



VistAWeb

Version 16.0
*(Patch WEBV*1*25)*

CPRS Access and Server Installation Guide

July 2012

Office of Information and Technology
Department of Veterans Affairs

Revision History

Date	Patch	Page(s)	Change(s)	Project Manager	Technical Writer
July 2012	WEBV*1*25 VistAWeb 16	Throughout	Updated section on new features in Version 16	Pam O'Reilly	Nick Metrokotsas
Dec 2011	15.0	throughout	Minor update changes based on developer review.	Richard Muse	JoAnn Green
Sept 2011	15.0	throughout	Minor update changes to reflect patch changes.	Richard Muse	JoAnn Green
June 2011	14.0	throughout	Minor update changes to reflect patch changes.	Richard Muse	JoAnn Green
September 2010	13.0	throughout	Minor update changes to reflect patch changes.	Jack Schram	JoAnn Green
June 2010	12.0	8	Updated CCOW link	Richard Muse	JoAnn Green
May 2010	12.0	throughout	Minor update changes to reflect patch changes.	Richard Muse	JoAnn Green
Jan 2010	11.0	throughout	Minor update changes to reflect patch changes.	Ed Ridley	JoAnn Green
Sept 2009	10.0	throughout	Minor update changes to reflect patch changes.	Ed Ridley	JoAnn Green
Jan 6 2009	9.0	39	Updated screen capture	Ed Ridley	JoAnn Green
Dec 23, 2008	9.0	throughout	Minor update changes to reflect patch changes.	Ed Ridley	JoAnn Green
10/30/08	8.0	39	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	27	Added Index Creation scripts to Appendix A	Ed Ridley	JoAnn Green
May 22, 2008	7.0/7.5	20	Added VistA Security to Server Installation Section	Scott Madsen	JoAnn Green
May 19, 2008	7.0/7.5	28-30	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,	Scott Madsen	Albert Fretz

Date	Patch	Page(s)	Change(s)	Project Manager	Technical Writer
			CprsUser_create.sql.txt, Log_create.sql.txt, LoggerTable_create.sql.txt, Requests_create.sql.txt, SpecialUsers_create.sql.txt, and updateCprsUsers.sql.txt		
February 2007	n/a	p. 7 p. 9	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 p. 7 p. 11 p. 20	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 distribution through the VDL. Changed VA Service Desk and e-mail address.	S. Madsen	M. Kelsey
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	G. Smith	M. Kelsey

Date	Patch	Page(s)	Change(s)	Project Manager	Technical Writer
			accounts and set up instructions for pre-release testing.		
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

REVISION HISTORY	II
TABLE OF CONTENTS	V
TABLE OF FIGURES	VI
VISTAWEB	7
INTRODUCTION.....	7
ASSUMPTIONS.....	7
KNOWN CONSTRAINTS.....	8
ACCESS TO VISTAWEB FROM CPRS VISTAWEB BUTTON.....	9
PROVIDING ACCESS TO VISTAWEB FROM THE TOOLS MENU.....	10
Tools Menu – Example Configuration	11
VISTAWEB INSTALLATION ON AN APPLICATION SERVER AND DATABASE SERVER...12	
SYSTEM REQUIREMENTS – HARDWARE	12
SYSTEM REQUIREMENTS – SOFTWARE.....	12
Application Configurations and Settings.....	13
BROKER SECURITY ENHANCEMENT CODE UPDATE CONFIGURATION STEPS	15
One Time Only:.....	15
SERVER INSTALLATION	17
WEB APPLICATION INSTRUCTIONS.....	17
SQL SERVER DATABASE SCHEMA INSTRUCTIONS.....	20
VISTA SECURITY	20
Patient Selection Restriction Setup	21
Tab Access Restriction Setup	22
Both Patient Selection and Tab Access Restrictions	23
APPENDIX A: DATABASE SCHEMA.....	25
LOG CREATION SCRIPT.....	26
CPRSUSERS CREATION SCRIPT	26
SPECIALUSERS CREATION SCRIPT	26
LOGDESC VIEW CREATION SCRIPT	26
LOGGERTABLE CREATION SCRIPT.....	26
BIGGERLOGGER CREATION SCRIPT.....	27
USERAUTH CREATION SCRIPT	27
CREATE INDEX FOR CPRS USERS	27
CREATE INDEX FOR SPECIALUSERS	27
CREATE INDEX FOR USERAUTH.....	27
APPENDIX B: CONFIGURATION BY A SCRIPT.....	28
SETUPAP.16.0.0.JS SCRIPT.....	28
VISTAWEB.PRODUCTION.16.0.0.PROPERTIES PROPERTIES FILE	34
EXAMPLE	36

Table of Figures

Figure 1: Setting Internet Explorer to Allow Pop-ups	9
Figure 2: VistAWeb from CPRS VistAWeb Button.....	9
Figure 3: Web Service Extension Settings	14
Figure 4: Web Application Settings	14
Figure 5: ASP.NET Tab in the Web App Properties Box	15
Figure 6: Configuration Screen Setup Script	19
Figure 7: Script Execution Output Example	20
Figure 8: User's CPRS tabs and patient selection settings.....	21
Figure 9: Effective restrictions and parameter string values	24

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 Bi-Directional Health Information Exchange (BHIE) system, the Health Data Repository II (HDR II) databases, and the Nationwide Health Information Network (NwHIN). (see <http://healthit.hhs.gov/portal/server.pt?open=512&mode=2&cached=true&objID=1142>).

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 directly from the VistaWeb 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/bc/templates_and_references.aspx

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:

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 this website for guidance about installation and configuration of the CCOW Desktop Components:*

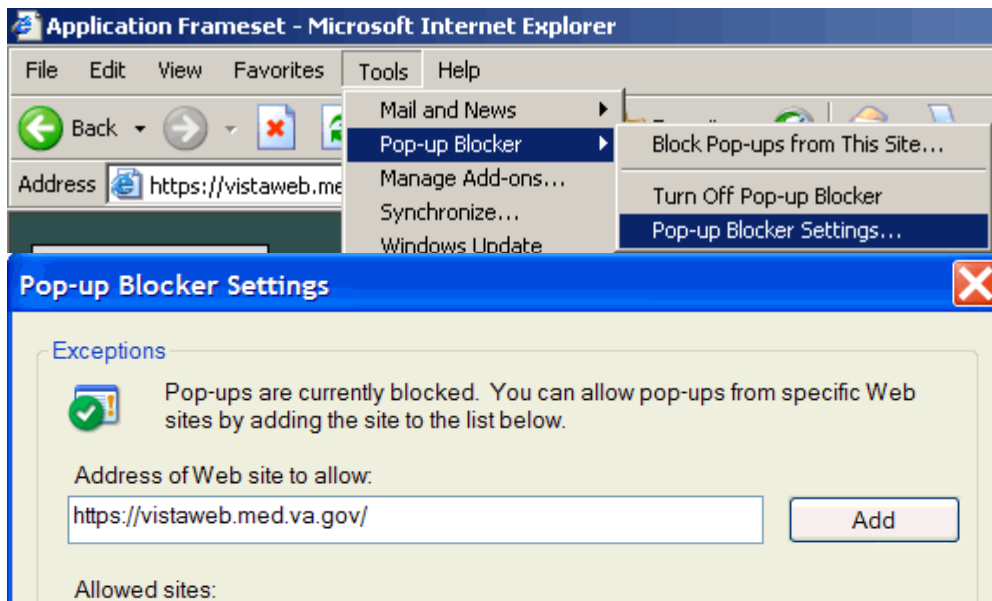
<http://vaww.eie.va.gov/SysDesign/HSED/CCOW/default.aspx>.

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 below).
4. Starting with version 10, VistAWeb uses version 2.0 of the .Net framework. The .Net framework can run side-by-side with different versions of the framework installed on the same machine. If .Net 2.0 is not installed, you should install the .Net 2.0 runtime available at <http://msdn.microsoft.com>.

Figure 1: Setting Internet Explorer to Allow Pop-ups

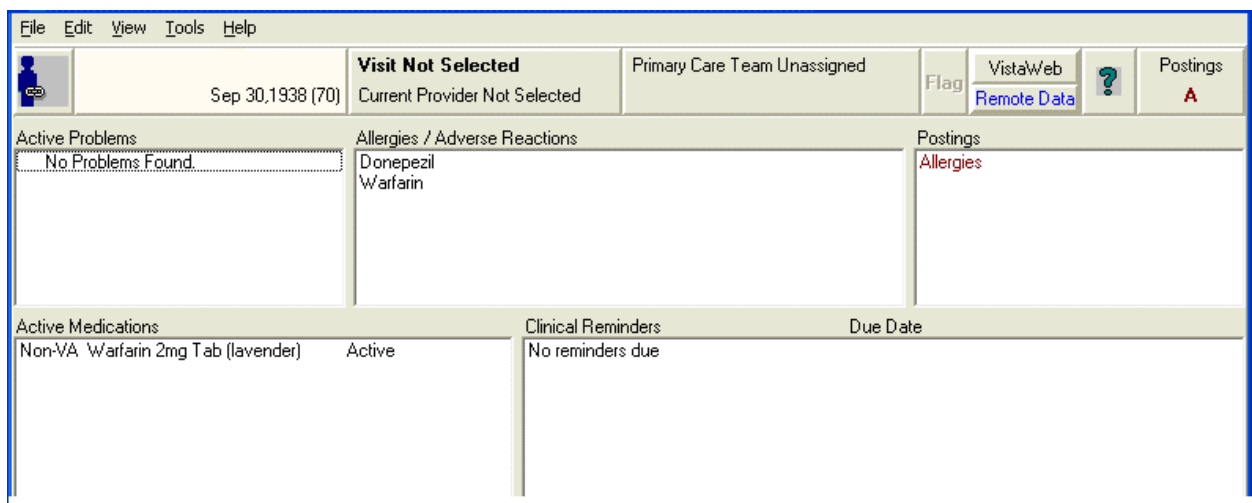


Note: Some links found in this installation guide go to sites or pages found on the VA intranet. These sites or pages are not accessible from outside the VA network.

Access to VistAWeb from CPRS VistAWeb 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 VistAWeb Button



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.)

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 Austin, Texas. Information on these production servers can be found at the following links:

Web Server

<http://vaww.sms.aac.va.gov/SMSReporting/MachDetails.asp?Machine=vaausnvweb200>

SQL Server

<http://vaww.sms.aac.va.gov/SMSReporting/MachDetails.asp?Machine=VAAUSNVWSQL200>

System Requirements – Software

The remainder of the instructions is for Windows 2003 installations; refer to the VW Production Installation Guide for installation on Windows 2008, a link to the guide is provided below:

http://vaww.oed.portal.va.gov/products/vler/nwhin/VistAWeb/Shared%20Documents/Increments/Increment%201/V16_16.1/Draft/Deployment/VW_Production_Install_Guide.docx

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
- For Windows Server 2003 Enterprise, verify or enter the registry values found in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` for `MaxUserPort` and `TcpTimedWaitDelay` with the following values:
 - 40000 (decimal) for `MaxUserPort` and
 - 120 (decimal) for `TcpTimedWaitDelay`
 - See the article at <http://msdn.microsoft.com/en-us/library/aa560610%28BTS.20%29.aspx> entitled “Avoiding TCP/IP Port Exhaustion” for more information
- Internet Information Services (IIS) 6.0 (installed by default as part of the Application Server role)
- .NET Framework 2.0
- 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)
- Make sure that the current version of ASP.NET is selected in the properties dialog of the virtual folder setup for VistAWeb version 14 (see Figure 5). It should be set to 2.0.50727.
- Configure Vistaweb to use SSL port 443 and use a security 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

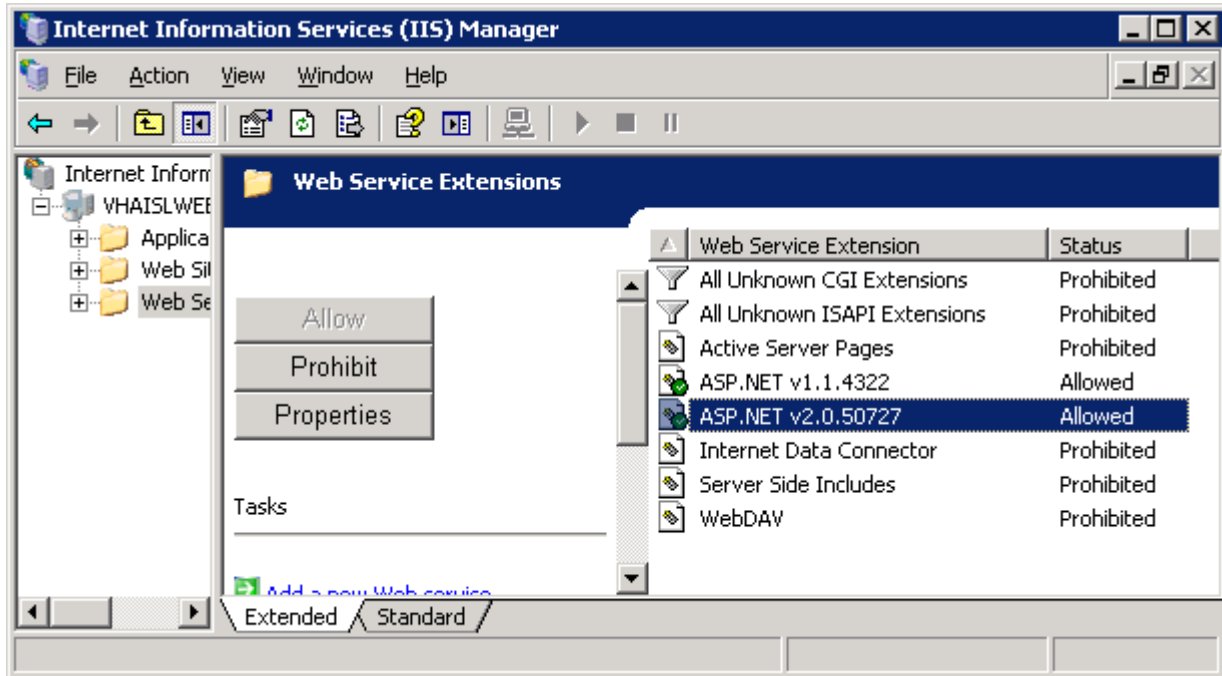


Figure 4: Web Application Settings

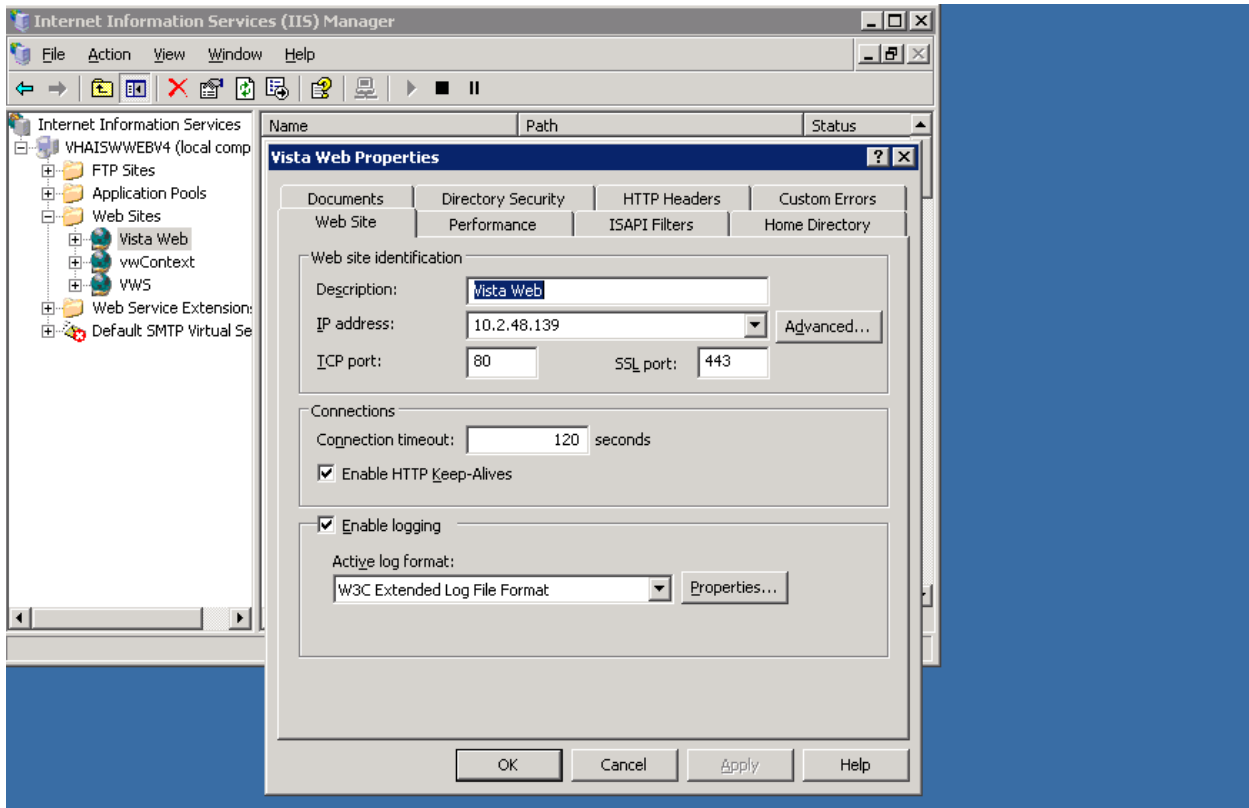
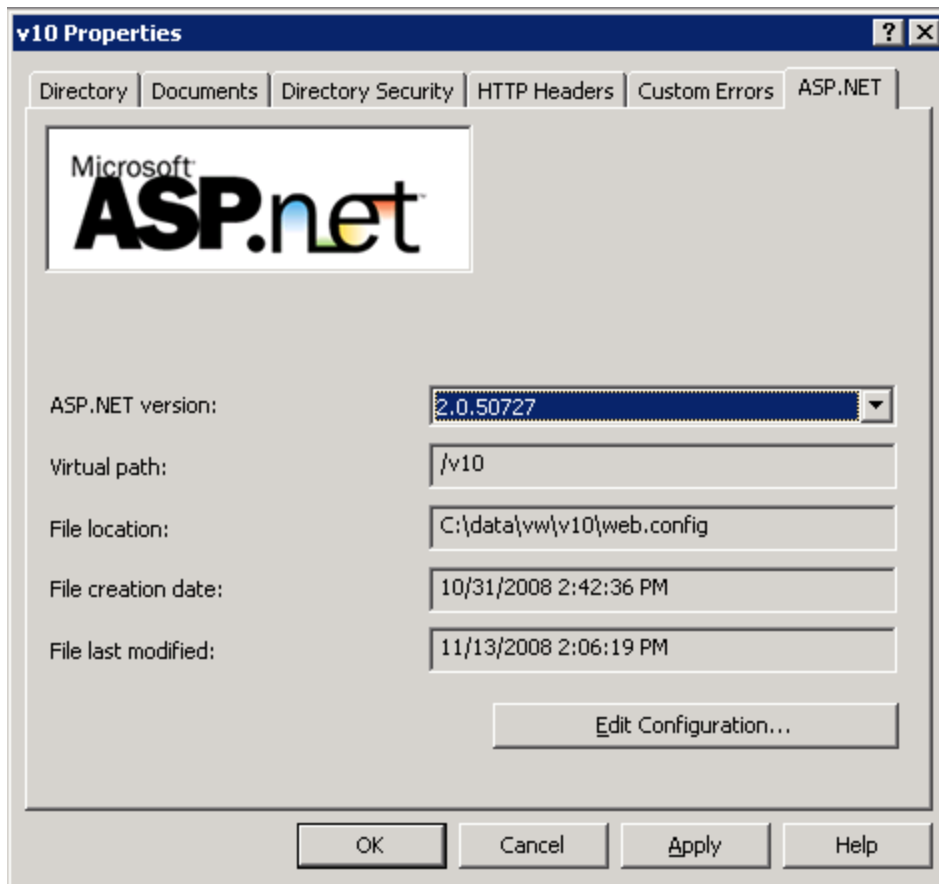


Figure 5: ASP.NET Tab in the Web App Properties Box



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. If this work has been done for version 8 or later, these steps can be skipped.

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_COMPLETE_15.n.n.tn.mm.dd.yyyy.zip` into a working folder of your choice. It is suggested that you name the folder with **v15** in the name (for example, `vw_v15`).
- 3 From your working directory, unzip `vistawebEMR_15.n.n.tn.mm.dd.yyyy.zip` into the target VW folder.
- 4 From your working directory, unzip `vistawebcontext_15.n.n.tn.mm.dd.yyyy.zip` into the target folder created in step 3 of the one-time BSE configuration mentioned above.
- 5 From your working directory, edit the file called `vistaweb.production.15.n.n.properties`, and change the following:
 - a. `securityPhrase=<BSE Application Security Phrase >`
 - b. `awivSecurityPhrase=<The security phrase for the AWIV VistA Imaging viewing component>`
 - c. `awivNotUsed=<true or false>` - use 'false' (without the quotes) if the AWIV component has been nationally released (and is generally available on all machines using this instance of VistAWeb); otherwise enter 'true' (without the quotes).
 - d. `db_server=<DNS name of your SQL Server instance>`
 - e. `db_databaseName=<Database name>`
 - f. `db_username=<Application user name>` - this is the user name used only by VistAWeb to login to the database and update data in the tables.
 - g. `db_password=<Application password>` - this is the password for the above Application user name.
 - h. If the log4net logging requires a different Application user name / password, remove the “#” character from the `#db_log4netUsername` and `#db_log4netPassword` lines, and enter those values respectively.
- 6 Open a command prompt at the same location as your working directory.
- 7 From the command prompt, run the configuration script **setupAp.15.n.n.js** against your target VistAWeb folder.
 - a. To do this, type in the following, and press <enter>: **cscript
setupAp.15.n.n.js
vistaweb.production.15.n.n.properties <target VW**

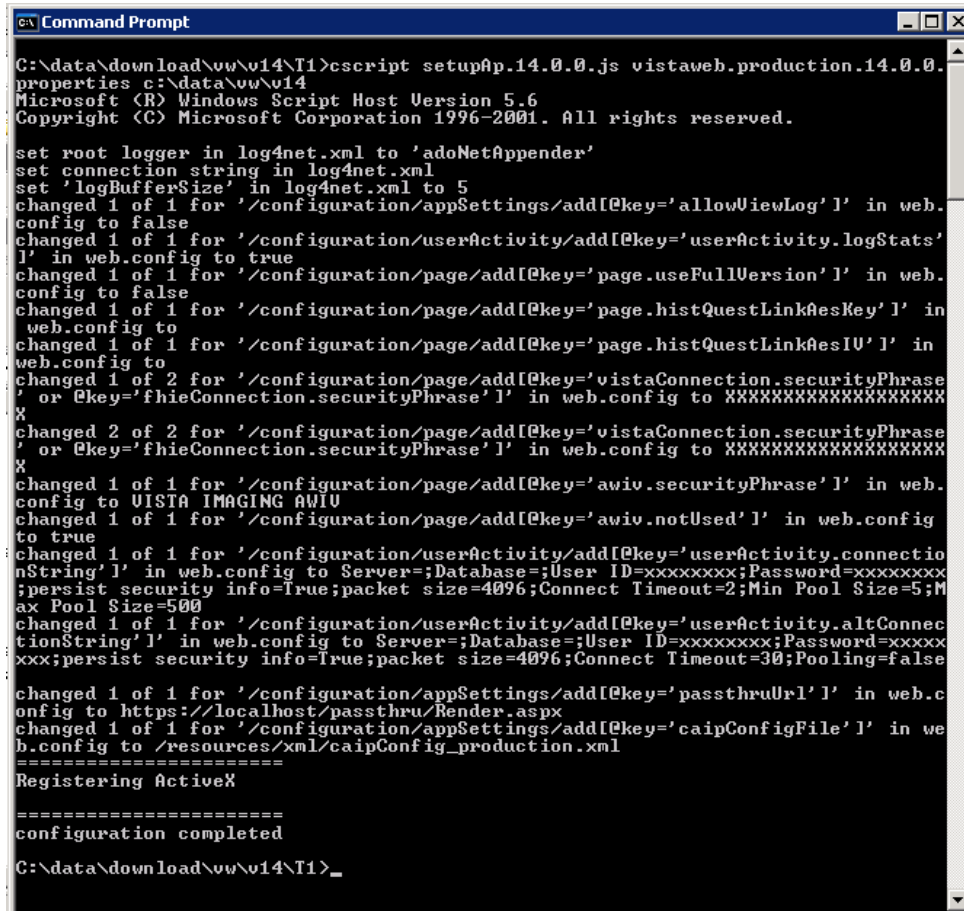
folder> , where 15.n.n is the version of VistAWeb that you're installing, and <target VW folder> is the name of the folder where all of the VistAWeb files were unzipped.

- i. For example, if the version of VistAWeb you were working with was 15.0.0, and your target VistAWeb folder was located at c:\data\vw, the command you would execute would be: **cscript setupAp.15.0.0.js vistaweb.production.15.0.0.properties c:\data\vw**
 - b. See figure 5 for a sample output from this script execution.
- 8 From the command prompt, run the configuration script **setupAp.15.n.n.js** against your target vwContext folder.
 - a. To do this, type in the following, and press <enter>: **cscript setupAp.15.n.n.js vistaweb.production.15.n.n.properties <target vwContext folder>**, where 15.n.n is the version of VistAWeb that you're installing, and <target vwContext folder> is the name of the folder where all of the vwContext files were unzipped.
 - i. For example, if the version of VistAWeb you were working with was 15.0.0, and your target vwContext folder was located at c:\data\vwContext, execute this command: **cscript setupAp.15.0.0.jsvistaweb.production.15.0.0.properties c:\data\vwContext**
 - b. See figure 6 for a sample output from this script execution
- 9 For Production purposes, a domain address was established for VistAWeb (<https://vistaweb.med.va.gov/>), with VistAWeb configured as its own website.
- 10 For Production purposes, a domain address was established for VistAWeb BSE context service (*vwContext* or *vistawebcontext*, at <http://vaww.context.vistaweb.med.va.gov>), with *vwContext* configured as its own web app (web site).
- 11 Set the default VistAWeb content page to "loginframeset.htm."
- 12 Grant "Write" permission to the *NETWORK SERVICE* user on the Windows 2003 server to the folder **~/resources/images/temp**. This is where temporary vitals graph images are stored for users who request them. The files that get placed in this folder are automatically managed by VistAWeb.
- 13 Start the *vistawebcontext* application pool and the *vistaweb* web application, then start the *vistaweb* application pool.
- 14 Start the *vistawebcontext* web application, then start the *vistaweb* application.
- 15 Copy *vhasites.xml* file from the prior VistAWeb instance into <vistaweb>/resources/xml.
- 16 Start <vistawebcontext>.
- 17 Start <vistaweb>.

A copy of the *setupAp.15.n.n.js* script and properties file is provided in Appendix B.

Note: There is an extra zip file in the build called `vistawebVwEncrypt_15.n.n.tn.mm.dd.yyyy.zip` which is only used for special configurations of VistAWeb on Windows 2008 servers. This capability is reserved for future versions of VistAWeb.

Figure 6: Configuration Screen Setup Script



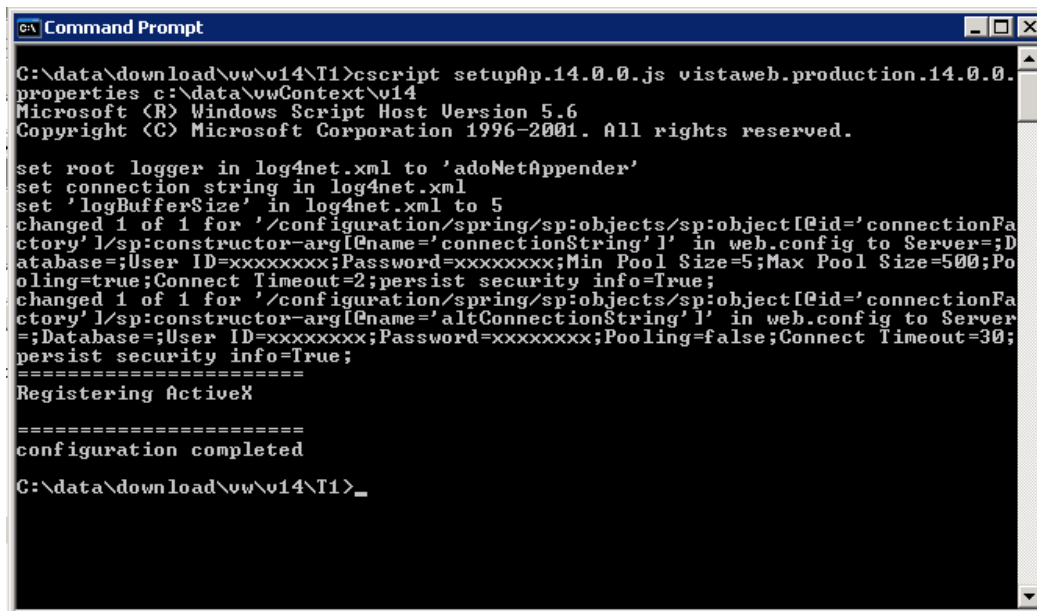
```
C:\data\download\vw\v14\T1>cscript setupAp.14.0.0.js vistaweb.production.14.0.0.
properties c:\data\vw\v14
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

set root logger in log4net.xml to 'adoNetAppender'
set connection string in log4net.xml
set 'logBufferSize' in log4net.xml to 5
changed 1 of 1 for '/configuration/appSettings/add[@key='allowViewLog']' in web.
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/page/add[@key='page.useFullVersion']' in web.
config to false
changed 1 of 1 for '/configuration/page/add[@key='page.histQuestLinkAesKey'] in
web.config to
changed 1 of 1 for '/configuration/page/add[@key='page.histQuestLinkAesIU']' in
web.config to
changed 1 of 2 for '/configuration/page/add[@key='vistaConnection.securityPhrase
' or @key='fhieConnection.securityPhrase']' in web.config to XXXXXXXXXXXXXXXXXXXX
X
changed 2 of 2 for '/configuration/page/add[@key='vistaConnection.securityPhrase
' or @key='fhieConnection.securityPhrase']' in web.config to XXXXXXXXXXXXXXXXXXXX
X
changed 1 of 1 for '/configuration/page/add[@key='awiv.securityPhrase']' in web.
config to VISTA IMAGING AWIU
changed 1 of 1 for '/configuration/page/add[@key='awiv.notUsed']' in web.config
to true
changed 1 of 1 for '/configuration/userActivity/add[@key='userActivity.connectio
nString']' in web.config to Server=;Database=;User ID=xxxxxxx;Password=xxxxxxx
;persist security info=True;packet size=4096;Connect Timeout=2;Min Pool Size=5;M
ax Pool Size=500
changed 1 of 1 for '/configuration/userActivity/add[@key='userActivity.altConnec
tionString']' in web.config to Server=;Database=;User ID=xxxxxxx;Password=xxxxx
xxx;persist security info=True;packet size=4096;Connect Timeout=30;Pooling=false

changed 1 of 1 for '/configuration/appSettings/add[@key='passthruUrl']' in web.c
onfig to https://localhost/passthru/Render.aspx
changed 1 of 1 for '/configuration/appSettings/add[@key='caipConfigFile']' in we
b.config to /resources/xml/caipConfig_production.xml
=====
Registering ActiveX
=====
configuration completed

C:\data\download\vw\v14\T1>_
```

Figure 7: Script Execution Output Example



```
C:\data\download\vw\v14\I1>cscript setupAp.14.0.0.js vistaweb.production.14.0.0.
properties c:\data\vwContext\v14
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 5
changed 1 of 1 for '/configuration/spring/sp:objects/sp:object[id='connectionFa
ctory' l/sp:constructor-arg[name='connectionString' l' in web.config to Server=;D
atabase=;User ID=xxxxxxx;Password=xxxxxxx;Min Pool Size=5;Max Pool Size=500;Po
oling=true;Connect Timeout=2;persist security info=True;
changed 1 of 1 for '/configuration/spring/sp:objects/sp:object[id='connectionFa
ctory' l/sp:constructor-arg[name='altConnectionString' l' in web.config to Server
=;Database=;User ID=xxxxxxx;Password=xxxxxxx;Pooling=false;Connect Timeout=30;
persist security info=True;
=====
Registering ActiveX
=====
configuration completed
C:\data\download\vw\v14\I1>_
```

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.)
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 configured on a site-by-site basis in VistAWeb.

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

Figure 8: User's CPRS tabs and patient selection settings



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

VW supports either a “NO” in the `RESTRICT PATIENT SELECTION` setting or a *blank* in the `RESTRICT PATIENT SELECTION`.

To require that the user has a “NO” in the `RESTRICT PATIENT SELECTION` setting, one would edit the `resources/xml/VhaSites.xml` file, in the `VhaSite` element that this restriction applies to. One would enter a `restriction="selectAll"` attribute on the `DataSource` element that has the VistA source IP and port. For example, for an entry called “CLE15”, to make this restriction active, one would specify:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="selectAll"/>
</VhaSite>
```

Listing 1

The entries in this attribute are not sensitive to case, punctuation, or whitespace. One could enter:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="...SelectAll "/>
</VhaSite>
```

Listing 2

And VistAWeb will function the same as the entry in

Listing 1.

Tab Access Restriction Setup

In order to restrict users from VistAWeb access based on the RPT or COR tabs from their FileMan user profile, one would change the entry for the given VistA in the `resources/xml/VhaSites.xml` file. One would add a `restriction="core"`, or `restriction="reports"`, or `restriction="reports,core"`, or `restriction="allTabs"` to allow users with COR, RPT, either COR or RPT, or both COR & RPT, respectfully.

For example, to require users in CLE15 to have the RPT tab authorization in their FileMan user profile, one would specify CLE15 in the sites file as:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="reports"/>
</VhaSite>
```

Listing 3

To require users in CLE15 to have at least the CORE *or* REPORTS tab authorization in their FileMan user profile, one would specify CLE15 in the sites file as:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="core,reports"/>
</VhaSite>
```

Listing 4

Case, order, and punctuation of the parameters don't matter either. It could be specified as:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="Reports CORE"/>
</VhaSite>
```

Listing 5

In order to require users to have both RPT and COR tabs, one would specify something like this:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="allTabs"/>
</VhaSite>
```

Listing 6

Both Patient Selection and Tab Access Restrictions

If you specify both restrictions, then both restrictions will be checked before access is allowed in the given VistA system.

For example, in order for VistAWeb to check for CPRS Tab access for RPT *and* COR *and* to check for Patient Selection restriction for the site CLE15, you could change the `/resources/xml/VhaSites.xml` file to the following:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="selectAll,allTabs"/>
</VhaSite>
```

Listing 7

If the restriction attribute is missing or completely invalid, then VistAWeb will act like it does in prior versions and not check any of these user authorization settings. For our CLE15 example:

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y" restriction="hi there"/>
</VhaSite>
```

Listing 8

or

```
<VhaSite name="CLE15" ID="982" moniker="CLE15">
  <DataSource protocol="VISTA" modality="HIS" source="10.4.229.27"
status="active" port="9593" test="Y"/>
</VhaSite>
```

Listing 9

Listing 8 and

Listing 9 would both allow users to access VistAWeb like it does in prior versions. To summarize, 8, below, contains a list of effective restrictions and the parameter string to use as the value in the restriction attribute to make this work.

Figure 9: Effective restrictions and parameter string values

Restriction	Parameter String for the restriction attribute
Let users log in to VistAWeb like in prior versions	= "", or simply take the restriction attribute out of the DataSource element.
Require users to just have the RESTRICT PATIENT SELECTION set to NO	= "selectAll"
Require users to just have the RPT tab in CPRS TAB ACCESS	= "report"
Require users to just have the COR tab in CPRS TAB ACCESS	= "core"
Require users to have either the COR tab <i>or</i> the RPT tab in CPRS TAB ACCESS	= "core,report"
Require users to have both the COR tab <i>and</i> the RPT tab in CPRS TAB ACCESS	= "allTabs"
Require users to have the RESTRICT PATIENT SELECTION set to NO <i>and</i> have the RPT tab in CPRS TAB ACCESS	= "selectAll,report"
Require users to have the RESTRICT PATIENT SELECTION set to NO <i>and</i> have the COR tab in CPRS TAB ACCESS	= "selectAll,core"
Require users to have the RESTRICT PATIENT SELECTION set to NO <i>and</i> have either the COR tab <i>or</i> RPT tab in CPRS TAB ACCESS	= "selectAll,core,report"
Require users to have the RESTRICT PATIENT SELECTION set to NO <i>and</i> have both the COR tab <i>and</i> RPT tab in CPRS TAB ACCESS	= "selectAll,allTabs"

Appendix A: Database Schema

- Database Name: EMR
- Database Tables:
 - Log
 - CprsUsers
 - SpecialUsers
 - LoggerTable
 - BiggerLogger
 - 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  
)
```

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
)

```

BiggerLogger Creation Script

```

CREATE TABLE [dbo].[BiggerLogger] (
    [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] [text] NULL,
    [Exception] [varchar] (2000) NULL
) ON [DEFAULT] TEXTIMAGE_ON [DEFAULT]

```

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
)

```

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

SetupAp.16.0.0.js 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 awivSecurityPhrase="";
var awivNotUsed="false";
var rootAppender="";
var logBufferSize="";
var log4netInternalDebug = "";
var caipConfigFile="";

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="";

var bhieReportCredentials_domain = "";
var bhieReportCredentials_username = "";
var bhieReportCredentials_password = "";
var passthruUrl = "";
var histQuestLinkAesKey = "";
var histQuestLinkAesIV = "";

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 ("awivSecurityPhrase"): {
                awivSecurityPhrase = newVal;
                break;
            }
            case ("awivNotUsed"): {
                awivNotUsed = newVal;
                break;
            }
            case ("rootAppender"): {
                rootAppender = newVal;
                break;
            }
            case ("logBufferSize"): {
                logBufferSize = newVal;
                break;
            }
            case ("log4netInternalDebug"): {
                log4netInternalDebug = newVal.toLowerCase();
                break;
            }
            case ("caipConfigFile"): {
                caipConfigFile = newVal;
            }
            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;
}
}
case ("bhieReportCredentials_domain"): {
    bhieReportCredentials_domain = newVal;
    break;
}
}
case ("bhieReportCredentials_username"): {
    bhieReportCredentials_username = newVal;
    break;
}
}
case ("bhieReportCredentials_password"): {
    bhieReportCredentials_password = newVal;
    break;
}
}
case ("passthruUrl"): {
    passthruUrl = newVal;
    break;
}
}
case ("histQuestLinkAesKey"): {
    histQuestLinkAesKey = newVal;
    break;
}
}
case ("histQuestLinkAesIV"): {
    histQuestLinkAesIV = 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);
}

if (log4netDomChanged) {
    var fileName = getLog4netFileName();
    fileSystem.DeleteFile(fileName);
    var newFile1 = fileSystem.OpenTextFile(fileName, 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/daoConfig/add[@key='protocol_CDS.chemHemExclud
e']",excludeChemHem);
    changeWebDom(webConfigDom,"/configuration/userActivity/add[@key='userActivity.logStats']
,logStats);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='page.useFullVersion']",useFullVe
rsion);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='page.histQuestLinkAesKey']",hist
QuestLinkAesKey);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='page.histQuestLinkAesIV']",histQ
uestLinkAesIV);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='vistaConnection.securityPhrase'
or @key='fhieConnection.securityPhrase']",securityPhrase);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='awiv.securityPhrase']",awivSecur
ityPhrase);
    changeWebDom(webConfigDom,"/configuration/page/add[@key='awiv.notUsed']",awivNotUsed);

    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);
        changeWebDom(webConfigDom,"/configuration/userActivity/add[@key='userActivity.connectionS
tring']", connectString1);
        changeWebDom(webConfigDom,"/configuration/userActivity/add[@key='userActivity.altConnecti
onString']", connectString2);
        changeWebDom(webConfigDom,"/configuration/daoConfig/add[@key='bhieReportCredentials.domai
n']",bhieReportCredentials_domain);
        changeWebDom(webConfigDom,"/configuration/daoConfig/add[@key='bhieReportCredentials.userN
ame']",bhieReportCredentials_username);
        changeWebDom(webConfigDom,"/configuration/daoConfig/add[@key='bhieReportCredentials.passw
ord']",bhieReportCredentials_password);
        changeWebDom(webConfigDom,"/configuration/appSettings/add[@key='passthruUrl']",passthruUr
l);
        changeWebDom(webConfigDom,"/configuration/appSettings/add[@key='caipConfigFile']",caipCon
figFile);
    }
else if (isVwContext(webConfigDom)) {
    /** vwContext **/
    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);
    changeWebDom(webConfigDom,
"/configuration/spring/sp:objects/sp:object[@id='connectionFactory']/sp:constructor-
arg[@name='connectionString']", connectString1);
    changeWebDom(webConfigDom,
"/configuration/spring/sp:objects/sp:object[@id='connectionFactory']/sp:constructor-
arg[@name='altConnectionString']", connectString2);
}
else {
    /** vwPassthru **/
    changeWebDom(webConfigDom,"/configuration/spring/sp:objects/sp:object[@type='Render.aspx'
]/sp:property[@name='DetailContentProcessor']/sp:object/sp:property[@name='Credentials']/sp:objec
t/sp:property[@name='Domain']",bhieReportCredentials_domain);
    changeWebDom(webConfigDom,"/configuration/spring/sp:objects/sp:object[@type='Render.aspx'
]/sp:property[@name='DetailContentProcessor']/sp:object/sp:property[@name='Credentials']/sp:objec
t/sp:property[@name='UserName']",bhieReportCredentials_username);
    changeWebDom(webConfigDom,"/configuration/spring/sp:objects/sp:object[@type='Render.aspx'
]/sp:property[@name='DetailContentProcessor']/sp:object/sp:property[@name='Credentials']/sp:objec
t/sp:property[@name='Password']",bhieReportCredentials_password);
}

if (log4netInternalDebug!=null && log4netInternalDebug.length>0) {
    changeWebDom(webConfigDom,"/configuration/appSettings/add[key='log4net.Internal.Debug']",
log4netInternalDebug);
}

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("Registering ActiveX");
var cShell = WScript.CreateObject("WScript.Shell");
cShell.Exec("regsvr32 /s " + getAwivActiveXName());

```

```

WScript.StdOut.WriteLine("");
WScript.StdOut.WriteLine("=====");
WScript.StdOut.WriteLine("configuration completed");
WScript.Quit(0);
//***** End of Program *****

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

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

function getAwivActiveXName() {
    return WScript.Arguments.Item(1) + "\\bin\\AWIVEncryption.ocx";
}

function getLog4netDom() {
    var fileName = getLog4netFileName();
    if (!(fileSystem.FileExists(fileName))) {
        WScript.StdOut.WriteLine("log4net.xml file not found in " + fileName);
        return null;
    }
    var xmlDom=getXmlDom();
    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=getXmlDom();
    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 getXmlDom() {
    var xmlDom = null;
    try {
        xmlDom = new ActiveXObject("MSXML2.DOMDocument.4.0");
    }
    catch (ex) {
        xmlDom = new ActiveXObject("MSXML2.DOMDocument.6.0");
    }
    xmlDom.async = false;
    xmlDom.validateOnParse=false;
    xmlDom.resolveExternals=false;
    xmlDom.setProperty("SelectionNamespaces", "xmlns:sp='http://www.springframework.net'");

    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 isVwContext(xmlDom) {
    var nodeList =
xmlDom.selectNodes("/configuration/spring/sp:objects/sp:object[@type='Resolve.aspx']");
    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);
    }
}

```

vistaweb.production.16.0.0.properties Properties File

```

excludeChemHem=true
allowViewLog=false
logStats=true
useFullVersion=false
securityPhrase=XXXXXXXXXXXXXXXXXXXX
awivSecurityPhrase=VISTA IMAGING AWIV
awivNotUsed=true
rootAppender=adoNetAppender
logBufferSize=5
caipConfigFile=/resources/xml/caipConfig_production.xml
log4netInternalDebug=false
passthruUrl=https://localhost/passthru/Render.aspx
#bhieReportCredentials_domain=vhamaster
#bhieReportCredentials_username=vistawebUser
#bhieReportCredentials_password=health_e_vet
#histQuestLinkAesKey=0123456789abcdef
#histQuestLinkAesIV=fedcba9876543210
#
db_server=
db_databaseName=
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:

```

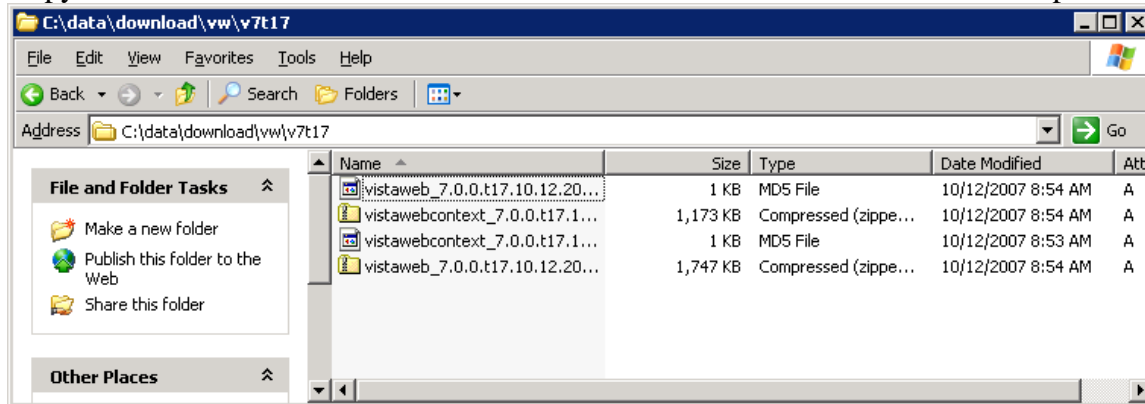
db_server=VHAISLVIS1.med.va.gov
db_databaseName=VW_MAIN
db_username=vwApp
db_password=E+I9345890s

```

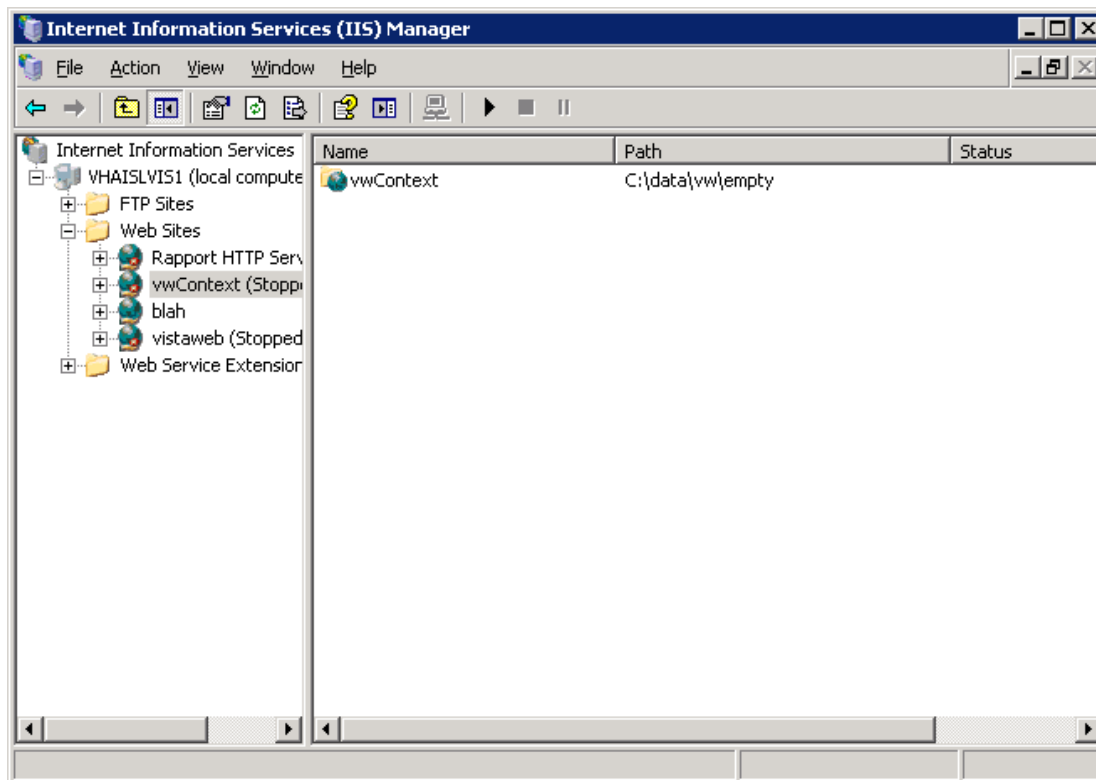
Example

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 “v16”.

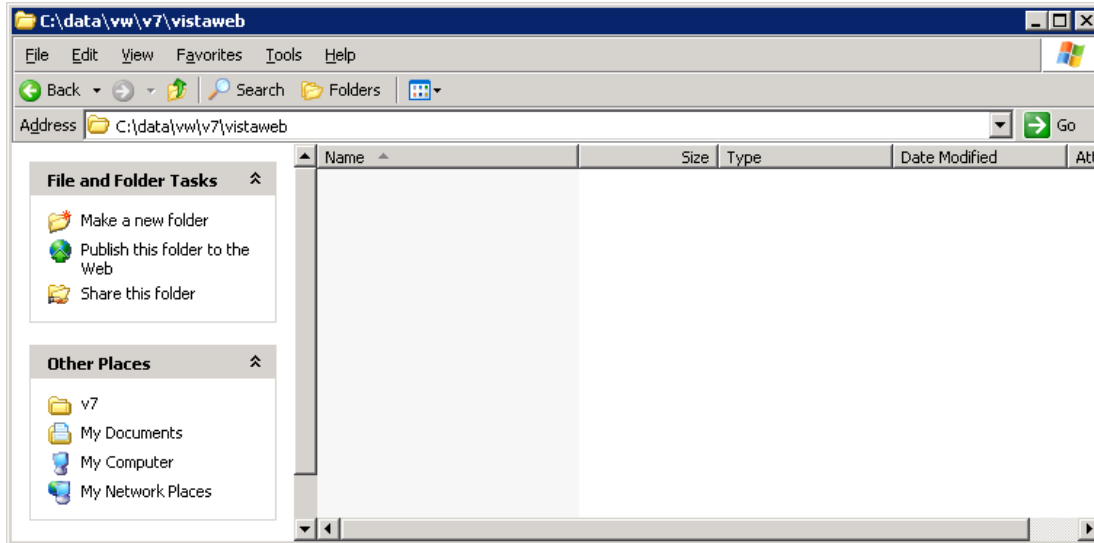
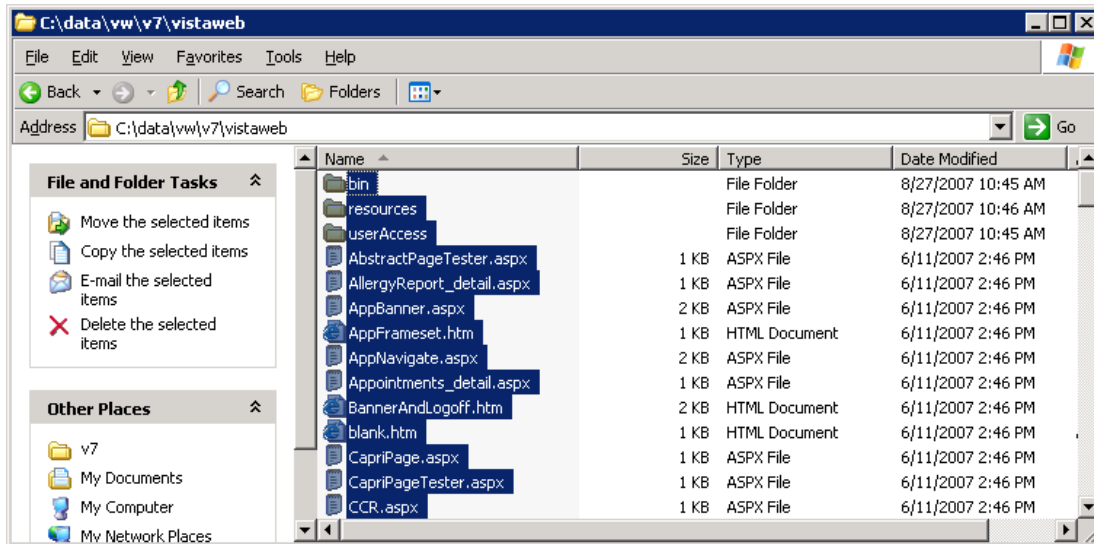
Copy the build files to the `c:\data\download\vw\v7t17` folder on the development server:



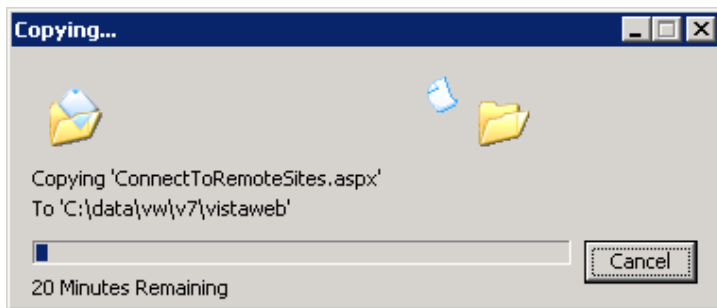
Right click on the `\vistaweb` and `\vistawebcontext` virtual folders and select **Stop** from the menu (in that order).



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\v7>cscript setupAp.js vistaweb.dev.properties c:\data\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='excludeChemMem']' 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\v7>_
```

Then run the script for vistaContext.

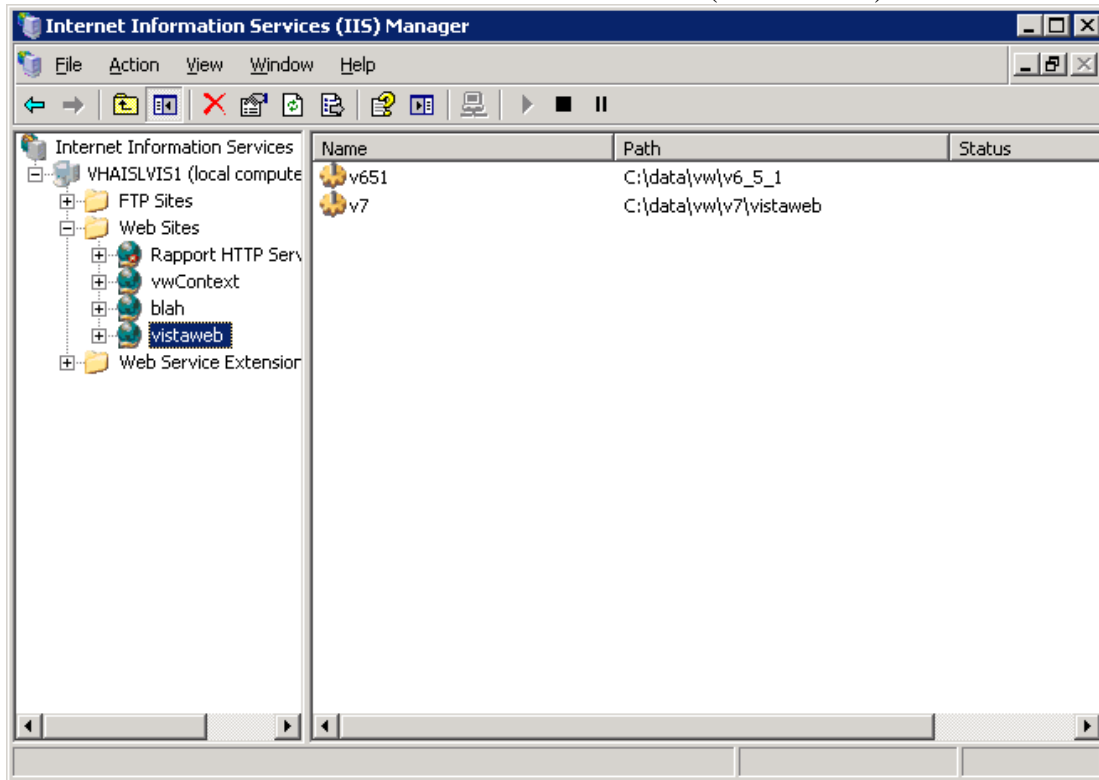
```
C:\WINDOWS\system32\cmd.exe

C:\data\v7>cscript setupAp.js vistaweb.dev.properties c:\data\v7Context\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\v7>_
```

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 the following screen appears:

