



## **VistAWeb**

*Version 8.0*  
*(Patch WEBV\*1\*13)*

---

# **CPRS Access and Server Installation Guide**

November 2008

Health Provider Systems  
Office of Information and Technology  
Department of Veterans Affairs

## Revision History

Date	Patch	Page(s)	Change(s)	Project Manager	Technical Writer
10/30/08	8.0	<a href="#">36</a>	Updated screen capture	Ed Ridley	JoAnn Green
10/8/08	8.0	throughout	Updated URLs, scripts, etc.	Ed Ridley	JoAnn Green
June 30, 2008	8.0	<a href="#">25</a>	Added Index Creation scripts to Appendix A	Ed Ridley	JoAnn Green
May 22, 2008	7.0/7.5	<a href="#">10</a>	Added VistA Security to Server Installation Section	Scott Madsen	JoAnn Green
May 19, 2008	7.0/7.5	<a href="#">26-30</a>	Updated Appendix B	Scott Madsen	JoAnn Green
December 3, 2007	7.0		Minor changes made by Steve Monson.	Scott Madsen	Albert Fretz
November 28, 2007	n/a	12, 6-9	Added Setup Script section taken from Technical Guide. Incorporated suggested changes from Vivian Parsons	Scott Madsen	Albert Fretz
November 21, 2007	n/a	1-3, 6-12	Minor update changes to reflect patch changes.	Scott Madsen (changes from Steve Monson)	Albert Fretz
October 18, 2007	n/a	6-8	Added section BSE Code Update.	Scott Madsen (from Steve Monson)	Albert Fretz
October 16, 2007	n/a	9-22	Updates and additions to the SQL files: UserAuth_create.sql.txt, CprsUser_create.sql.txt, Log_create.sql.txt, LoggerTable_create.sql.txt, Requests_create.sql.txt, SpecialUsers_create.sql.txt, and updateCprsUsers.sql.txt	Scott Madsen	Albert Fretz
February 2007	n/a	<a href="#">p. 1</a> <a href="#">p. 2</a>	Added note about internal VA network links. Repeated note about internal VA network links.	S. Madsen	M. Kelsey
November 2006	WEBV*1*8 VistAWeb 6.0	n/a <a href="#">p. 1</a> <a href="#">p. 5</a>	Made slight organizational changes to improve document flow. Modified intro to better reflect functionality and access to VistAWeb. Updated link to CPRS manual at new VDL URL. Removed Installation Instructions for Pre-release Testing because they were inaccurate and/or misleading, as well as unsuitable for public	S. Madsen	M. Kelsey

Date	Patch	Page(s)	Change(s)	Project Manager	Technical Writer
			distribution through the VDL. Changed VA Service Desk and e-mail address.		
		<a href="#">p. 10</a>			
7/11/06	WEBV*1*7 VistAWeb V5	1, 3  9  12	Added description of additional method of using VistAWeb from Remote Data Available button in CPRS.  Removed references to the VistAWebDocs application, which is no longer included in the EMR.zip file. Added two new scripts to Appendix A.	S. Madsen	M. Kelsey
4/6/06	WEBV*1*6	None	For software release 4.5.1, minor software changes were made to facilitate the processing and display of data.	S. Madsen	R. Merrill
3/20/06	WEBV*1*5	2, 9	Supplemental information on pop-ups was added and the VistAWeb contact information was changed to the National Help Desk. The <i>Managing Future Updates</i> section was removed— a new process for updates following EVS standard operating procedures is being instituted.	S. Madsen	M. Kelsey
9/28/05	WEBV*1*4	Multiple	Accepted previous changes and made minor edits.	G. Smith	J. Green
8/22/05	WEBV*1*4	2, 4	Added new guidance on test accounts and set up instructions for pre-release testing.	G. Smith	M. Kelsey
5/16/05	n/a	Multiple	Changed Title and other wording to indicate that CPRS access to VistAWeb is done through a URL link rather than local installation. Removed reference to Special Users DB and Requests DB.	G. Smith	M. Kelsey
5/4/05	n/a	All	Added <i>Known Constraints</i> , replaced URL references, and made minor edits and format changes.	G. Smith	M. Kelsey
2/23/05	n/a	All	Revised flow		
2/18/05	n/a	5	Fixed date, removed URL (VISN CIO will provide URL)		
2/10/05	n/a	1, 9	Removed URL referenced on Page 1; Removed reference to Special User scripts on page 9		
1/31/05	Informational Patch number OR*3*230	All	Initial Installation Guide for use with beta test version		



# Table of Contents

<b>REVISION HISTORY .....</b>	<b>II</b>
<b>TABLE OF CONTENTS .....</b>	<b>V</b>
<b>TABLE OF FIGURES .....</b>	<b>VI</b>
<b>VISTAWEB .....</b>	<b>1</b>
INTRODUCTION.....	1
ASSUMPTIONS.....	1
KNOWN CONSTRAINTS .....	2
ACCESS TO VISTAWEB FROM CPRS REMOTE DATA AVAILABLE BUTTON.....	3
PROVIDING ACCESS TO VISTAWEB FROM THE TOOLS MENU .....	3
Tools Menu – Example Configuration.....	4
<b>VISTAWEB INSTALLATION ON AN APPLICATION SERVER AND DATABASE SERVER ....</b>	<b>5</b>
SYSTEM REQUIREMENTS – HARDWARE.....	5
Components that Apply to both Web and Database Servers .....	5
Web Server Components .....	5
Database Server Components .....	6
SYSTEM REQUIREMENTS – SOFTWARE .....	6
Application Configurations and Settings .....	6
BROKER SECURITY ENHANCEMENT CODE UPDATE CONFIGURATION STEPS.....	8
One Time Only:.....	8
<b>SERVER INSTALLATION .....</b>	<b>9</b>
WEB APPLICATION INSTRUCTIONS .....	9
SQL SERVER DATABASE SCHEMA INSTRUCTIONS.....	10
VISTA SECURITY.....	10
Patient Selection Restriction Setup .....	11
Tab Access Restriction Setup.....	11
Both Patient Selection and Tab Access Restrictions.....	12
<b>APPENDIX A: DATABASE SCHEMA .....</b>	<b>13</b>
LOG CREATION SCRIPT .....	14
CPRSUSERS CREATION SCRIPT.....	14
SPECIALUSERS CREATION SCRIPT .....	14
REQUESTS CREATION SCRIPT .....	14
LOGDESC VIEW CREATION SCRIPT .....	14
LOGGERTABLE CREATION SCRIPT .....	15
USERAUTH CREATION SCRIPT.....	15
UPDATE CPRS SCRIPT .....	15
<b>APPENDIX B: CONFIGURATION BY A SCRIPT.....</b>	<b>26</b>
SETUP SCRIPT .....	26
PROPERTIES FILE.....	31
EXAMPLE .....	33

## Table of Figures

Figure 1: Setting Internet Explorer to Allow Pop-ups .....	2
Figure 2: VistAWeb from CPRS Remote Data Available Button.....	3
Figure 3: Web Service Extension Settings.....	7
Figure 4: Web Application Settings.....	7

# VistAWeb

## Introduction

Veterans Health Information Systems and Technology Architecture (VistA) VistAWeb is an intranet web application used to review remote patient information found in VistA, the Federal Health Information Exchange (FHIE) system, and the Health Data Repository (HDR) databases. To a large extent, VistAWeb mirrors the reports behavior of the Computerized Patient Record System (CPRS) and Remote Data View (RDV). However, by permitting a more robust and timely retrieval of remote-site patient data, VistAWeb is also an enhancement to CPRS/RDV.

There are three ways to access VistAWeb. VistAWeb can be made available by adding it to the CPRS Tools Menu, and it can be selected as the default method of retrieving data from the Remote Data Available button in CPRS. These two methods are referred to as CPRS-*spawned* versions of VistAWeb. They are compliant with the Health Level 7 (HL7) Clinical Context Object Workgroup (CCOW) standards and therefore maintain context with the patient selected in CPRS. As a third option, VistAWeb can be accessed in a standalone mode by entering the uniform resource locator (URL) link (<https://vistaweb.med.va.gov/>) in the Internet Explorer address bar.

**Note:** *Some links found in this installation guide go to sites or pages found on the VA intranet. These sites or pages are not accessible from outside the VA network.*

The standalone version of VistAWeb is connected to neither CPRS nor the clinical context management application. Standalone VistAWeb serves an important function for users who have been granted special access to multiple sites, such as for National Programs, Veterans Administration (VA) researchers, and others. VistAWeb was also made available more broadly, though temporarily, to assist clinical staff with the retrieval of patient information from the sites affected by damage caused by hurricane Katrina.

Refer to the VistAWeb User Manual for a detailed description on access and use of VistAWeb from CPRS and as a standalone application process.

## Assumptions

This installation guide is intended for system administrators (specifically, web administrators) who are assumed to possess the technical knowledge of how to configure and interact with application servers. This document also assumes the necessary security hardening guidelines have already been implemented. (Refer to the Office of Cyber and Information Security link below for information pertaining to security requirements.)

[https://vaww.infoprotection.va.gov/portal/server.pt?open=17&objID=4283&DirMode=1&parentname=Dir&parentid=3&mode=2&in\\_hi\\_userid=2&cached=true](https://vaww.infoprotection.va.gov/portal/server.pt?open=17&objID=4283&DirMode=1&parentname=Dir&parentid=3&mode=2&in_hi_userid=2&cached=true)

VistAWeb is not installed at each local site; it is installed on an application server. A link to the application may be incorporated into the existing CPRS Tools Menu at the local site. The instructions provided in this guide identify the required configuration settings for VistAWeb use from the CPRS Tools Menu. Additional reference material may be viewed in the CPRS GUI Technical Manual by selecting the following link:

[http://www.va.gov/vdl/documents/Clinical/Comp\\_Patient\\_Recrd\\_Sys\\_\(CPRS\)/cprsguitem.pdf](http://www.va.gov/vdl/documents/Clinical/Comp_Patient_Recrd_Sys_(CPRS)/cprsguitem.pdf)

## Known Constraints

There are known constraints in the installation and use of VistAWeb:

1. VistAWeb is a CCOW-compliant application. If VistAWeb is launched from CPRS on a PC without the CCOW-compliant Vergence Desktop Components installed, a message will be displayed saying “VistAWeb is CCOW compliant and has been unsuccessful in locating a CCOW vault. Please contact your local IRM for assistance.” VistAWeb will then exit.

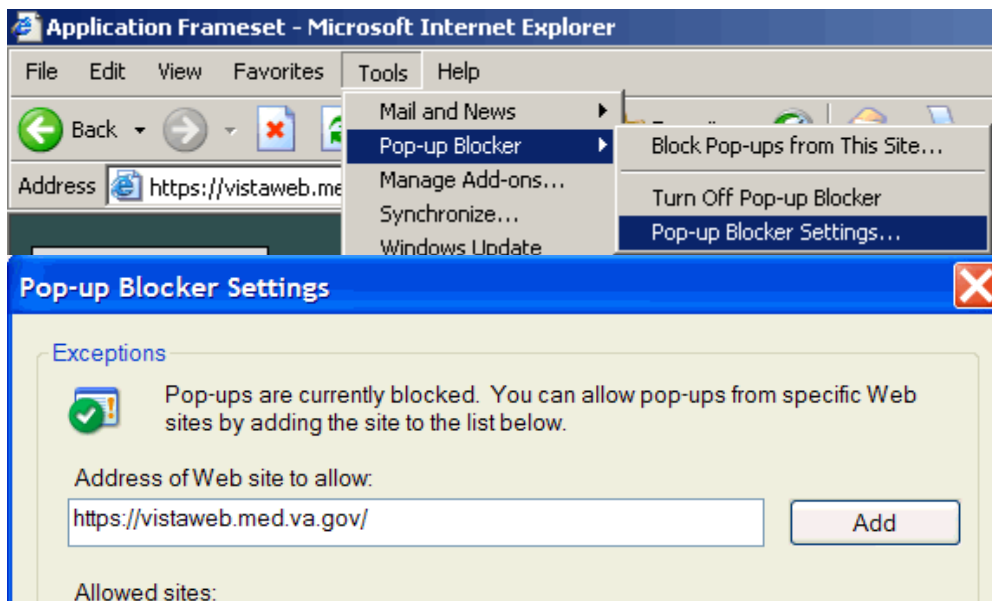
**Note:** *IRM staff may refer to the VistAWeb Informational Patch OR\*3\*230 for guidance on where to find information and who to talk to about installation and configuration of the CCOW Desktop Components.*

2. Access to VistAWeb in a test account should **not** be made available to general users. Access **should** be made available in a production account. Logging in to VistAWeb in test accounts will require the user to enter the IP address and port number of the test system. Access to VistAWeb in a test account should be restricted to IRM staff for limited testing purposes only.

**Note:** *Using VistAWeb to look up test patients may produce confusing results. No two sites ever have the same test patients. Using a test patient in a production account may seem to work okay, but can cause VistAWeb to error out as it attempts to reconcile a test patient at multiple sites.*

3. VistAWeb uses pop-ups. Field facilities that have chosen to turn off pop-ups on desktops will need to allow them for VistAWeb. In *Internet Explorer* in the *Tools* menu pull-down, select *Pop-up Blocker > Pop-up Blocker Settings*, type the VistAWeb URL in the “Address of Web site to allow:” box, and click the “Add” button (Figure 1).

**Figure 1: Setting Internet Explorer to Allow Pop-ups**



**Note:** *Some links found in this installation guide go to sites or pages found on the VA intranet. These sites or pages are not accessible from outside the VA network.*



## Access to VistAWeb from CPRS Remote Data Available Button

A VistAWeb button is now available next to the Remote Data Available button (Figure 2), and CPRS will launch VistAWeb for you. Additionally, when VistAWeb is launched by CPRS, patient context is maintained. This means that VistAWeb will change patients whenever you do a patient selection in CPRS.

Figure 2: VistAWeb from CPRS Remote Data Available Button

The screenshot shows a CPRS interface with a menu bar (File, Edit, View, Tools, Help) and a patient information section. The patient information includes a name (Sep 30, 1938 (70)), a status (Visit Not Selected - Current Provider Not Selected), and a Primary Care Team Unassigned. A 'Flag' button is visible, along with a 'VistaWeb' button and a 'Remote Data' button. The 'VistaWeb' button is highlighted. Below the patient information, there are sections for 'Active Problems' (No Problems Found), 'Allergies / Adverse Reactions' (Donepezil, Warfarin), 'Postings' (Allergies), 'Active Medications' (Non-VA Warfarin 2mg Tab (lavender) Active), and 'Clinical Reminders' (No reminders due).

## Providing Access to VistAWeb from the Tools Menu

A site may use the Tools Menu to give users access to other client software from within CPRS. The parameter, ORWT TOOLS MENU, is used to set up the list of software that appears on the menu. This parameter may be set up for the site, then overridden as appropriate at the division, service, and user levels.

Each item entered on the Tools Menu should have the form:

NAME=COMMAND

NAME represents what the user will see on the menu for that line item. An ampersand “&” may also be used in front of a letter to allow keyboard access to the menu item.

COMMAND may be an entry that is executable by Windows. It may be any file that has a Windows file association.

*For example:*

Name=Command: **&CPRSInfo=http://vista.med.va.gov/cprs/index.html**

*For VistAWeb:*

Name=Command:

**VistaWeb="https://vistaweb.med.va.gov/toolspage.aspx?q9gtw0=<StationID>&xqi4z=%DFN&yiicf=%DUZ"**, where the < > symbols are removed and "StationID" is replaced by the user's actual station ID. (A listing of station IDs can be found on the Facilities Locator & Leadership Directory page at [http://vaww1.va.gov/directory/guide/rpt\\_fac\\_list.cfm](http://vaww1.va.gov/directory/guide/rpt_fac_list.cfm).)

In the following example extracted from the CPRS GUI Technical Manual, note that **CPRSInfo** did not require an executable file to be identified. Since Windows understands hypertext transfer protocol (HTTP), it will launch the workstation's default browser and navigate to the address. Also note the quotation marks in the VistA Terminal (VT) example. A path that contains space characters (like C:\Program Files\...) must be surrounded by quotation marks. Entries on the command line may also contain parameters. The LOCALVAMC is the name of a KEA! session, which is passed as a command line parameter.

## Tools Menu – Example Configuration

```
Select General Parameter Tools Option:  ep  Edit Parameter Values
                                         --- Edit Parameter Values ---
Select PARAMETER DEFINITION NAME:  orwt TOOLS MENU    CPRS GUI Tools
MenuORWT TOOLS MENU may be set for the following:
    1  User          USR    [choose from NEW PERSON]
    2  Location      LOC    [choose from HOSPITAL LOCATION]
    2.5 Service      SRV    [choose from SERVICE/SECTION]
    3  Division      DIV    [REGION 5]
    4  System        SYS    [OEC.ISC-SLC.VA.GOV]
Enter selection: 1  User    NEW PERSON
Select NEW PERSON NAME:  CPRSPROVIDER,TEN            TC

----- Setting ORWT TOOLS MENU for User: CPRSPROVIDER,TEN -----
Select Sequence: 1
Are you adding 1 as a new Sequence? Yes//    YES
Sequence: 1//    1
Name=Command:  &Notepad=Notepad.exe
Select Sequence: 2
Are you adding 2 as a new Sequence? Yes//    YES
Sequence: 2//    2
Name=Command:  &CPRSInfo=http://vista.med.va.gov/cprs/index.html
Select Sequence: 3
Are you adding 3 as a new Sequence? Yes//    YES
Sequence: 3//    3
Name=Command:  &Vista="C:\Program Files\Attachmate\KEA! VT\keavt.exe" LOCALVAMC
Select Sequence:
```

It is also possible to pass context-sensitive parameters. These are parameters that are entered as placeholders and then converted to the appropriate values at runtime. These placeholder parameters are:

%SRV            = Server name for the current broker connection

%PORT	= Port number for the current broker connection
%MREF	= M code giving the global reference where the patient DFN is stored
%DFN	= The actual DFN of the currently selected patient
%DUZ	= Internal entry number of the current user

So, if you have another application that needs to know, for example, the identity of the current user and currently selected patient, you could list %DUZ and %DFN as parameters in the command that executes that program.

## VistAWeb Installation on an Application Server and Database Server

The remainder of the installation guide describes the one-time installation of the VistAWeb application on a sole application server cluster and the one-time installation of the SQL server database on a database server cluster. It is divided into the following four sections:

1. System Requirements – Hardware
  - Components that apply to both web and database servers
  - Web server components
  - Database server components
2. System Requirements – Software
  - Application configurations and settings
3. Installation Instructions—instructions both for the web application server and SQL server database.
4. Appendix A: Database Schema—database specifications (written for SQL Server 2000)

### **System Requirements – Hardware**

The servers that run VistAWeb are configured in Silver Spring, Maryland. The basic components for the web servers and the database servers are listed below.

### **Components that Apply to both Web and Database Servers**

- Dell PowerEdge 4210 Rack with KVM (16-port switch)
- Two SAN adapter cards, and Emulex FC HBA with Power Path licenses

### **Web Server Components**

- Two Dell PowerEdge 6650s
- Dual-CPU 2.2 GHz processors
- 8 GB RAM
- Five 36-GB SCSI hard drives
- Dual network interface cards

## Database Server Components

- Two Dell PowerEdge 6650s
- Quad-CPU 2.2 GHz processors
- 8 GB RAM
- Five 36-GB SCSI hard drives
- Dual network interface cards

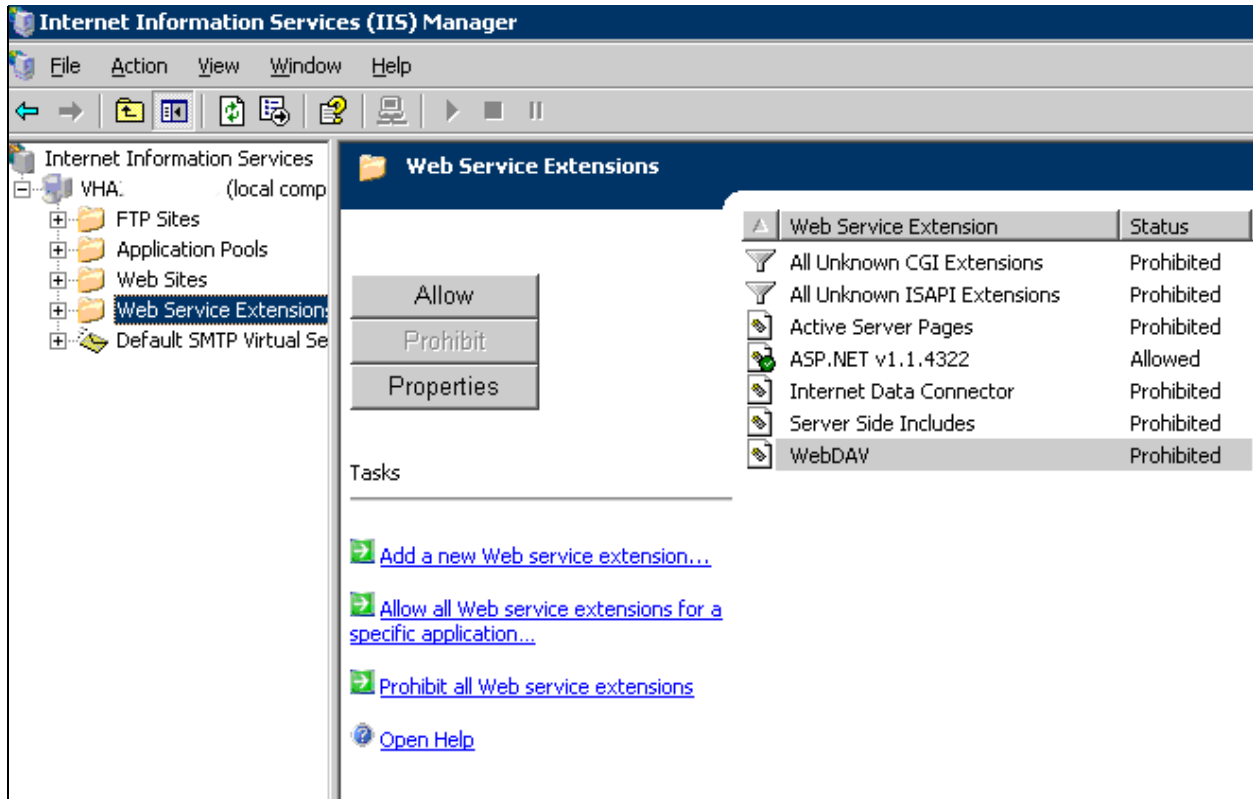
## System Requirements – Software

The basic web application server and database server software configurations are listed below.

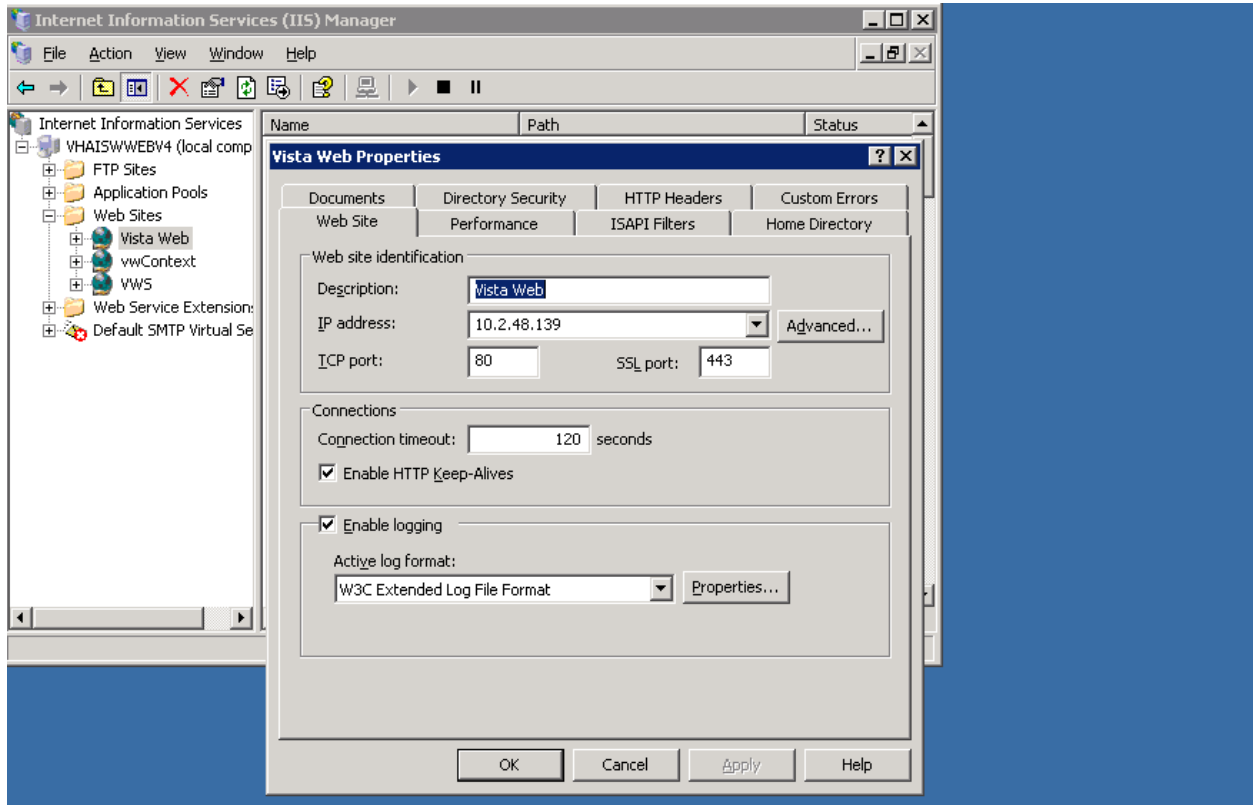
## Application Configurations and Settings

- Windows Server 2003 Enterprise, configured with the role of application server
- Internet Information Services (IIS) 6.0 (installed by default as part of the Application Server role)
- Microsoft Visual J#.NET 2003 runtime component
- .NET Framework 1.1 (part of the Windows Server 2003 operating system default installation)
- .NET Framework 1.1 is installed by default on Windows 2003 systems. Service packs and updates to all three components are available through Microsoft Windows update (<http://windowsupdate.microsoft.com>).
- Web Extension Services set to allow ASP.NET extensions (see Figure 3)
- Create web site configured to use DNS Name (vistaweb.med.va.gov) (see Figure 4)
- Configure Vistaweb to use SSL port 443 and use Verisign certificate
- Enable W3c Extended File Log Format
- SQL Server 2000 (The database does not need to run on the same server as the web application.)

**Figure 3: Web Service Extension Settings**



**Figure 4: Web Application Settings**



## **Broker Security Enhancement Code Update Configuration Steps**

This version of VistAWeb incorporates changes that allow it to operate under the VistA Broker security model called the Broker Security Enhancement (BSE). To operate using this model, the following changes need to be made to upgrade an instance of VistAWeb prior to version 7.

### **One Time Only:**

1. Back up the SQL Server database. *This step cannot be skipped!*
2. Create a login in SQL Server to allow *only inserting* into the *LoggerTable* table.
3. Add 10.2.48.11 as an additional IP on the Windows Server. (This can be added in the Network Properties or configured as a virtual IP if using a Microsoft Cluster.)
4. Create a segregated Application pool named vwContext in IIS 6.0. Configure the application pool to run in multiple web gardens.
5. Create a web application for the new BSE context service; name it VWContext. Configure the web application to use 10.2.48.11 and unsecured (non-SSL) port 12181. Point this web application to use the Application Pool vwContext.
  - a. Specify a new MIME type for this application for “\*.do” items to be handled exactly the same way \*.aspx pages are handled.
  - b. This web application will run as an HTTP application (and *not* HTTPS).
6. Run the script entitled “Update CPRS Script” from appendix A against the database in SQL Server; this will add a column to one of the tables and populate the column with the data.

# Server Installation

The installation instructions are divided into two sections—the web application and the SQL server database schema. Note that for production purposes, the installation of both the web application and the SQL server database need only be performed once.

## **Web Application Instructions**

Install the VistAWeb application on the server following steps 1-5 below

- 1 Remove prior versions of VW (and vwContext, if it exists.)
- 2 Unzip vistaweb\_<version>.<date>.zip into the target VW folder.
- 3 Unzip vistawebcontext\_<version>.<date>.zip into the target folder created in step 3 of the one-time BSE configuration mentioned above.
- 4 Change the <vistaweb>/resources/xml/log4net.xml file to add the login ID and password from step 2 above. Strip out the text “Provider=SQLOLEDB” from the connection string.
- 5 Change the <vistawebcontext>/resources/xml/log4net.xml to add the login ID and password from step 2 above. Strip out the text “Provider=SQLOLEDB” from the connection string.
- 6 Change the <vistaweb>/web.config file
  - a. Change userActivity.connectionString to specify the SQL Server database.
  - b. Change version.useFullVersion to “false”.
  - c. Change allowViewLog to “false”.
  - d. Change excludeChemHem to “true”.
- 7 Change the <vistawebcontext>/web.config file to specify the SQL Server database (set it to the same value as what’s in 6.a.)
- 8 For Production purposes, a domain address was established for VistAWeb (<https://vistaweb.med.va.gov/>), with VistAWeb configured as its own website.
- 9 Set the default VistAWeb content page to “loginframeset.htm.”
- 10 Grant “Write” permission to the *NETWORK SERVICE* user on the Windows 2003 server to the folder **~/resources/images/temp**.
- 11 Start the vistawebcontext application pool and the vistaweb web application, then start the vistaweb application pool.
- 12 Start the vistawebcontext web application, then start the vistaweb application.
- 13 Copy vhasites.xml file from the prior VistAWeb instance into <vistaweb>/resources/xml.
- 14 Start <vistawebcontext>
- 15 Start <vistaweb>

The configuration in steps 4 – 7 can be done by using the script and properties file provided in Appendix B.

## SQL Server Database Schema Instructions

A database schema is required for maintaining the VistAWeb log files and identity of sites from which a user can perform patient lookups. Appendix A contains the database specifications (written for SQL Server 2000). For security purposes, the user account and password that is needed by the VistAWeb application will not be included in this document.

1. Create a database called EMR, or something similar.
2. Run the SQL scripts in Appendix A.
3. Add the user account used by VistAWeb (contact the VA Service Desk for information regarding the account/password at 1-888-596-4357 or by e-mail at [VASD@va.gov](mailto:VASD@va.gov).)
4. Grant the provided user account “dbowner” rights to the EMR database.

## VistA Security

VistAWeb can optionally handle VistA security similarly to the way CPRS does, by requiring the user to have CPRS patient selection and/or CPRS GUI tab access in order to gain access to patient reports through VistAWeb. This setup is global in nature; that is, if set up, all users would have to have one or both of these settings in order to be able to login to VistAWeb.

You can see a user’s CPRS tabs and patient selection settings in the user management area of VistA FileMan.

```

Edit an Existing User
NAME: VWUSER,ONE Page 4 of 5
RESTRICT PATIENT SELECTION: NO OE/RR LIST:
CPRS TAB ACCESS:
  Name Description Effective Date Expiration Date
  RPT Reports tab. MAY 21,2008
Exit Save Next Page Refresh
Enter a command or '^' followed by a caption to jump to a specific field.
COMMAND: 1(024,010) Press <PF1>H for help Insert
```



Please note that VistAWeb supports restriction based on a “NO” or a *blank* in the `RESTRICT PATIENT SELECTION` field (it does not currently support “YES”.) It also supports `TAB ACCESS` using either the `COR` or `RPT` entries.

## Patient Selection Restriction Setup

In order to restrict VistAWeb access based on patient selection set to `NO` or *blank*, you would make a change to the `/resources/contexts/pageContext.xml` file to include a message for patient selection not allowed. You do this by locating the `vistaLoginHelper` object within this file, and enter a message string between the `<value>` tags in the `ptSelectionNotAllowedMessage` field. When there is a message in that field, VistAWeb will check to see if the user has CPRS patient selection defined as `NO` or *blank* in the user settings before allowing access. (If there is a `YES` in that field, then VistAWeb access will not be allowed.)

For example; in order for VistAWeb to check the user’s patient selection ability and display an error message, you could make the change in the `pageContext.xml` file as indicated in red below:

```
<object id="vistaLoginHelper"
type="gov.va.med.vistaweb.data.vista.LoginHelper,EMR.net">
  <constructor-arg name="ptSelectionNotAllowedMessage"><value>Patient
selection list flag not set; access denied</value></constructor-arg>
  <constructor-arg
name="cprsTabsNotAssignedMsg"><value></value></constructor-arg>
  <constructor-arg name="hasCoreTabs" value="false"/>
  <constructor-arg name="hasReportTab" value="false"/>
  <constructor-arg name="requireBothTabs" value="false"/>
</object>
```

When users without this access try to use VistAWeb, they would see the message as an error and not be allowed further access.

## Tab Access Restriction Setup

In order to restrict access to VistAWeb based on Tab access entries, you would need to modify the `/resources/contexts/pageContext.xml` file to include a message for Tab access not allowed. You do this by locating the `vistaLoginHelper` object within this file, and enter a message string between the `<value>` tags in the `cprsTabsNotAssignedMsg` field. When there is a message in that field, you must also set the value attribute of `hasCoreTabs` or `hasReportTab` to `true` in order to check for those respective tabs. If both `hasCoreTabs` and `hasReportTab` value attributes are set to `true`, and the `requireBothTabs` value is set to `false`, VistAWeb access is allowed if the user has access to *either* tabs. If the `requireBothTabs` value is set to `true` and the `hasCoreTabs` and `hasReportTab` value attributes are set to `true`, then the user must have *both* tabs in order to use VistAWeb.

If either `hasCoreTabs`’s or `hasReportTab`’s value attribute is set to `false`, then the `requireBothTabs` setting has no effect.

For example, in order for VistAWeb to check the users' tab access, to make sure they have either COR or RPT tabs, you could make the change in the [pageContext.xml](#) file as indicated in red below:

```
<object id="vistaLoginHelper"
type="gov.va.med.vistaweb.data.vista.LoginHelper,EMR.net">
  <constructor-arg
name="ptSelectionNotAllowedMessage"><value></value></constructor-arg>
  <constructor-arg name="cprsTabsNotAssignedMsg"><value>CPRS Tab Usage
Not Defined: Access Denied</value></constructor-arg>
  <constructor-arg name="hasCoreTabs" value="true"/>
  <constructor-arg name="hasReportTab" value="true"/>
  <constructor-arg name="requireBothTabs" value="false"/>
</object>
```

## Both Patient Selection and Tab Access Restrictions

If you specify both restrictions, then both restrictions will be checked before access is allowed.

For example, in order for VistAWeb to check for CPRS Tab access for RPT *and* COR *and* to check for Patient Selection restriction, you could change the [/resources/contexts/pageContext.xml](#) file to the following:

```
<object id="vistaLoginHelper"
type="gov.va.med.vistaweb.data.vista.LoginHelper,EMR.net">
  <constructor-arg name="ptSelectionNotAllowedMessage"><value>Patient
selection list flag not set; access denied</value></constructor-arg>
  <constructor-arg name="cprsTabsNotAssignedMsg"><value>CPRS Tab Usage
Not Defined: Access Denied</value></constructor-arg>
  <constructor-arg name="hasCoreTabs" value="true"/>
  <constructor-arg name="hasReportTab" value="true"/>
  <constructor-arg name="requireBothTabs" value="true"/>
</object>
```

## Appendix A: Database Schema

- Database Name: EMR
  - Database Tables:
    - Log
    - CprsUsers
    - SpecialUsers
    - Requests
    - LoggerTable
    - UserAuth
  - Views:
    - LogDesc

## **Log Creation Script**

```
CREATE TABLE [dbo].[Log] (  
    [id] [numeric](19, 0) IDENTITY (1, 1) NOT NULL ,  
    [requestDate] [datetime] NULL ,  
    [remoteAddr] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [userId] [numeric](19, 0) NULL ,  
    [userName] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [userSitecode] [varchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [userSitename] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [requestPage] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [requestSitecode] [varchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [requestSitename] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [patientID] [numeric](19, 0) NULL ,  
    [patientName] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [patientSensitivity] [tinyint] NULL ,  
    [message] [varchar] (1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL  
)
```

## **CprsUsers Creation Script**

```
CREATE TABLE [dbo].[CprsUsers] (  
    [UserID] [numeric](19, 0) IDENTITY(1,1) NOT NULL ,  
    [Sitecode] [varchar] (3) NOT NULL ,  
    [SiteName] [varchar] (80) ,  
    [DUZ] [varchar] (50) NOT NULL ,  
    [SSN] [varchar] (9) NOT NULL ,  
    [Name] [varchar] (100) NOT NULL  
)
```

## **SpecialUsers Creation Script**

```
CREATE TABLE [dbo].[SpecialUsers] (  
    [RecID] [numeric](19, 0) IDENTITY (1, 1) NOT NULL ,  
    [UserSiteId] [varchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [DUZ] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [UserName] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [Site] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [Reason] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [ActiveDate] [datetime] NULL ,  
    [DeactiveDate] [datetime] NULL  
)
```

## **Requests Creation Script**

```
CREATE TABLE [dbo].[Requests] (  
    [requestID] [numeric](19, 0) IDENTITY (1, 1) NOT NULL ,  
    [userID] [varchar] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,  
    [text] [varchar] (500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL  
)
```

## **LogDesc View Creation Script**

```
CREATE VIEW dbo.LogDesc  
AS  
SELECT      *  
FROM        dbo.Log  
ORDER BY id DESC
```

## **LoggerTable Creation Script**

```
CREATE TABLE [dbo].[LoggerTable] (  
    [Id] [int] IDENTITY (1, 1) NOT NULL,  
    [Date] [datetime] NOT NULL,  
    [Thread] [varchar] (255) NOT NULL,  
    [Level] [varchar] (50) NOT NULL,  
    [Logger] [varchar] (255) NOT NULL,  
    [Message] [varchar] (4000) NOT NULL,  
    [Exception] [varchar] (2000) NULL  
)
```

## **UserAuth Creation Script**

```
CREATE TABLE [dbo].[UserAuth] (  
    [sessionId] [varchar] (80) NOT NULL,  
    [sessionType] [varchar] (20) NOT NULL,  
    [effectiveDate] [datetime] NOT NULL,  
    [inactiveDate] [datetime],  
    [status] [varchar] (10),  
    [name] [varchar] (100) NOT NULL,  
    [userId] [numeric](19, 0) NOT NULL  
)
```

## **Update CPRS Script**

This script is used to upgrade the CprsUsers table when moving from VistAWeb v6.x to v7 or higher.

```
/****** Object: Table [dbo].[CprsUsers]      Script Date: 05/22/2007  
11:52:06 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_PADDING ON  
GO  
alter table [dbo].[CprsUsers] ADD [SiteName] [varchar](100)  
go  
  
update [dbo].[CprsUsers]  
set SiteName='Togus, ME'  
where SiteCode='402';  
  
update [dbo].[CprsUsers]  
set SiteName='White River Junction, VT'  
where SiteCode='405';  
  
update [dbo].[CprsUsers]  
set SiteName='Bedford, MA'  
where SiteCode='518';  
  
update [dbo].[CprsUsers]  
set SiteName='Boston HCS'  
where SiteCode='523';  
  
update [dbo].[CprsUsers]  
set SiteName='Manchester, NH'  
where SiteCode='608';
```

```

update [dbo].[CprsUsers]
set SiteName='Northampton, MA'
where SiteCode='631';

update [dbo].[CprsUsers]
set SiteName='Providence, RI'
where SiteCode='650';

update [dbo].[CprsUsers]
set SiteName='Connecticut HCS'
where SiteCode='689';

update [dbo].[CprsUsers]
set SiteName='Upstate NY HCS'
where SiteCode='528';

update [dbo].[CprsUsers]
set SiteName='Bronx, NY'
where SiteCode='526';

update [dbo].[CprsUsers]
set SiteName='New Jersey HCS'
where SiteCode='561';

update [dbo].[CprsUsers]
set SiteName='Hudson Valley HCS'
where SiteCode='620';

update [dbo].[CprsUsers]
set SiteName='NY HCS'
where SiteCode='630';

update [dbo].[CprsUsers]
set SiteName='Northport, NY'
where SiteCode='632';

update [dbo].[CprsUsers]
set SiteName='Wilmington, DE'
where SiteCode='460';

update [dbo].[CprsUsers]
set SiteName='Altoona, PA'
where SiteCode='503';

update [dbo].[CprsUsers]
set SiteName='Butler, PA'
where SiteCode='529';

update [dbo].[CprsUsers]
set SiteName='Clarksburg, WV'
where SiteCode='540';

update [dbo].[CprsUsers]
set SiteName='Coatesville, PA'
where SiteCode='542';

update [dbo].[CprsUsers]
set SiteName='Erie, PA'

```

```

where SiteCode='562';

update [dbo].[CprsUsers]
set SiteName='Lebanon, PA'
where SiteCode='595';

update [dbo].[CprsUsers]
set SiteName='Philadelphia, PA'
where SiteCode='642';

update [dbo].[CprsUsers]
set SiteName='Pittsburgh HCS'
where SiteCode='646';

update [dbo].[CprsUsers]
set SiteName='Wilkes Barre, PA'
where SiteCode='693';

update [dbo].[CprsUsers]
set SiteName='Maryland HCS'
where SiteCode='512';

update [dbo].[CprsUsers]
set SiteName='Martinsburg, WV'
where SiteCode='613';

update [dbo].[CprsUsers]
set SiteName='Washington, DC'
where SiteCode='688';

update [dbo].[CprsUsers]
set SiteName='Beckley, WV'
where SiteCode='517';

update [dbo].[CprsUsers]
set SiteName='Durham, NC'
where SiteCode='558';

update [dbo].[CprsUsers]
set SiteName='Fayetteville, NC'
where SiteCode='565';

update [dbo].[CprsUsers]
set SiteName='Hampton, VA'
where SiteCode='590';

update [dbo].[CprsUsers]
set SiteName='Asheville, NC'
where SiteCode='637';

update [dbo].[CprsUsers]
set SiteName='Richmond, VA'
where SiteCode='652';

update [dbo].[CprsUsers]
set SiteName='Salem, VA'
where SiteCode='658';

```

```

update [dbo].[CprsUsers]
set SiteName='Salisbury, NC'
where SiteCode='659';

update [dbo].[CprsUsers]
set SiteName='Atlanta, GA'
where SiteCode='508';

update [dbo].[CprsUsers]
set SiteName='Augusta, GA'
where SiteCode='509';

update [dbo].[CprsUsers]
set SiteName='Birmingham, AL'
where SiteCode='521';

update [dbo].[CprsUsers]
set SiteName='Charleston, SC'
where SiteCode='534';

update [dbo].[CprsUsers]
set SiteName='Columbia, SC'
where SiteCode='544';

update [dbo].[CprsUsers]
set SiteName='Dublin, GA'
where SiteCode='557';

update [dbo].[CprsUsers]
set SiteName='Central Alabama HCS'
where SiteCode='619';

update [dbo].[CprsUsers]
set SiteName='Tuscaloosa, AL'
where SiteCode='679';

update [dbo].[CprsUsers]
set SiteName='Bay Pines, FL'
where SiteCode='516';

update [dbo].[CprsUsers]
set SiteName='Bay Pines CIO Test'
where SiteCode='998';

update [dbo].[CprsUsers]
set SiteName='Miami, FL'
where SiteCode='546';

update [dbo].[CprsUsers]
set SiteName='West Palm Beach, FL'
where SiteCode='548';

update [dbo].[CprsUsers]
set SiteName='N. Florida/S. Georgia HCS'
where SiteCode='573';

update [dbo].[CprsUsers]
set SiteName='San Juan, PR'

```



```

where SiteCode='672';

update [dbo].[CprsUsers]
set SiteName='Tampa, FL'
where SiteCode='673';

update [dbo].[CprsUsers]
set SiteName='Huntington, WV'
where SiteCode='581';

update [dbo].[CprsUsers]
set SiteName='Lexington, KY'
where SiteCode='596';

update [dbo].[CprsUsers]
set SiteName='Louisville, KY'
where SiteCode='603';

update [dbo].[CprsUsers]
set SiteName='Mountain Home, TN'
where SiteCode='621';

update [dbo].[CprsUsers]
set SiteName='Tennessee Valley HCS'
where SiteCode='626';

update [dbo].[CprsUsers]
set SiteName='Memphis, TN'
where SiteCode='614';

update [dbo].[CprsUsers]
set SiteName='Chillicothe, OH'
where SiteCode='538';

update [dbo].[CprsUsers]
set SiteName='Cincinnati, OH'
where SiteCode='539';

update [dbo].[CprsUsers]
set SiteName='Dayton, OH'
where SiteCode='552';

update [dbo].[CprsUsers]
set SiteName='Cleveland, OH'
where SiteCode='541';

update [dbo].[CprsUsers]
set SiteName='Columbus, OH'
where SiteCode='757';

update [dbo].[CprsUsers]
set SiteName='Saginaw, MI'
where SiteCode='655';

update [dbo].[CprsUsers]
set SiteName='Battle Creek, MI'
where SiteCode='515';

```

```

update [dbo].[CprsUsers]
set SiteName='Detroit, MI'
where SiteCode='553';

update [dbo].[CprsUsers]
set SiteName='Indianapolis, IN'
where SiteCode='583';

update [dbo].[CprsUsers]
set SiteName='Ann Arbor, MI'
where SiteCode='506';

update [dbo].[CprsUsers]
set SiteName='Danville, IL'
where SiteCode='550';

update [dbo].[CprsUsers]
set SiteName='Northern Indiana HCS'
where SiteCode='610';

update [dbo].[CprsUsers]
set SiteName='Milwaukee, WI'
where SiteCode='695';

update [dbo].[CprsUsers]
set SiteName='Hines, IL'
where SiteCode='578';

update [dbo].[CprsUsers]
set SiteName='Iron Mountain, MI'
where SiteCode='585';

update [dbo].[CprsUsers]
set SiteName='North Chicago, IL'
where SiteCode='556';

update [dbo].[CprsUsers]
set SiteName='Tomah, WI'
where SiteCode='676';

update [dbo].[CprsUsers]
set SiteName='Jesse Brown VAMC'
where SiteCode='537';

update [dbo].[CprsUsers]
set SiteName='Madison, WI'
where SiteCode='607';

update [dbo].[CprsUsers]
set SiteName='Columbia, MO'
where SiteCode='543';

update [dbo].[CprsUsers]
set SiteName='Kansas City, MO'
where SiteCode='589';

update [dbo].[CprsUsers]
set SiteName='St. Louis, MO'

```

```

where SiteCode='657';

update [dbo].[CprsUsers]
set SiteName='Topeka, KS'
where SiteCode='677';

update [dbo].[CprsUsers]
set SiteName='Leavenworth, KS'
where SiteCode='686';

update [dbo].[CprsUsers]
set SiteName='Alexandria, LA'
where SiteCode='502';

update [dbo].[CprsUsers]
set SiteName='Little Rock, AR'
where SiteCode='598';

update [dbo].[CprsUsers]
set SiteName='Fayetteville, AR'
where SiteCode='564';

update [dbo].[CprsUsers]
set SiteName='Jackson, MS'
where SiteCode='586';

update [dbo].[CprsUsers]
set SiteName='Houston, TX'
where SiteCode='580';

update [dbo].[CprsUsers]
set SiteName='Muskogee, OK'
where SiteCode='623';

update [dbo].[CprsUsers]
set SiteName='Oklahoma City, OK'
where SiteCode='635';

update [dbo].[CprsUsers]
set SiteName='Shreveport, LA'
where SiteCode='667';

update [dbo].[CprsUsers]
set SiteName='Biloxi, MS'
where SiteCode='520';

update [dbo].[CprsUsers]
set SiteName='New Orleans, LA'
where SiteCode='629';

update [dbo].[CprsUsers]
set SiteName='Central Texas HCS'
where SiteCode='674';

update [dbo].[CprsUsers]
set SiteName='South Texas HCS'
where SiteCode='671';

```

```

update [dbo].[CprsUsers]
set SiteName='North Texas HCS'
where SiteCode='549';

update [dbo].[CprsUsers]
set SiteName='Amarillo, TX'
where SiteCode='504';

update [dbo].[CprsUsers]
set SiteName='El Paso, TX'
where SiteCode='756';

update [dbo].[CprsUsers]
set SiteName='Albuquerque, NM'
where SiteCode='501';

update [dbo].[CprsUsers]
set SiteName='Prescott, AZ'
where SiteCode='649';

update [dbo].[CprsUsers]
set SiteName='Tucson, AZ'
where SiteCode='678';

update [dbo].[CprsUsers]
set SiteName='Big Spring, TX'
where SiteCode='519';

update [dbo].[CprsUsers]
set SiteName='Phoenix, AZ'
where SiteCode='644';

update [dbo].[CprsUsers]
set SiteName='Denver, CO (HAC)'
where SiteCode='741';

update [dbo].[CprsUsers]
set SiteName='Eastern Colorado HCS'
where SiteCode='554';

update [dbo].[CprsUsers]
set SiteName='Montana HCS'
where SiteCode='436';

update [dbo].[CprsUsers]
set SiteName='Cheyenne, WY'
where SiteCode='442';

update [dbo].[CprsUsers]
set SiteName='Grand Junction, CO'
where SiteCode='575';

update [dbo].[CprsUsers]
set SiteName='Sheridan, WY'
where SiteCode='666';

update [dbo].[CprsUsers]
set SiteName='Salt Lake City, UT'

```

```

where SiteCode='660';

update [dbo].[CprsUsers]
set SiteName='Anchorage, AK'
where SiteCode='463';

update [dbo].[CprsUsers]
set SiteName='Boise, ID'
where SiteCode='531';

update [dbo].[CprsUsers]
set SiteName='Walla Walla, WA'
where SiteCode='687';

update [dbo].[CprsUsers]
set SiteName='Portland, OR'
where SiteCode='648';

update [dbo].[CprsUsers]
set SiteName='Spokane, WA'
where SiteCode='668';

update [dbo].[CprsUsers]
set SiteName='Puget Sound HCS'
where SiteCode='663';

update [dbo].[CprsUsers]
set SiteName='Roseburg, OR'
where SiteCode='653';

update [dbo].[CprsUsers]
set SiteName='White City OR'
where SiteCode='692';

update [dbo].[CprsUsers]
set SiteName='Northern California HCS'
where SiteCode='612';

update [dbo].[CprsUsers]
set SiteName='San Francisco, CA'
where SiteCode='662';

update [dbo].[CprsUsers]
set SiteName='Fresno, CA'
where SiteCode='570';

update [dbo].[CprsUsers]
set SiteName='Honolulu, HI'
where SiteCode='459';

update [dbo].[CprsUsers]
set SiteName='Palo Alto HCS'
where SiteCode='640';

update [dbo].[CprsUsers]
set SiteName='Reno, NV'
where SiteCode='654';

```

```

update [dbo].[CprsUsers]
set SiteName='Manila, PI'
where SiteCode='358';

update [dbo].[CprsUsers]
set SiteName='West Los Angeles, CA'
where SiteCode='691';

update [dbo].[CprsUsers]
set SiteName='Loma Linda, CA'
where SiteCode='605';

update [dbo].[CprsUsers]
set SiteName='Long Beach, CA'
where SiteCode='600';

update [dbo].[CprsUsers]
set SiteName='San Diego, CA'
where SiteCode='664';

update [dbo].[CprsUsers]
set SiteName='Las Vegas, NV'
where SiteCode='593';

update [dbo].[CprsUsers]
set SiteName='Fargo, ND'
where SiteCode='437';

update [dbo].[CprsUsers]
set SiteName='Des Moines, IA'
where SiteCode='555';

update [dbo].[CprsUsers]
set SiteName='Iowa City, IA'
where SiteCode='584';

update [dbo].[CprsUsers]
set SiteName='Minneapolis, MN'
where SiteCode='618';

update [dbo].[CprsUsers]
set SiteName='Central Plains HCS'
where SiteCode='636';

update [dbo].[CprsUsers]
set SiteName='Sioux Falls, SD'
where SiteCode='438';

update [dbo].[CprsUsers]
set SiteName='St. Cloud, MN'
where SiteCode='656';

update [dbo].[CprsUsers]
set SiteName='Black Hills HCS'
where SiteCode='568';

update [dbo].[CprsUsers]
set SiteName='DoD'

```

```
where SiteCode='200';

update [dbo].[CprsUsers]
set SiteName='HDR'
where SiteCode='CDS';

update [dbo].[CprsUsers]
set SiteName='Claims System'
where SiteCode='100';
GO
SET ANSI_PADDING OFF
```

### **Create Index for CPRS Users**

```
CREATE INDEX [CprsUsers] ON [dbo].[CprsUsers]([UserID]) ON [PRIMARY]
```

### **Create index for SpecialUsers**

```
CREATE INDEX [SpecialUsers] ON [dbo].[SpecialUsers]([RecID]) ON [PRIMARY]
```

### **Create index for UserAuth**

```
CREATE INDEX [UserAuth_sessionUser] ON [dbo].[UserAuth]([sessionId],
[userId]) ON [PRIMARY]
```

## Appendix B: Configuration by a Script

### Setup Script

```
if (WScript.Arguments.Length<2) {
    WScript.Stdout.WriteLine("no parameters specified; you must specify 'cscript setupAp.js
<property_file_name> <directory_path>');
    WScript.Quit(1);
}

var fileSystem = new ActiveXObject("Scripting.FileSystemObject");

if (!fileSystem.FileExists(WScript.Arguments.Item(0))) {
    WScript.Stdout.WriteLine("the property file '" + WScript.Arguments.Item(0) + "' could not
be found");
    WScript.Quit(1);
}

var targetDir=WScript.Arguments.Item(1);

var propFile=fileSystem.GetFile(WScript.Arguments.Item(0));

var propStream=propFile.OpenAsTextStream(1, -2);

var allowViewLog="false";
var excludeChemHem="true";
var logStats="true";
var useFullVersion="true";
var securityPhrase="";
var rootAppender="";
var logBufferSize="";
var log4netInternalDebug = "";

var db_server = "someServer";
var db_databaseName = "someDatabase";
var db_username = "";
var db_password = "";

var db_webUsername="";
var db_webPassword="";
var db_webConnectionString1="";
var db_webConnectionString2="";

var db_log4netUsername="";
var db_log4netPassword="";
var db_log4netConnectionString1="";
var db_log4netConnectionString2="";

var db_contextUsername="";
var db_contextPassword="";
var db_contextConnectionString1="";
var db_contextConnectionString2="";

while (!propStream.AtEndOfStream) {
    var line=propStream.ReadLine();
    if (line.substring(0,1)=="#") {
        continue;
    }
    var divIndex=line.indexOf("=");
    if (divIndex<0) {
        WScript.Stdout.WriteLine("***Begin
Message*****");
        WScript.Stdout.WriteLine("    no '=' found; skipping line ==>" + line);
        WScript.Stdout.WriteLine("***End
Message*****");
        continue;
    }
    var leftSide=line.substr(0,divIndex);
    var newVal=line.substr(divIndex+1);
    switch(leftSide) {
```



```

case ("allowViewLog"): {
    allowViewLog = newVal.toLowerCase();
    break;
}
case ("logStats"): {
    logStats = newVal.toLowerCase();
    break;
}
case ("excludeChemHem"): {
    excludeChemHem = newVal.toLowerCase();
    break;
}
case ("useFullVersion"): {
    useFullVersion = newVal.toLowerCase();
    break;
}
case ("securityPhrase"): {
    securityPhrase = newVal;
    break;
}
case ("rootAppender"): {
    rootAppender = newVal;
    break;
}
case ("logBufferSize"): {
    logBufferSize = newVal;
    break;
}
case ("log4netInternalDebug"): {
    log4netInternalDebug = newVal.toLowerCase();
    break;
}
case ("db_server"): {
    db_server = newVal;
    break;
}
case ("db_databaseName"): {
    db_databaseName = newVal;
    break;
}
case ("db_webUsername"): {
    db_webUsername = newVal;
    break;
}
case ("db_webPassword"): {
    db_webPassword = newVal;
    break;
}
case ("db_webConnectionString1"): {
    db_webConnectionString1 = newVal;
    break;
}
case ("db_webConnectionString2"): {
    db_webConnectionString2 = newVal;
    break;
}
case ("db_log4netUsername"): {
    db_log4netUsername = newVal;
    break;
}
case ("db_log4netPassword"): {
    db_log4netPassword = newVal;
    break;
}
case ("db_log4netConnectionString1"): {
    db_log4netConnectionString1 = newVal;
    break;
}
case ("db_log4netConnectionString2"): {
    db_log4netConnectionString2 = newVal;
    break;
}
case ("db_contextUsername"): {

```

```

        db_contextUsername = newVal;
        break;
    }
    case ("db_contextPassword"): {
        db_contextPassword = newVal;
        break;
    }
    case ("db_contextConnectionString1"): {
        db_contextConnectionString1 = newVal;
        break;
    }
    case ("db_contextConnectionString2"): {
        db_contextConnectionString2 = newVal;
        break;
    }
    case ("db_username"): {
        db_username = newVal;
        break;
    }
    case ("db_password"): {
        db_password = newVal;
        break;
    }
    default: {
        WScript.Stdout.WriteLine("***Begin
Message*****");
        WScript.Stdout.WriteLine("Setting for '" + leftSide + "' ignored: " +
line);
        WScript.Stdout.WriteLine("***End
Message*****");
    }
}

}

if (db_webUsername.length<1) db_webUsername = db_username;
if (db_log4netUsername.length<1) db_log4netUsername = db_username;
if (db_contextUsername.length<1) db_contextUsername = db_username;

if (db_webPassword.length<1) db_webPassword = db_password;
if (db_log4netPassword.length<1) db_log4netPassword = db_password;
if (db_contextPassword.length<1) db_contextPassword = db_password;

var log4netDom=getLog4netDom();
var log4netDomChanged = false;
var nodeList = log4netDom.selectNodes("/log4net/root/appender-ref");
if (nodeList.length>0 && rootAppender!=null && rootAppender.length>0) {
    nodeList.item(0).setAttribute("ref", rootAppender);
    WScript.Stdout.WriteLine("set root logger in log4net.xml to '" + rootAppender + "'");
    log4netDomChanged = true;
}

nodeList = log4netDom.selectNodes("/log4net/appender[@name='adoNetAppender']/connectionString");
if (nodeList.length>0) {
    log4netDomChanged = true;
    nodeList.item(0).setAttribute("value",
stuffConnectionString(db_log4netConnectionString1,db_server,db_databaseName,db_log4netUsername,db
_log4netPassword));
    WScript.Stdout.WriteLine("set connection string in log4net.xml");
}

nodeList = log4netDom.selectNodes("/log4net/appender[@name='adoNetAppender']/bufferSize");
if (nodeList.length>0 && logBufferSize!=null && logBufferSize.length>0) {
    log4netDomChanged = true;
    nodeList.item(0).setAttribute("value", logBufferSize);
    WScript.Stdout.WriteLine("set 'logBufferSize' in log4net.xml to " + logBufferSize);
}

nodeList =
log4netDom.selectNodes("/configuration/appSettings/add[key='log4net.Internal.Debug']");
if (nodeList.length>0 && log4netInternalDebug!=null && log4netInternalDebug.length>0) {
    log4netDomChanged = true;
    nodeList.item(0).setAttribute("value", log4netInternalDebug);
}

```

```

        WScript.Stdout.WriteLine("set 'log4netInternalDebug' in log4net.xml to " +
log4netInternalDebug);
    }

    if (log4netDomChanged) {
        var fileName1 = getLog4netFileName();
        fileSystem.DeleteFile(fileName1);
        var newFile1 = fileSystem.OpenTextFile(fileName1, 8, true);
        newFile1.Write(log4netDom.xml);
        newFile1.Close();
    }

    var webConfigDom=getWebConfigDom();
    if (isVistaweb(webConfigDom)) {
        changeWebDom(webConfigDom,"/configuration/appSettings/add[@key='allowViewLog']",allowView
Log);
        changeWebDom(webConfigDom,"/configuration/appSettings/add[@key='excludeChemHem']",exclude
ChemHem);
        changeWebDom(webConfigDom,"/configuration/userActivity/add[@key='userActivity.logStats'
],logStats);
        changeWebDom(webConfigDom,"/configuration/version/add[@key='version.useFullVersion'
],use
FullVersion);
        changeWebDom(webConfigDom,"/configuration/daoConfig/add[@key='vistaConnectionFactory.secu
rityPhrase' or @key='fhieConnectionFactory.securityPhrase']",securityPhrase);
        var connectString1 =
stuffConnectionString(db_webConnectionString1,db_server,db_databaseName,db_webUsername,db_webPass
word);
        var connectString2 =
stuffConnectionString(db_webConnectionString2,db_server,db_databaseName,db_webUsername,db_webPass
word);

        var nodeList =
webConfigDom.selectNodes("/configuration/userActivity/add[@key='userActivity.connectionString'
]);
        if (nodeList.length>0) {
            nodeList.item(0).setAttribute("value", connectString1);
            WScript.Stdout.WriteLine("set 'userActivity.connectionString' in web.config");
        }
        nodeList =
webConfigDom.selectNodes("/configuration/userActivity/add[@key='userActivity.altConnectionString'
]");
        if (nodeList.length>0) {
            nodeList.item(0).setAttribute("value", connectString2);
            WScript.Stdout.WriteLine("set 'userActivity.altConnectionString' in web.config");
        }
    }
    else {
        /** Must be vwContext that we're configuring */
        var connectString1 =
stuffConnectionString(db_contextConnectionString1,db_server,db_databaseName,db_contextUsername,db
_contextPassword);
        var connectString2 =
stuffConnectionString(db_contextConnectionString2,db_server,db_databaseName,db_contextUsername,db
_contextPassword);

        var nodeList =
webConfigDom.selectNodes("/configuration/spring/objects/object[@id='connectionFactory']/construct
or-arg[@name='connectionString' ]");
        if (nodeList.length>0) {
            nodeList.item(0).setAttribute("value", connectString1);
            WScript.Stdout.WriteLine("set 'connectionFactory.connectionString' in
web.config");
        }
        nodeList =
webConfigDom.selectNodes("/configuration/spring/objects/object[@id='connectionFactory']/construct
or-arg[@name='altConnectionString' ]");
        if (nodeList.length>0) {
            nodeList.item(0).setAttribute("value", connectString2);
            WScript.Stdout.WriteLine("set 'connectionFactory.altConnectionString' in
web.config");
        }
    }
}

```

```

nodeList = webConfigDom.selectNodes("/configuration/system.web/compilation[@debug='true']");
if (nodeList.length>0) {
    nodeList.item(0).setAttribute("debug","false");
    WScript.StdOut.WriteLine("set 'compilation.debug' in web.config to 'false'");
}

nodeList = webConfigDom.selectNodes("/configuration/system.web/trace[@enabled='true']");
if (nodeList.length>0) {
    nodeList.item(0).setAttribute("enabled","false");
    WScript.StdOut.WriteLine("set 'trace.enabled' in web.config to 'false'");
}

var fileName2 = getWebConfigName();
fileSystem.DeleteFile(fileName2);
var newFile2 = fileSystem.OpenTextFile(fileName2, 8, true);
newFile2.Write(webConfigDom.xml);
newFile2.Close();

WScript.StdOut.WriteLine("=====");
WScript.StdOut.WriteLine("configuration completed");
WScript.Quit(0);

function getLog4netFileName() {
    return WScript.Arguments.Item(1) + "\\resources\\xml\\log4net.xml";
}

function getWebConfigName() {
    return WScript.Arguments.Item(1) + "\\web.config";
}

function getLog4netDom() {
    var fileName = getLog4netFileName();
    if (!(fileSystem.FileExists(fileName))) {
        WScript.StdOut.WriteLine("log4net.xml file not found in " + fileName);
        return null;
    }
    var xmlDom=new ActiveXObject("MSXML2.DOMDocument.4.0");
    xmlDom.async=false;
    xmlDom.validateOnParse=false;
    xmlDom.resolveExternals=false;
    xmlDom.load(fileName);
    if (xmlDom.parseError.errorCode!=0) {
        WScript.StdOut.WriteLine("could not open log4net.xml");
        return null;
    }
    xmlDom.setProperty("SelectionLanguage","XPath");
    return xmlDom;
}

function getWebConfigDom() {
    var fileName = getWebConfigName();
    if (!(fileSystem.FileExists(fileName))) {
        WScript.StdOut.WriteLine("web.config file not found in " + fileName);
        return null;
    }
    var xmlDom=new ActiveXObject("MSXML2.DOMDocument.4.0");
    xmlDom.async=false;
    xmlDom.validateOnParse=false;
    xmlDom.resolveExternals=false;
    xmlDom.load(fileName);
    if (xmlDom.parseError.errorCode!=0) {
        WScript.StdOut.WriteLine("could not open web.config");
        return null;
    }
    xmlDom.setProperty("SelectionLanguage","XPath");
    return xmlDom;
}

function isVistaweb(xmlDom) {
    var nodeList =
xmlDom.selectNodes("/configuration/userActivity/add[@key='userActivity.logStats']");
    if (nodeList!=null && nodeList.length>0) return true;
    return false;
}

```

```

}

function replaceStringWith(originalString, stringToFind, stringToStuff) {
    var index = originalString.indexOf(stringToFind);
    if (index<0) return originalString;
    return originalString.substring(0,index) + stringToStuff +
originalString.substring(index+stringToFind.length);
}

function stuffConnectionString(connectionString, serverName, dbName, userName, password) {
    return
replaceStringWith(replaceStringWith(replaceStringWith(replaceStringWith(connectionString, "$server
$", serverName), "$database$", dbName), "$username$", userName), "$password$", password);
}

function changeWebDom(xmlDom, xpath, newValue) {
    var nodeList = xmlDom.selectNodes(xpath);
    if (nodeList.length<1) {
        return;
    }
    for(var i=0;i<nodeList.length;i++) {
        nodeList.item(i).setAttribute("value", newValue);
        WScript.StdOut.WriteLine("changed " + (i+1) + " of " + nodeList.length + " for '"
+ xpath + "' in web.config to " + newValue);
    }
}

```

## Properties File

```

excludeChemHem=true
allowViewLog=false
logStats=true
useFullVersion=false
securityPhrase=XXXXXXXXXXXXXXXXXXXX
rootAppender=adoNetAppender
logBufferSize=5
#log4netInternalDebug=false
#
db_server=vhaiswsqlv1.vha.med.va.gov
db_databaseName=EMR
db_username=xxxxxxx
db_password=xxxxxxx
#
# In all connection strings, use the placeholders $server$, $database$, $username$, and
$password$ for each of those items respectively, when needed
#
# Use the other username and password fields to override the general username and password fields
above, for separate usernames
#     and passwords for the different connections, if desired.
#
#db_webUsername=
#db_webPassword=
db_webConnectionString1=Server=$server$;Database=$database$;User
ID=$username$;Password=$password$;persist security info=True;packet size=4096;Connect
Timeout=2;Min Pool Size=5;Max Pool Size=500
#db_webConnectionString2=Server=$server$;Database=$database$;User
ID=$username$;Password=$password$;persist security info=True;packet size=4096;Connect
Timeout=30;Pooling=false
#
#db_log4netUsername=
#db_log4netPassword=
db_log4netConnectionString1=Server=$server$;Database=$database$;integrated security=false;persist
security info=True;User ID=$username$;Password=$password$
#db_log4netConnectionString2=
#
#db_contextUsername=
#db_contextPassword=
db_contextConnectionString1=Server=$server$;Database=$database$;User
ID=$username$;Password=$password$;Min Pool Size=5;Max Pool Size=500;Pooling=true;Connect
Timeout=2;persist security info=True;
db_contextConnectionString2=Server=$server$;Database=$database$;User
ID=$username$;Password=$password$;Pooling=false;Connect Timeout=30;persist security info=True;

```

### One-time Setup

- 1) Save the above setup script as **setupAp.js**, and save the properties file as **vistaweb.production.properties**
- 2) Enter the DNS name of your database server, the database name, the application user name and password for your VistAWeb instance in the db\_server, db\_databaseName, db\_username, and db\_password fields respectively.
- 3) Save the file.

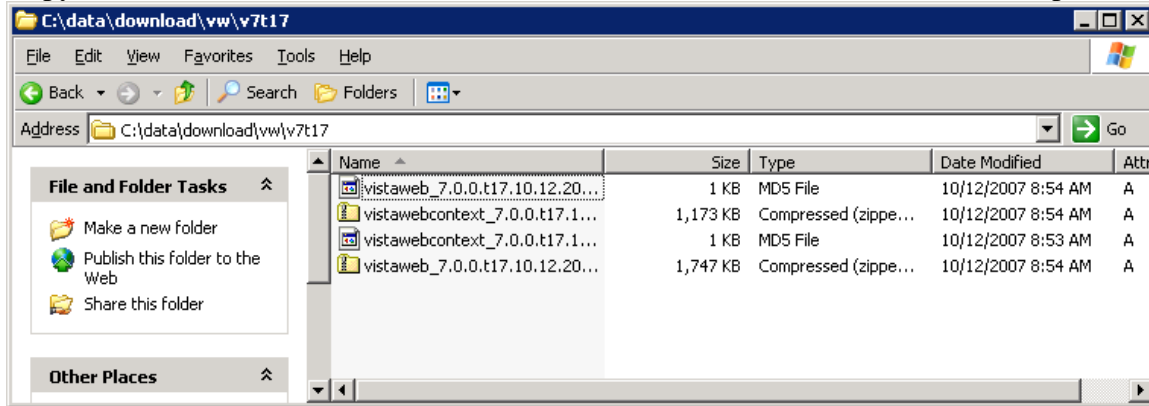
For example, if your server was named VHAISLVIS1.med.va.gov, your database was named VW\_MAIN, and the application user name and password were vwApp and E+I9345890s, the properties file entry would look like this:

```
db_server=VHAISLVIS1.med.va.gov
db_databaseName=VW_MAIN
db_username=vwApp
db_password=E+I9345890s
```

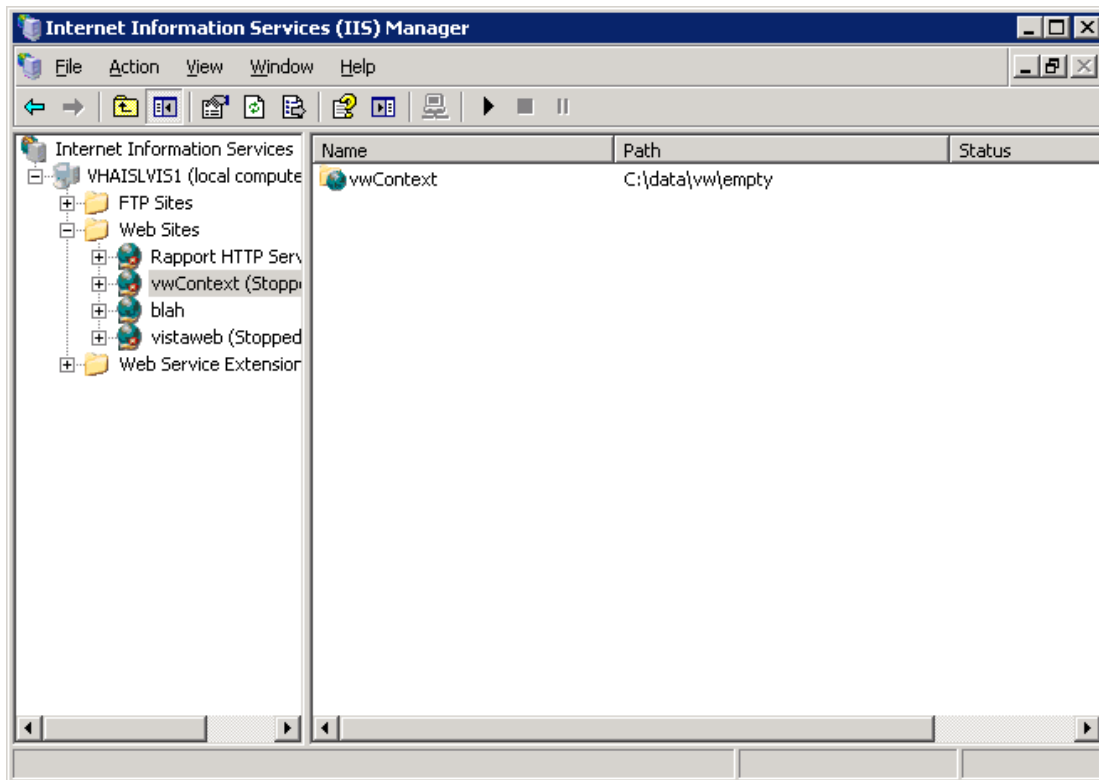
## Example

The example below shows how configuration via the script can be done on a development server. Even though the example screenshots and output below say “v7”, the same procedure applies to “v8”.

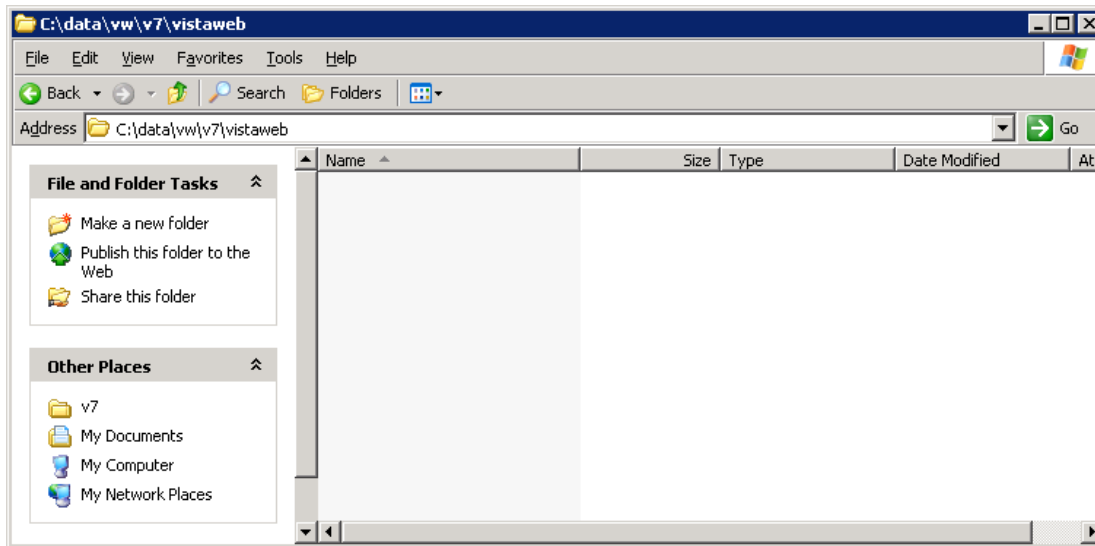
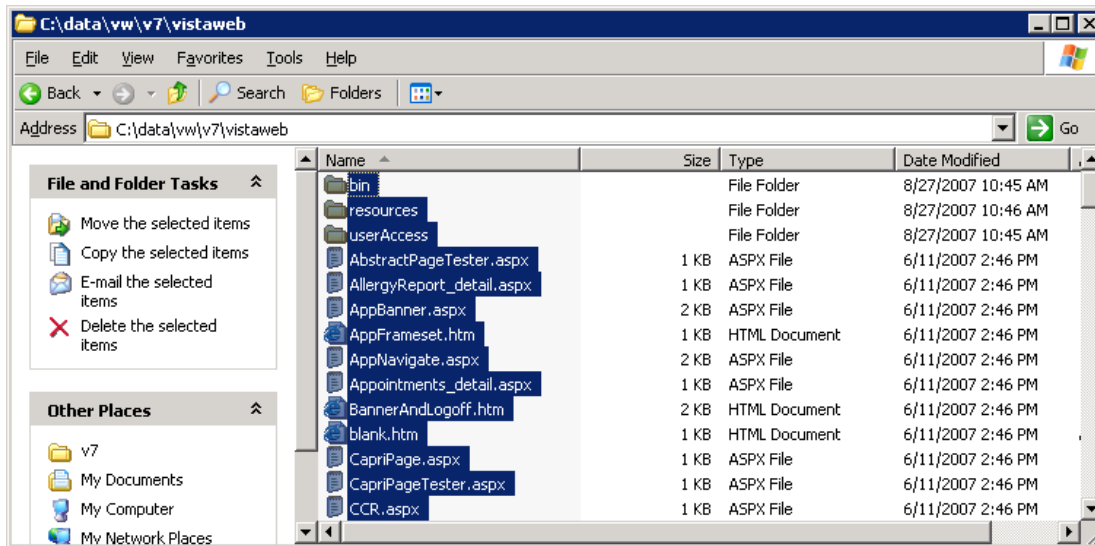
Copy the build files to the `c:\data\download\vw\v7t17` folder on the development server:



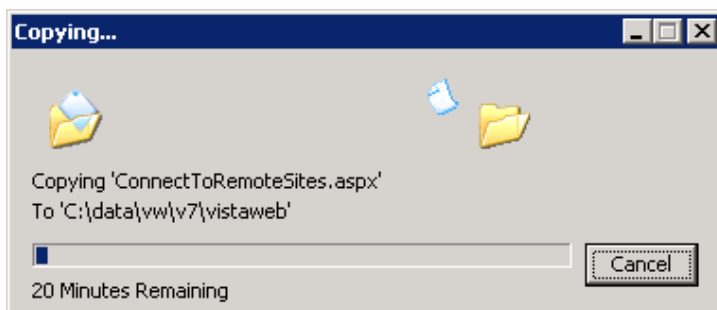
Right click on the `\vistaweb` and `\vistawebcontext` virtual folders and select **Stop** from the menu (in that order).



Highlight all the files in the program folder for `\vistaweb` and `\vistawebcontext` and press <Delete>.



Open the zip file for \vistaweb and \vistacontext and copy each one to their respective program folders.



Open a command window.

Navigate to where the `setupAp.js` and `vistaWeb.dev.properties` files are located and run the script for VistAWeb.



```
C:\WINDOWS\system32\cmd.exe

C:\data\vw>cscript setupAp.js vistaweb.dev.properties c:\data\vw\v7
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

set root logger in log4net.xml to 'adoNetAppender'
set connection string in log4net.xml
set 'logBufferSize' in log4net.xml to 1
changed 1 of 1 for '/configuration/appSettings/add[@key='allowViewLog']' in web.
config to true
changed 1 of 1 for '/configuration/appSettings/add[@key='excludeChemHem']' in we
b.config to false
changed 1 of 1 for '/configuration/userActivity/add[@key='userActivity.logStats'
] in web.config to true
changed 1 of 1 for '/configuration/version/add[@key='version.useFullVersion']' i
n web.config to true
changed 1 of 2 for '/configuration/daoConfig/add[@key='vistaConnectionFactory.se
curityPhrase' or @key='fhieConnectionFactory.securityPhrase']' in web.config to
MY NAME IS UIS01
changed 2 of 2 for '/configuration/daoConfig/add[@key='vistaConnectionFactory.se
curityPhrase' or @key='fhieConnectionFactory.securityPhrase']' in web.config to
MY NAME IS UIS01
set 'userActivity.connectionString' in web.config
=====
configuration completed

C:\data\vw>_
```

Then run the script for vistaContext.

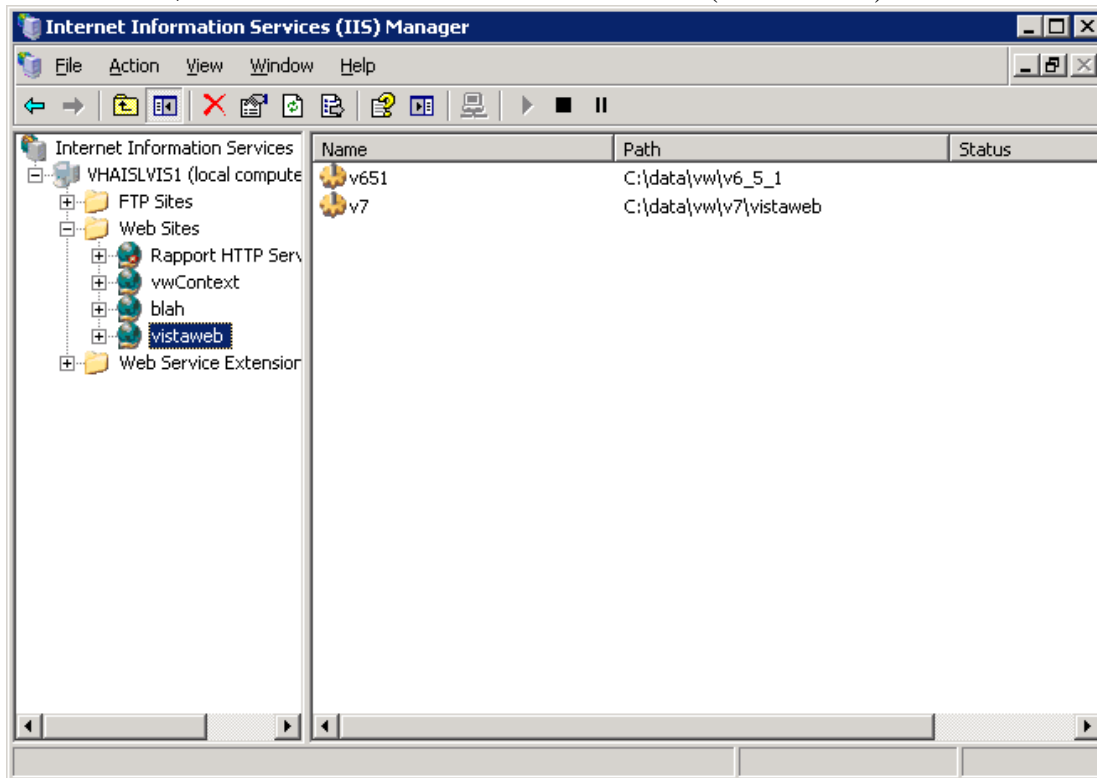
```
C:\WINDOWS\system32\cmd.exe

C:\data\vw>cscript setupAp.js vistaweb.dev.properties c:\data\vwContext\v7
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

set root logger in log4net.xml to 'adoNetAppender'
set connection string in log4net.xml
set 'logBufferSize' in log4net.xml to 1
set 'connectionFactory.connectionString' in web.config
set 'connectionFactory.altConnectionString' in web.config
=====
configuration completed

C:\data\vw>_
```

Go to IIS Manager and right click on the \vwContext virtual folder and click **Start**, then right click on the \vistaWeb virtual folder and click **Start** (in that order).



Point the browser to the development box, and voila!

