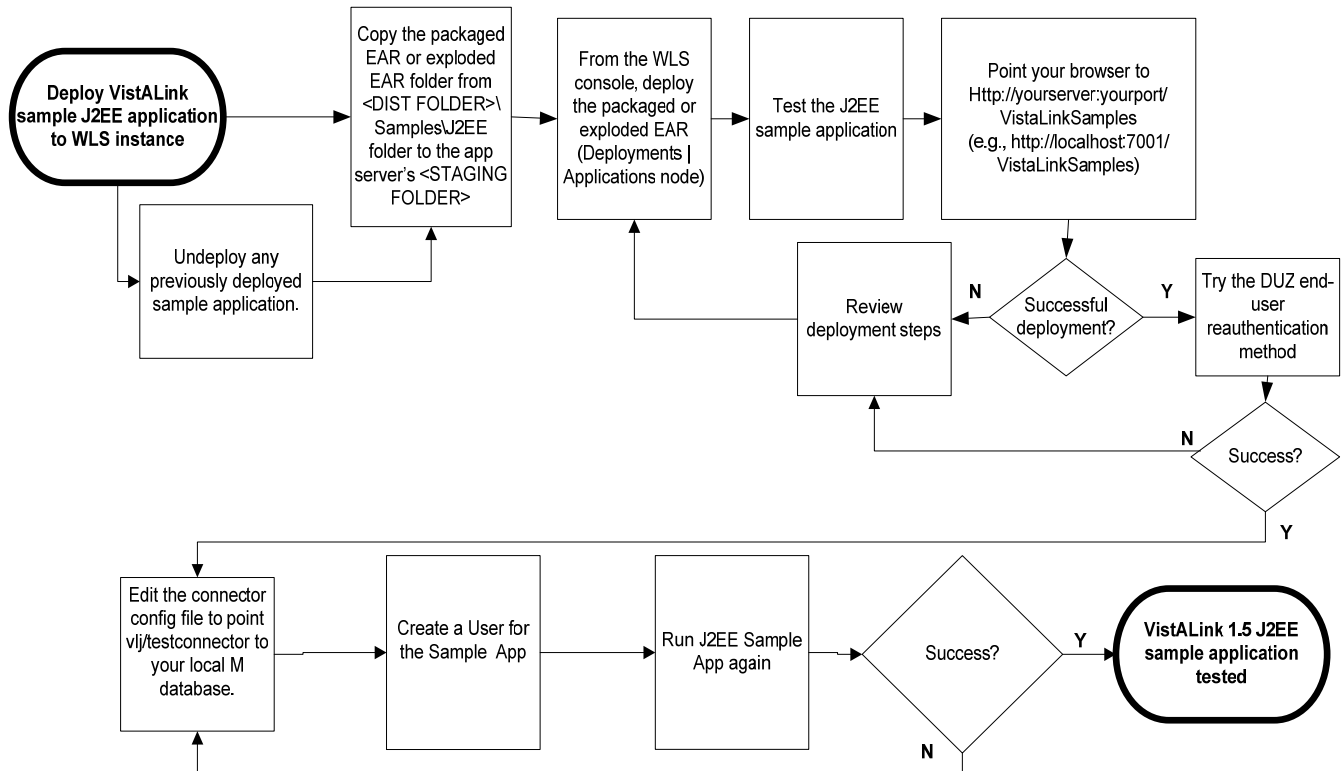
Figure 7. Flowchart for Sample Application Deployment

You can find the sample application as part of the VistALink distribution zip file, in the **<DIST FOLDER>/samples/J2EE folder**. Both packaged and exploded EAR formats are provided.

The sample J2EE application is configured to use the VistALink adapter with the JNDI lookup name of "vlj/testconnector." The `vlj/testconnector` should be deployed and operational on the same server that the sample J2EE application is installed on. It can be pointed to any VistA/M system. A default configuration is provided that points this connector to the VistALink demo VistA/M server at the Albany OI Field Office.

To deploy the sample J2EE application:

1. If a previous version of the VistALink sample application is deployed, undeploy it completely.

2. Copy either the packaged EAR file or the exploded EAR folder for the sample application from the **<DIST FOLDER>/samples/J2EE** folder to the **<APPLICATION STAGING FOLDER>.**

3. Using the WebLogic console to deploy either the packaged or exploded EAR (via the `<domain name> | Deployments | Applications` node):

   a. Navigate to where you copied the packaged and exploded EAR, e.g., the <APPLICATION STAGING FOLDER>

   b. Select either the packaged or exploded EAR

   c. If using managed server configuration, select as the target server the server where the `vlj/testconnector` is deployed

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 8. VistALink Sample Application

> **ⓘ** Loading the top page of the application means only that the VistALink sample application has been deployed to the app server. At this point it does not mean that it has used VistALink to connect to an VistA/M system's VistALink listener.

3. Choose a re-authentication method. The choices are VPID, Application Proxy, DUZ and CCOW. At this time, the CCOW option is experimental, so choose one of the other methods.

4. You are now asked to supply end-user re-authentication identification (DUZ or VPID and division) and to specify the connector to use. If your `vlj/testconnector` connector is pointing at the VistALinkDemo system, you can accept the defaults.

   If this connector is pointing at your own M system, you need to supply the DUZ or VPID of a valid user on your system. This user must hold the [XOBV VISTALINK TESTER] "B"-type option.

Please enter end-user re-authentication identification
(this is *different* than the connector login credentials specified in ra.xml)

VPID:	joe_test_0001
DUZ(2) division*:	11000
Connector to use:	⦿ Use "eis/vlj/testconnector" <resource-ref> ○ Use institution mapping to lookup connector for specified division ○ Select from deployed connectors: – select –
	Submit

* Every reauthentication must explicitly specify the user division. As all Health*e*Vet VistA applications should be multi-divisional-aware, all such applications track what division a user is logged on under. Pass this value as the external station number of the division, e.g., "523AZ". This will be set into DUZ(2), and corresponds to STATION NUMBER field (#99) value for the division, as standardized in the VistA Institution file (#4).

Figure 9. Sample Application Re-authentication Page

5. Press **Submit**.

    The sample J2EE application will now attempt to execute a set of remote procedure calls (RPCs) on the connector module retrieved from JNDI under the name "vlj/testconnector," using the end-user re-authentication credentials specified.

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

Report Generated at Thu Sep 15 16:00:06 EDT 2005.

Credentials Used:

	type	value
	vpid:	joe_test_0001
application proxy name:	null	
	duz:	null
access code:	null	
verify code:	null	
	division:	11000
CCOW handle:	null	
connector used (JNDI name):	eis/vlj/testconnector	

All RPCs executed successfully?: true

RPC Results:

```
XOBV TEST PING Results:
Ping Successful!

XOBV TEST WORD PROCESSING Results:
==================
upon this continent, a new nation, conceived in liberty,
and dedicated to the proposition that 'all men are
created equal'.

Now we are engaged in a great civil war, testing whether that
nation, or any nation so conceived, and so dedicated, can long
endure. We are met on a great battle field of that war. We come
to dedicate a portion of it, as a final resting place for those
who died here, that the nation might live. This we may, in all
propriety do. But, in a larger sense, we can not dedicate -- we
can not consecrate -- we can not hallow, this ground -- The brave
men, living and dead, who struggled here, have hallowed it, far
above our poor power to add or detract. The world will little
note,nor long remember what we say here; while it can never
forget what they did here."
```

try again

Figure 10. VistALink J2EE Sample Application Results Page

3.8 Testing the Sample Application with Your Own VistA/M Server

If this is a first-time installation, you have so far installed a single VistALink connector on the J2EE system. This connector is configured to use the default VistALink configuration file entry with the "vlj/testconnector" JNDI name. The `vlj/testconnector` entry is configured by default to access the VistALink demo VistA/M system at the Albany OI Field Office.

If you have a VistA/M system that you are ready to connect to (other than this demo system), you can use the VistALink sample application to point to your own VistA/M system and test the successful operation of both sides of your connector (J2EE and M). This is a four-step process:

1. Reconfigure the `vlj/testconnector` adapter to access your own VistA/M listener.

2. Use an existing user or create new one for the sample application.

3. Grant the "B"-type option to XOBVTESTER,APPLICATION PROXY.

4. Run the VistALink sample Web application.

3.8.1 Reconfigure "vlj/testconnector" Adapter to Access Your Own VistA/M Listener

To reconfigure the `vlj/testconnector` adapter to access your own VistA/M listener:

1. Obtain the following information from your VistA/M system administrator:

   - Access code and verify code for the connector proxy user
   - IP address of the listener
   - Port of the listener

2. Locate your VistALink configuration file:

   ```
   <HEV CONFIGURATION FOLDER>\gov.va.med.vistalink.connectorConfig.xml
   ```

3. Under the `vlj/testconnector` JNDI name, update the `access-code, verify-code, ip,` and `port` values to match those of your VistA/M system.

4. Set the `encrypted` value to "false."

5. Using the WebLogic console, either restart your J2EE server or stop and redeploy the connector.

> **i** You can also use set the VistALink Configuration Editor to make the changes in steps 3 and 4.

3.8.2 Create or Use an Existing VistA/M System User for the Sample App

To run the sample Web application (DUZ and VPID re-authentication pages) against your own VistA/M system, you need to either create a Kernel user account or use an existing one. The characteristics required for this user are:

- A known, valid DUZ or VPID identifier for an end-user on your VistA/M system
- An [XOBV VISTALINK TESTER] "B"-type option is assigned to the user
- A valid station number under which that user can log into your VistA/M system

  If the user has one or more divisions specified in their NEW PERSON file (#200) "DIVISION" multiple, a valid station number must be the station number for one of these divisions. Otherwise, the valid division for the user is the station number of DEFAULT INSTITUTION, in the KERNEL SYSTEM PARAMETERS file (#8989.3).

Your VistA/M system manager may need to create this user for you and provide you with the identifier and division information. You will be prompted to enter these values when running the sample Web application.

3.8.3 Grant "B"-Type Option to the Application Proxy User

The sample Web application is the Application Proxy re-authentication page. To run it against your own VistA/M system, you need to grant the "B"-type option to the application proxy user (which was added to the NEW PERSON file (#200) as part of the VistALink 1.5 install).

1. Grant the "B"-type option "XOBV VISTALINK TESTER" to the application proxy user XOBVTESTER,APPLICATION PROXY

3.8.4 Run the Sample Web App

You can use each of the pages (DUZ, VPID, or Application Proxy) for which you have set up a user to run RPCs with the Sample web application. The results page for each will report whether the test, which executes a series of RPCs against your VistA/M system, has been successful or not. The value for **All RPCs Executed Successfully** will show "true" rather than "false," as shown in the example below:

Report Generated at Wed Apr 06 16:28:40 PDT 2005.

Credentials:

type	value
vpid:	null
application proxy name:	null

type	value
duz:	12345
access code:	null
verify code:	null
division:	523
CCOW handle:	null
connector used (JNDI name):	vlj/testconnector

All RPCs executed successfully?: true

RPC Results:
XOBV TEST PING Results:
Ping Successful!

3.9 Troubleshooting

For help with problems encountered running VistALink, see the "Troubleshooting VistALink" section of the *VistALink 1.5 System Management Guide*.

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.

Running any Java 2 Standard Edition (J2SE) application consists first of the following steps:

1. Install the correct Java Runtime Environment (JRE)

2. Make all supporting Java libraries available

3. Install the application

4. Set up the application classpath

5. Set up any configuration files required by the application

The figure below shows the flow of steps for setting up and running the VistALink 1.5 SwingTester sample application for J2SE.
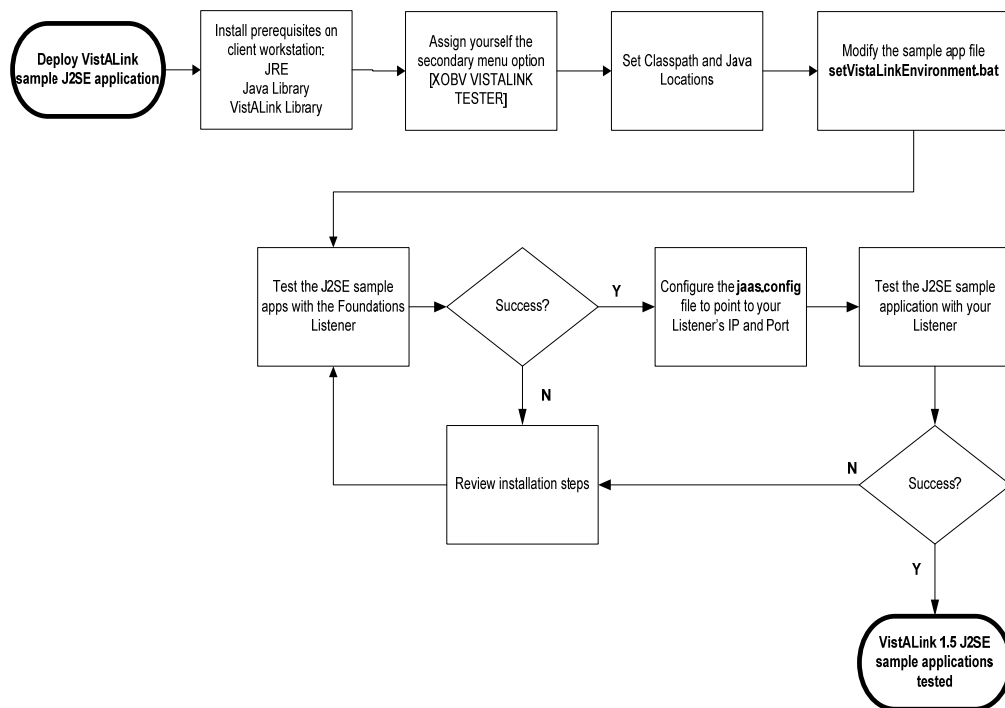
Figure 11. Flowchart for Running J2SE Sample Applications

Installation Instructions

Installing the Java Runtime Environment (JRE)

VistALink requires the J2SE Java Runtime Environment (JRE) 1.4.1 (or higher) or the Java Development Kit (JDK) to be installed on the client workstation.

Installing the J2SE Sample Application Files

To install the J2SE Sample Application files:

1. Create a directory, to hold the sample application files, e.g.,

    ```
    c:\Program Files\vistalink\samples, for the sample
    application
    ```

2. Copy the contents of the \samples\J2SE folder in the distribution file to

    ```
    c:\Program Files\vistalink\samples
    ```

Copying Java Libraries

VistALink requires certain supporting libraries to be available on the client workstation:

1. You need either **weblogic.jar** or **j2ee.jar**. Do one of the following:

    *   Download and install the 1.3.x J2EE SDK (http://java.sun.com/j2ee/sdk_1.3/), to get j2ee.jar (then the SDK can be un-installed)

    *   If you have access to an installed WebLogic server, you can just use **weblogic.jar** from the WebLogic server installation directory's **lib** subdirectory.

2. Copy the following library files to the same folder you copied the J2SE Sample Application files to:

    *   **j2ee.jar or weblogic.jar** – Sources: the directory where J2EE 1.3.x runtime was installed, or weblogic server's \lib folder

    *   **jaxen-dom.jar** and **jaxen-core.jar** – Source: VistALink distribution zip file, <DIST FOLDER>\rar\ExplodedVistaLinkRAR\lib folder

    *   **saxpath.jar** – Source: VistALink distribution zip file, <DIST FOLDER>\rar\ExplodedVistaLinkRAR\lib folder

    *   **log4j-1.2.8.jar** – Source: VistALink distribution zip file, <DIST FOLDER>\rar\ExplodedVistaLinkRAR\lib folder

Copying VistALink Libraries

Copy the following files from the VistALink distribution jars folder to the same folder you copied the J2SE Sample Application files to (e.g., **c:\program files\vistalink\samples**):

- vljConnector-1.5.0.nnn.jar
- vljFoundationsLib-1.5.0.nnn.jar
- vljSecurity-1.5.0.nnn.jar

Granting 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).

Note: 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/.

Setting Classpath and Java Locations

Three batch files are supplied in the samples folder of the distribution, one for each of the three sample applications:

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

In addition, a fourth batch file (**setVistaLinkEnvironment.bat**) is supplied that sets the classpath and the location of the **Java.exe** executable to use on your workstation. This fourth batch file is called by each of the other three batch files listed above. So to configure the classpath and Java executable location for your workstation, you need modify only this one file. The content of this file, as distributed, is:

```
REM -- you may need to adjust the locations of the various jars and
REM    other files to match the locations of these files on your
REM    system.
REM
REM -- set directory with bin subdirectory containing java.exe
REM -- (don't include the \bin subdirectory)
set JAVA_HOME=c:\j2sdk1.4.2_08
REM
REM -- classpath for J2EE (j2ee.jar or weblogic.jar)
REM CLASSPATH=./weblogic.jar
set CLASSPATH=./j2ee.jar
REM
REM -- classpath for XML libraries
```

```
set CLASSPATH=%CLASSPATH%;./jaxen-core.jar
set CLASSPATH=%CLASSPATH%;./jaxen-dom.jar
set CLASSPATH=%CLASSPATH%;./saxpath.jar
REM
REM -- classpath for Log4J
set CLASSPATH=%CLASSPATH%;./log4j-1.2.8.jar
REM
REM -- classpath for VistALink (replace version #s if different)
set CLASSPATH=%CLASSPATH%;./vljConnector-1.5.0.nnn.jar
set CLASSPATH=%CLASSPATH%;./vljFoundationsLib-1.5.0.nnn.jar
set CLASSPATH=%CLASSPATH%;./vljSecurity-1.5.0.nnn.jar
REM
REM -- classpath for sample app  (replace version # if different)
REM #) -- (assumes the samples jar is in the current directory)
set CLASSPATH=%CLASSPATH%;./vljSamples-1.5.0.nnn.jar
```

Modifying Sample Application Batch Files

1. Modify the **setVistaLinkEnvironment** batch file 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.
    Choose version 1.4.1 or higher.

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

    ```
    REM -- set the directory location containing java.exe executable
    REM -- (don't include the \bin subdirectory)
    set JAVA_HOME=c:\j2sdk1.4.1_02
    ```

    If you wish to verify that you have correctly modified the batch file, look in the bin
    directory of the JAVA_HOME environment variable and use the **java –version**
    command to determine what version of the JRE you are running. Use the example
    below as a guide.

    ```
    C:\>CD\j2sdk1.4.1_02\jre\

    C:\j2sdk1.4.1_02\jre>CD BIN

    C:\j2sdk1.4.1_02\jre\bin>java -version

    java version "1.4.2"
    Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2-b28)
    Java HotSpot(TM) Client VM (build 1.4.2-b28, mixed mode)
    ```

2. Modify the **setVistaLinkEnvironment** batch file to match the locations of the various
    supporting library jar files needed to run the sample application. You need to specify
    the locations of each of the **J2EE (or weblogic)**, **JAXEN**, **Log4J**, **saxpath**, and
    VistALink library jar files.

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 clear CLASSPATH and set CLASSPATH for J2EE
set CLASSPATH=./j2ee.jar
```

Running 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.

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**.
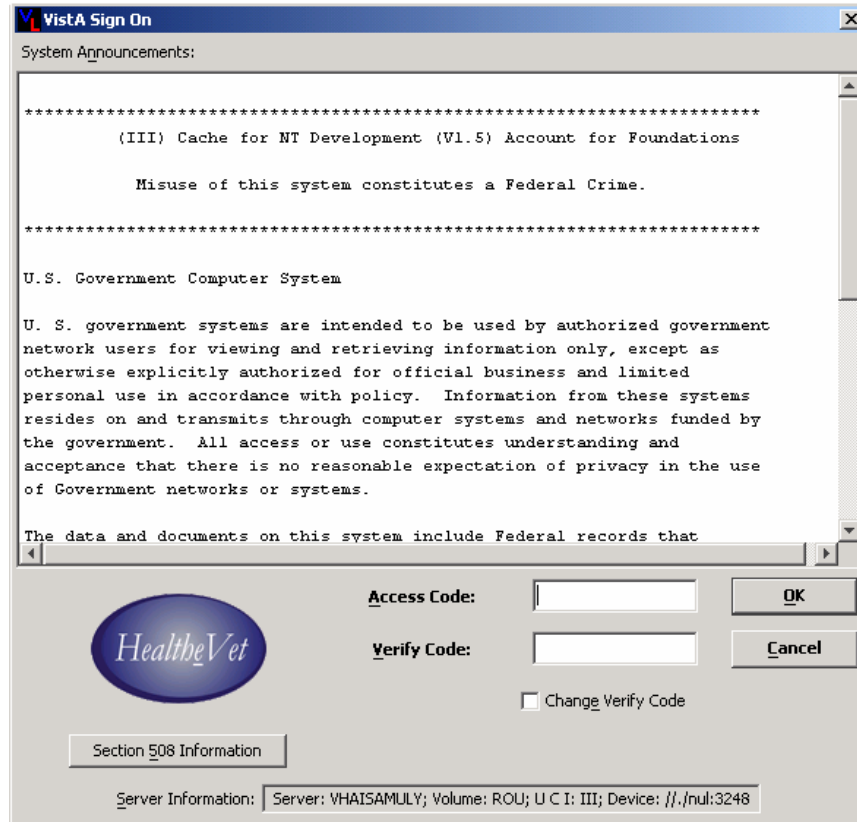
Figure 12. Test Program Access/Verify Code Entry

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.
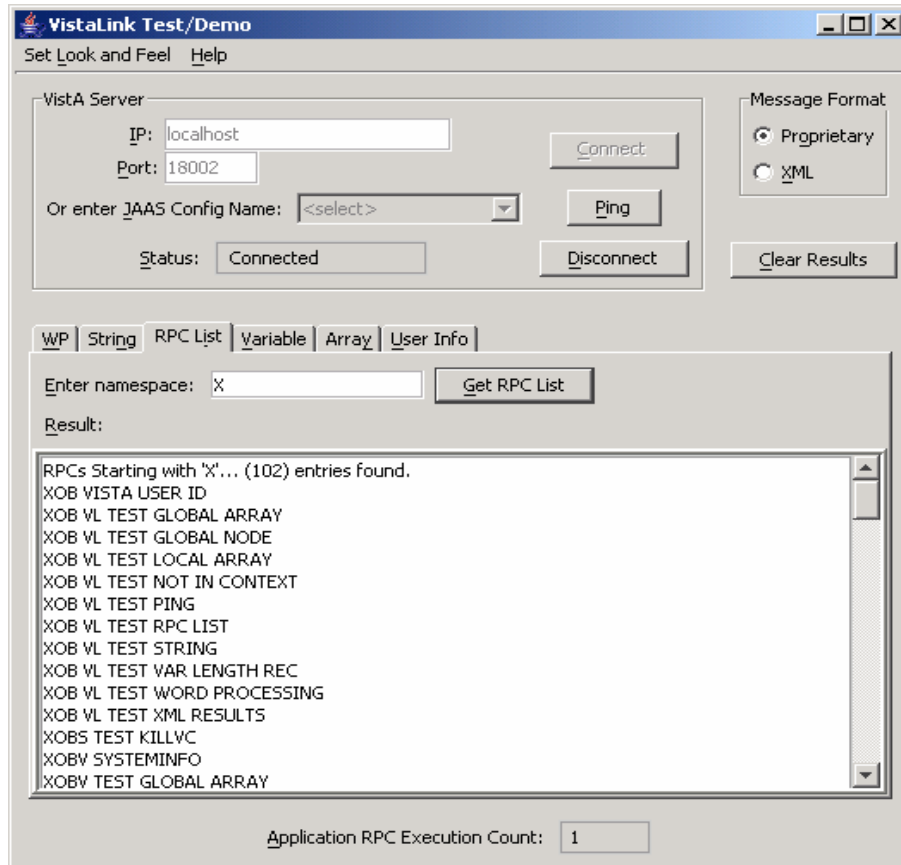
Figure 13. SwingTester RPC List

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.
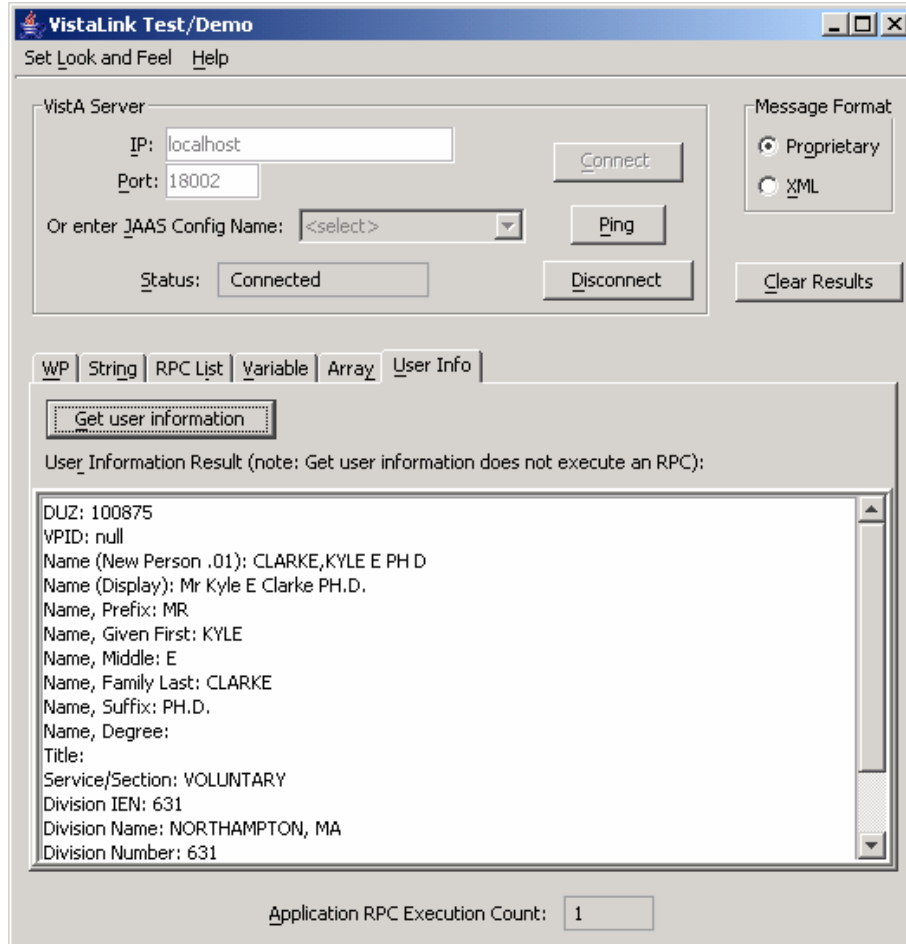
Figure 14. Test Program User Information

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.

Note: **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 **log4jconfig.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 **log4jconfig.xml** file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="myConsoleAppender1" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%-4r [%t] %-5p class %C method %M
line number %L category %c %x - %m%n"/>
    </layout>
  </appender>
  <root>
    <priority value ="info" />
    <appender-ref ref="myConsoleAppender1"/>
  </root>

</log4j:configuration>
```

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).

Note: 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.

Table 1. VistALink Sample Application Loggers

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

Appendix A: Installing and Running J2SE Sample Applications

VistALink 1.5.0
Installation Guide Draft

Appendix B:
Using a Single Server as a Windows Service

Developers may want to set up WebLogic as a Windows Service in their development environment. (This is not a production solution.)

Follow the guidelines in the WebLogic Server documentation for setting up WebLogic Server as a Windows service. The service will be called "beasvc." In the WLS 8.1 documentation, you can find these instructions at

> http://edocs.bea.com/wls/docs81/adminguide/winservice.html)

under "Setting Up a WebLogic Server Instance as a Windows Service."

For VistALink, follow the steps described for installing a single server in the section of this guide titled, "Installing the VistALink 1.5 Adapter(s)" – but modify **installService.cmd** rather than **startWebLogic.cmd**.

Appendix C:
Checksum Information

The routine name and corresponding checksum value for each M routine contained within the VistALink 1.5 software package are listed below.

Routine	CHECK^XTSUMBLD Checksum	CHECK1^XTSUMBLD Checksum
Package: XOBU 1.5		
XOBUENV	value = 13174235	value = 63357205
XOBUM	value = 6398625	value = 36451931
XOBUM1	value = 5453174	value = 26753206
XOBUPOST	value = 532482	value = 854598
XOBUPRE	value = 1471175	value = 3506590
Package: XOBV 1.5		
XOBVLIB	value = 6831152	value = 29313547
XOBVLJU	value = 1960856	value = 6703339
XOBVLL	value = 4145247	value = 17549397
XOBVLT	value = 7984287	value = 33177137
XOBVPOST	value = 10626166	value = 74532391
XOBVPRE	value = 448563	value = 705681
XOBVRH	value = 4065674	value = 12086865
XOBVRM	value = 1906083	value = 4104338
XOBVRMX	value = 1567727	value = 3191236
XOBVRPC	value = 11555639	value = 55905021
XOBVRPCI	value = 9506456	value = 31701719
XOBVRPCX	value = 4846365	value = 22552757
XOBVSKT	value = 5856297	value = 19405670
XOBVSYSI	value = 9172263	value = 49378284
XOBVTCP	value = 4854518	value = 27664282
XOBVTCPL	value = 2857826	value = 9020728
XOBVTLS	value = 2347267	value = 4233837
Package: XOBS 1.5		
XOBSCAV	value = 13785453	value = 52453815
XOBSCAV1	value = 28453677	value = 79287263
XOBSCAV2	value = 22768250	value = 62164548
XOBSCI	value = 535269	value = 707070
XOBSRA	value = 9526226	value = 62375119
XOBSRA1	value = 1388857	value = 2895023
XOBSRAKJ	value = 2733043	value = 10254848

Appendix C: Checksum Information

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 BEA 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.
Anonymous Software Directories	Directories where VHA application, documentation, and patch files are placed for distribution.
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.
Caché	Caché is an M environment, a product of InterSystems Corp.
CCOW	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.
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*.
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 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.
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 Site Parameters file. 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	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.
File #18	System file #18 was the precursor to the KERNEL SYSTEMS PARAMETERS file, 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.
Global	A multi-dimensional data storage structure -- the mechanism for persistent data storage in a MUMPS database.
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.
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 Site Parameters file. 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 1.5 release 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.
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.
J2EE	*Java 2 Enterprise Edition.* A standard suite of technologies for developing distributed, multi-tier, enterprise applications.
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.
JDK	*Java Development Kit.* A set of programming tools for developing Java applications.
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.
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 operating system that runs on various types of hardware platforms. 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 BEA WebLogic domain that is not an

	administration server, i.e., not used to configure all other server instances in the domain.
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	The New Person file contains information for all valid users on an M 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.
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.
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.
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 BEA WebLogic Systems.
XOB Namespace	The VistALink namespace. All VistALink programs and their elements begin with the characters "XOB."

VistALink 1.5.0
Installation Guide Draft