HealtheVet Web Services Client (HWSC) 1.0

Systems Management Guide

October 2016

Department of Veterans Affairs (VA)

Office of Information and Technology (OI&T)

Enterprise Program Management Office (EPMO)

Revision History

Date	Version	Description	Author
10/20/2016	2.0	Tech Edits: • Added the "Orientation" section. • Updated the "Introduction" section. • Updated the "Using the Web Server Manager" section for Patch XOBW*1.0*4. • Updated the "Using SSL/TLS and Certificate-Based Authentication with HWSC" section for Patch XOBW*1.0*4. • Converted Word document to .docx format. • Reformatted document to follow latest documentation standards and formatting rules. Also, formatted document for online presentation vs. print presentation (i.e., for double-sided printing). These changes include: o Revised section page setup. o Removed section headers. o Revised document footers. o Removed blank pages between sections. o Revised all heading style formatting. • Updated organizational references (e.g., "Product Development [PD]" to "Enterprise Program Management Office [EPMO]"). • Redacted document for the following information: o Names (replaced with role and initials). o Production IP addresses and ports. o VA Intranet websites. o Server geographic locations and node names.	HealtheVet Web Services Client (HWSC) Project Team
02/--/2011	1.0	HWSC Version 1.0 Initial release	Product Development Services Security Program HWSC development team. Albany, NY OIFO:

Date	Version	Description	Author
			• Developers—M. K. & L. D. Bay Pines, FL OIFO: • Development Manager—C. S. Oakland, CA OIFO: • Developers—K. C. & J. G. • SQA—G. S. • Tester—P. S. • Tech Writer—S. S.

Table of Contents

List of Figures

List of Tables

Orientation

How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of the HealtheVet Web Services Client (HWSC) software and the functionality it provides for Veterans Information Systems and Technology Architecture (VistA).

Intended Audience

The intended audience of this manual is the following stakeholders:

- Enterprise Program Management Office (EPMO)—VistA legacy development teams.
- System Administrators—System administrators at Department of Veterans Affairs (VA) sites who are responsible for computer management and system security on the VistA M Servers.
- Information Security Officers (ISOs)—Personnel at VA sites responsible for system security.
- Product Support (PS)—Personnel who support Kernel-related products.

Disclaimers

Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is *not* subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed freely provided that any derivative works bear some notice that they are derived from it.

CAUTION: Kernel routines should *never* be modified at the site. If there is an immediate national requirement, the changes should be made by emergency Kernel patch. Kernel software is subject to FDA regulations requiring Blood Bank Review, among other limitations. Line 3 of all Kernel routines states:

Per VHA Directive 2004-038, this routine should not be modified

CAUTION: To protect the security of VistA systems, distribution of this software for use on any other computer system by VistA sites is prohibited. All requests for copies of Kernel for *non*-VistA use should be referred to the VistA site's local Office of Information Field Office (OIFO).

Documentation Disclaimer

This manual provides an overall explanation of using Kernel; however, no attempt is made to explain how the overall VistA programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA Internet and Intranet SharePoint sites and websites for a general orientation to VistA. For example, visit the Office of Information and Technology (OI&T) Enterprise Program Management Office (EPMO) Intranet Website.

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

Documentation Conventions

This manual uses several methods to highlight different aspects of the material:

- Various symbols are used throughout the documentation to alert the reader to special information. Table 1 gives a description of each of these symbols:

Table 1: Documentation Symbol Descriptions

Symbol	Description
	**NOTE / REF:** Used to inform the reader of general information including references to additional reading material.
	**CAUTION / RECOMMENDATION / DISCLAIMER:** Used to caution the reader to take special notice of critical information.

- Descriptive text is presented in a proportional font (as represented by this font).
- "Snapshots" of computer commands and online displays (i.e., screen captures/dialogues) and computer source code, if any, are shown in a *non*-proportional font and can be enclosed within a box.
    - User's responses to online prompts are **boldface** and (optionally) highlighted in yellow (e.g., **<Enter>**).
    - Emphasis within a dialogue box is **boldface** and (optionally) highlighted in blue (e.g., STANDARD LISTENER: RUNNING).
    - Some software code reserved/key words are **boldface** with alternate color font.
    - References to "**<Enter>**" within these snapshots indicate that the user should press the **Enter** key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing **<PF1>**.
    - Author's comments are displayed in italics or as "callout" boxes.

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

- This manual refers to the M programming language. Under the 1995 American National Standards Institute (ANSI) standard, M is the primary name of the MUMPS programming language, and MUMPS is considered an alternate name. This manual uses the name M.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field/file names, and security keys (e.g., the XUPROGMODE security key).

  NOTE: Other software code (e.g., Delphi/Pascal and Java) variable names and file/folder names can be written in lower or mixed case (i.e., CamelCase).

How to Obtain Technical Information Online

Exported VistA M Server-based software file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.

NOTE: Methods of obtaining specific technical information online are indicated where applicable under the appropriate section.

REF: See the *Kernel Technical Manual* for further information.

Help at Prompts

VistA M Server-based software provides online help and commonly used system default prompts. Users are encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of VistA M Server-based software.

Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the List File Attributes option [DILIST] on the Data Dictionary Utilities menu [DI DDU] in VA FileMan to print formatted data dictionaries.

REF: For details about obtaining data dictionaries and about the formats available, see the "List File Attributes" chapter in the "File Management" section in the *VA FileMan Advanced User Manual*.

Assumptions

This manual is written with the assumption that the reader is familiar with the following:

- VistA computing environment:
  - Kernel 8.0—VistA M Server software
  - Remote Procedure Call (RPC) Broker 1.1—VistA M Server software
  - VA FileMan 22.0 (and higher) data structures and terminology—VistA M Server software
  - VistALink 1.6—VistA M Server and Application Server software
- Linux or Microsoft® Windows environment

- Java Programming language:
  - Java Integrated Development Environment (IDE)
  - J2SE™ Development Kit (JDK)
  - Java Authentication and Authorization Services (JAAS) programming
- M programming language
- WebLogic 9.2 or 10.x Application Server

Reference Materials

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

- *HWSC 1.0 Installation Guide*
- *HWSC 1.0 Systems Management Guide* (this manual)
- *HWSC 1.0 Developer's Guide*
- *HWSC 1.0 Patch XOBW*1.0*4 Release Notes*
- *HWSC 1.0 Patch XOBW*1.0*4 Installation, Back-Out, and Rollback Guide*
- *HWSC 1.0 Patch XOBW*1.0*4 Security Configuration Guide*

VistA documentation is made available online in Microsoft® Word format and in Adobe® Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe® Acrobat Reader, which is freely distributed by Adobe® Systems Incorporated at: http://www.adobe.com/

VistA documentation can be downloaded from the VA Software Document Library (VDL): http://www.va.gov/vdl/

REF: HWSC manuals are located on the VDL at: http://www.va.gov/vdl/application.asp?appid=180

VistA documentation and software can also be downloaded from the Product Support (PS) Anonymous Directories.

1 Introduction

1.1 HWSC Overview

HealtheVet Web Services Client (HWSC) uses Caché's Web services client to invoke Web service methods on external servers and retrieve results. It provides helper methods and classes to improve the use of the Web service client in Veterans Health Information Systems and Technology Architecture (VistA).

1.1.1 HWSC Features

HWSC acts as an adjunct to the Web services client functionality provided in Caché, by:

- Leveraging Caché's platform-provided Web services client capabilities.
- Adding a file and user interface (UI) to manage the set of external Web server endpoints (IP, port, etc.)
- Adding a file and UI to register and manage the set of external Web services.
- Providing runtime application programming interface (API) to invoke a specific Web service on a specific Web server.
- Providing a runtime API to facilitate error processing in a VistA environment.
- Providing a deployment API to install/register a Web service proxy from a Web Services Description Language (WSDL) file.
- Providing a management UI including the ability to "ping" (test) a given Web service/server combination from VistA M.
- Supporting both Simple Object Access Protocol (SOAP)- and Representational State Transfer (REST)-style Web services.
- Fostering consistent implementation of VistA M Web service consumers.

2 HWSC Management Functions

HealtheVet Web Services Client (HWSC) provides several management screens allowing you to create and manage the Web server and Web service information needed by VistA applications to consume external Web services. The management screens are:

- Web Server Manager
- Web Service Manager
- Lookup Key Manager

2.1 Using the Web Server Manager

You can use the XOBW WEB SERVER MANAGER option to call up the HWSC Web Server Manager. You should use this tool to enter Web server information.

NOTE: Programmer access (DUZ(0)= "@") is required to use this option.

Figure 1: Using the XOBW WEB SERVER MANAGER Option

```
Web Server Manager            Apr 18, 2007@16:07:54          Page:    1 of 1
                        HWSC Web Server Manager
                       Version: 1.0     Build: xx

 ID    Web Server Name           IP Address or Domain Name:Port
 1    *Oakland Test Server1      vhaisxsysa.vha.med.va.gov:7111
 2    *Oakland Test Server2      vhaisxsysb.vha.med.va.gov:7112
 3    *Oakland Test Server3      vhaisxsysc.vha.med.va.gov:7113
 4    *Oakland Test Server4      vhaisxsysd.vha.med.va.gov:7114


          Legend:  *Enabled
AS  Add Server                          TS  Test Server
ES  Edit Server                         WS  Web Service Manager
DS  Delete Server                       CK  Check Web Service Availability
EP  Expand Entry                        LK  Lookup Key Manager
Select Action:Quit//
```

Table 2 summarizes the actions available in the Web Server Manager.

Table 2: Web Server Manager Actions

Action	Description
AS (Add Server)	Add a new entry in the WEB SERVER file (18.12).
DS (Delete Server)	Delete an entry from the WEB SERVER file (18.12).
ES (Edit Server)	Edit an entry in the WEB SERVER file (18.12).
EP (Expand Entry)	View all information about a particular entry in the WEB SERVER file (18.12).
TS (Test Server)	If the XOBT sample application is installed, it runs some of its tags to call sample external Web services. Disabled if the sample application (XOBT) is *not* installed. **REF:** For more information on the XOBT sample, see the *HWSC*

Action	Description
	1.0 Developer's Guide.
WS (Web Service Manager)	Invoke the Web Service Manager screen.
CK (Check Web Service Availability)	Check availability for each Web service authorized/assigned to the Web server.
LK (Lookup Key Manager)	Invoke the Lookup Key Manager screen.

When you add or edit a Web server, you are prompted for the information shown in Table 3.

Table 3: Web Server Fields

Field	Description
Name	Name to identify the Web server entry. The name *must* be 3-30 characters in length.
Server	The full domain name (for DNS) or IP address of the Web service server.
Port	The TCP/IP port of Web service server.
Default Http Timeout	A default http timeout to use for outgoing requests made to this server. The default value is 30.
Status	Select either ENABLED or DISABLED.
Security Credentials	
Login Required?	If a login is required, enter **YES** (allows editing of username and password).
Username	Name of the authorized user in the security realm on the Web server.
Edit Password?	Enter "**Y**" if you wish to change the password; otherwise, "**N**".
	SSL Setup
SSL Enabled	Determines whether SSL/TLS is enabled for the Web server.
SSL Configuration	Name of Caché SSL configuration to use for this Web server.
SSL Port	SSL port number to use for this Web server.
Authorize Web Services	
Select Web Service	Select one of the Web services listed, or enter a new one. A Web service *must* be "authorized" by entering here, to be used with this Web server.
Status	Select ENABLED or DISABLED.

2.2 Using Web Service Manager

Use the Web Service Manager to enter or modify information for Web services that M applications access.

NOTE: VistA applications that install Web service clients will probably automatically create Web service entries for the external Web services they are accessing.

NOTE: Programmer access (DUZ(0)= "**@**") is required to use this option.

To display the Web Service Manager, select the WS action in the Web Server Manager (see "Using the Web Server Manager"). In addition to adding a new service, you can edit or delete a previous entry, or display complete information previously entered for a particular service.

Figure 2: Using Web Service Manager

```
Web Service Manager          May 09, 2007@14:27:19          Page:    1 of    1
HWSC Web Service Manager
Version: 1.0       Build: xx

 ID    Web Service Name           Type   URL Context Root
 1     XOBT TESTER REST SERVICE   REST   hwscrestservice
 2     XOBT TESTER WEB SERVICE    SOAP   hwscwebservices/TesterWebService


          Enter ?? for more actions
AS  Add Service
ES  Edit Service
DS  Delete Service
EP  Expand Entry
Select Action:Quit//
```

Table 4 summarizes the actions available in the Web Service Manager.

Table 4: Web Service Manager Actions

Action	Description
AS (Add Service)	Add a new entry to the WEB SERVICE file (18.02).
DS (Delete Service)	Delete an entry from the WEB SERVICE file (18.02).
ES (Edit Service)	Edit an entry in the WEB SERVICE file (18.02).
EP (Expand Entry)	View all information about a particular entry in the WEB SERVICE file (18.02).

There are two types of Web services supported by HWSC:

- Representational State Transfer (REST)
- Simple Object Access Protocol (SOAP)

When you register a new Web service, you are prompted for slightly different information depending on the type, as shown in Table 5.

Table 5: Web Service Fields

Service Type	Field	Description
REST	Name	Name to identify the Web service entry. The name *must* be *non*-numeric, 3-30 characters long, and starting *without* punctuation.
	Date Registered	Date the entry was registered (created).
	Service Type	Choose **REST**.
	Context Root	The context root of the Web service.
	Availability Resource	A "resource" to append to the context root to create a URL that can be used to check if the Web service is available.
SOAP	Name	Name to identify the Web service entry. The name *must* be *non*-numeric, 3-30 characters long, and starting *without* punctuation.
	Date Registered	Date the entry was registered (created).
	Service Type	Choose **SOAP**.
	Proxy Class Name	Name of the Caché Object class that is the Web service client proxy, as created by the Caché WSDL compiler.
	Context Root	The context root of the Web service.
	Availability Resource	A "resource" to append to the context root to create a URL that can be used to check if the Web service is available.

2.3 Using the Lookup Key Manager

The Lookup Key Manager is a tool for sites to use to associate a mnemonic name with a particular server. Applications use the mnemonic at runtime to retrieve the Web server.

Figure 3: Using the Lookup Key Manager

```
Lookup Key Manager            Apr 20, 2007@12:52:21          Page:    1 of 1
HWSC Web Server Lookup Key Manager
Version: 1.0     Build: xx

Filters:  Key = <no filter>     Server = <no filter>
 ID    Lookup Key Name [Sorted By]              Web Server Name
 1     XOBZ DEMO SERVER                             Oakland Test Server3
 2     XOBZ ID SERVER                           Oakland Test Server1
 3     XOBZ MESSAGE SERVER                   Oakland Test Server1
 4     XOBZ PATIENT SERVER                          Oakland Test Server2


          Enter ?? for more actions
AK  Add Key                              SS  Switch Sort
EK  Edit Key                             FK  Filter Key
DK  Delete Key                           FS  Filter Server
EP  Expand Entry
Select Action:Quit//
```

Table 6 summarizes the actions available in the Lookup Key Manager.

Table 6: Lookup Key Manager Actions

Action	Description
AK (Add Key)	Add a new entry to the WEB SERVER LOOKUP KEY file (18.13).
DK (Delete Key)	Delete an entry from the WEB SERVER LOOKUP KEY file (18.13).
EK (Edit Key)	Edit an entry in the WEB SERVER LOOKUP KEY file (18.13).
EP (Expand Entry)	View all information about a particular entry in the WEB SERVER LOOKUP KEY (18.13).
FK (Filter Key)	Limit the list of lookup keys displayed by the key manager. The user specifies text to be used as a filter against the beginning characters of the key values. Only matching keys are listed. This protocol is also used to clear a key filter if one is currently being applied.
FS (Filter Server)	Limit the list of Web server entries displayed by the key manager. The user specifies text to be used as a filter against the beginning characters of the key values. Only matching keys are listed. This protocol is also used to clear a server filter if one is currently being applied.
SS (Switch Sort)	Switch sorting between “key” and “server” in the list of Web server lookup keys.

When you enter a new lookup key, you are prompted for the information shown in Table 7.

Table 7: Lookup Key Manager Fields

Field	Description
Key Name	The name for the lookup key *must* be 3-30 characters in length.
Brief Description	The description *must* be 2-50 characters in length.
Associated Web Server Name	The WEB SERVER NAME with which to associate this key. Lookups made on the key return this Web server.

3 Security

3.1 Using SSL/TLS and Certificate-Based Authentication with HWSC

HealtheVet Web Services Client (HWSC) HyperText Transfer Protocol (HTTP) connections can be secured with Secure Socket Layer/Transport Layer Security (SSL/TLS). Doing so makes those connections much more secure, by encrypting the authentication handshake as well as the message contents.

REF: For more information about using SSL/TLS (on Windows or Linux systems), contact the Help Desk and file a support ticket with the HWSC Product Support team.

In addition, HWSC supports two authentication mechanisms:

- HTTP Basic authentication
- Certificate-based authentication

As of Patch XOBW*1.0*4, HWSC enabled the use of Secure Socket Layer/Transport Layer Security (SSL/TLS) encryption and certificate-based authentication on OpenVMS systems. This allows VistA applications to make Hypertext Transfer Protocol (Secure) HTTP(S) connections from VistA to remote HTTP(S) servers. HWSC uses a Caché library that makes HTTP or HTTPS requests.

REF: For more information and configuration setup instructions, see the *HWSC 1.0 Patch XOBW*1.0*4 Security Configuration Guide*.

3.2 Securing a Web Service Using HTTP Basic Authentication

To use HTTP Basic Authentication (with or without SSL/TLS), some setup is required. The J2EE and M server administrators need to perform the necessary steps for HTTP Basic Authentication:

REF: For more information and configuration setup instructions, see the *HWSC 1.0 Patch XOBW*1.0*4 Security Configuration Guide*.

4 Troubleshooting

4.1 HWSC Availability Checking

The Web Server Manager "Check Availability" action is probably the best tool to troubleshoot Web services problems. It is one way to test the validity of the Web server and Web service configuration; it lists all authorized/assigned Web services and tests each one individually.

REF: For more information, see the "Supporting HWSC Availability Checking" section in the *HWSC 1.0 Developer's Guide*.

Assuming that the Web service entry is set up with a valid "Availability Resource," this option connects to the URL composed of the server, port, context root, and availability resource, and reports any errors encountered. Some examples are provided below for a few error types.

Figure 4: Unsuccessful Availability Check—Listener Down

```
Web Service Availability      May 18, 2007@11:46:56        Page:    1 of    1
Web Server:
 4    *VHAISXSYSA                vhaisxsysa:7111

  1  Unable to retrieve '?wsdl' for XOBT TESTER WEB SERVICE
    o  ERROR #6059: Unable to open TCP/IP socket to server vhaisxsysa:7111
```

Figure 5: Unsuccessful Availability Check—Authorization Failure; HTTP Error Code 401

```
Web Service Availability      May 18, 2007@11:54:12        Page:    1 of    1
Web Server:
 13   * VHAISXSYSB                vhaisxsysb:7111

  1  Unable to retrieve '?wsdl' for XOBT TESTER WEB SERVICE
    o  HTTP Response Status Code: 401
  2  Unable to retrieve '/available' for XOBT TESTER REST SERVICE
    o  HTTP Response Status Code: 401
```

Figure 6: Successful Availability Check

```
Web Service Availability      May 18, 2007@11:49:08        Page:    1 of    1
Web Server:
  5   *VHAISXSYSC                vhaisxsysc:7111

  1  XOBT TESTER WEB SERVICE is available
  2  XOBT TESTER REST SERVICE is available
```

4.2 Runtime Errors Due to Configuration Issues

4.2.1 Caché Error #5005: Cannot Open File

On VMS systems, the VMS accounts used by end-users *must* have RWED access to the directory used by Caché for creating temporary files. Otherwise, calls to external Web services made in a given end-user's process can fail. Runtime errors can be of the form:

- Cannot open file “WREK2XEPAXY.stream”

  Or:

- ERROR #5005: Cannot open file “TRMUAJZVAQT.stream”

Where “WREK2XEPAXY.stream” and “TRMUAJZVAQT.stream” are examples of randomly assigned temporary file names used by Caché.

REF: For detailed information on how to address this issue, see the “Pre-Installation Preparation” section in the *HWSC 1.0 Installation Guide*.

4.2.2 zDelete Errors

On VMS systems, VMS accounts used for end-user processes need to have adequate VMS process parameters (quotas). If VMS process parameters are *not* set to at least the minimum values *recommended* by InterSystems, calls to external Web services made in a given end-user's process can fail. Error messages can include the phrase:

"Error: <*FUNCTION*>zDelete^%ooLibrary.File.1"

This error is usually caused (at a lower level) by a VMS process quota exceeded error.

REF: For detailed information on how to address this issue, see the “Pre-Installation Preparation” section in the *HWSC 1.0 Installation Guide*.

5 Appendix A—HWSC Error Codes

Error code entries are contained in the DIALOG file (#.84). Table 8 lists the dialog entries used in the HWSC software:

Table 8: HWSC Error Codes

Dialog Number	Short Description
186001	(reserved for future use)
186002	Web Server Disabled
186003	Web Service not registered to server
186004	Web Service disabled for Web server
186005	Web Server not defined
186006	Web Service not defined
186007	Web Service is wrong type.
186008	Invalid Server Lookup Key
186009	Server Lookup Key Missing Association

Glossary

Term	Description
AA	Authentication and Authorization.
Business Delegate	A business delegate acts as a representative of the client components and is responsible for hiding the underlying implementation details of the business service. It knows how to look up and access the business services.
Certificate Authority (CA)	"A certificate authority (CA) is an entity that creates and then 'signs' a document or file containing the name of a user and his public key. Anyone can verify that the file was signed by no one other than the CA by using the public key of the CA. By trusting the CA, one can develop trust in a user's public key. The trust in the certification authority's public key can be obtained recursively. One can have a certificate containing the certification authority's public key signed by a superior certification authority (Root CA) that he already trusts. Ultimately, one need only trust the public keys of a small number of top-level certification authorities. Through a chain of certificates (Sub CAs), trust in a large number of users' signatures can be established. A broader application of digital certification includes not only name and public key but also other information. Such a combination, together with a signature, forms an extended certificate. The other information may include, for example, electronic-mail address, authorization to sign documents of a given value, or authorization to sign other certificates."[1] Currently, the Department of Veterans Affairs (VA) uses VeriSign, Inc. as the Certificate Authority (CA).
Cryptography	The system or method used to write or decipher messages in code (see "Encryption" and "Decryption").
CSR	Certificate Signing Request.
Decryption	Using a secret key to unscramble data or messages previously encrypted with a cipher or code so that they are readable. In some cases, encryption algorithms are one directional (i.e., they only encode and the resulting data cannot be unscrambled).
Encryption	Scrambling data or messages with a cipher or code so that they are unreadable without a secret key. In some cases, encryption algorithms are one directional (i.e., they only encode and the resulting data cannot be unscrambled).
HTTP Protocol	Hyper Text Transfer Protocol is the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
HWSC	HealtheVet Web Services Client is a support framework that offers VistA/M applications real-time, synchronous client access to n-tier (J2EE) Web services through the supplied M-based and Caché APIs.

[1] DEA website (http://www.deadiversion.usdoj.gov/ecomm/csos/archive/conops.pdf): "Public Key Infrastructure Analysis Concept of Operations," Section 3.4.3 "Public Key - The I in PKI."

Term	Description
Intermediate CA	Intermediate Certificate Authority. Currently, the Department of Veterans Affairs (VA) uses VeriSign, Inc. as the Certificate Authority (CA). VeriSign requires the use of a CA Intermediate Certificate. The CA Intermediate Certificate is used to sign the peer's (server) certificate. This provides another level of validation-managed PKI for SSL.
J2EE	The Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing multi-tier enterprise applications. J2EE defines components that function independently, that can be deployed on servers, and that can be invoked by remote clients. The J2EE platform is a set of standard technologies and is not itself a language. The current J2EE platform is version 1.4.
PKI	Public Key Infrastructure technology adds the following security services to an electronic ordering system: • Confidentiality—only authorized persons have access to data. • Authentication—establishes who is sending/receiving data. • Integrity—the data has not been altered in transmission. • *Non*-repudiation—parties to a transaction cannot convincingly deny having participated in the transaction.[2]
Private Certificate	This is the certificate that contains both the user's public and private keys. This certificate resides on a smart card.
Public Certificate	This is the certificate that contains the user's public key. This certificate resides in a file or database.
REST	Representational State Transfer (REST) is an architectural style for simplified Web services, based on accessing resources via HTTP.
Root CA	Root Certificate Authority. In cryptography and computer security, a root certificate is an unsigned public key certificate, or a self-signed certificate, and is part of a public key infrastructure scheme. The most common commercial variety is based on the ITU-T X.509 standard. Normally an X.509 certificate includes a digital signature from a Certificate Authority (CA), which vouches for correctness of the data contained in a certificate. Root certificates are implicitly trusted. Currently, the Department of Veterans Affairs (VA) uses VeriSign, Inc. as the Certificate Authority (CA).
Service Facade	The Service Façade acts as the server-side bridge between the Business Delegate and the capability. The Service Façade is responsible for taking a request from the delegate and doing any translation necessary to invoke the capability and provide the response to the delegate.
Servlet Container	A servlet is managed by a servlet container (formerly referred to as servlet engine.) The servlet container is responsible for loading and instantiating the servlets and then calling init(). When a request is received by the servlet container, it decides what servlet to call in accordance with a configuration file. A famous example of a servlet container is Tomcat. The servlet Container calls the servlet's service() method and passes an instance of ServletRequest and ServletResponse. Depending on the

[2] DEA Web site (http://www.deadiversion.usdoj.gov/ecomm/csos/archive/conops.pdf): "Public Key Infrastructure Analysis Concept of Operations," Section 3.3 "Security."

Term	Description
	request's method (mostly GET and POST), service calls doGet() or doPost(). These passed instances can be used by the servlet to find out who the remote user is, if and what HTTP POST parameters have been set and other characteristics. Together with the Web server (or application server) the servlet container provides the HTTP interface to the world. It is also possible for a servlet container to run standalone (without Web server), or to even run on a host other than the Web server. [3]
Signed Certificate	The Signed Certificate (a.k.a. self-signed certificate) is the peer's (server) digital certificate. Currently, the Department of Veterans Affairs (VA) uses VeriSign, Inc. as the Certificate Authority (CA) to sign (validate) digital certificates. VeriSign, Inc. requires the use of CA Root and Intermediate Certificates. The Subject and Issuer have the same content when signed by VeriSign; the issuer has VeriSign's content.
SOAP	Simple Object Access Protocol (SOAP) is a protocol for exchanging structured information over a network, often via HTTP.
SSL	Secure Socket Layer. A low-level protocol that enables secure communications between a server and a browser. It provides communication privacy.
TLS	Transport Layer Security. Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols which provide secure communications on the Internet for such things as Web browsing, e-mail, Internet faxing, instant messaging and other data transfers. There are slight differences between SSL 3.0 and TLS 1.0, but the protocol remains substantially the same.
Web Service	A Web resource meant to be consumed over a network via HTTP, by an autonomous program.
WebLogic	An Oracle® product; WebLogic Server 8.1, 9.2, and 10.x is a J2EE- v1.3-certified application server for developing and deploying J2EE enterprise applications.
WSDL	Web Services Definition Language. “WSDL is an XML-based service description on how to communicate using Web services. The WSDL defines services as collections of network endpoints, or ports. WSDL specification provides an XML format for documents for this purpose. The abstract definition of ports and messages is separated from their concrete use or instance, allowing the reuse of these definitions. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Messages are abstract descriptions of the data being exchanged, and port types are abstract collections of supported operations. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding, where the messages and operations are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the Web service. WSDL is often used in combination with SOAP and XML Schema to provide Web services over the Internet. A client program connecting to a

[3] From the *ADP –Analyse, Design & Programing GmbH* website: http://www.adp-gmbh.ch/java/servlets/container.html

Term	Description
	Web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL."[4]

REF: For a list of commonly used terms and definitions, see the OI&T Master Glossary VA Intranet Website.

For a list of commonly used acronyms, see the VA Acronym Lookup Intranet Website.

[4] http://en.wikipedia.org/wiki/Web_Services_Description_Language