

Medical Domain Web Services (MDWS)

Version 2.0

C3-C1 Conversion Project

Installation Guide

(MWVS*2)



September 2011

Department of Veterans Affairs
Office of Information and Technology (OI&T)
Office of Enterprise Development (OED)

Revision History

Date	Revision	Description	Author
April 2010	1.0	Initial version for v 2.0	J Confer L Harmon C Beynon
May 2010	1.1	Added updates	J Confer C Beynon
July 2010		Changed dates	C Beynon
September 2010		Changed date to September	C Beynon
October 2010		Changed dates to October	C Beynon
November 2010	1.2	Revised for ORT comments	J Confer
December 2010		Changed dates to January 2011	CBeynon
December 2010	1.3	<ul style="list-style-type: none"> • Added v2.0 to the MDWS Database • Added back out procedures per EIE input • Per EIE feedback, combined elements of the Production Operations Manual with the Install Guide • See last section of Install Guide 	JRogers
December 2010	1.4	Added MDWS BSE patch XU*8*557 to section <i>Pre-install Instructions and Preparation</i>	CBeynon
May 2011	1.5	<ul style="list-style-type: none"> • Added (MWVS*2) namespace • Changed dates to May 2011 	CBeynon
June 2011	1.6	Updated the IG with instructions from JM/Medora	CBeynon
July 2011	1.7	Prepped for national release, ESE Checklist	CBeynon
September 2011	1.8	Changed dates to September 2011 for release	CBeynon

Table of Contents

Introduction	1
History	1
Deployment Overview	3
Prior to Installation	4
Requirements.....	4
Configuring the MDWS Database	5
Installation Instructions	6
Obtaining the Software.....	6
Installing MDWS	6
After Installation	11
Updating the ZIPCodes Table	11
Troubleshooting MDWS	21
Uninstalling MDWS.....	21
Normal Procedures	21
Sample of an Error Message	22
Production Issue History	22
Potential Troubleshooting Steps.....	22
Failover MDWS Deployment	23
Automated Solution.....	23
Manual Solution	23
Symptoms, Diagnoses, and Possible Solutions	24
Production Operations Manual Section	25

Introduction

Medical Domain Web Services (MDWS) (pronounced *meadows*) is a suite of Service Oriented Architecture (SOA) middle-tier web services that exposes medical domain functionality, Medical Domain Objects (MDO). MDWS is equipped with the capacity to virtualize any legacy Veterans Health Information Systems and Technology Architecture (VistA) Remote Procedure Call (RPC) as a web service. A web service is an Application Programming Interface (API), which uses Simple Object Access Protocol (SOAP), the standardized protocol to communicate with subscribed client applications.

History

Historically, the Department of Veteran Affairs (VA) developers use a standard, 2-tier (client/server) architecture to develop applications, such as the Computerized Patient Record System (CPRS) and the Remote Procedure Call (RPC) Broker. CPRS communicates to VistA through the RPC Broker.

1. Client - The top tier, or frontend, is the user interface (such as CPRS).
2. Server - The bottom tier, or backend, is the data source (a single VistA system).

MDWS evolved from the field development that Joe Gillon created with MDO at Ann Arbor Veterans Affairs Medical Center (VAMC). MDO is easier to implement/utilize than the traditional methods of accessing the VistA Legacy systems (such as the RPC Broker).

- MDO is a library of data structures with behaviors in the medical domain. It is an improvement over the Delphi RPC Broker by building in business rules to free other developers from implementing the same requirements in each application.
- MDO is written in C#.NET.
- MDO is capable of accessing a VistA system, enabling it to communicate directly with any VistA system and use all the standard local CPRS RPCs.
- MDO is capable of multi-site queries, allowing it to read data from all relevant VistA systems in parallel in the time it takes to receive data from one system.

The browser-based Electronic Medical Record Graphical User Interface (EMR GUI)/VistAWeb (VW) was developed to demonstrate MDO. VW not only demonstrated MDO, but also featured patient-centric data rather than geo-centric data. VW became a national Class 1 (C1) application in 2005.

VistAWeb Services (VWS) was developed to take MDO to Java 2 Platform, Enterprise Edition (J2EE), when it was realized that most clients can consume SOAP web services. Trying to produce J2EE web services proved painful, error-prone and time consuming. However, writing web services in the top level domain .NET was simple. VWS became a C#.NET web service exposing a pure Java library. Several web services were produced under VWS, as development moved toward a new set of web services with a new MDO written in C#.

The new service, MDWS, exposes MDO and provides transparent multi-site accessibility, while enforcing business rules. Although MDWS is not yet certified as C1 software, MDWS received a waiver from Systems Engineering for the C1 deployment of Suicide Hotline. MDWS will be the catalyst to make several VA mission critical systems operational in addressing compliance with VA requirements and White House/DHS mandates.

1. Healthcare-Associated Infection & Influenza Surveillance System (HAISS) program tools
2. Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE)
3. QcPathfinder
4. Bed Management Solutions (BMS) and other web-based applications

The current object set in MDWS focuses primarily on clinical information. Future development efforts may include other patient administrative areas, financial areas, etc. Much of the medical data comes from VistA,

- where data domain objects, such as Allergy, Medication, LabResult, etc., are created from the results of one or more VistA RPCs.
- where data comes from a relational source, the objects are created from recordsets.
- where data comes from XML sources the objects are created by parsing the Document Object Model (DOM).

Using MDO's data structures and behaviors, MDWS interacts with a variety of data sources. MDWS queries several VA data sources for clinical data.

1. All the VistA systems
2. Master Patient Index (MPI)
3. Structured Query Language (SQL)
4. Extensible Markup Language (XML)
5. Health Level 7 (HL7)
6. Some Planning System Support Group (PSSG) sources

MDWS is used by a variety of field-developed products and is a component of several notable C1 efforts implemented across the Enterprise.

1. Adverse Drug Reaction
<http://vhaannscml.v11.med.va.gov/trac/medora/wiki/Clients/ADR>
2. Apollo (CPRS Re-engineering (AViVA)
<http://trac.medora.va.gov/web/wiki/Projects/Apollo>
3. Athena
<http://trac.medora.va.gov/web/wiki/Clients/Athena>
4. BHIE
<http://trac.medora.va.gov/web/wiki/Clients/BHIE>
5. Chronic Disease Management
<http://trac.medora.va.gov/web/wiki/Clients/CDM>
6. Crisis Center (web service behind Suicide Hotline and Homeless Hotline)
<http://medora.sharepoint.med.va.gov/sites/crisiscenter/default.aspx>
7. Diversions
<http://medora.sharepoint.med.va.gov/sites/diversions/default.aspx>
8. Electrophysiology Reporting
Ann Arbor
9. EMERSE
<http://trac.medora.va.gov/web/wiki/Clients/EMERSE>

10. MOVE
<http://www.move.va.gov/Default.asp>
11. MyHealtheVet
<http://www.myhealth.va.gov/>
12. Mynapin (used in demonstrations)
<http://www.kabotintl.com/products.php?ProdCatID=7>
13. National Utilization Management Integration (NUMI)
<http://medora.sharepoint.med.va.gov/sites/utilizationmgt/default.aspx>
14. PatientFinder
<http://medora.sharepoint.med.va.gov/sites/PatientFinder/default.aspx>
15. Traumatic Brain Injury
<http://trac.medora.va.gov/web/wiki/Clients/TBI>

Deployment Overview

MDWS requires an application server with 2gb of RAM and with hard drive space as follows:

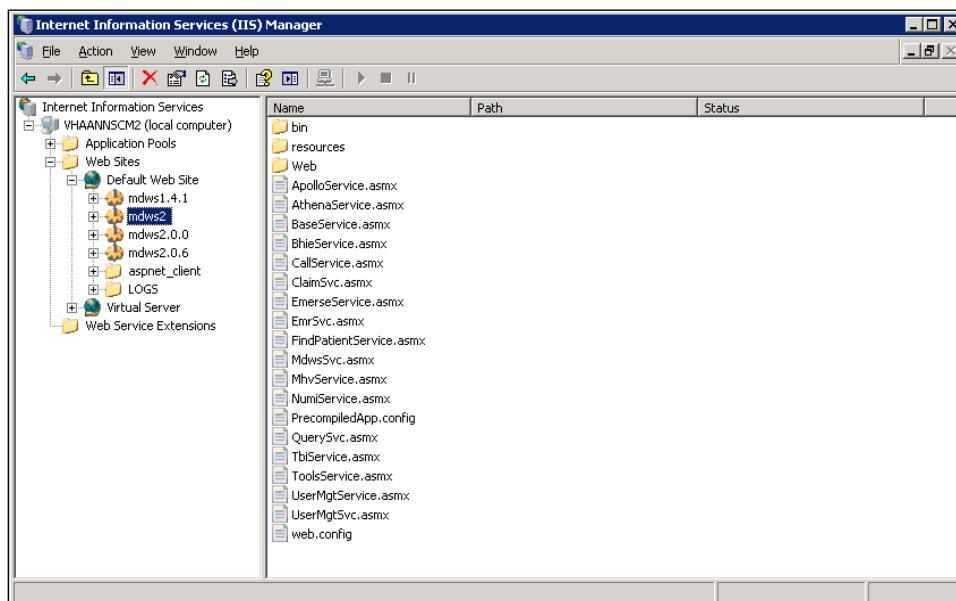
- MDWS Application: 50mb
- MDWS Database (if local): 200mb free space after SQL Server 2005 installation
- MDWS Database logs (if local), managed by an administrator, at least 300mb+

Prior to Installation

Requirements

- Windows Server 2003 (compatible with Server 2008 but some manual tweaks are needed)
- .NET Framework 3.5
- WSE 3.0 (Web Service Enhancements 3.0)
- SQL Server 2005 or 2008 (optional for non-BSE installations)
There is logging and some geographical data capabilities with an SQL database.
- IIS 6.0
 - a. IIS should be relatively close to default settings.
 - b. There should only be one web site.
MDWS is installed in a new virtual directory.
 - c. Multiple versions of MDWS can co-exist on the same server.

Note: Make sure ASP.NET 2.0 Web Service Extensions are enabled.



Internet Information Services (IIS) Manager

Configuring the MDWS Database

If you are a Broker Security Enhancement (BSE) client or you want the extra feature an SQL database affords, complete the following steps before installing MDWS.

1. Identify your SQL server and create a **MDWS** database.
2. Download the SQL scripts that coincide with your MDWS version from:
<ftp://downloads.medora.va.gov/mdws/SQL>
3. In the following order, execute the scripts on your new MDWS database:
 - a. `Session.sql`
 - b. `MdwsSessions?.sql`
 - c. `MdwsSessionRequests?.sql`
4. Create an SQL account with write privileges to your new MDWS database.
5. Specify the account and SQL information when you install MDWS.

Installation Instructions

Obtaining the Software

1. Use your favorite FTP client to download the MDWS.msi Windows installation file from <ftp://downloads.medora.va.gov/mdws>
Username: **anonymous**
Password: no password
2. Save the installation file to the server with the requirements on page 4.

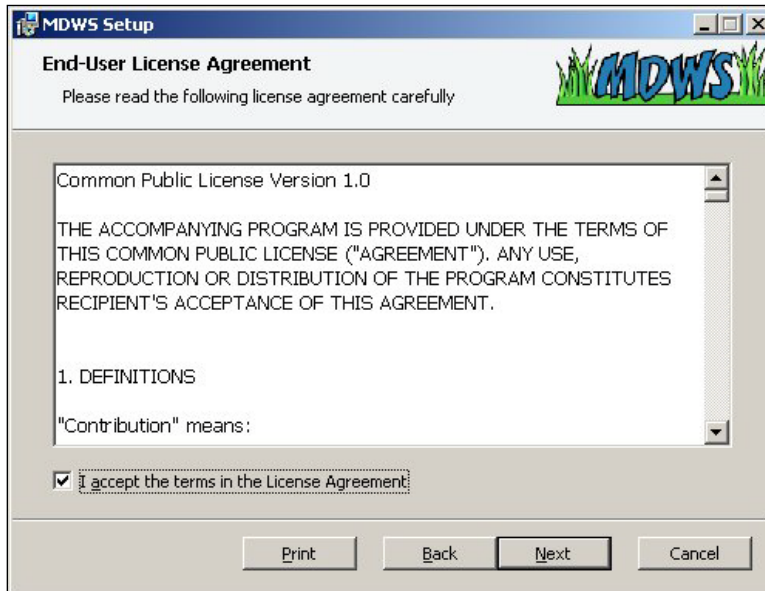
Installing MDWS

1. To begin the MDWS installation process, double-click the file.
2. On the Welcome to the MDWS Setup Wizard, click **Next**.



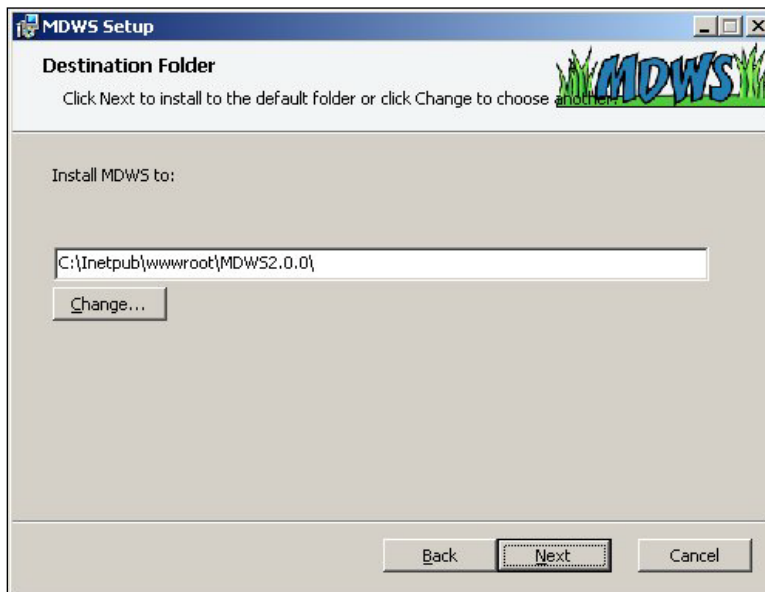
MDWS Setup Wizard - Welcome

3. To agree to the license agreement, select the *I accept the terms in the License Agreement* check box.
4. Click **Next**.



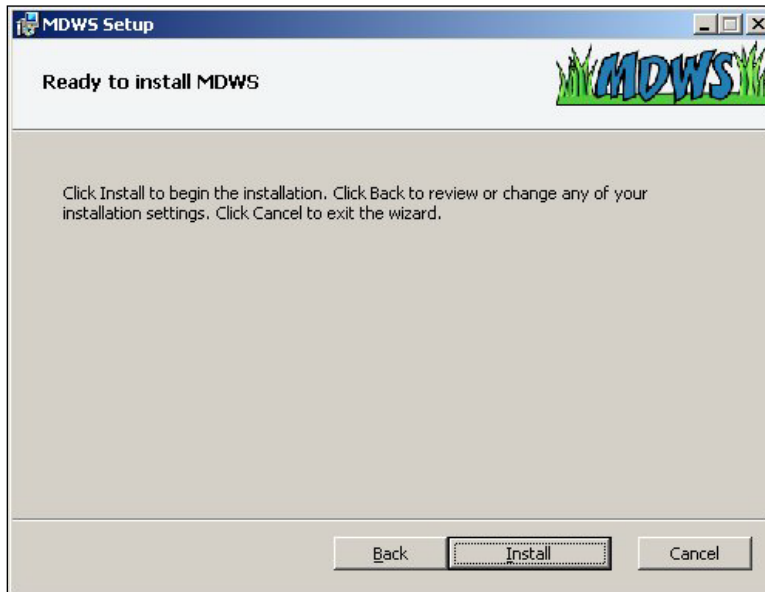
MDWS Setup Wizard – End-User License Agreement

5. Leave the default as your installation directory path; you can select your own installation directory path.
6. Click **Next**.




MDWS Setup Wizard – Destination Folder

7. To begin the installation of MDWS, click **Install**.



MDWS Setup Wizard – Ready to Install MDWS

8. The Administration Console will pop up as the installer completes.
9. Use the Administration Console to configure your installation settings.
 - If you are installing MDWS in a sandbox environment, you can use the **Check a Vista Connection** tool on the Administration Console to verify the listener is available.
 - To permanently save your test system, save the connection parameters in your **VhaSites.xml** file.


 Medical Domain Web Services

MDWS Configuration

Production Installation: True
 False

Log MDWS Sessions: True
 False

MDWS Sessions Log Level: info
 debug

Check a Vista connection

IP Address:

Port:

SQL Configuration

SQL Server Path:

SQL Server Database:

SQL Server Username:

SQL Server Password:

Facade Configuration

Facade Name:

Sites File Name:

Production: True
 False

Version:

Logged in as VHA11\VHAANNMewtoJ1

MDWS Setup Wizard – Administration Console

10. When the installation of MDWS is complete, click **Finish**.



MDWS Setup Wizard – Completed

After Installation

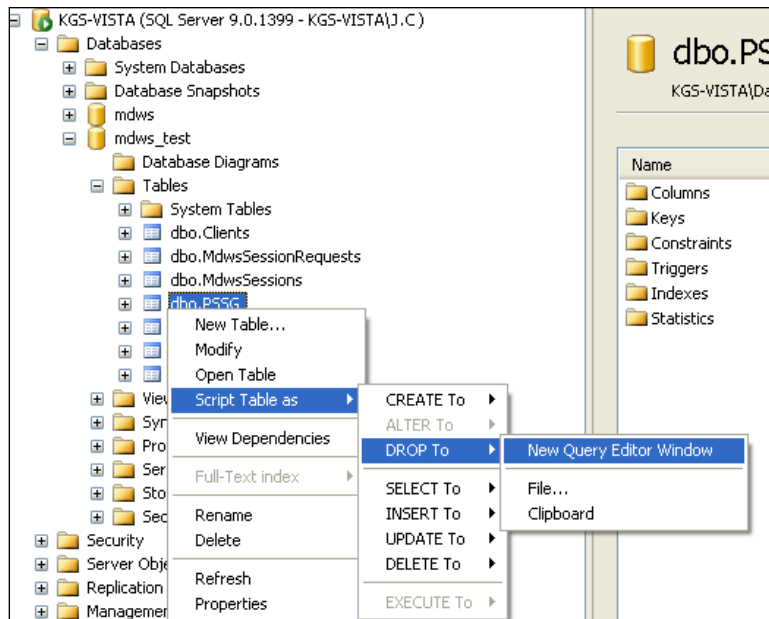
Updating the ZIPCodes Table

Note: MDWS uses a zip code database from a paid subscription to **ZIPCodeDownload**.

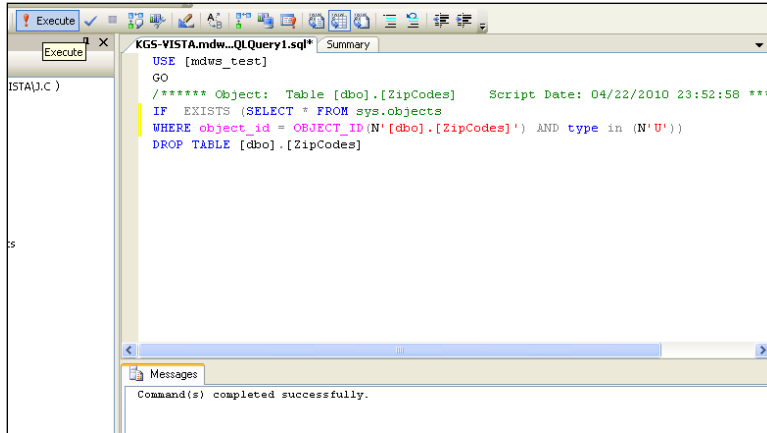
- If Class 1 support needs the subscription, contact the MDWS development team for the file.
- The file may be in an Excel or Access database, if so, alter the following steps to account for the different file format.

If the zip code data requires updating:

1. Obtain the update for the zip code data file from the source, **ZIPCodeDownload** at www.ZIPCodeDownload.com
2. Save the file to a local drive.
3. In the database tree view, follow the path: **Databases>mdws_test>Tables>dbo.PSSG**.
4. To drop the ZIPCodes table (or delete), select **Script Table as**, select **DROP To**, and select **New Query Editor Window**.
5. The Query Editor window displays.

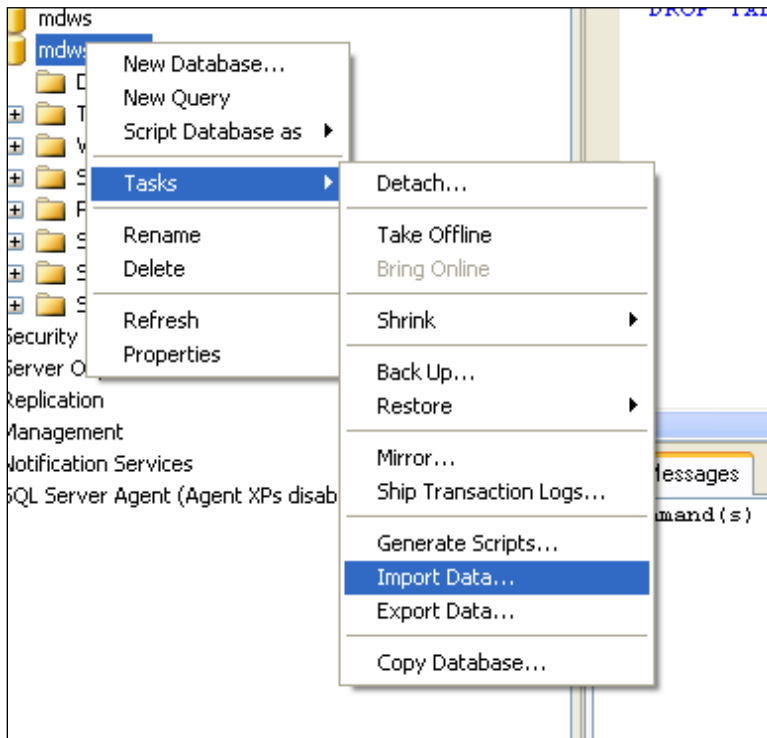


Screen capture of a Server database tree mapping to New Query Editor Window



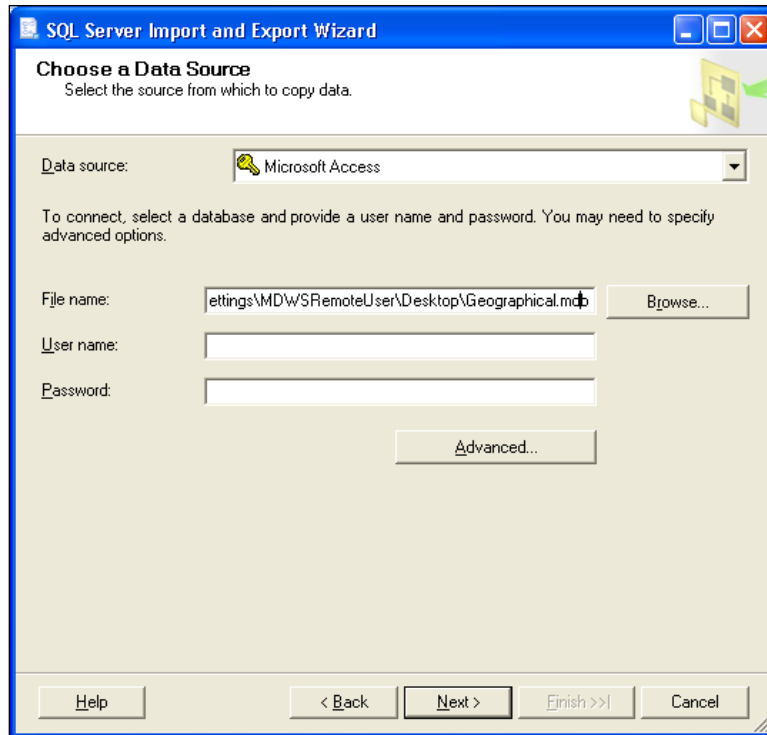
Screen capture of an Execute reply to a query

6. In the database tree view, select **mdws_test**, **Tasks**, and **Import Data...**



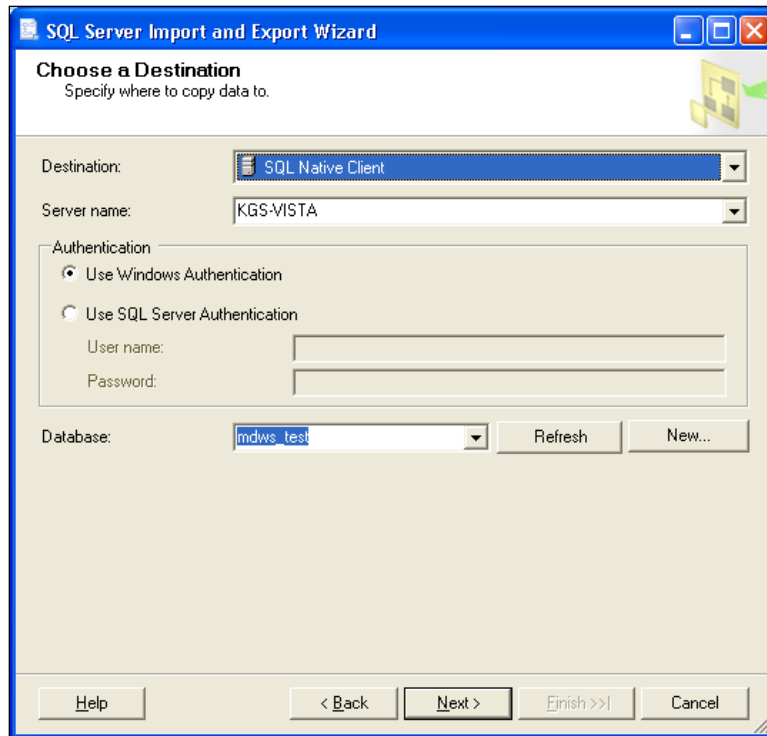
Screen capture of a Server database tree mapping to Import Data...

7. Open the SQL Server Import and Export Wizard.
 - a. On the Choose a Data Source window from the **Data source** drop-down text box, select **Microsoft Access**.
 - i. To locate the **File name** of the downloaded file, click **Browse**.
 - ii. Click **Next**.



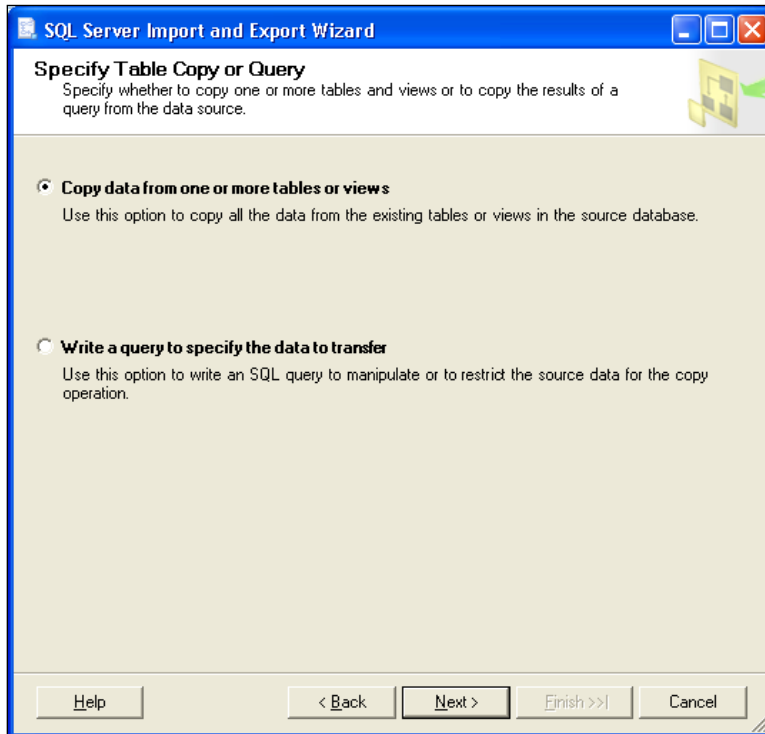
**Screen capture of the SQL Server Import and Export Wizard window
Choose a Data Source**

- b. On the Choose a Destination window from the **Destination** drop-down text box, select **SQL Native Client**.
 - i. Confirm that the **Server name** and **Database** are appropriate.
 - ii. In the Authentication section, confirm the **Use Windows Authentication** radio button is selected.
 - iii. Click **Next**.



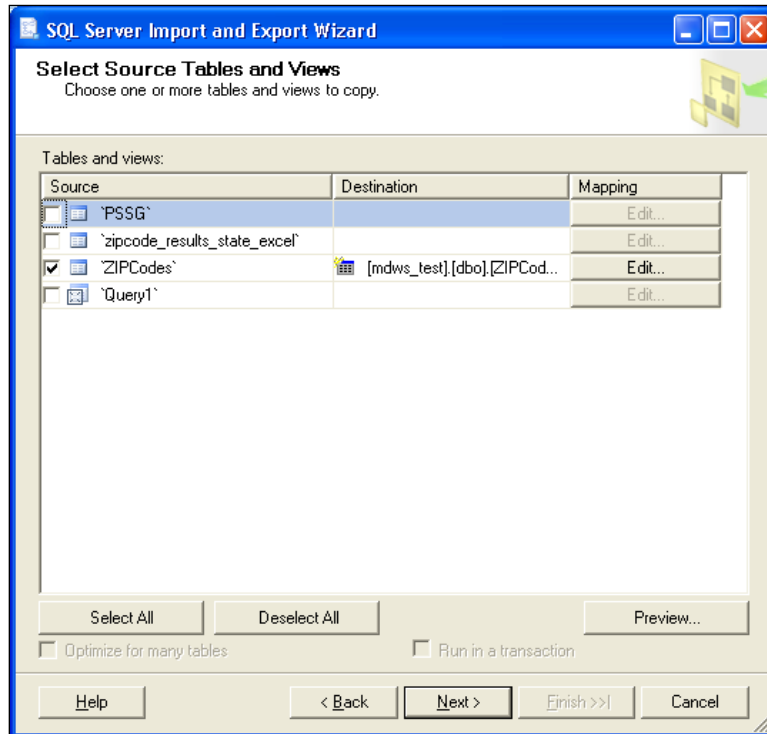
**Screen capture of the SQL Server Import and Export Wizard window
Choose a Destination**

- c. On the Specify Table Copy or Query window, select the **Copy data from the existing tables or views in the source database** radio button and click **Next**.



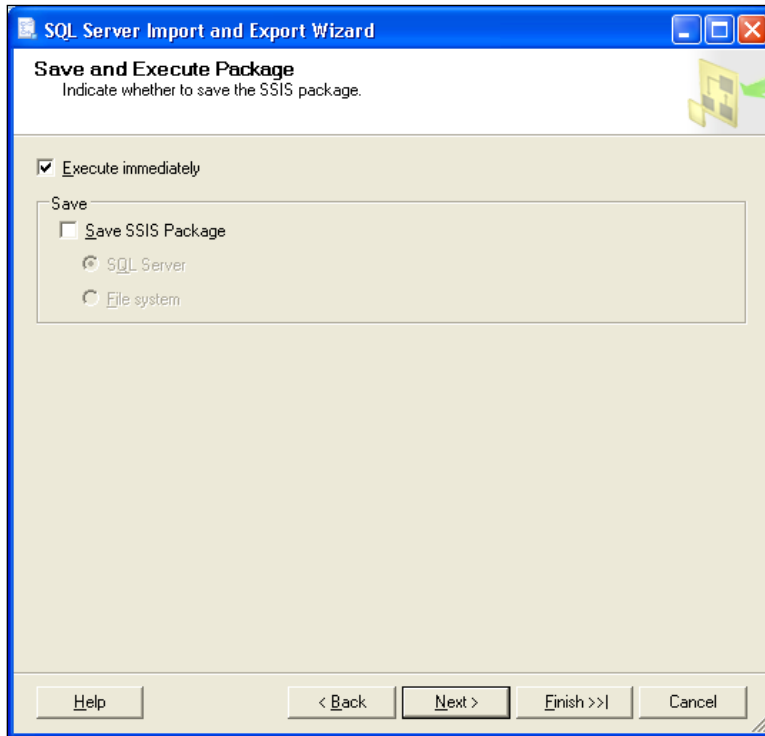
**Screen capture of the SQL Server Import and Export Wizard window
Specify Table Copy or Query**

- d. On the Select Source Tables and Views window, select the source table, **ZIPCodes** check box and click **Next**.



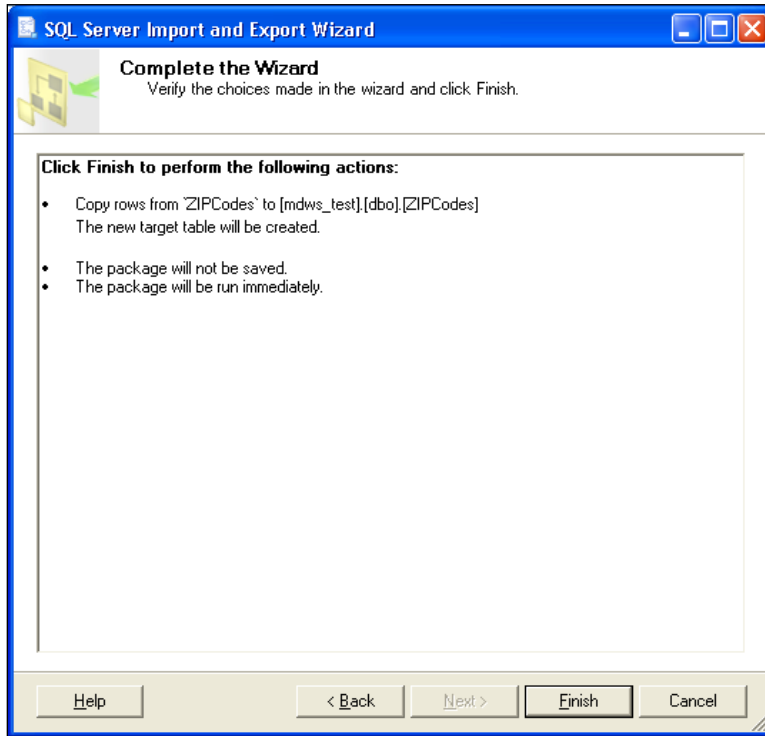
**Screen capture of the SQL Server Import and Export Wizard window
Select Source Tables and Views**

- e. On the Save and Execute Package window, select the **Execute immediately** check box and click **Next**.



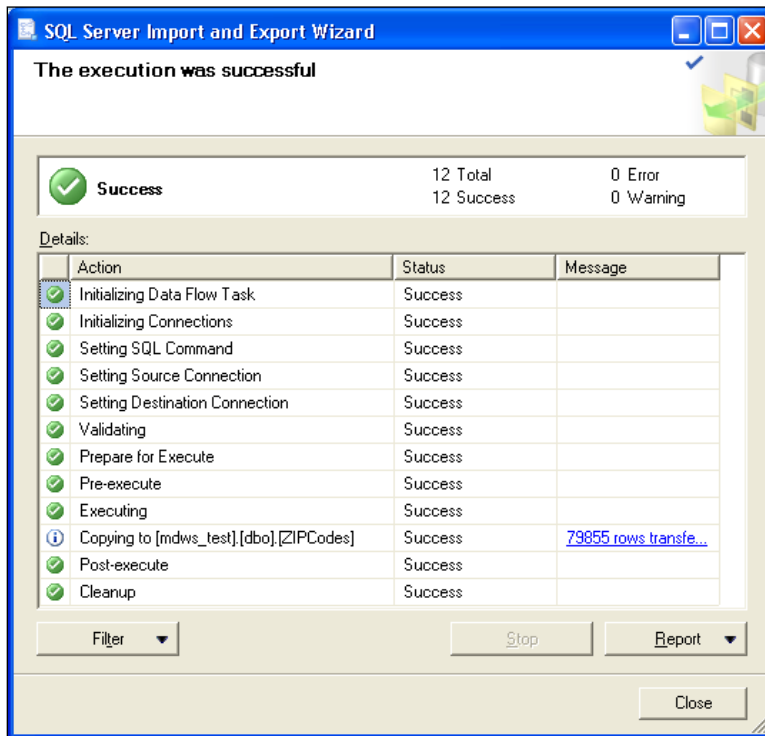
**Screen capture of the SQL Server Import and Export Wizard window
Save and Execute Package**

- f. On the Complete the Wizard window, review the source/target tables and click **Finish**.



**Screen capture of the SQL Server Import and Export Wizard window
Complete the Wizard**

- g. With successful transfer,  Success displays, all steps  complete. Click **Close**.



**Screen capture of the SQL Server Import and Export Wizard window
The execution was successful**

h. In the database tree view, verify the **dbo.ZIPCodes** table displays with data.

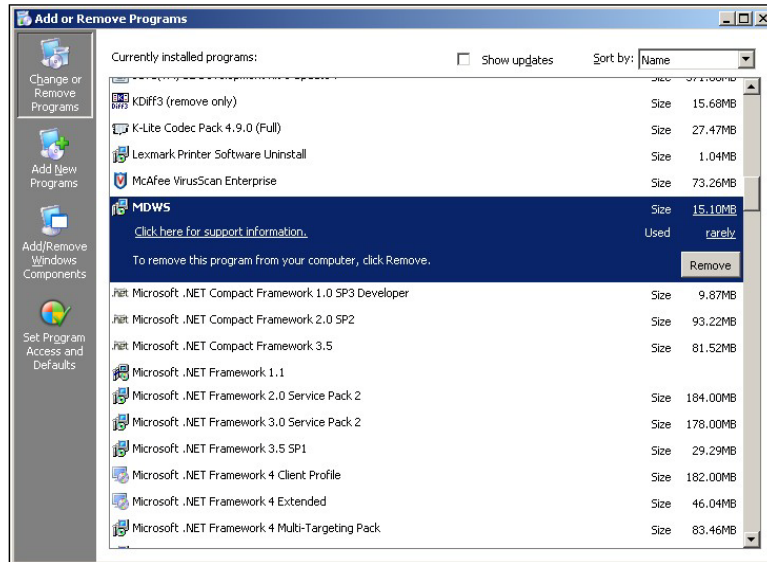
ZIPCode	ZIPType	CityName	CityType	CountyName	CountyFIPS
71291	S	West Monroe	D	Ouachita	22073
71292	S	Bawcomville	N	Ouachita	22073
71292	S	Brownsville	N	Ouachita	22073
71292	S	Cheniere	N	Ouachita	22073
71292	S	Lapine	N	Ouachita	22073
71292	S	Luna	N	Ouachita	22073
71292	S	Olinkraft	N	Ouachita	22073
71292	S	Siegle	N	Ouachita	22073
71292	S	West Monroe	D	Ouachita	22073
71294	P	West Monroe	D	Ouachita	22073
71295	S	Bushes	N	Franklin	22041
71295	S	Liddieville	N	Franklin	22041
71295	S	Swampers	N	Franklin	22041
71295	S	Winnsboro	D	Franklin	22041
71301	S	Alex	N	Rapides	22079

**Screen capture of a Server database tree
Table – dbo.ZIPCodes tab**

Troubleshooting MDWS

Uninstalling MDWS

1. Click **Start**.
2. Select **Control Panel**.
3. Double-click **Add or Remove Programs**.
4. Select **MDWS**.
5. Click **Remove**.



Control Panel>Add or Remove Programs

Normal Procedures

In general, to troubleshoot any problem, check the following sources:

1. Browse to the local web services and make sure the Web Service Definition Language (WSDL) displays.
2. Run the connection test page.

Sample of an Error Message

from MDWS to a requesting client application

```
<?xml version="1.0" encoding="utf-8" ?>
- <TaggedInpatientStayArray xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://mdws.medora.va.gov/EmrSvc">
- <fault>
  <type />
  <message>There are no open connections</message>
  <stackTrace />
  <suggestion />
</fault>
<count>0</count>
</TaggedInpatientStayArray>
```

Production Issue History

MDWS has never had a production problem. MDWS shared an application pool at the C3 level with VistAWeb. Problems with the VistAWeb application caused a brief loss of connectivity for MDWS clients, until IIS was restarted.

Future productions issues are to be added to this document in the following table.

Date	Cause	Resolution

Potential Troubleshooting Steps

1. In IIS, recycle the application pool in which MDWS resides.
2. Restart IIS.
3. Look for server events or server changes (anti-virus, group policies, etc.).

Failover MDWS Deployment

The client application(s) are responsible for pointing to a failover MDWS deployment and is not directly related to restoring a failed MDWS instance. The client can accomplish the failover in two ways: automated and manual.

Automated Solution

The automated solution is more complex from a software development standpoint, but has the advantage of being a near instantaneous resolution to a primary MDWS failure.

In an automated failover environment, when the primary endpoint no longer responds to requests, the client application switches from the primary well known MDWS endpoint to a well known backup or failover MDWS endpoint.

- It is imperative the client application support team is made aware a switch was made to a backup service.
- The client application developer must architect this notification into their software.

Manual Solution

In a manual failover environment, when the primary well known MDWS endpoint becomes unavailable, the client application developer must manually modify their code or configuration files.

- The client application support team can be made aware of the failure automatically by including code that notifies the necessary personnel when the primary MDWS instance becomes unavailable.
- The client application support team usually settles on the simplest solution, which is to wait for users to report the failure. Then the support team begins troubleshooting, determines the failure is MDWS related, and points the client application to a well-known failover endpoint.

Symptoms, Diagnoses, and Possible Solutions

<p>1.</p>	<p>Symptom</p> <p>MDWS WSDL not viewable locally</p> <p>http://localhost/mdws/CallService.asmx</p> <p>404 Page Not Found</p> <hr/> <p>Diagnoses and Solutions</p> <p>IIS Default web site configuration likely incorrect</p>
<p>2.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>3.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>4.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>5.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>6.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>7.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>
<p>8.</p>	<p>Symptom</p> <p>This room left blank intentionally for future solutions</p> <hr/> <p>Diagnoses and Solutions</p> <p>This room left blank intentionally for future solutions</p>

Production Operations Manual Section

Per Enterprise Testing Services feedback, there is enough overlap between the POM and the Installation Guide that the questions can be combined into the Installation Guide. Questions posed by ETS with input from the Class 1 and Class 3 development teams follows.

1. **Present physical and logical system descriptions**

MDWS is employing a distributed approach and each site has its own unique hardware.

2. **Address vendor, version and license information (if applicable) associated with COTS (zip code) components**

The licensing aspect is N/A. The current method of obtaining the COTS zip code file via subscription service will continue to be controlled through Ann Arbor and Joel Mewton when MDWS goes Class 1.

3. **Address procedures for monitoring daily process flows and threshold encroachments (if applicable)**

A web service is a web application, so every transaction is recorded in the IIS log. MDWS does not have details like user or patient IDs or type of data fetched, but for daily workload, it will be sufficient.

4. **Address any scheduled maintenance activities (if applicable)**

N/A

There are no scheduled maintenance activities. Backups are done.

5. **Address exception handling and escalation procedures (if applicable)**

The error messages are designed to be understood by Development staff. There is no need for error messages that can be read by end-users or Tier 1/2 support staff. If Development staff at any of the sites has questions, they will contact Joel Mewton in Ann Arbor.