Medical Domain Web Services (MDWS)

Version 2.0

C3-C1 Conversion Project

Systems Management 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**
February 2010	1.0	Initial version for v 2.0	J Confer L Harmon K Bonner C Beynon
May 2010	1.1	Added content	J Confer
July 2010		Changed dates	C Beynon
September 2010		Changed dates to September	C Beynon
September 2010	1.2	Clarified that this project is for a Class 1 conversion analysis	J Rogers
October 2010	1.2	Changed dates to October	CBeynon
December 2010	1.2	Changed dates to January 2011	CBeynon
May 2011	1.3	Changed dates to May 2011 Added (MWVS*2) namespace	CBeynon
June 2011	1.4	Updated the SMG with comments from JM/Medora	CBeynon
July 2011	1.5	Prepped for national release, ESE Checklist	CBeynon
September 2011	1.6	Changed dates to September 2011 for September release	CBeynon

Table of Contents

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.

The purpose behind this project is to evaluate the feasibility of and make recommendations to Enterprise Infrastructure Engineering (EIE) and VA management about taking the MDWS service to a Class 1 status. As a Class 1 service, MDWS will be fully certified, nationally released and supported by the Office of Enterprise Development.

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 (HAIISS) 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://vhaannscm1.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/BHI
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

Orientation

MDWS 2.0 Overview

The Medical Domain Web Services (MDWS) 2.0 is a middleware service that provides data services to other internal VA applications. It is written in C#, uses the Microsoft 2.0 .NET Framework, and runs on Microsoft Internet Information Services 6.0. It allows applications written in any language to access specific data resources through its external SOAP interface.

MDWS consists of external web methods, a set of data access objects for its resources termed Medical Data Objects (MDO), and a local SQL Server Database for logging and configuration tables.

- The external web methods adhere to SOAP and are defined by a WSDL that changes as little as possible. When the WSDL changes, so do the client applications dependent upon it.
- The MDO contain the data access and business rules. The objects connect to various data sources and return patient data.
- The SQL Server 2005 Database requires periodic updates to incorporate zip code and geographic data that MDWS requires.

Document Overview

This manual provides information on the management of MDWS 2.0 database and application server(s). It contains detailed information on .NET application server management, institution files, security, logging, and troubleshooting. Its intended audience includes server administrators and specialists at VHA hosting facilities like the intended C1 hosting at AITC.

The installation and maintenance instructions presented here assume the use of Windows as the client operating system. MDWS 2.0 does not support any flavor of *nix and is a Microsoft only platform.

- **Terminology**
  The terms site and VISN are used interchangeably to refer to VHA sites; however, while all VISNs are considered sites, there are some sites that are not a VISN.
- **Text Conventions**
  File names and directory names are set off from other text using bold font (e.g., **config.xml**). Bold is also used to indicate GUI elements, such as tab, field, and button names (e.g., press **Delete**).

Additional Resources

MDWS Web Site

The MDWS website summarizes MDWS architecture and functionality and presents status updates:
http://medora.sharepoint.med.va.gov/sites/mdws/default.aspx

MDWS Developer

Joel Mewton – Ann Arbor VAMC

MDWS Documentation Set

The following documents are provided in the MDWS 2.0 documentation set:

- *MDWS 2.0 Installation Guide*: Provides detailed instructions for setting up, installing, and configuring the MDWS 2.0 application and database on Windows 2003 Server. Its intended audience includes server administrators and IT specialists, and C# application developers.
- *MDWS 2.0 Systems Management Guide*: Contains detailed information on IIS 6.0 application server management, institution files, security, logging, and troubleshooting.
- *MDWS 2.0 Developer's Guide*: Contains detailed information about workstation setup, build procedures, institution files, subscribed applications, facades, and other information pertinent to developers learning the MDWS application.
- *MDWS 2.0 Release Notes*: Lists all new features included in each MDWS 2.0 release.

Microsoft Internet Information Services 6.0

MDWS 2.0 was tested and is supported on Microsoft IIS 6.0 only. Microsoft product documentation is found at the following website:
http://technet.microsoft.com/en-us/windowsserver/default.aspx.

Database Information

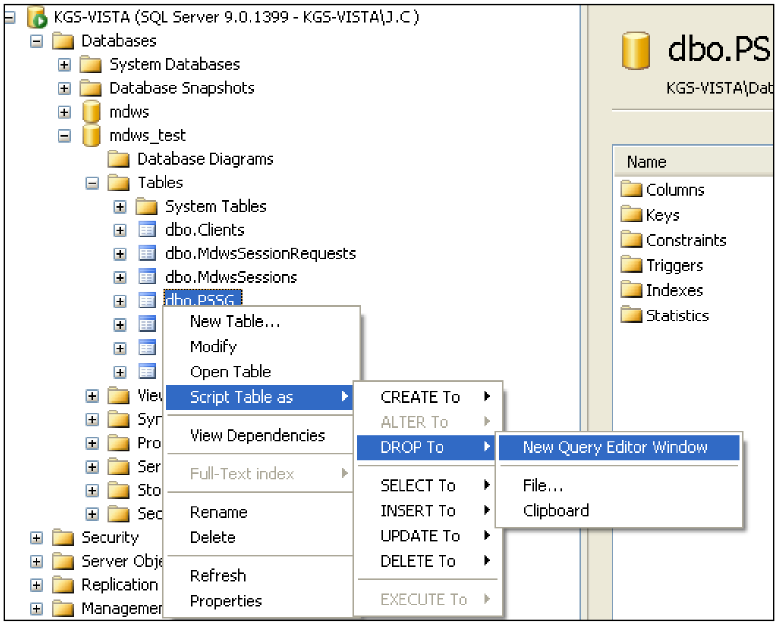
Updating the PSSG Table

If the Planning Systems Support Group (PSSG) data requires updating:

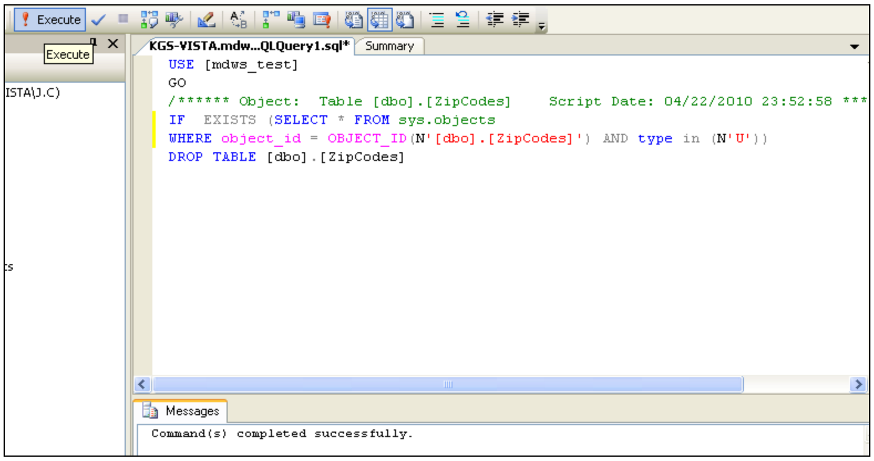
1. Obtain the PSSG data file from the link **Complete zip code extract** at http://vaww.pssg.med.va.gov/PSSG/search_zipcode4.html
2. Save the file to a local drive.

> **Note:** The exact filename varies, because the group that provides the file updates it. Currently, the format is **ZIP_mm_yyyy**.

3. In the database tree view, follow the path: Databases>mdws_test>Tables>dbo PSSG
4. To drop the PSSG 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

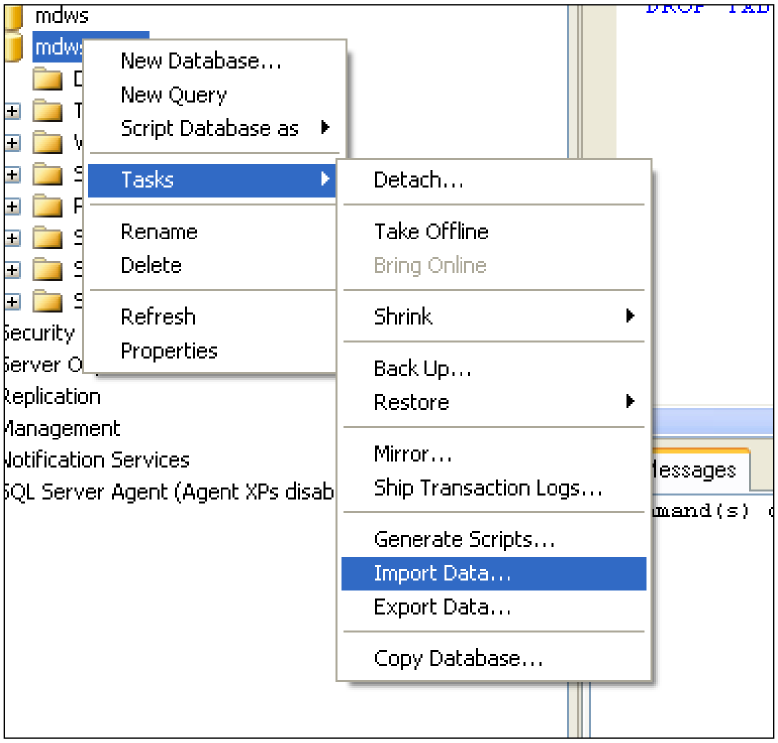


```
USE [mdws_test]
GO
/****** Object:  Table [dbo].[ZipCodes]    Script Date: 04/22/2010 23:52:58 ***
IF  EXISTS (SELECT * FROM sys.objects
WHERE object_id = OBJECT_ID(N'[dbo].[ZipCodes]') AND type in (N'U'))
DROP TABLE [dbo].[ZipCodes]
```

```
Command(s) completed successfully.
```

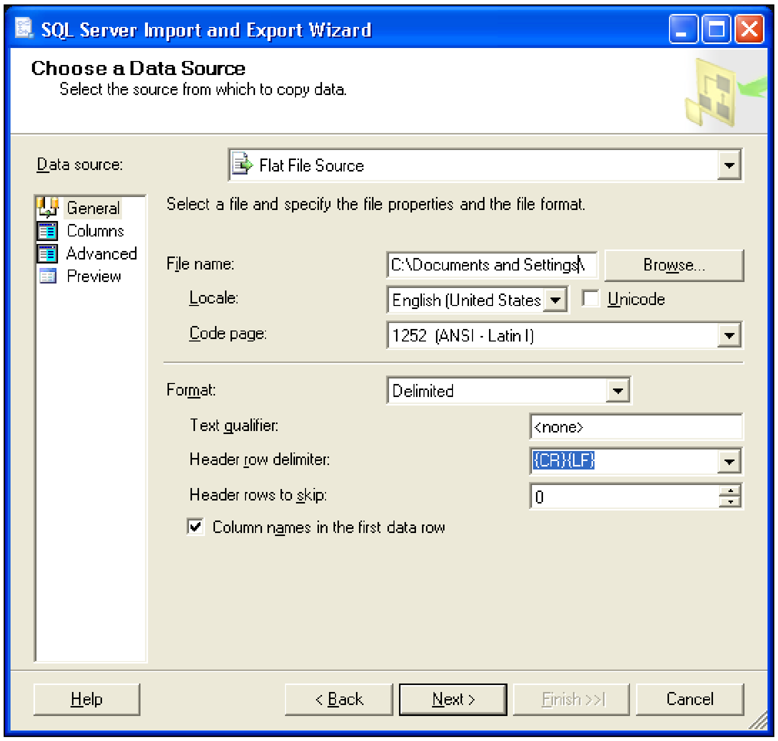
Screen capture of an Execute reply to a query

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 **Flat File Source**.
      i. To locate the File name of the downloaded file, click Browse.
      ii. In the Format section, select the Column names in the first data row check box.
      iii. Click Next.



Choose a Data Source

Select the source from which to copy data.

Data source: Flat File Source

Select a file and specify the file properties and the file format.

File name: C:\Documents and Settings\

Locale: English (United States

□ Unicode

Code page: 1252 (ANSI - Latin I)

Format: Delimited

Text qualifier: <none>

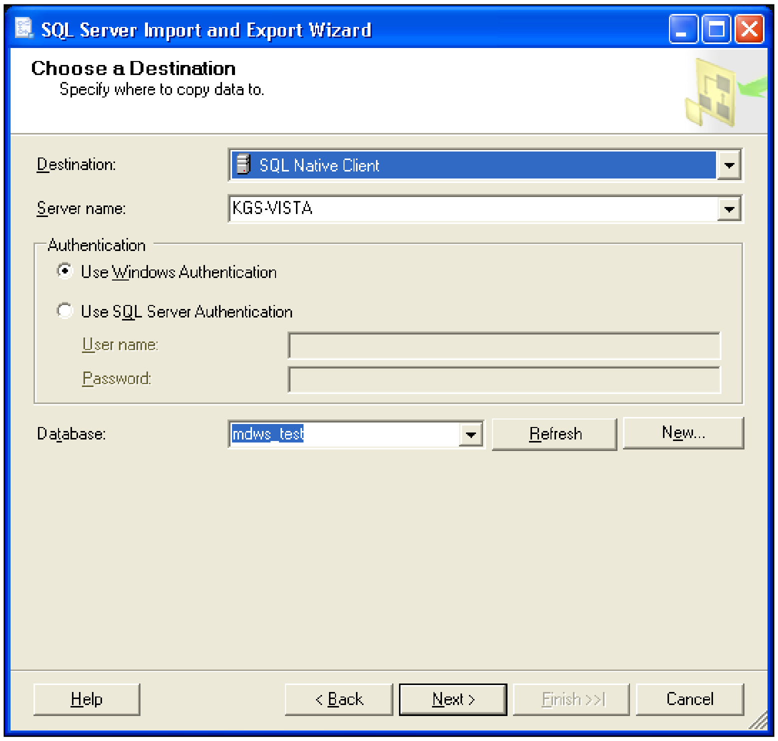
Header row delimiter: {CR}{LF}

Header rows to skip: 0

☑ Column names in the first data row

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**.



SQL Server Import and Export Wizard

Choose a Destination

Specify where to copy data to.

Destination: SQL Native Client

Server name: KGS-VISTA

Authentication

● Use Windows Authentication

○ Use SQL Server Authentication

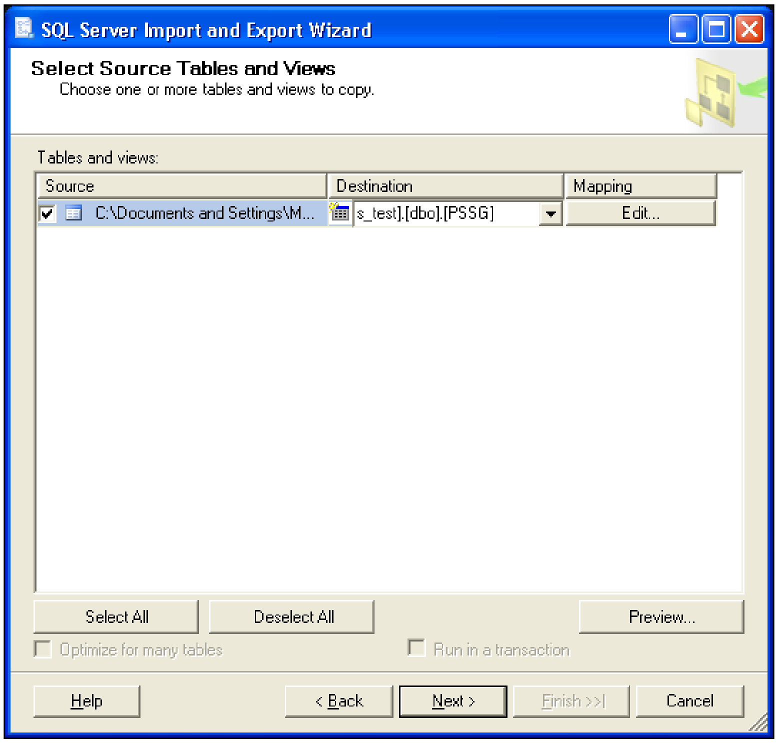
User name:

Password:

Database: mdws_test

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

c. On the Select Source Tables and Views window, rename the **Destination** table to **PSSG** and click **Next**.



SQL Server Import and Export Wizard

Select Source Tables and Views

Choose one or more tables and views to copy.

Tables and views:

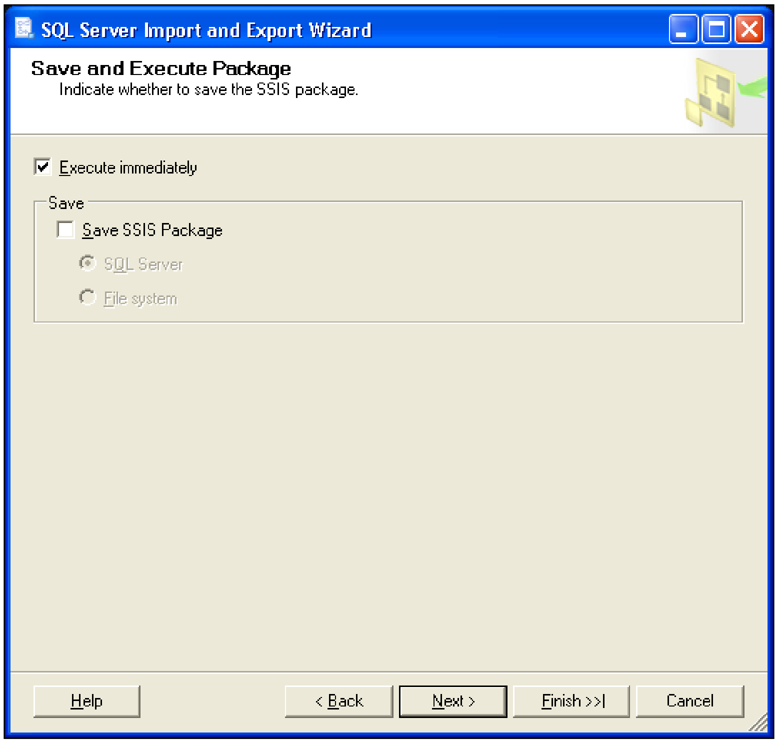
Source	Destination	Mapping
☑ C:\Documents and Settings\M...	s_test].[dbo].[PSSG]	Edit...

□ Optimize for many tables

□ Run in a transaction

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

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


SQL Server Import and Export Wizard

Save and Execute Package

Indicate whether to save the SSIS package.

☑ Execute immediately

Save

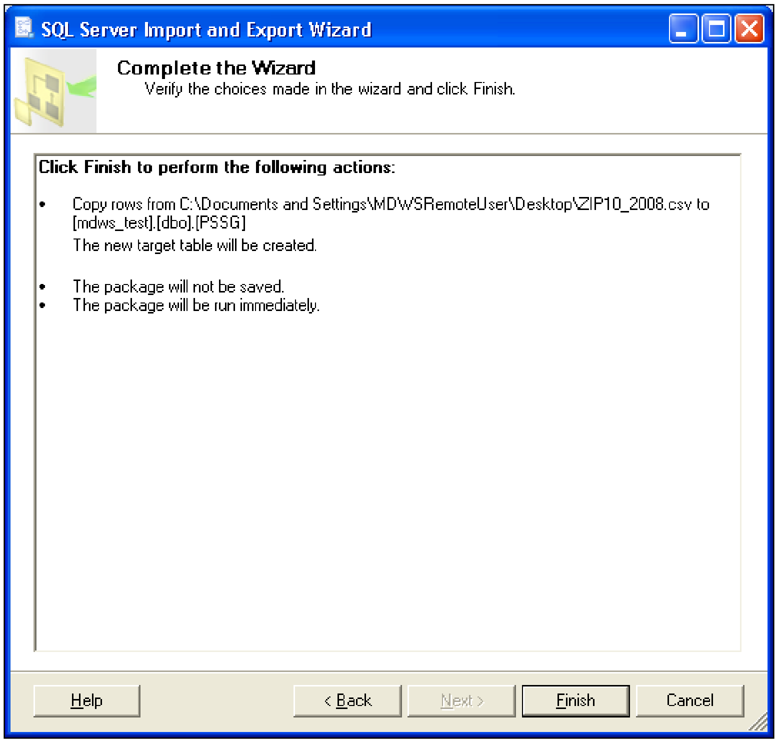
□ Save SSIS Package

● SQL Server

○ File system

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

e. On the Complete the Wizard window, review the source/destination locations and click **Finish**.



Complete the Wizard

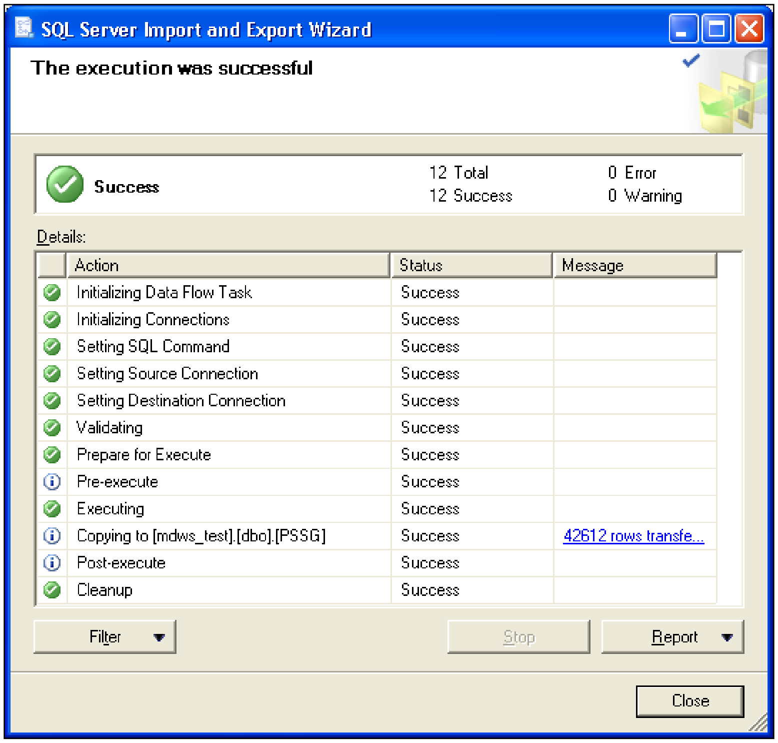
Verify the choices made in the wizard and click Finish.

Click Finish to perform the following actions:

- Copy rows from C:\Documents and Settings\MDWSRemoteUser\Desktop\ZIP10_2008.csv to [mdws_test].[dbo].[PSSG]
  The new target table will be created.
- The package will not be saved.
- The package will be run immediately.

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

f. With successful execution, ✓ Success displays, all steps ✓ complete. Click **Close**.



The execution was successful

Success

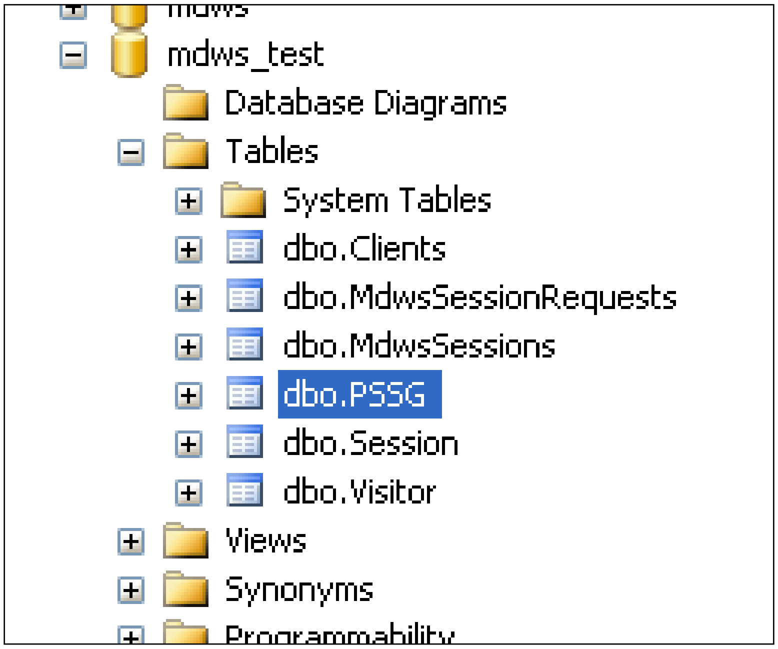
12 Total
12 Success
0 Error
0 Warning

Details:

	Action	Status	Message
✓	Initializing Data Flow Task	Success	
✓	Initializing Connections	Success	
✓	Setting SQL Command	Success	
✓	Setting Source Connection	Success	
✓	Setting Destination Connection	Success	
✓	Validating	Success	
✓	Prepare for Execute	Success	
ⓘ	Pre-execute	Success	
✓	Executing	Success	
ⓘ	Copying to [mdws_test].[dbo].[PSSG]	Success	42612 rows transfe...
ⓘ	Post-execute	Success	
✓	Cleanup	Success	

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

8. In the database tree view, verify the **dbo.PSSG** table displays.



Screen capture of a Server database tree

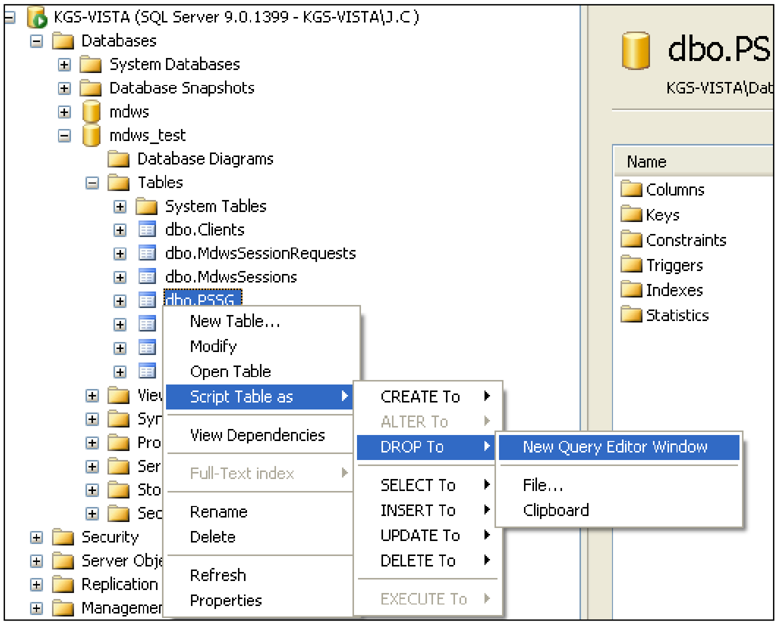
Updating the ZIPCodes Table

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

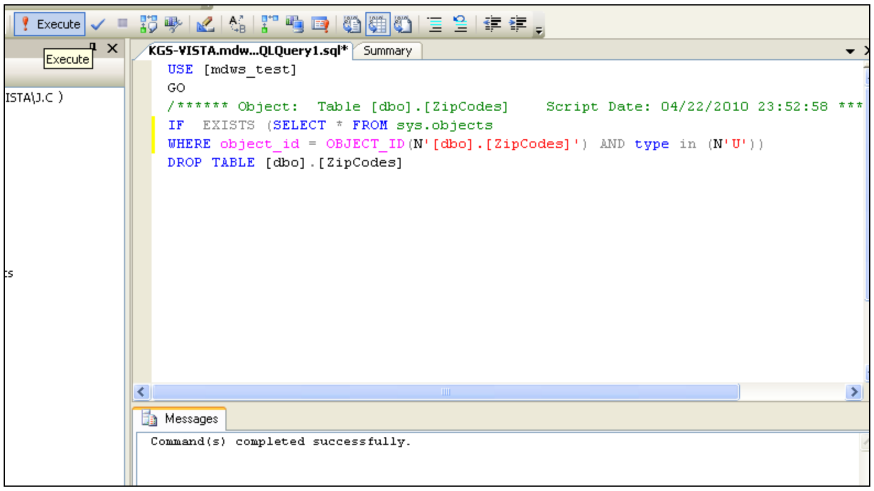
- If Class 1 support needs the subscription, contact the MDWS development team for the file.
- The file may be in an Excel or Access database, if so, alter the following steps to account for the different file format.

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

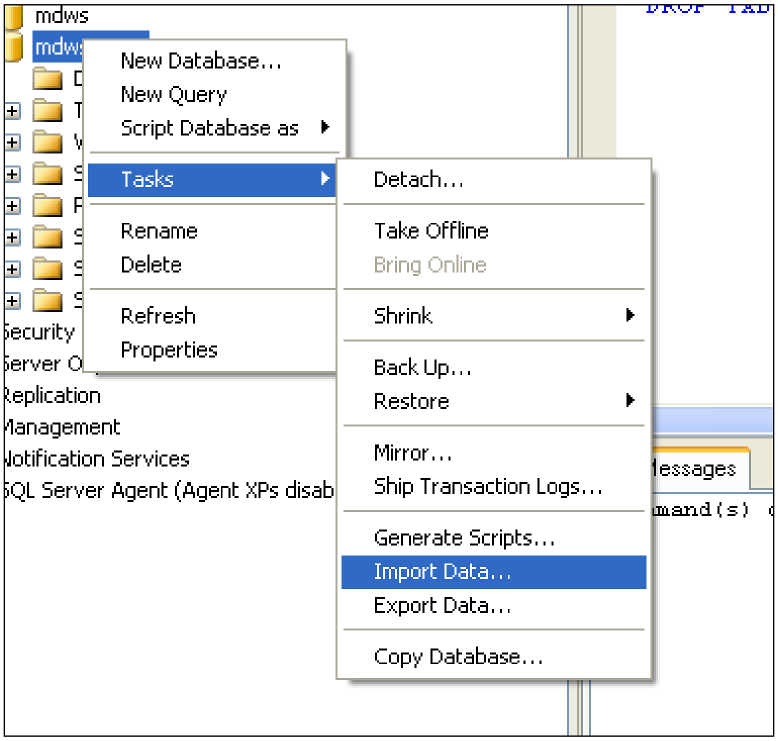


```
USE [mdws_test]
GO
/****** Object:  Table [dbo].[ZipCodes]    Script Date: 04/22/2010 23:52:58 ***
IF  EXISTS (SELECT * FROM sys.objects
WHERE object_id = OBJECT_ID(N'[dbo].[ZipCodes]') AND type in (N'U'))
DROP TABLE [dbo].[ZipCodes]
```

```
Command(s) completed successfully.
```

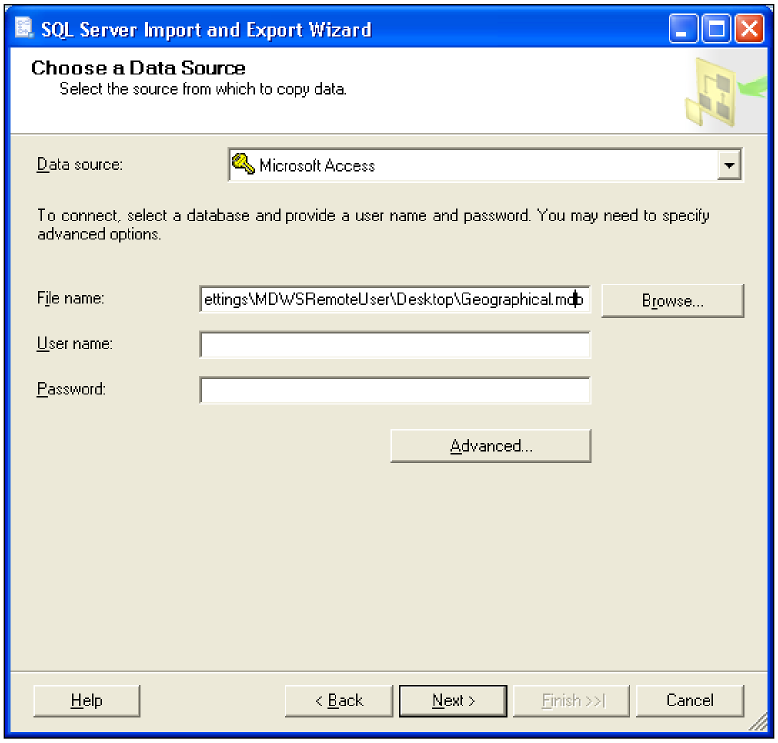
Screen capture of an Execute reply to a query

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**.



SQL Server Import and Export Wizard

Choose a Data Source

Select the source from which to copy data.

Data source: Microsoft Access

To connect, select a database and provide a user name and password. You may need to specify advanced options.

File name: ettings\MDWSRemoteUser\Desktop\Geographical.mdb

User name:

Password:

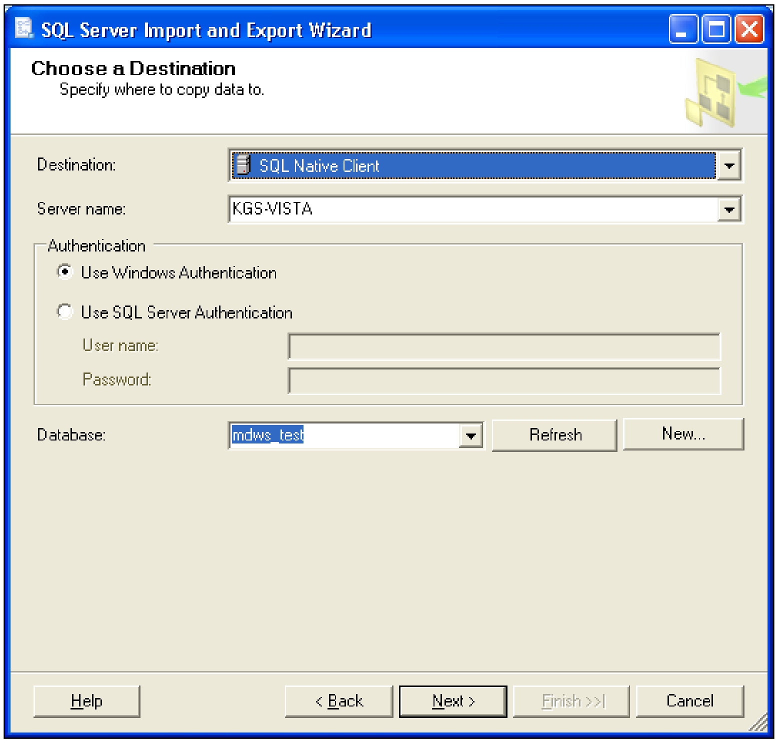
Screen capture of the SQL Server Import and Export Wizard window Choose a Data Source

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**.



Specify where to copy data to.

Destination: SQL Native Client

Server name: KGS-VISTA

Authentication

● Use Windows Authentication

○ Use SQL Server Authentication

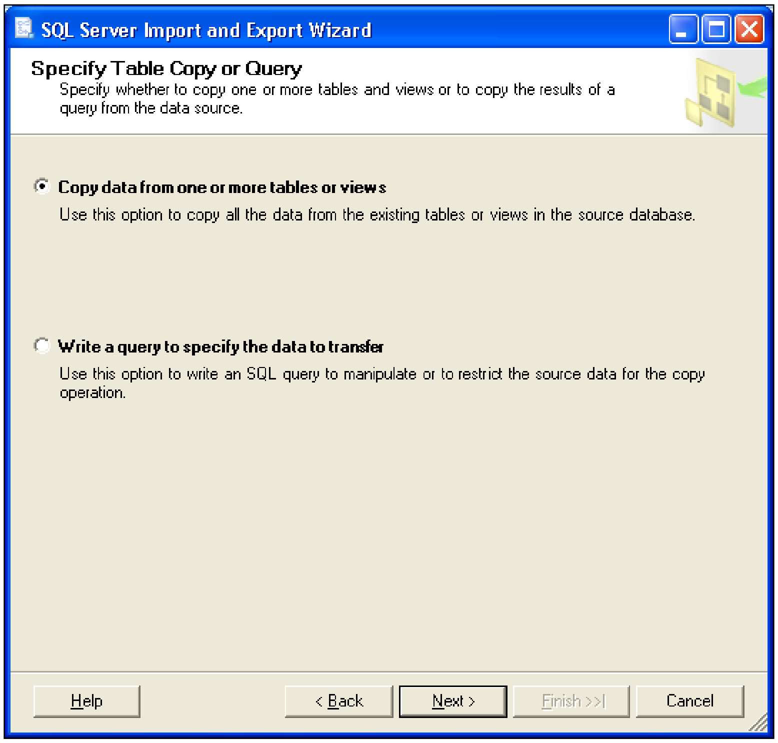
User name:

Password:

Database: mdws_test

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**.



Specify Table Copy or Query

Specify whether to copy one or more tables and views or to copy the results of a query from the data source.

● Copy data from one or more tables or views

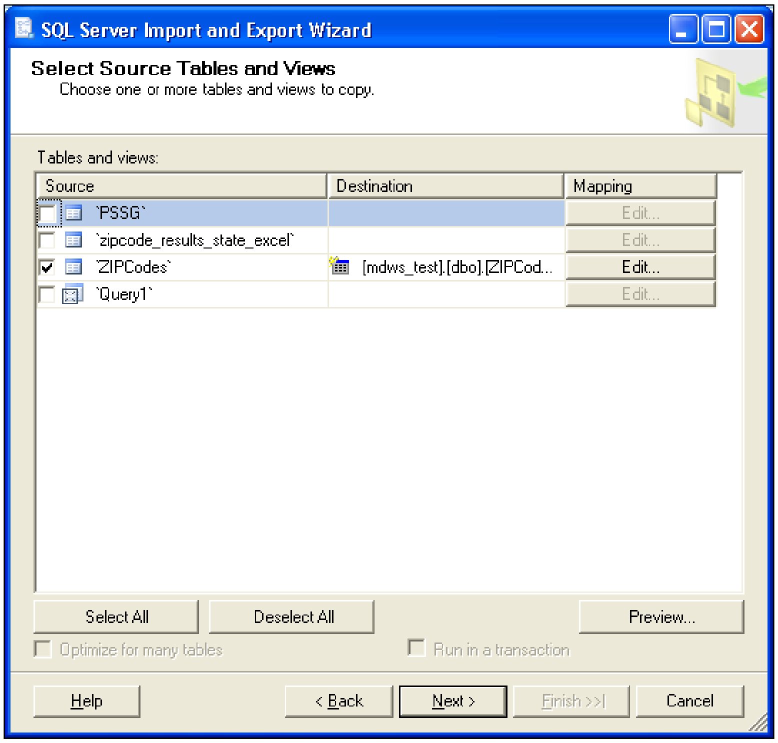
Use this option to copy all the data from the existing tables or views in the source database.

○ Write a query to specify the data to transfer

Use this option to write an SQL query to manipulate or to restrict the source data for the copy operation.

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**.



SQL Server Import and Export Wizard

Select Source Tables and Views

Choose one or more tables and views to copy.

Tables and views:

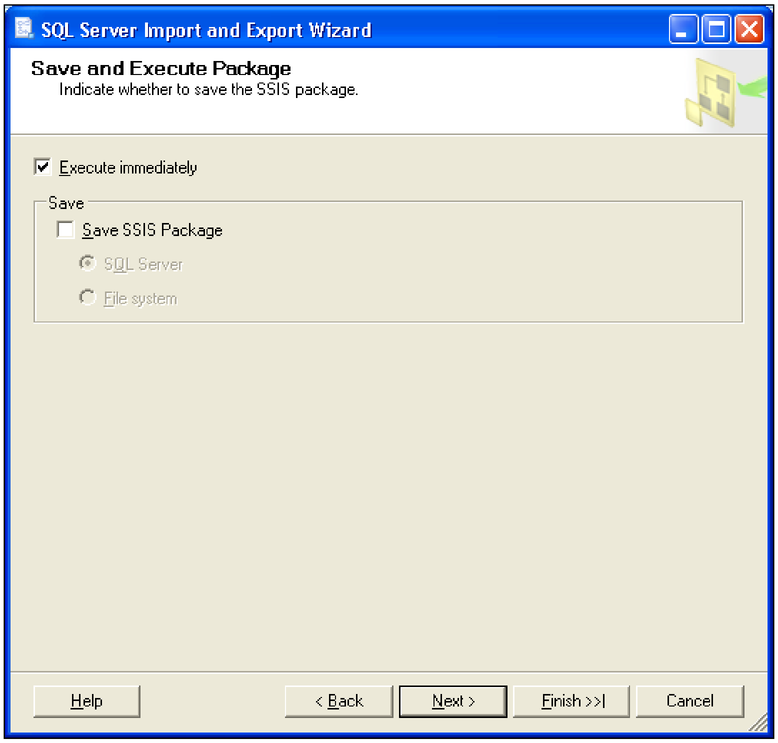
	Source	Destination	Mapping
□	`PSSG`		Edit...
□	`zipcode_results_state_excel`		Edit...
☑	`ZIPCodes`	[mdws_test].[dbo].[ZIPCod...	Edit...
□	`Query1`		Edit...

□ Optimize for many tables

□ Run in a transaction

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**.



SQL Server Import and Export Wizard

Save and Execute Package

Indicate whether to save the SSIS package.

☑ Execute immediately

Save

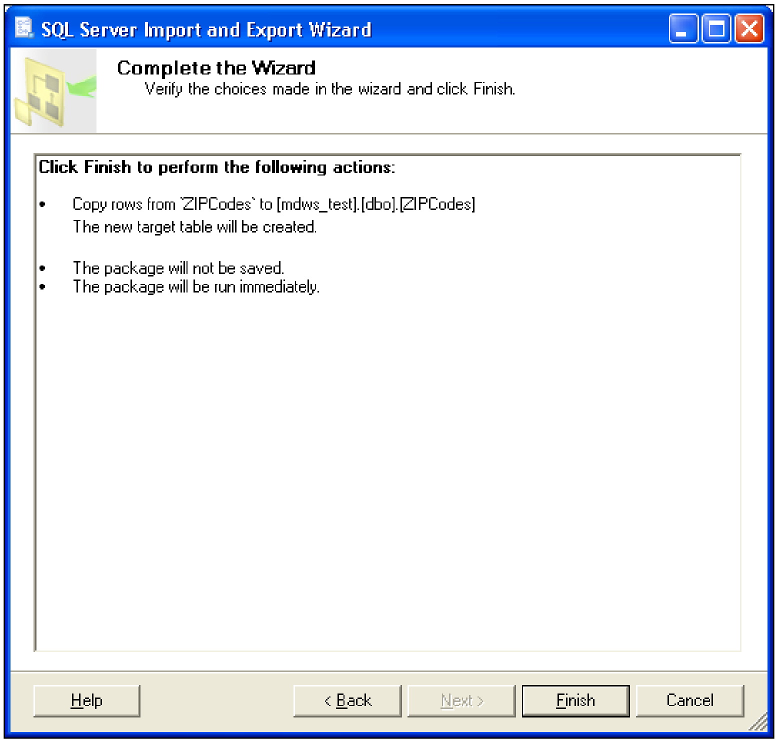
☐ Save SSIS Package

● SQL Server

○ File system

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**.



Complete the Wizard

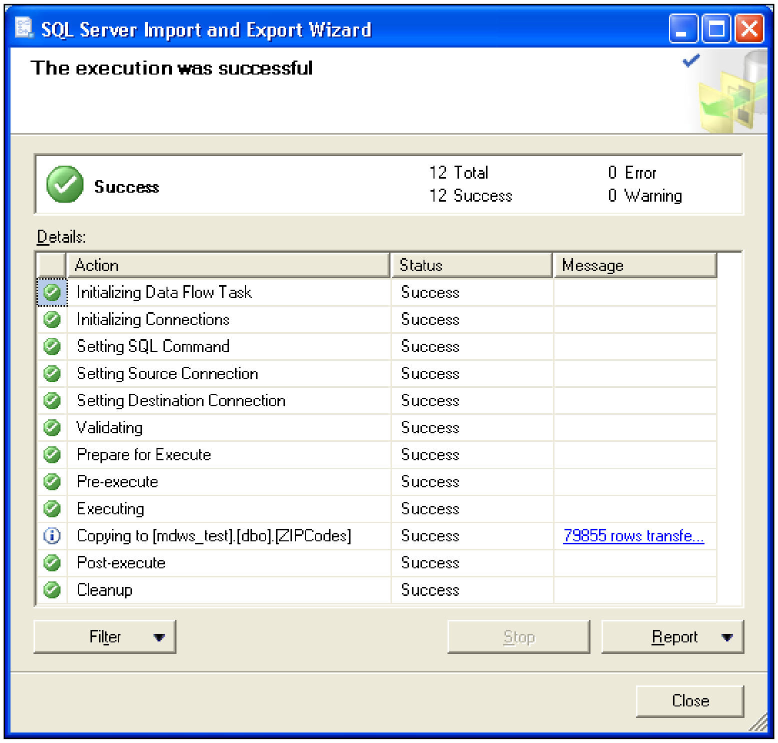
Verify the choices made in the wizard and click Finish.

Click Finish to perform the following actions:

- Copy rows from `ZIPCodes` to [mdws_test].[dbo].[ZIPCodes]
  The new target table will be created.
- The package will not be saved.
- The package will be run immediately.

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**.



The execution was successful

✓ Success

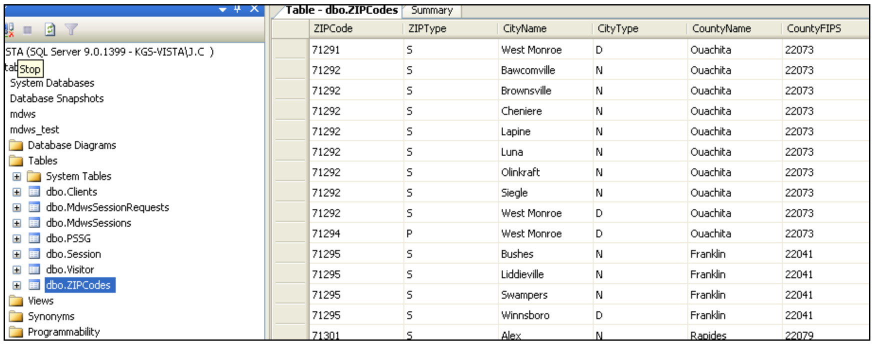
12 Total 0 Error
12 Success 0 Warning

Details:

	Action	Status	Message
✓	Initializing Data Flow Task	Success	
✓	Initializing Connections	Success	
✓	Setting SQL Command	Success	
✓	Setting Source Connection	Success	
✓	Setting Destination Connection	Success	
✓	Validating	Success	
✓	Prepare for Execute	Success	
✓	Pre-execute	Success	
✓	Executing	Success	
ⓘ	Copying to [mdws_test].[dbo].[ZIPCodes]	Success	79855 rows transfe...
✓	Post-execute	Success	
✓	Cleanup	Success	

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.



STA (SQL Server 9.0.1399 - KGS-VISTA\J.C)
tab Stop
System Databases
Database Snapshots
mdws
mdws_test
Database Diagrams
Tables
System Tables
dbo.Clients
dbo.MdwsSessionRequests
dbo.MdwsSessions
dbo.PSSG
dbo.Session
dbo.Visitor
dbo.ZIPCodes
Views
Synonyms
Programmability

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

Security

MDWS Passwords

There are a couple of passwords for the MDWS system stored in plain text on the application server in a configuration file in the **Resources/Conf** directory. *It is important to limit access to this directory*.

> **Note:** A future version of MDWS may encrypt these passwords inside the MDWS local database.

File with sensitive passwords:
mdws.conf

Setup and Configuration

MDWS Application Server Configuration

MSI Overview

MDWS application will be distributed in a .msi file (Microsoft Installer file) that contains everything needed to install and run the application.

MSI Physical Layout

The contents of the installed MDWS deployment package are as follows:

```
(root)           MDWS web service definitions:
          - EmrSvc
          - CallService
          - MhvService
          - NumiService
bin\      .dll libraries:
          - mdo.dll
          - mdws.dll
          - log4net.dll
          - other supporting .dll

css\             Cascading Style Sheets:
          - jquerty.treeview
          - screen

images\      Images for dashboard page.

js\    JavaScript for dashboard page.

obj\      Debugging objects.
Resources\conf   MDWS.conf configuration file

resources\lib\   Supporting application files.

resources\xml\   Site configuration files
          -VhaSites
          -TestVhaSites
          -Other files per façade
```

MDWS Deployment Overview

For each site you want your MDWS instance to connect to, first make sure MDWS exists in one of the **/resource/xml/*sites.xml** files. The **mdws.conf** file contains the configuration information for each **façade** indicating which façade **to point to the files of multiple sites.** It is easy to change a single façade towards another (e.g., test) system by changing its entry in the configuration file.

The **web.config** contains configuration information specific to the application server. The connection information for MDWS resides in the local database. A login will be created for the database and this is where the user and password are defined.

For a server already set up with a working MDWS instance, the basic, high-level steps to deploy a new minor version of MDWS are as follows:

1. Obtain the deployment package.
2. Shut down IIS.
3. Copy the deployment package over the existing websites folder.
4. Use the MDWS test page to test links to all sites.

   Note: This feature may still be in development.

5. Update the MDWS database with new Zip Code and PSSG data, if applicable.
6. Ensure any temporary .NET cache files are deleted.

The configurable settings for deployment of any MDWS instance are contained in the following locations:

- **/resources/xml/*.xml**
  contains xml definitions of data sources, both VHA sites and others
- **/resources/conf/mdws.conf**
  contains application server-specific definitions, including database connection information

MDWS Config Files

MDWS web services are broken up into façades. Façades control the selection of web methods to which an application has access. Each façade's data sources can be controlled independently through the **mdws.conf** file. The Administration Console provides an interface for editing the **mdws.conf** file.

VhaSites.xml Format

```
<?xml version="1.0" encoding="UTF-8"?>
  <VhaVisnTable xmlns=http://med.va.gov/vistaweb/sitesTable
                xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
                xsi:schemaLocation="http://med.va.gov/vistaweb/sitesTable
VhaSites.xsd">
  <VhaVisn name="VA New England Health Care System" ID="1">
    <VhaSite name="Togus, ME" ID="402" moniker="TOG">
      <DataSource modality="HIS" protocol="VISTA" source="VISTA.TOGUS.MED.VA.GOV"
                  status="active" port="19215"/>
    </VhaSite>
  </VhaVisn>
  <VhaVisn name="MPI" ID="500">
    <VhaSite name="MPI" ID="500" moniker="MPI">
      <DataSource protocol="HL7" modality="MPI" source="MPI-AUSTIN.MED.VA.GOV"
                  status="active" port="5500" />
    </VhaSite>
  </VhaVisn>
</VhaVisnTable>
```

Note: Although each tag is VhaVisn and VhaSite, consider VISN as a Region descriptor for VISNs and a placeholder for non-VISN sources.

Tag	Field	Description
VhaVisn	name	Region ID text string
	ID	Region ID integer
VhaSite	name	City, State Abbreviation
	ID	Site ID
	moniker	Site Abbreviation
DataSource	modality	
	protocol	Represents the communication protocol of the data source
	source	URL of data source
	status	Active or inactive
	port	Port number of data source

IIS 6.0 Console

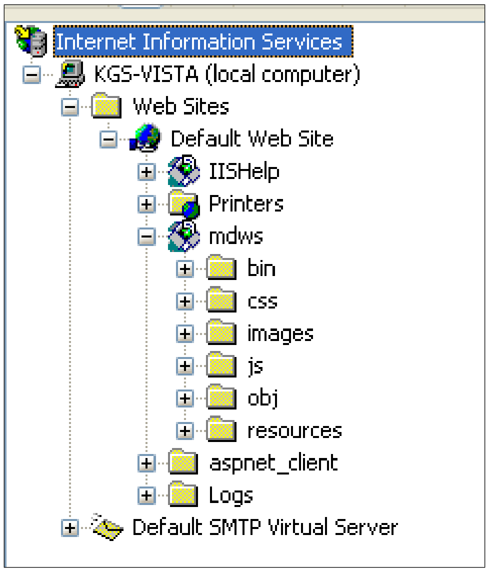
Overview

MDWS is managed using the Internet Information Services (IIS) 6.0 Microsoft Management Console (MMC).

To launch the IIS console,
Click **Start->All Programs->Administrative Tools->Internet Information Services**

Console Navigation Tree

The IIS console displays the installation branch, probably called/ named **mdws_zzz**, in the standard console navigation tree.

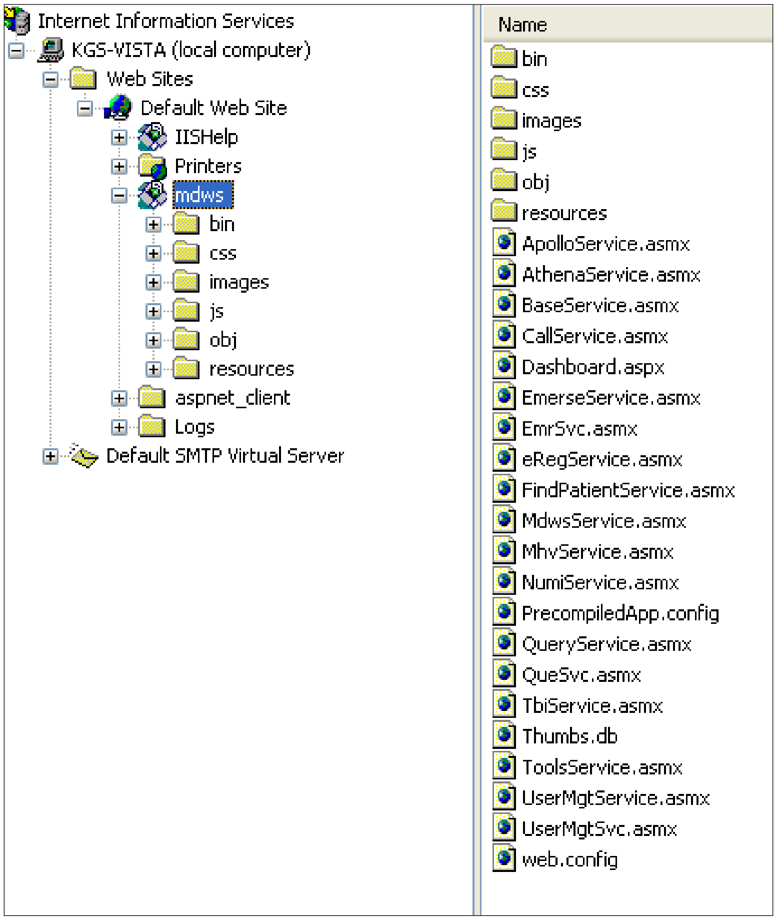


Screen capture of a Web Service database tree

Server-specific Information

Under the Web Sites node, the navigation tree displays one node for each MDWS instance in your application server. You can use these nodes to display information about MDWS deployments on each instance of the application. Because deployments are per server, deployment and status information may vary between servers.

Click a web site node and all the files associated with that site’s root directory display on the left side.

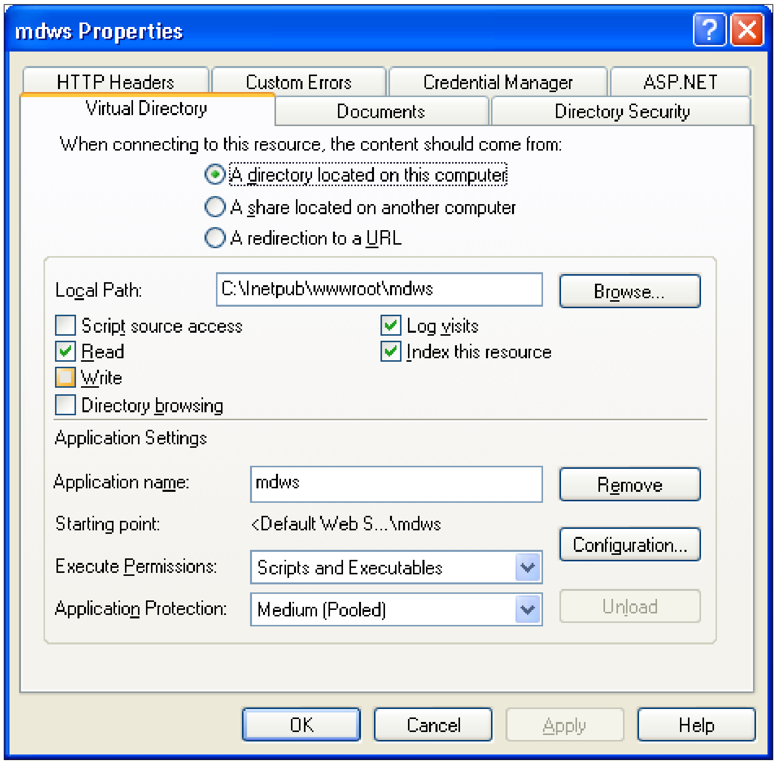


Name
bin
css
images
js
obj
resources
ApolloService.asmx
AthenaService.asmx
BaseService.asmx
CallService.asmx
Dashboard.aspx
EmerseService.asmx
EmrSvc.asmx
eRegService.asmx
FindPatientService.asmx
MdwsService.asmx
MhvService.asmx
NumiService.asmx
PrecompiledApp.config
QueryService.asmx
QueSvc.asmx
TbiService.asmx
Thumbs.db
ToolsService.asmx
UserMgtService.asmx
UserMgtSvc.asmx
web.config

Screen capture of a Web Service database tree
Review MDWS installation

MDWS IIS Property Tabs

The information provided on the Properties tabs are for the non-default value tabs. The HTTP Headers, Custom Errors, and Documents tabs do not change.



When connecting to this resource, the content should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local Path: C:\Inetpub\wwwroot\mdws

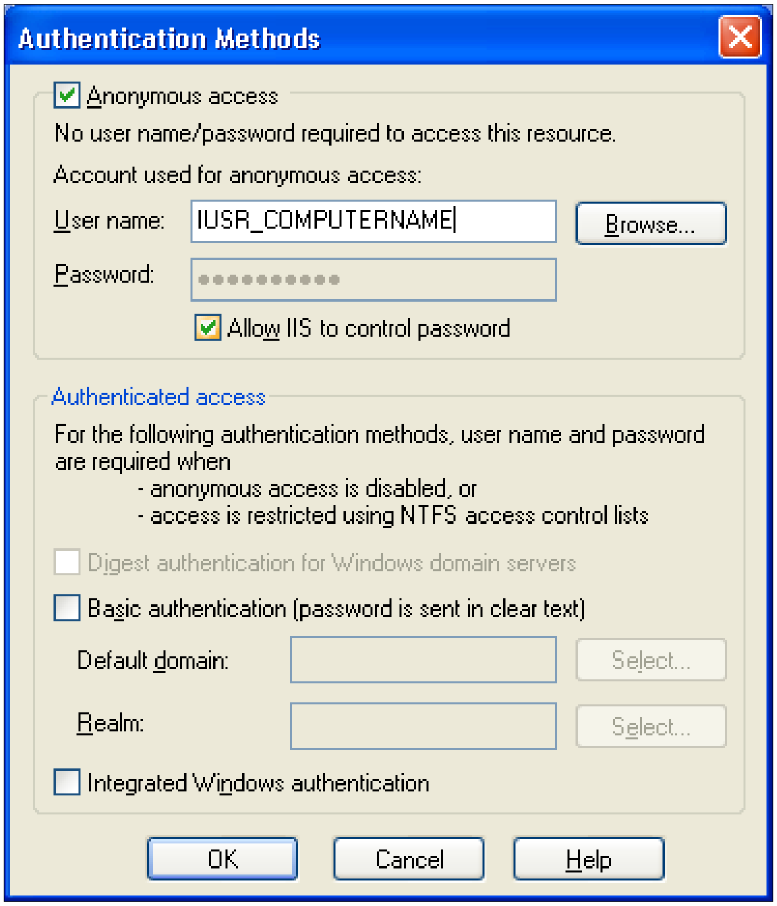
☐ Script source access
☑ Read
☐ Write
☐ Directory browsing
☑ Log visits
☑ Index this resource

Application Settings

Application name: mdws
Starting point: <Default Web S...\mdws
Execute Permissions: Scripts and Executables
Application Protection: Medium (Pooled)

Screen capture of the mdws Properties window
Virtual Directory tab

Use the Authentication Methods information to verify the configuration of your web site(s).



Authentication Methods

☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: IUSR_COMPUTERNAME

Password: ●●●●●●●●●●

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when
- anonymous access is disabled, or
- access is restricted using NTFS access control lists

□ Digest authentication for Windows domain servers

□ Basic authentication (password is sent in clear text)

Default domain:

Realm:

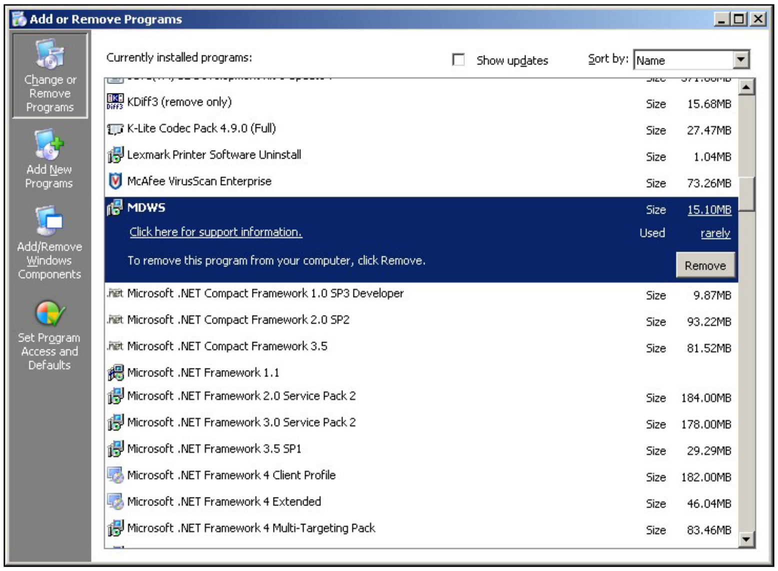
□ Integrated Windows authentication

Screen capture of the Authentication Methods window

Troubleshooting MDWS

Uninstalling MDWS

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


Add or Remove Programs
Change or Remove Programs
Add New Programs
Add/Remove Windows Components
Set Program Access and Defaults
Remove

Currently installed programs: □ Show updates Sort by: Name

KDiff3 (remove only)	Size	15.68MB
K-Lite Codec Pack 4.9.0 (Full)	Size	27.47MB
Lexmark Printer Software Uninstall	Size	1.04MB
McAfee VirusScan Enterprise	Size	73.26MB
MDWS	Size	15.10MB
Click here for support information.	Used	rarely
To remove this program from your computer, click Remove.		
Microsoft .NET Compact Framework 1.0 SP3 Developer	Size	9.87MB
Microsoft .NET Compact Framework 2.0 SP2	Size	93.22MB
Microsoft .NET Compact Framework 3.5	Size	81.52MB
Microsoft .NET Framework 1.1		
Microsoft .NET Framework 2.0 Service Pack 2	Size	184.00MB
Microsoft .NET Framework 3.0 Service Pack 2	Size	178.00MB
Microsoft .NET Framework 3.5 SP1	Size	29.29MB
Microsoft .NET Framework 4 Client Profile	Size	182.00MB
Microsoft .NET Framework 4 Extended	Size	46.04MB
Microsoft .NET Framework 4 Multi-Targeting Pack	Size	83.46MB

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

1.	**Symptom**
	`MDWS WSDL not viewable locally`
	http://localhost/mdws/CallService.asmx
	`404 Page Not Found`
	Diagnoses and Solutions
	IIS Default web site configuration likely incorrect

2.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
3.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
4.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
5.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
6.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
7.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions
8.	**Symptom** `This room left blank intentionally for future solutions` **Diagnoses and Solutions** This room left blank intentionally for future solutions

Glossary

Term	Definition
AITC	Austin Information Technology Center
AViVA	A Virtual Instance of VistA Architecture
BHIE	Bi-directional Health Information Exchange
BMS	Bed Management Solutions
Caché	An ‘M’ based product (by InterSystems) which has been selected as the next generation VistA platform
CAPRI	Compensation and Pension Records Interchange
C1	Class 1
C3	Class 3
CCOW	Clinical Context Object Workgroup
Client Applications	Client applications can be written in Delphi. Visual Basic, C#, HTML/Javascript, PHP, etc.
COM	Component Object Model
Connectivity	Connectivity provides connection to the Master Patient Index (MPI), Structured Query Language (SQL) and Extensible Markup Language (XML).
COTS	Commercial Off the Shelf
CPRS	Computerized Patient Record System
DAO	Data Access Objects
DFN	Data File Number A patient’s local identifier, the internal entry number in file #2
DHS	Department of Homeland Security
DICOM	Digital Imaging and Communication in Medicine
DUZ	A user’s local identifier, the internal entry number in file #200
EIE	Enterprise Infrastructure Engineering
ESSENCE	Electronic Surveillance System for the Early Notification of Community-based Epidemics
Façade	A set of useable features made available for applications of a certain type. Each façade is a partial class composed of methods from multiple source libraries combined in a logical grouping for the consumer type (e.g., Patient, Provider, Util)
FIPS	Federal Information Processing Standard
GOTS	Government Off The Shelf
GUI	Graphical User Interface
HAIISS	Healthcare Associated Infection and Influenza Surveillance System
HDR	Health Data Repository
HL7	Health Level 7
ICD	International Classification of Diseases
ICN	The patient’s national identifier, Integration Control Number
IDE	Integrated Development Environment

Term	Definition
IEN	Internal Entry Number
J2EE	Java 2 Platform, Enterprise Edition defines the standard for developing multitier enterprise applications
MDO	Medical Domain Objects MDO is the middle-tier SOA used by MDWS to access multiple VistA sites without further credentialing, which works entirely through existing Remote Procedure Calls (RPCs)
MDWS	Medical Domain Web Services
MPI	Master Patient Index
MRI	Magnetic Resonance Imaging
MSI	Microsoft Installer (file)
MWSV	NameSpace assigned to Medical Domain Web Services (MDWS) by DBA
.NET	The Microsoft .NET Framework is a software framework that can be installed on computers running Microsoft Windows operating systems
NOK	Next Of Kin
NUMI	National Utilization Management Integration
OEF	Operation Enduring Freedom
OIF	Operation Iraqi Freedom
PHR	Patient Health Re cord
PSSG	Planning System Support Group
RPC	Remote Procedure Call
RSD	Requirements Specification Document
SIA	Security Integration Agreement
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture is a flexible set of design principles used during the phases of systems development and integration. A deployed SOA-based architecture will provide a loosely integrated suite of *services* that can be used within multiple business domains.
SOAP	Server Oriented Architecture Protocol
SQL	Structured Query Language
TIU	Text Integration Utility
UDDI	Universal Description, Discovery and Integration A registry that enables a developer to shop for MDWS pre-fabricated web services
VA	Department of Veterans Affairs
VAMC	Department of Veterans Affairs Medical Center
VHA	Veterans Health Administration
VistA	Veterans Health Information Systems and Technology Architecture An enterprise-wide information system built around an electronic health record used throughout the Department of Veterans Affairs medical system.
VW	VistA Web
VWS	VistA Web Services

Term	Definition
WSDL	Web Service Description Language (WSDL) is a document that provides a common language to describe: • what the web service does • what functionality it can provide • what data it can deliver A developer can click WSDL and generate the code automatically.
XML	Extensible Markup Language