



VISTALINK RELEASE NOTES

Version 1.6

October 2019

Department of Veterans Affairs
Office of Information and Technology
Product Development

Revision History Date	Revised Pages	Patch Number	Description
10/24/2019	N/A	XOBV/XOBS*1.6*4	No Changes
01/14/2019	6-2	XOBV/XOBS*1.6*3	Added 2 factor authentication RPC call.

Contents

1. Introduction	1-1
1.1. About VistALink.....	1-1
1.2. WebLogic Updates Project.....	1-1
1.3. New Product Web Site	1-1
1.4. For More Information.....	1-1
2. M Updates	2-1
2.1. Include Client IP address in error traps for Failed Connections	2-1
2.2. Elimination of SETMSG+5^XOBVRH (Telnet Connection) Hard Error	2-1
2.3. Added Support for Launching Cache/Linux Listeners via XINETD.....	2-1
2.4. New Connection Manager (aka 'Zapper').....	2-1
3. VistALink Adapter Updates.....	3-3
3.1. WebLogic 9/10 Compatibility	3-3
3.2. RAR Changes Summary since Previous Version.....	3-3
3.3. New Recommendation: Deploy VistALink Jars as J2EE Shared Libraries (Production Systems Only).....	3-4
3.4. Revised VistALink MBeans.....	3-4
3.5. VistALink Adapters No Longer Cache DNS Lookups for M Systems.....	3-4
3.6. Automatic ServerType Detection	3-5
3.7. Updated Sample log4j Configuration Files	3-5
3.8. Added Support Needed to Connect to VistALink Listeners Through an SSH (Secure Shell) Tunnel.....	3-5
3.8.1. Security: SSH Needs Client IP Address.....	3-5
3.8.2. Added Support for Deployment Toolkit	3-6
4. Institution Mapping Updates	4-1
4.1. Leading '0' Allowed in Primary Station Number.....	4-1
4.2. Elimination of primaryStationSuffix Attribute	4-1
4.3. New Pluggable Institution Mapping Rules.....	4-1
5. VistALink Administration Console Updates.....	5-1
5.1. WebLogic 9/10 Compatibility	5-1
5.2. WebLogic 10.3 Compatibility	5-1
5.3. Console Honors Monitor Role Restrictions	5-1

5.4.	Configuration Editor Access/Verify Code Encryption Changes.....	5-1
5.5.	Elimination of the Upload/Download Configuration File Feature.....	5-2
5.6.	New JNDI Name Recommendations.....	5-2
6.	Developer API Updates.....	6-1
6.1.	Exception Nesting Changes in VistALink 1.6	6-1
6.2.	VistALink Deprecated APIs.....	6-1
6.3.	VistALink New APIs.....	6-2
7.	J2SE Client/Server Updates	7-3
7.1.	Java 5 or Greater Required.....	7-3
7.2.	Elimination of Unnecessary J2SE Logger Error-Level Message.....	7-3
7.3.	CCOW User Context Update	7-3
7.4.	Smaller Geronimo Jar Replaces Need for J2EE/WebLogic Jar for Clients.....	7-3
7.5.	Section 508 Compliance: Differentiate Focus on Change Verify Code Check Box.....	7-3

1. Introduction

1.1. About VistALink

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

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

- The adapter libraries use the Java 2 Enterprise Edition (J2EE) Connector Architecture (J2CA) 1.5 specification to integrate Java applications with legacy systems.
- The M listener process receives and processes requests from client applications.
- Java applications can call Remote Procedure Calls (RPCs) on the M server, executing RPC Broker RPCs on the M server without modification.

1.2. WebLogic Updates Project

In support of the Department of Veterans Affairs Information Technology application Modernization effort, the three applications VistALink, Fat-client Kernel Authentication and Authorization (FatKAAT), and Kernel Authentication and Authorization for the Java 2 Enterprise Edition (KAAJEE) have been developed. Based on the direction of the Technical Review Model (TRM) and in order to support applications that upgrade to the new WebLogic Server versions 9.2 and 10.x, this project is required. The scope of the project is to upgrade these three applications to work with the WebLogic Server versions 9.2, 10.x, and 11g.

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

1.3. New Product Web Site

New VistALink Web site (<http://vaww.vista.med.va.gov/vistalink/>) summarizes VistALink architecture and functionality and gives status updates for all VistALink products.

1.4. For More Information

For more information on any new feature discussed in the release notes, please see the VistALink documentation set on the VA Software Document Library at the following address:

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

2. M Updates

2.1. Include Client IP address in error traps for Failed Connections

For failed connections (e.g., when a port scanner or telnet client tries to connect to your listener) where an error is logged, the variable XWBTIP is now defined in the error trap, and contains the client IP address. This will help VistA sites determine the source of the errors/connection failures.

2.2. Elimination of SETMSG+5^XOBVRH (Telnet Connection) Hard Error

Formerly when connecting to a VistALink listener via telnet (e.g., to test listener connectivity) a hard M error (SETMSG+5^XOBVRH) would be set into the Kernel error trap. Now a “soft” error (dialog #184001) is displayed and logged in the error trap instead.

2.3. Added Support for Launching Cache/Linux Listeners via XINETD

A new entry point has been added (CACHELNX^XOBVTCP) for operations staff to support running VistALink listeners at the Linux operating system level via xinetd, when running Caché on Linux. As this still an M system configuration that VA OI&T is experimenting with at the time of writing, VistALink's support is also experimental at the current point in time.

2.4. New Connection Manager (aka 'Zapper')

The new "Connection Manager" functionality in VistALink 1.6 provides a UI screen to view all VistALink processes connected to the M system, including the associated user, and actions to kill either a selected process or all connected VistALink processes. This functionality may be useful during software installations, for example, where it might be desirable to get all VistALink processes off of the system.

3. VistALink Adapter Updates

3.1. WebLogic 9/10 Compatibility

VistALink 1.6 adapters have been updated to be compatible with:

- WebLogic 9.2, 10.x and 11g
- Java 5
- J2EE 1.4+
- J2EE Connector Architecture 1.5

3.2. RAR Changes Summary since Previous Version

- **vljFoundationsLib and vljConnector jars:** New versions provide a J2CA 1.5 compliant adapter implementation.
- **Deployment Descriptor Changes:** The format of both the ra.xml and weblogic-ra.xml descriptors is different. Existing adapters' deployment descriptors need to be updated. Using the provided sample/template descriptors, only weblogic-ra.xml will need to be modified when creating a VistALink RAR:
 - **META-INF/ra.xml:** The template version provided in the VistALink distribution is updated to be J2CA 1.5 compliant. In most cases, it no longer needs any further editing when creating a RAR.
 - **META-INF/weblogic-ra.xml:** The template version provided in the VistALink distribution is updated to be J2CA 1.5 compliant. It must be edited for each RAR deployed.
 - **META-INF/MANIFEST.MF:** The template version provided in the VistALink distribution is updated to support use of J2EE Shared Libraries by RARs.
- **lib/saxpath.jar, lib/jaxen-core.jar, and lib/jaxen-dom.jar** are no longer needed in the RAR as their functionality has been replaced by the XPath implementation introduced in Java 5.
- **lib/xbean.jar** no longer needed in the RAR since an implementation of XML Beans is included in WebLogic 9/10.
- **Linked adapters** (link-ref mechanism) no longer supported for VistALink 1.6 adapters.
- **Development System Classloading:** Automatic classloading of all jars in RAR is now supported by WebLogic 9.x/10. Library jars included in RAR no longer need to be manually added to the server classpath.
- **Production System Classloading:** Oracle (formerly known as BEA) recommends removing all jars from RARs on production systems, and deploying them as J2EE shared libraries instead (to reduce resource consumption – replacement for resource-saving aspect of the 8.1 link-ref mechanism). When deployed as J2EE shared libraries, the jars supporting VistALink RARs no longer need to be manually added to the server classpath.

3.3. New Recommendation: Deploy VistALink Jars as J2EE Shared Libraries (Production Systems Only)

As a replacement for the WebLogic 8.1 “Linked Adapters” feature (not supported for upgraded adapters like VistALink 1.6), Oracle recommends removing all jars from RARs on production systems, and deploying each one as a J2EE shared library instead. This addresses the reduced resource consumption benefit formerly provided by using the 8.1 link-ref mechanism. Therefore, on production systems only, we recommend that the following jars be removed from all RARs and deployed (once) as J2EE shared libraries:

Jar Name	J2EE Shared Library Deployment Name
vljConnector-1.6.0.jar	vljConnector(1.6,1.6)
vljFoundationsLib-1.6.0.jar	vljFoundationsLib(1.6,1.6)
log4j-1.2.13.jar	log4j-1

Two different versions of an example RAR are provided in the distribution zip file, one structured for development system deployments, the other for production deployments:

- **rar\Rar-Dev** (contains all needed jars)
- **rar\Rar-Prod** (contains no jars; MANIFEST.MF file looks for jars as J2EE shared libraries)

3.4. Revised VistALink MBeans

VistALink MBeans have been revamped, with one MBean per running adapter, allowing third-party management tools to more easily monitor on VistALink-specific error thresholds (e.g., authentication failures). VistALink provides two MBean types that can be monitored:

- **VistaLinkConnector MBean**
 Deployed: One per deployed VistALink adapter, per JVM
 MBean Type: VistaLinkConnector
 ObjectName: gov.va.med:Name=VistaLink_XXXXX,Type=VistaLinkConnector
 (where XXXXX is the JNDI name of the adapter)
- **VistaLinkInstitutionMapping MBean**
 Deployed: One instance per JVM where VistALink adapters are deployed
 Type: VistaLinkInstitutionMapping
 ObjectName: gov.va.med:Name=VistaLinkInstitutionMapping,Type=VistaLinkInstitutionMapping

3.5. VistALink Adapters No Longer Cache DNS Lookups for M Systems

A major area of concern with VA facility system administrators using VistALink has been updating VistALink adapters based on DNS updates for M systems. In VistALink 1.5, adapters cached DNS

information obtained for an M system, and that information stayed cached until either the J2EE system restarted or the adapter was undeployed/redeployed.

In 1.6, VistALink adapters no longer cache DNS lookups. As long as the underlying JVM is also configured to not cache DNS lookups (by default it is not!), VistALink 1.6 adapters will always use a live DNS lookup each time a new connection is created.

3.6. Automatic ServerType Detection

J2EE server type detection is now automatic for WebLogic servers. Passing the JVM argument -Dgov.va.med.environment.servertype=weblogic is no longer necessary on WebLogic servers in most configurations. If passed, however, the JVM argument still overrides automatic detection. The gov.va.med.environment.Environment.getServerType() Application Program Interface (API) does auto-detection (for WebLogic only) by examining the classloader hierarchy.

3.7. Updated Sample log4j Configuration Files

The names and purposes of the sample log4j config files in the distribution have changed with the move to WebLogic 9.x/10. The new sample log4j configuration files are located in the log4j directory inside the VistALink distribution zip file, as follows:

File Name	Description
log4jSampleJ2EEConfig.xml	Example configuration for a J2EE system, turning DEBUG-level logging on for VistALink classes of interest on a J2EE system. NOTE: Turning on 'debug' level can adversely affect system performance.
log4jSampleJ2SEConfig.xml	Example configuration for a J2SE (Java client-M server) application, turning DEBUG-level logging on for VistALink classes of interest for client/server applications.

3.8. Added Support Needed to Connect to VistALink Listeners Through an SSH (Secure Shell) Tunnel

3.8.1. Security: SSH Needs Client IP Address

Pilot work done using Secure Shell (SSH), allowing data to be exchanged using a secure channel between two networked devices, revealed that the client IP address must be explicitly passed from the client. VistALink has been updated to do this, similar to how the Remote Procedure Call (RPC) Broker handles client IP address detection.

3.8.2. Added Support for Deployment Toolkit

Restructured the ZIP file based on recommendations to support Deployment Toolkit (DTK) functionality for WebLogic Application Server 9.x/10.x.

4. Institution Mapping Updates

4.1. Leading '0' Allowed in Primary Station Number

"Primary Stations" for connector configurations can now start with '0' (e.g., to match a primary station of '050' as configured in the FOIA Caché distribution.)

4.2. Elimination of `primaryStationSuffix` Attribute

The rarely-used *primaryStationSuffix* connector attribute (for AITC 200-series station numbers, in the `gov.va.med.vistalink.connectorConfig.xml` file) no longer necessary. This change should affect only the Austin Information Technology Center (AITC) 200-series (i.e., AITC) station numbers. Any such primary station numbers should now be entered entirely within the "primaryStation" attribute in the configuration file, including any suffix (e.g., '200M').

4.3. New Pluggable Institution Mapping Rules

New for non-VA users of VistALink: Organization-specific rules for parsing primary station numbers (validity checking and lookup) are now externalized in a new `IPrimaryStationRules` implementation. Non-VA users of VistALink can provide their own implementation, and override the default VA-specific rules implementation via a new JVM argument, `"-Dgov.va.med.vistalink.primary-station-rules-class"`.

5. VistALink Administration Console Updates

5.1. WebLogic 9/10 Compatibility

WebLogic's console plug-in mechanism changed from 8.1 to versions 9.x/10. The VistALink console plug-in has been updated to be compatible with WebLogic 9.x/10.

5.2. WebLogic 10.3 Compatibility

Due to difficulties encountered integrating the VistALink administration console with the WebLogic 10.3 console (specifically its navigation tree and tab set), the VistALink administration console is also provided as a standalone web application, in an EAR file. Therefore, for WebLogic 10.3 we recommend installing the VistALink administration console as a standalone web application, not integrated with the WebLogic console. The standalone EAR can also be used on 9 and 10.0 of WebLogic if desired.

5.3. Console Honors Monitor Role Restrictions

WebLogic's console supports four roles: Administrators, Deployers, Operators and Monitors. The 1.5 VistALink administration console did not distinguish between these roles, particularly the Monitors role, and it allowed write access (i.e., VistALink's configuration editor) to the otherwise read-only Monitors role. The 1.6 VistALink administration console restricts users in the Monitors role to read-only access.

Note: To support access by users in the Monitors role, the system administrator currently needs to explicitly declare the "Listen Address" attribute in the WebLogic domain configuration, for each server in the domain. Otherwise a JMX remote call that is needed, fails, under Monitor privileges alone.

5.4. Configuration Editor Access/Verify Code Encryption Changes

Encryption of the access and verify codes in the configuration file (`gov.va.med.vistalink.connectorConfig.xml`) adds an additional level of security to help protect connector proxy user credentials. The access/verify code connector attributes in the configuration file are encrypted when:

- A specific connector configuration is loaded by a deployed adapter on a given J2EE server
- A specific connector definition is edited or created inside the Configuration Editor

In addition, the following new encryption-related features have been added to the 1.6 VistALink administration console:

- The number of currently unencrypted entries in the configuration file is listed on the main Configuration Editor page
- An "Encrypt Unencrypted Entries" button allows you to encrypt all unencrypted entries

A new encryption type ("scoped") has been added:

- “scoped” encryption type changes the encryption of access codes to include a WebLogic domain-specific token, making it harder to decrypt the access/verify codes in a WebLogic domain other than the one the configuration file was encrypted in.
- The new file-wide <connectors> attribute "encryptionScoped" in gov.va.med.vistalink.connectorConfig.xml reflects whether entries have been encrypted with domain-scoped encryption or not.
- New console action to change encryption type to scoped. The original encryption type (“not scoped”) is supported as well.
- When the console is used to change the encryption type, it will decrypt and then re-encrypt all currently encrypted access/verify codes, and also change the new "encryptionScoped" file-wide attribute to "true".

5.5. Elimination of the Upload/Download Configuration File Feature

The configuration file upload/download feature is no longer supported in the new VistALink administration console.

5.6. New JNDI Name Recommendations

We recommend that the JNDI name should not contain most punctuation characters. If editing the JNDI name with the Configuration Editor, the following punctuation characters are allowed: - _ / \ () []

Additionally, we recommend not using the ‘/’ punctuation character if you plan to write WLST scripts to monitor VistALink MBeans (in WLST this character is used for another purpose that interferes with accessing an MBean, and VistALink MBean names are based on adapter JNDI names.)

6. Developer API Updates

6.1. Exception Nesting Changes in VistALink 1.6

Previous versions of VistALink were designed to work with Java version 1.3, which did not support nested exceptions. To compensate, nested exception capability was built into all VistALink exception classes.

Now that Java versions 1.4 and above support nested exceptions "out of the box", VistALink's nested exception facilities have been deprecated, and VistALink's classes have been internally modified to store nested exceptions using Java's nested exception functionality instead.

Any code that currently relies on VistALink's deprecated nested exception facilities to obtain nested exception information should switch to using the now-standard Java nested exception facilities.

6.2. VistALink Deprecated APIs

The following APIs have been marked for deprecation in VistALink 1.6:

- `gov.va.med.exceptionFOUNDATIONSExceptionInterface` (and implementers):
 - `getFullStackTrace()` method
 - `getNestedException()` method
- `gov.va.med.exceptionFOUNDATIONSException` (and descendants):
 - `getFullStackTrace()` method
 - `getNestedException()` method
- `gov.va.med.vistalink.adapter.cci.VistaLinkResourceException` (and descendants):
 - `getFullStackTrace()` method
 - `getNestedException()` method
- `gov.va.med.vistalink.security.VistaLoginModuleException` (and descendants):
 - `getFullStackTrace()` method
 - `getNestedException()` method
- `gov.va.med.exception.ExceptionUtils`:
 - `String getFullStackTrace()` method
 - `Throwable getNestedExceptionByClass()` method
- `gov.va.med.xml.XmlUtilities`:
 - `String convertXmlToStr()` method
 - `Document getDocumentForXmlString()` method
 - `Document getDocumentForXmlInputStream()` method
 - `Attr getAttr()` method
 - `Node getNode()` method

- String XML_HEADER field

6.3. VistALink New APIs

Pursuant to Memorandum for Implementation of Federal Personal Identity Verification (PIV) Credentials for Federal Employee and Contractor Access to VA IT Systems (VAIQ# 7614373) 2 Factor Authentication (2FA) was implemented to connect to a VistA system, VistALink is being updated to add an RPC that accepts the SAML token and validates the user log on in the same manner as the RPC Broker software does for web applications. RPC XOBV VALIDATE SAML calls tag SAML for routine XOBVSAML then returns an ARRAY that either contains the users DUZ if successful, or returns "1^error message" and subsequent failure or success messages for the consuming application in the lower subscripts. The RPC receives a SAML token to validate a user logon to VistA of type literal specified as parameter DOC that is the SAML token with a maximum data length of 32000 characters.

7. J2SE Client/Server Updates

7.1. Java 5 or Greater Required

Because VistALink 1.6's jars, used both on WebLogic 9/10, and for client/server connectivity, are compiled with Java 5, new client/server applications using VistALink 1.6 will need to be compiled with and run under a minimum of Java 5 as well.

7.2. Elimination of Unnecessary J2SE Logger Error-Level Message

An unnecessary error-level log4j log message meaningful for J2EE environments only is no longer logged when running in client/server mode in a J2SE environment.

7.3. CCOW User Context Update

CCOW logins now add in the user's DUZ(2) as a second “^”-piece in the user.id.logon.vistalogon context item, for both client/server and FatKAAT logins. Doing this makes VistALink consistent with the RPC Broker's user context population, and supports functionality needed by FatKAAT.

7.4. Smaller Geronimo Jar Replaces Need for J2EE/WebLogic Jar for Clients

The VistALink distribution now includes a small jar (37K), `geronimo-j2ee-connector_1.5_spec-1.1.1.jar`, from the Apache Geronimo J2EE server. Distribution of this jar completely satisfies VistALink's need on the J2SE client side for an implementation of J2EE's `javax.resource.ResourceException` class, replacing the former need for J2SE installers to either place `weblogic.jar` on the client side, or go to the Sun web site and download the Sun App Server, to obtain a single (large) file, `j2ee.jar`. Neither Sun's `j2ee.jar` nor `weblogic.jar` had been included in previous VistALink distributions for two reasons, 1) licensing restrictions, and 2) size.

7.5. Section 508 Compliance: Differentiate Focus on Change Verify Code Check Box

Due to a known defect in Java Swing (http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4985801), and as defined by Section 508 compliance, it might be difficult for end-users to distinguish when focus occurs on the Change Verify Code check box in the login dialog window when using the Windows “Look and Feel” High Contrast Black color scheme.

