|  |  |
| --- | --- |
|  | **Unidev** |
|  | **Unified Development, Inc.**  16690 Swingley Ridge Road  Suite 260  Chesterfield, MO 63017  (636) 532-4424 |

|  |
| --- |
|  |
|  |

**Duncan Solutions, Inc**.

Installing and Configuring PEMS and RBAC

February 2014

Revision 1.2

**Prepared by:**

Ron Howard

Senior Systems Engineer

rhoward@unidev.com

Unified Development

**Table of Contents**

[1 Overview 6](#_Toc379357711)

[1.1 Goal 6](#_Toc379357712)

[2 Prerequisites and Target Audience 7](#_Toc379357713)

[2.1 Prerequisites 7](#_Toc379357714)

[2.2 Target Audience 7](#_Toc379357715)

[2.3 Guidelines 8](#_Toc379357716)

[3 Installation Files 9](#_Toc379357717)

[4 RBAC Database 10](#_Toc379357718)

[4.1 Creation 10](#_Toc379357719)

[4.2 RBAC Database Instance Population 12](#_Toc379357720)

[4.2.1 Create Objects 12](#_Toc379357721)

[4.2.2 Create Initial Data 15](#_Toc379357722)

[5 Create Admin User and Menus 17](#_Toc379357723)

[5.1 Execute Installer Program 17](#_Toc379357724)

[6 RBAC Database – Post-Populate 20](#_Toc379357725)

[6.1 Post-Populate 20](#_Toc379357726)

[7 PEMS Website Installation 22](#_Toc379357727)

[7.1 Create PEMS Website 22](#_Toc379357728)

[7.1.1 Create Site 22](#_Toc379357729)

[7.1.2 Configure App Pool 24](#_Toc379357730)

[7.2 Publish PEMS Website Source 25](#_Toc379357731)

[7.3 Deploy PEMS Website Source 30](#_Toc379357732)

[7.4 PEMS Website Directory Rights 30](#_Toc379357733)

[8 Discount Website Installation 31](#_Toc379357734)

[8.1 Create Discount Website 31](#_Toc379357735)

[8.1.1 Create Site 31](#_Toc379357736)

[8.1.2 Create HTTPS Binding 33](#_Toc379357737)

[8.1.3 Configure App Pool 35](#_Toc379357738)

[8.2 Publish and Deploy Discount Source 36](#_Toc379357739)

[8.3 Deploy PEMS Website Source 40](#_Toc379357740)

[8.3.1 Deploy Additional Lib Directory 41](#_Toc379357741)

[9 Configuration Files – PEMS 42](#_Toc379357742)

[9.1 mailSettings.config 42](#_Toc379357743)

[9.2 connectionStrings.config 42](#_Toc379357744)

[9.2.1 RBAC Connections 43](#_Toc379357745)

[9.2.2 PEMS (RipNet) Connections 44](#_Toc379357746)

[9.2.3 Reports Connection 44](#_Toc379357747)

[9.2.4 Maintenance Connection 45](#_Toc379357748)

[9.3 NLog.config 46](#_Toc379357749)

[9.3.1 Targets 46](#_Toc379357750)

[9.3.2 Rules 46](#_Toc379357751)

[9.4 web.config 46](#_Toc379357752)

[9.4.1 <appSettings> section 46](#_Toc379357753)

[9.4.2 Elmah Settings 49](#_Toc379357754)

[9.5 Publishing PEMS Config Files 50](#_Toc379357755)

[10 Configuration Files - Discount 52](#_Toc379357756)

[10.1 mailSettings.config 52](#_Toc379357757)

[10.2 connectionStrings.config 52](#_Toc379357758)

[10.2.1 RBAC Connection 52](#_Toc379357759)

[10.2.2 PEMS (RipNet) Connections 53](#_Toc379357760)

[10.3 web.config 54](#_Toc379357761)

[10.3.1 <appSettings> section 54](#_Toc379357762)

[10.4 Publishing Discount Config Files 55](#_Toc379357763)

[11 Starting the Websites 56](#_Toc379357764)

[12 Testing PEMS Website 57](#_Toc379357765)

[12.1 Initial Access 57](#_Toc379357766)

[12.1.1 Troubleshooting 57](#_Toc379357767)

[12.2 Initial Login 59](#_Toc379357768)

[12.2.1 Troubleshooting 62](#_Toc379357769)

[12.3 Creating First PEMS Customer 62](#_Toc379357770)

[13 Testing Discount Website 63](#_Toc379357771)

[13.1 Initial Access 63](#_Toc379357772)

[13.1.1 Troubleshooting 63](#_Toc379357773)

[13.2 Test Login 65](#_Toc379357774)

[13.2.1 Troubleshooting 67](#_Toc379357775)

[14 Conclusion 68](#_Toc379357776)

**Table of Figures**

[Figure 1 RBAC Database Create Response 11](#_Toc379356478)

[Figure 2 RBAC Database Create Output 11](#_Toc379356479)

[Figure 3 RBAC Database Build Response 12](#_Toc379356480)

[Figure 4 RBAC Database Build Output 13](#_Toc379356481)

[Figure 5 Example RBAC Tables in SSMS 14](#_Toc379356482)

[Figure 6 RBAC Database Populate Response 15](#_Toc379356483)

[Figure 7 RBAC Database Populate Output 16](#_Toc379356484)

[Figure 8 Duncan.PEMS.Install.exe Console Output 18](#_Toc379356485)

[Figure 9 Duncan.PEMS.Install.exe Log File Output 19](#_Toc379356486)

[Figure 10 Duncan.PEMS.Install.exe Console Output with Error 19](#_Toc379356487)

[Figure 11 Duncan.PEMS.Install.exe Log File Output with Error 19](#_Toc379356488)

[Figure 12 RBAC Database PostPopulate Response 21](#_Toc379356489)

[Figure 13 RBAC Database PostPopulate Output 21](#_Toc379356490)

[Figure 14 - IIS PEMS Website Creation 22](#_Toc379356491)

[Figure 15 - IIS PEMS Site Details 23](#_Toc379356492)

[Figure 16 Navigate to PEMS App Pool 24](#_Toc379356493)

[Figure 17 - IIS PEMS .NET Version and Process Model 25](#_Toc379356494)

[Figure 18 - PEMS Publish Step 1 26](#_Toc379356495)

[Figure 19 - PEMS Publish Step 2 27](#_Toc379356496)

[Figure 20 - PEMS Publish Step 3 28](#_Toc379356497)

[Figure 21 - PEMS Publish Step 4 29](#_Toc379356498)

[Figure 22 Typical Publish Result 29](#_Toc379356499)

[Figure 23 - IIS Discount Website Creation 31](#_Toc379356500)

[Figure 24 - IIS Discount Site Details 32](#_Toc379356501)

[Figure 25 Discount Site Bindings 33](#_Toc379356502)

[Figure 26 Discount Edit Site Bindings 34](#_Toc379356503)

[Figure 27 Discount Edit Site Bindings HTTPS 34](#_Toc379356504)

[Figure 28 Navigate to Discount App Pool 35](#_Toc379356505)

[Figure 29 - IIS Discount .NET Version and Process Model 36](#_Toc379356506)

[Figure 30 - Discount Publish Step 1 37](#_Toc379356507)

[Figure 31 - Discount Publish Step 2 38](#_Toc379356508)

[Figure 32 - Discount Publish Step 3 39](#_Toc379356509)

[Figure 33 - Discount Publish Step 4 40](#_Toc379356510)

[Figure 34 Verify both sites running 56](#_Toc379356511)

[Figure 35 Inital PEMS Website Test 57](#_Toc379356512)

[Figure 36 Error in SQL User Name or Password - Duncan.Membership.Connector 58](#_Toc379356513)

[Figure 37 Configuration Error in SQL Initial Catalog Name - Duncan.Membership.Connector 58](#_Toc379356514)

[Figure 38 Configuration Error in SQL Server Name - Duncan.Membership.Connector 59](#_Toc379356515)

[Figure 39 Testing Admin login 60](#_Toc379356516)

[Figure 40 Testing Admin login - Required Password Change 61](#_Toc379356517)

[Figure 41 Configuration Error in SQL Server Name – PEMRBACEntities 62](#_Toc379356518)

[Figure 42 Initial Discount Website Test 63](#_Toc379356519)

[Figure 43 Configuration Error in SQL Initial Catalog Name – RbacEntities 64](#_Toc379356520)

[Figure 44 Configuration Error in SQL User Name or Password – RbacEntities 64](#_Toc379356521)

[Figure 45 Configuration Error in SQL Data Source – RbacEntities 65](#_Toc379356522)

[Figure 46 Discount Website Login Test 66](#_Toc379356523)

[Figure 47 Discount Website Login Test Result 67](#_Toc379356524)

[Figure 48 Configuration Error in Membership.Connector Connection String 67](#_Toc379356525)

**Revision** History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Description | Revision Date |
| 1.00 | Ron Howard | Initial version | 11/13/2013 |
| 1.01 | Ron Howard | Added ***pems.roles.required.#*** *in appSettings* | 11/21/2013 |
| 1.10 | Ron Howard | Added section on PEMS\_RBAC\_Database\_PostPopulate.sql | 02/05/2014 |
| 1.20 | Ron Howard | Added note on write rights for website App\_Data directory | 02/05/2014 |

# Overview

## Goal

This document is intended as a guide to the installation and configuration of the web applications, PEMS and Discount, and Role-Based Access Control (RBAC) database that support the PEMS System. This document is intended to be used in conjunction with the installation files generated and provided by the PEMS Visual Studio 2012 Project. These installation files may be created by the Visual Studio project or provided as a package of files.

This document will guide you through the creation of an instance of the RBAC database, the population of the RBAC database, installation of the PEMS and Discount web sites, creation of the administration user and initialization of the PEMS admin web site.

Upon completion of these instructions you should be able to navigate to an instance of the PEMS Administration website and begin the creation of a PEMS customer.

# Prerequisites and Target Audience

## Prerequisites

These instructions assume the following prerequisites have been met.

1. SQL Server 2008 R2 has been installed and a user is available that has rights to create databases and execute SQL scripts that create database tables. The name of this server, and optionally, instance, will be referenced as **[SQL\_SERVER]** in the remainder of this document. These instructions assume the user has Windows credentials to access SQL Server. If using SQL user/password then user must modify script execution as required.
2. A SQL login has been created with the name of “Duncan” and an appropriately complex password has been created. The SQL user name, “Duncan”, will be referenced as **[SQL\_USER\_NAME]** and the password will be referenced as **[SQL\_USER\_PASSWORD]** in the remainder of this document.
3. An instance of IIS Version 7.5 is installed on appropriate hardware and a user with access to create and configure a web site is available.
4. .NET Framework 4.5 is installed on IIS Server.
5. Any required security Certificates have been obtained and installed in IIS.
6. Access to the configuration-managed **Duncan.PEMS.Install** project or access to a PEMS install package.
7. Access to the configuration-managed **Duncan.PEMS.Web** project or access to a PEMS Website install package.
8. Access to the configuration-managed **Duncan.Discount.Web** project or access to a PEMS Website install package.
9. Acquisition of a public domain with which to make the PEMS website available for public use. This document does not cover that process.
10. Acquisition of a public domain with which to make the Discount website available for public use. This document does not cover that process.

## Target Audience

The target audience of this document is person or persons who have experience in:

1. Microsoft SQL Server administration and understand rights, database creation and administration, and are able to used either SSMS or SQL command line interface.
2. Visual Studio 2012 and web site publishing.
3. IIS 7+ Administration and web site creation.

## Guidelines

Installation should proceed in the order given in this document. If a step fails, do not continue installation.

Rollback/retry instructions are given at each step, if applicable.

# Installation Files

Installation files can be obtained either by creating them via the *Duncan.PEMS.Install* or by obtaining a PEMS install package which was created by another user with access to the *Duncan.PEMS.Install* project.

This document will assume that the package has been provided and extracted to a directory. All files will be referenced from the root of this directory. This directory will be referred to as **INSTALL\_ROOT**.

The contents of the install package are:

**INSTALL\_ROOT** – Executable file, assemblies, and configuration file for the installer program.

**INSTALL\_ROOT\ConfigFiles\Discount** – Configuration file samples for the Discount website. These files are provided as examples for required configuration files.

**INSTALL\_ROOT\ConfigFiles\PEMS** – Configuration file samples for the PEMS website. These files are provided as examples for required configuration files.

**INSTALL\_ROOT\Documentation** – This document

**INSTALL\_ROOT\lib** – Folder containing required assembly(ies) for Izenda internal report viewer version. This (these) file(s) will be copied to the published directory structure of PEMS website during installation.

**INSTALL\_ROOT\rbac** –XML file to configure RBAC authorization and menus for the Admin website.

**INSTALL\_ROOT\Scripts** – SQL Scripts that create and populate the initial instance of RBAC

# RBAC Database

The RBAC database is created and populated by execution of three scripts.

*PEMS\_RBAC\_Database\_Create.sql* – Creates the database instance.

*PEMS\_RBAC\_Database\_Build.sql* – Builds the tables, stored procedures, etc. of RBAC.

*PEMS\_RBAC\_Database\_Populate.sql* – Populates required initial data of RBAC.

## Creation

Select a name for the RBAC database. This name will be used in creation of the database, the scripts used to create and populate tables and the configuration of the web site instance. A recommended name is PEMSRBAC.

The database can be created by either using the Sql Server Management Studio (SSMS) UI or similar management tool or by updating and executing **INSTALL\_ROOT\Scripts**\***PEMS\_RBAC\_Database\_Create.sql*** script. Make the following edits to the script.

* Change all instances of the string “PEMS\_RBAC\_DATABASE\_NAME “to the name of the database. Recommended name is “PEMSRBAC”
* Change all instances of the string “PEMS\_RBAC\_LOGICAL\_NAME” to the logical name of the database. Recommended logical name is “PEMSRBAC-Prod”
* Change all instances of the string “PEMS\_RBAC\_DATABASE\_PATH” to a valid path where the database file will be created. Do not end with “/” or “\”.
* Change all instances of the string “PEMS\_RBAC\_FILE\_NAME” to the name of the database file name desired.

Execute the copy of the script. Open a command prompt and execute the following SQL command.

**sqlcmd -S [SERVER]\[INSTANCE] -i [PATH]\PEMS\_RBAC\_Database\_Create.sql -o [PATH]\PEMS\_RBAC\_Database\_Create.log**

[SERVER]\[INSTANCE] – Name of the SQL Server and, optionally, instance name. This will be the **[SQL\_SERVER]** as referenced in Section 2.1.

[PATH] – Fully qualified path to *PEMS\_RBAC\_Database\_Create.sql*. Should be **INSTALL\_ROOT\Scripts\*PEMS\_RBAC\_Database\_Create.sql***

Example command window where **INSTALL\_ROOT** is “C:\temp\PEMS”. The execution of this command should produce no response to command window. You should simply be returned to the command prompt.

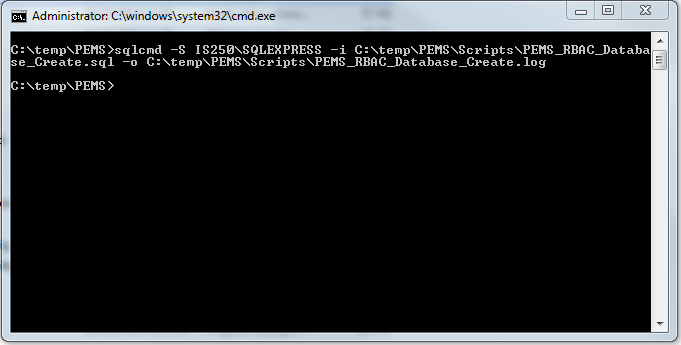


Figure 1 RBAC Database Create Response

Review the output file, **[PATH]\PEMS\_RBAC\_Database\_Create.log,** to check for any errors. The output file should only contain one line.

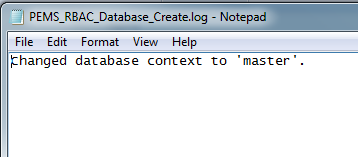


Figure 2 RBAC Database Create Output

Using SSMS, verify that your database was created as expected. Do not forget to refresh the tree in SSMS to be sure you are looking at the most recent view of existing databases.

If you encountered an error either at the command prompt or in the log file or do not see the database in SSMS, resolve the error and rerun the script. Do not continue on until a database is successfully created.

## RBAC Database Instance Population

### Create Objects

The database objects can be created by updating and executing **INSTALL\_ROOT\Scripts**\***PEMS\_RBAC\_Database\_Build.sql*** script. Make the following edits to the script.

* Change all instances of the string “PEMS\_RBAC\_DATABASE\_NAME” to the name of the database.

Execute the copy of the script. Open a command prompt and execute the following SQL command.

**sqlcmd -S [SERVER]\[INSTANCE] -i [PATH]\PEMS\_RBAC\_Database\_Build.sql -o [PATH]\PEMS\_RBAC\_Database\_Build.log**

[SERVER]\[INSTANCE] – Name of the SQL Server and, optionally, instance name. This will be the **[SQL\_SERVER]** as referenced in Section 2.1.

[PATH] – Fully qualified path to *PEMS\_RBAC\_Database\_Build.sql.* Should be **INSTALL\_ROOT\Scripts\*PEMS\_RBAC\_Database\_Build.sql***

Example command window where **INSTALL\_ROOT** is “C:\temp\PEMS”. The execution of this command should produce no response to command window. You should simply be returned to the command prompt.

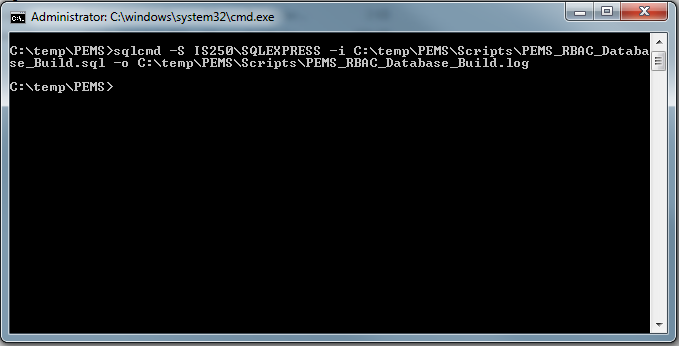


Figure 3 RBAC Database Build Response

Review the output file, **[PATH]\PEMS\_RBAC\_Database\_Build.log,** to check for any errors. The output file should only contain six lines as shown below. The last five lines are warnings that can be ignored.

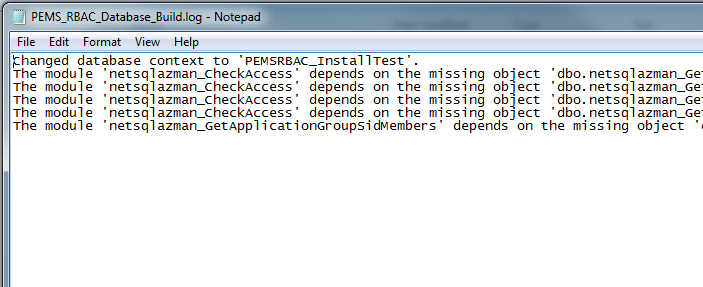


Figure 4 RBAC Database Build Output

Using SSMS, verify that your database tables were created as expected. Do not forget to refresh the tree in SSMS to be sure you are looking at the most recent view of existing tables. You should see something like this. This figure may not represent the latest set of tables and is given as an example only.

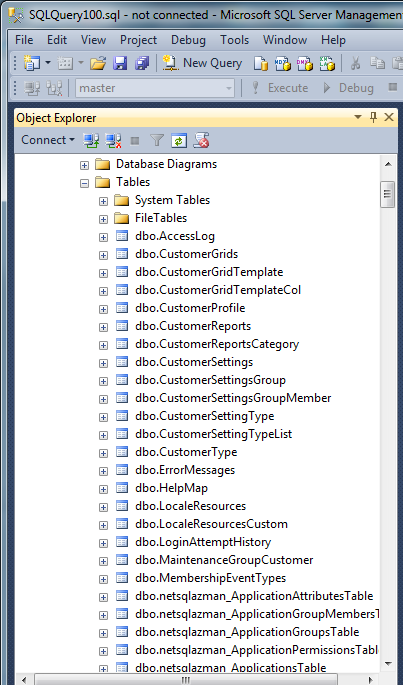


Figure 5 Example RBAC Tables in SSMS

If you encountered an error either at the command prompt or in the log file or do not see the database tables in SSMS, resolve the error and rerun the script. Do not continue on until database objects are successfully created.

### Create Initial Data

The database can be populated by updating and executing **INSTALL\_ROOT\Scripts**\***PEMS\_RBAC\_Database\_Populate.sql*** script. Make the following edits to the script.

* Change all instances of the string “PEMS\_RBAC\_DATABASE\_NAME” to the name of the database.

Execute the copy of the script. Open a command prompt and execute the following SQL command.

**sqlcmd -S [SERVER]\[INSTANCE] -i [PATH]\PEMS\_RBAC\_Database\_Populate.sql -o [PATH]\PEMS\_RBAC\_Database\_Populate.log**

[SERVER]\[INSTANCE] – Name of the SQL Server and, optionally, instance name. This will be the **[SQL\_SERVER]** as referenced in Section 2.1.

[PATH] – Fully qualified path to *PEMS\_RBAC\_Database\_Populate.sql* Should be **INSTALL\_ROOT\Scripts\*PEMS\_RBAC\_Database\_Populate.sql***

Example command window where **INSTALL\_ROOT** is “C:\temp\PEMS”. The execution of this command should produce no response to command window. You should simply be returned to the command prompt.

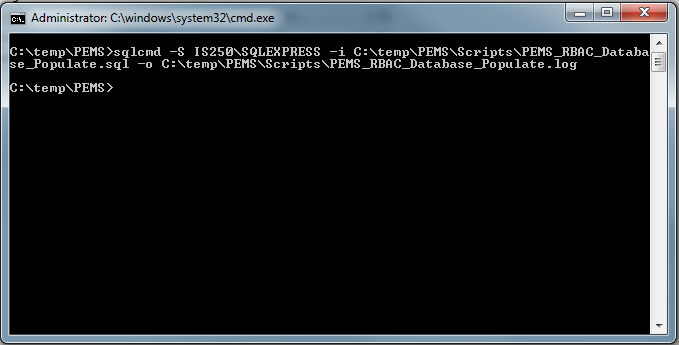


Figure 6 RBAC Database Populate Response

Review the output file, **[PATH]\PEMS\_RBAC\_Database\_Populate.log,** to check for any errors. The output file should contain the “Change database…” line and many repeated lines indicating “(1 rows affected)” as shown below. Be sure there are no errors in the log file.

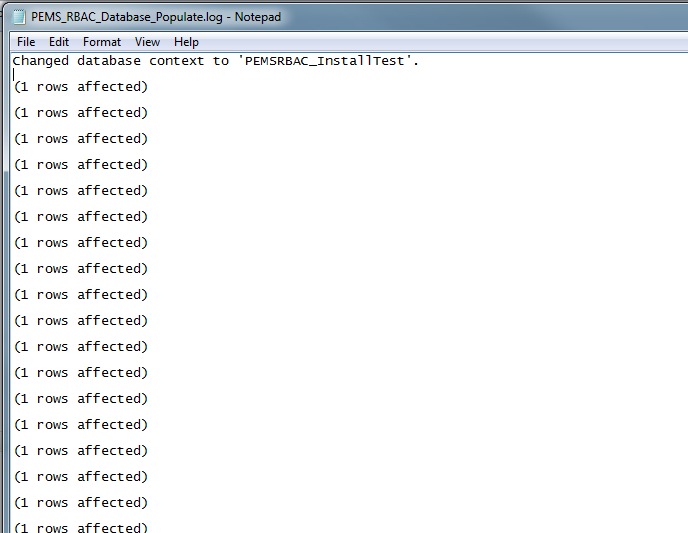


Figure 7 RBAC Database Populate Output

Review the output file to check for any errors. If errors were encountered, the populate script is out-of-date. Contact the PEMS development team for the latest scripts.

# Create Admin User and Menus

The installer program, **Duncan.PEMS.Install.exe**, will create the administration user and set up the menu and rights for the PEMS Admin site. This program is a console application and uses command-line parameters to configure the Admin instance. This program can be found in the **INSTALL\_ROOT** of the installation package.

Edit the **INSTALL\_ROOT*\Duncan.PEMS.Install.exe.config*** file and update the two connection strings, **Duncan.Membership.Connector** and **PEMRBACEntities,** to point to the new database created above.

<connectionStrings>

<add name="Duncan.Membership.Connector" connectionString="Data Source=[SERVER]\[INSTANCE];Initial Catalog=[PEMS\_RBAC\_DATABASE\_NAME];User ID=[USER\_NAME];password=[USER\_PASSWORD];" providerName="System.Data.SqlClient" />

<add name="PEMRBACEntities" connectionString ="metadata=res://\*/RBAC.PEMRBACEntities.csdl|res://\*/RBAC.PEMRBACEntities.ssdl|res://\*/RBAC.PEMRBACEntities.msl;provider=System.Data.SqlClient;provider

connection string=&quot;data source=[SERVER]\[INSTANCE];

initial catalog=[PEMS\_RBAC\_DATABASE\_NAME];user id=[USER\_NAME];password=[USER\_PASSWORD];MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

[SERVER] – Name of the SQL Server

[INSTANCE] – Name of the target SQL Server instance

[PEMS\_RBAC\_DATABASE\_NAME] – Name of database

[USER\_NAME] – Name of database user. Is configured as “Duncan” in the *PEMS\_RBAC\_Database\_Build.sql* script.

[USER\_PASSWORD] – Password assigned to the “Duncan” user.

## Execute Installer Program

Open a command (cmd) window in the **INSTALL\_ROOT** of the installation package. This is where **Duncan.PEMS.Install.exe** is located. The following command and switches will create an administration user named ‘*Admin*’ with a default password of ‘*password*’. This is the user and initial password used to log on to the Admin website in later step. The initial logon process will require a new password so the value of ‘*password*’ at this point does not represent a security issue. If you wish, replace ‘*password*’ with your own value but DO NOT FORGET IT.

This command also create the initial authorizations and menu settings for the Admin website.

Edit the two values indicated and execute the following at the command prompt.

**Duncan.PEMS.Install.exe --admin-user Admin --admin-pwd password --admin-email [ADMIN\_EMAIL] --admin-site-template [XML\_PATH]\Duncan.Auth.Admin.xml --admin-site-id 1 --verbose --logfile log.txt**

[ADMIN\_EMAIL] - Appropriate e-mail address for PEMS Administrator.

[XML\_PATH] – Path to the **Duncan.Auth.Admin.xml** file. Should be located in **INSTALL\_ROOT\rbac.**

The –verbose switch will emit process progress to the console. In the above example, the following console output is expected.

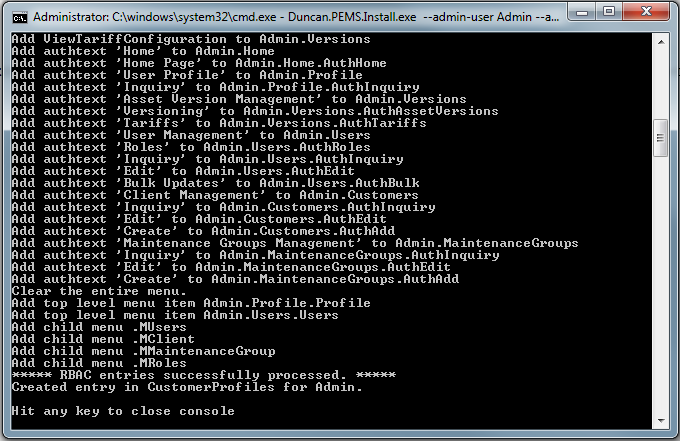
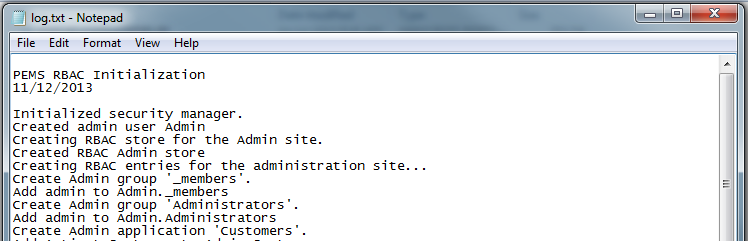


Figure 8 Duncan.PEMS.Install.exe Console Output

The –logfile option will write out all process progress logs to the indicated file. Examine that file for any errors. For example, if an error was encountered, the following console output example indicates the issue.



…

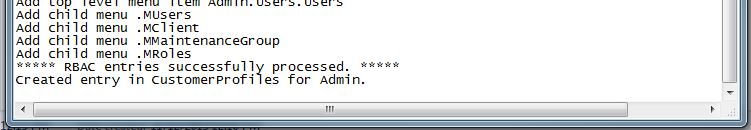


Figure 9 Duncan.PEMS.Install.exe Log File Output

If errors were encountered the console output and the log file will indicate a description of the error. The following are examples of an error with the RBAC xml file.

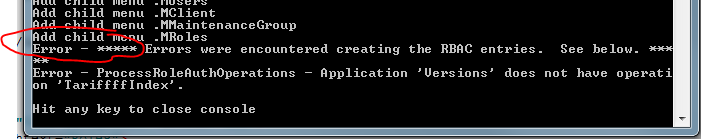


Figure 10 Duncan.PEMS.Install.exe Console Output with Error

Same error if –logfile option was selected.

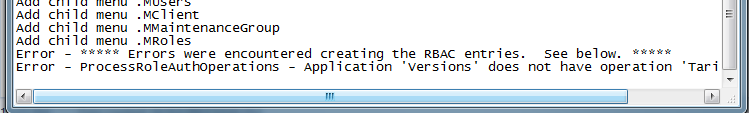


Figure 11 Duncan.PEMS.Install.exe Log File Output with Error

Typical errors encountered will be with the **Duncan.Auth.Admin.xml**. This document does not cover the configuration of authorizations and menus via XML. Please see ‘PEMS Authentication and Menus Users Guide.docx’ for more details.

# RBAC Database – Post-Populate

The RBAC database now contains initial data for the Admin site and Admin user. Since the Admin site could be instantiated with an id chosen by the installation admin, a post-creation script must be run to create the default password for new users created for the Admin site. This script is:

*PEMS\_RBAC\_Database\_PostPopulate.sql* – Populates default password for Admin site.

## Post-Populate

The default password for the Admin site can either be defaulted to the value created in the database population script (*PEMS\_RBAC\_Database\_Populate.sql*) or it can be selected by the installation person by editing *PEMS\_RBAC\_Database\_PostPopulate.sql* script before execution.

The default password script for the Admin site can be database can be executed either using the Sql Server Management Studio (SSMS) UI or similar management tool or by updating and executing **INSTALL\_ROOT\Scripts**\***PEMS\_RBAC\_Database\_PostPopulate.sql*** script. Make the following edits to the script.

* Change all instances of the string “PEMS\_RBAC\_DATABASE\_NAME “to the name of the database. Recommended name is “PEMSRBAC”

Execute the copy of the script. Open a command prompt and execute the following SQL command.

**sqlcmd -S [SERVER]\[INSTANCE] -i [PATH]\PEMS\_RBAC\_Database\_PostPopulate.sql -o [PATH]\PEMS\_RBAC\_Database\_ PostPopulate.log**

[SERVER]\[INSTANCE] – Name of the SQL Server and, optionally, instance name. This will be the **[SQL\_SERVER]** as referenced in Section 2.1.

[PATH] – Fully qualified path to *PEMS\_RBAC\_Database\_PostPopulate.sql*. Should be **INSTALL\_ROOT\Scripts\*PEMS\_RBAC\_Database\_* PostPopulate*.sql***

Example command window where **INSTALL\_ROOT** is “C:\temp\PEMS”. The execution of this command should produce no response to command window. You should simply be returned to the command prompt.

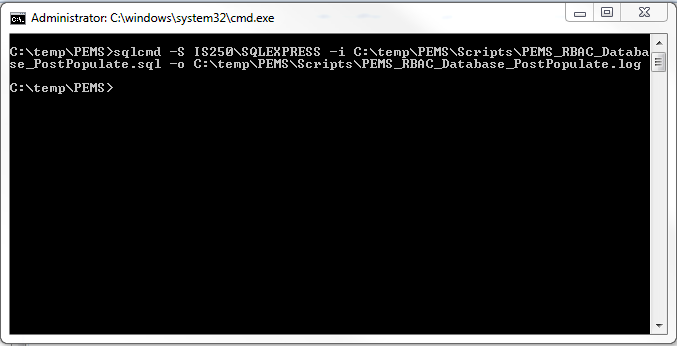


Figure 12 RBAC Database PostPopulate Response

Review the output file, **[PATH]\PEMS\_RBAC\_Database\_PostPopulate.log,** to check for any errors. The output file should only contain three lines. The last line shows what the default password for any new users created in the Admin site will be. In this example, the default password is set to ‘Duncan1!’. If you do not select your own password, this is the password chosen.

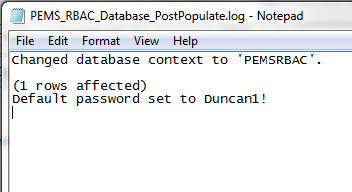


Figure 13 RBAC Database PostPopulate Output

If you encountered an error either at the command prompt or in the log file or do not see the database in SSMS, resolve the error and rerun the script. Do not continue on until a default password is successfully created.

# PEMS Website Installation

## Create PEMS Website

The creation of the PEMS website requires planning and the acquisition of a domain with which to make the website available for public use. This document does not cover that process. It is assumed that a domain has been determined at this point.

**[WEB\_HOST\_NAME]** will serve as a place holder to indicate usage of the selected domain. In this example, the host name will be *Duncan.pems.com*.

Determine the physical path where the web site source files will be installed. A typical location is a directory located under C:\inetpub\wwwroot. Make a determination where the web site source files will be located and create a directory there.

**[PHYSICAL\_PATH] –** Placeholder for physical location ofthe PEMS web site. This is where the source files will be copied to in later steps and where the website will reference under IIS. In this example, the physical location will be *C:\inetpub\wwwroot\pems*.

### Create Site

Access the IIS Management Console on the appropriate web server that is going to host the PEMS installation. Create the PEMS Web Site instance in IIS.

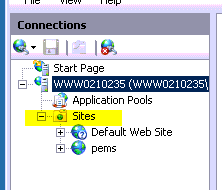


Figure 14 - IIS PEMS Website Creation

To create the web site, right-click on ‘Sites’ and select ‘Add Web Site…” to create a new web site. Fill out the indicated fields below.

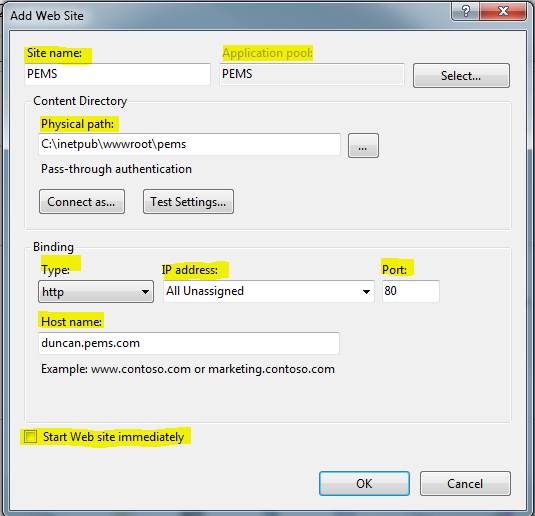


Figure 15 - IIS PEMS Site Details

Enter the following fields:

**Site name** – Name as it will appear in IIS. Recommend calling site “PEMS”.

**Application pool** – IIS will automatically assign a new Application Pool. Note the name for later use.

**Physical path** – Physical path on local machine where web site files will be located. You will be publishing the PEMS web site files to the Physical path later in this installation. This path will be referenced as **[PHYSICAL\_PATH]** in this document**.**

**Host name** – Optional host name. Not required if this site will handle all web traffic on this instance of IIS. The Host name will be the root web site name that will be published to the customers. This host will be determined by Duncan and appropriately registered. This host name will be referenced by **[WEB\_HOST\_NAME]** in this document.

**Binding: Type** – Leave as http.

**Binding: IP address** – Leave as ‘All Unassigned’ unless server has been configured with multiple NIC cards (IP addresses). Verify this setting with the network administrator.

**Binding: Port** – Leave this set at 80 unless requested by network administrator to use other port. A different port may be assigned based on other network attributes and security settings.

**Start Web site immediately** – Uncheck this. Site will be started later in the install process.

Click ‘OK’ to apply settings and create PEMS website.

### Configure App Pool

Configure the website App Pool via IIS Manager.

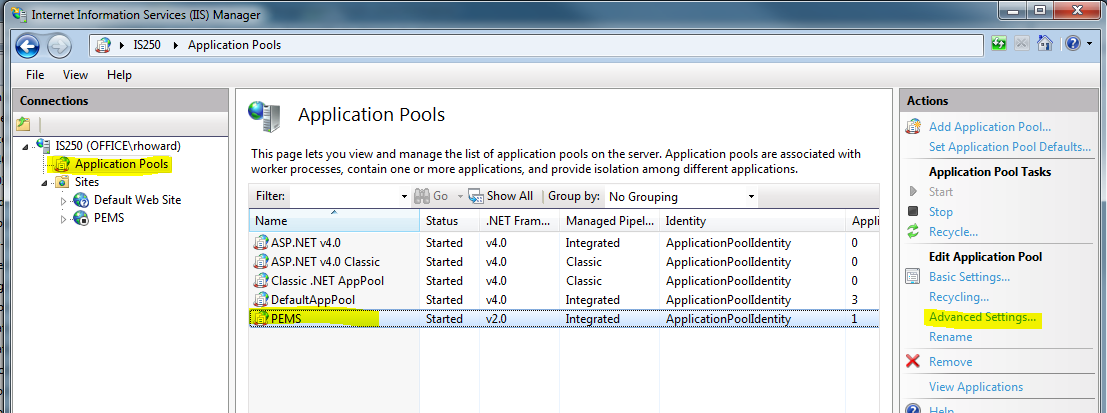


Figure 16 Navigate to PEMS App Pool

Click on **Application Pool** under Connections on left side of IIS Manager. Next, click on the appropriate app pool as noted from the creation of the web site. In this example it is ‘PEMS’. Finally, click on **Advanced Settings…**  on the right hand Actions column. This will bring up the following dialog.

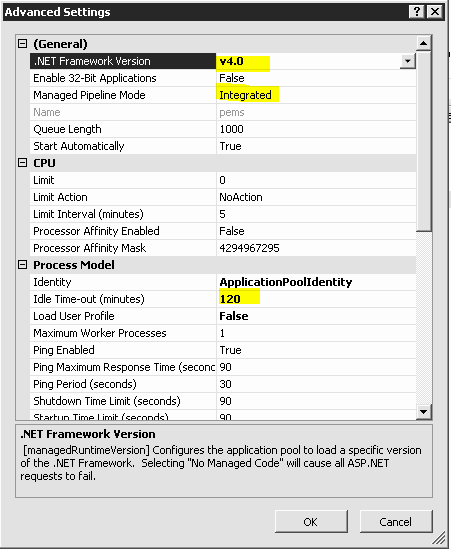


Figure 17 - IIS PEMS .NET Version and Process Model

Edit the application pool for the web site. Under Advanced Settings for the application app pool, set the following:

.**NET Framework Version** – v4.0

**Managed Pipeline Mode** – Integrated

**Idle Time-out (minutes)** – 120

Click ‘OK’. These basic settings will create and enable the PEMS Site.

## Publish PEMS Website Source

If the PEMS Website was provided as an installation package, you can skip this section. This section will create the package of files required to create the PEMS Website.

Create a PEMS web site publish package with Visual Studio 2012 Web Site Publish facility. Obtain the latest configuration-managed version of the **Duncan.PEMS.Web** project and open it in Microsoft Visual Studio 2012. Right-click on the project and select **Publish…**  The following dialog will open.

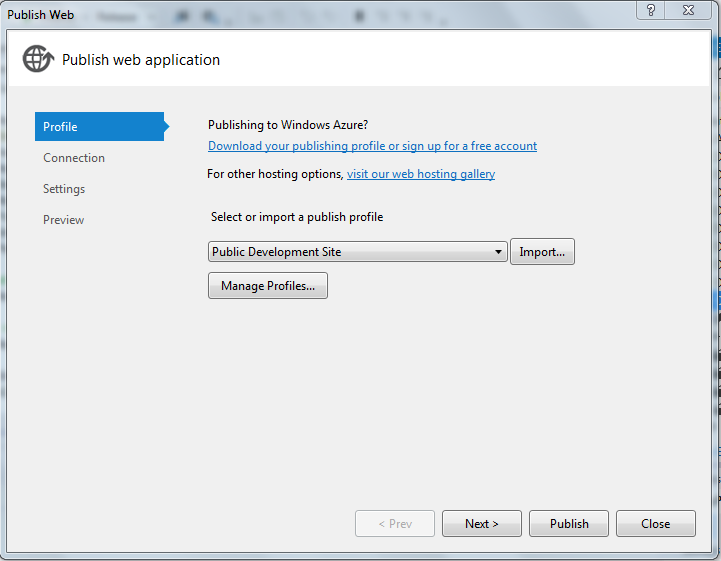


Figure 18 - PEMS Publish Step 1

Select an existing publish profile or create a new profile. Click ‘Next >’.

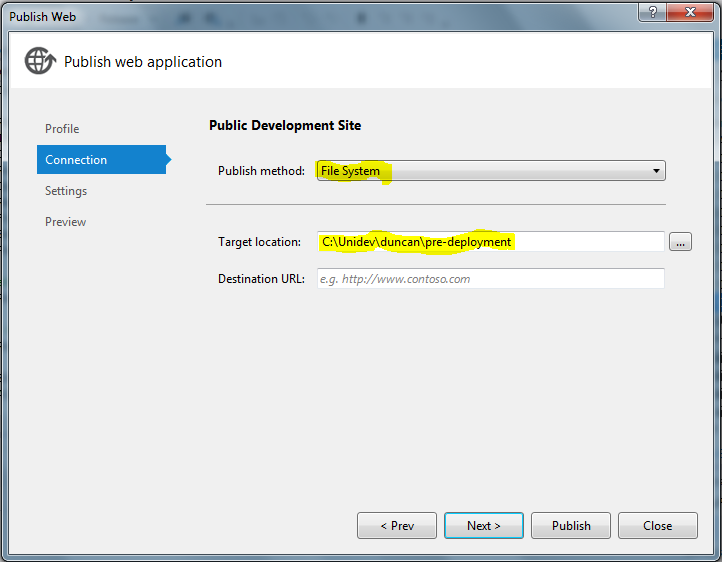


Figure 19 - PEMS Publish Step 2

Select **Publish method** of ‘File System’. Select a **Target location** on your local computer. Destination URL is not required. Click ‘Next >’.

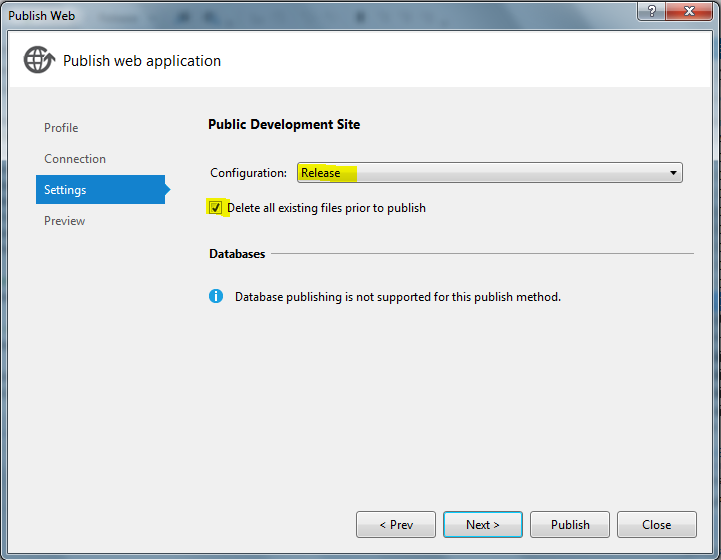


Figure 20 - PEMS Publish Step 3

Select the **Configuration** that will be published. Generally you should be selecting ‘Release’. It is recommended to check **Delete all existing files prior to publish** to create a clean release. Click ‘Next >’.

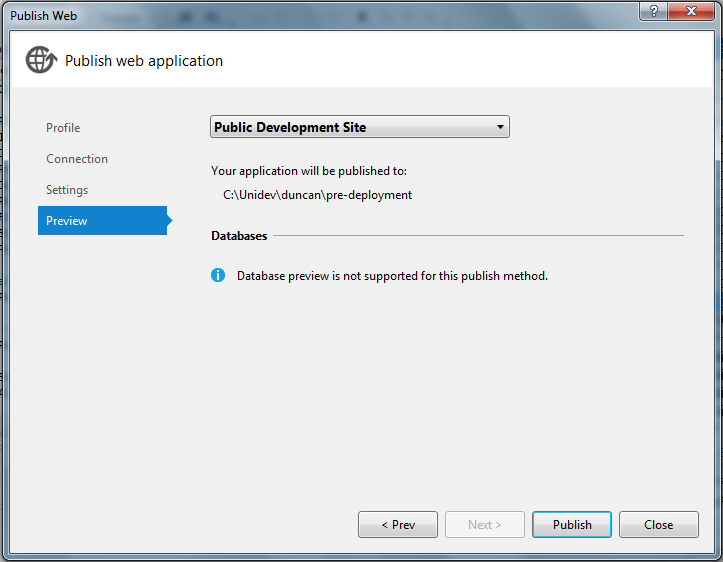


Figure 21 - PEMS Publish Step 4

Review your settings then click ‘Publish’. This will publish all of the files required to create an instance of PEMS on an IIS instance. Visual Studio will indicate progress and result of publish under the ‘Output’ tab at the bottom.

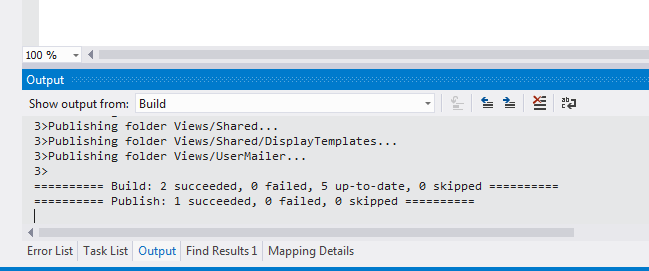


Figure 22 Typical Publish Result

## Deploy PEMS Website Source

If PEMS Website files were provided as a package, navigate to file location and copy all files and folders.

If PEMS Website has just been published as indicated above, navigate to **Target location** for the published files and copy all files and directories from this directory.

Paste the copied files into the **Physical path ([PHYSICAL\_PATH])** location configured in IIS above. If this is an initial installation copy all of the files including web.config, connectionStrings.config, mailSettings.config, and NLog.config.

## PEMS Website Directory Rights

The PEMS web application requires that IIS user (the user that IIS executes under) have write rights to the ‘App\_Data’ directory of PEMS.

Open Windows Explorer, navigate to **[PHYSICAL\_PATH]**/App\_Data folder and grant write rights to that folder and its child folders.

# Discount Website Installation

## Create Discount Website

The creation of the Discount websites requires planning and the acquisition of a domain with which to make each website available for public use. This document does not cover that process. It is assumed that a domain has been determined at this point.

Access the IIS Management Console on the appropriate web server that is going to host the Discount installation. Create the Discount Web Site instance in IIS. There is a separate instance of the Discount site for each customer.

**[DISCOUNT\_WEB\_HOST\_NAME]** will serve as a place holder to indicate usage of the selected domain. In this example, the host name will be *discount.pems.com*.

Determine the physical path where the web site source files will be installed. A typical location is a directory located under C:\inetpub\wwwroot. Make a determination where the web site source files will be located and create a directory there.

**[DISCOUNT\_PHYSICAL\_PATH] –** Placeholder for physical location ofthe PEMS web site. This is where the source files will be copied to in later steps and where the website will reference under IIS. In this example, the physical location will be *C:\inetpub\wwwroot\discount.NSC*.

### Create Site

Access the IIS Management Console on the appropriate web server that is going to host the Discount installation. Create the Discount Website instance in IIS.

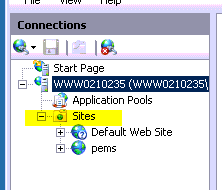


Figure 23 - IIS Discount Website Creation

To create the web site, right-click on ‘Sites’ and select ‘Add Web Site…” to create a new web site.

CREATE HTTP Version then add binding to HTTPS after.

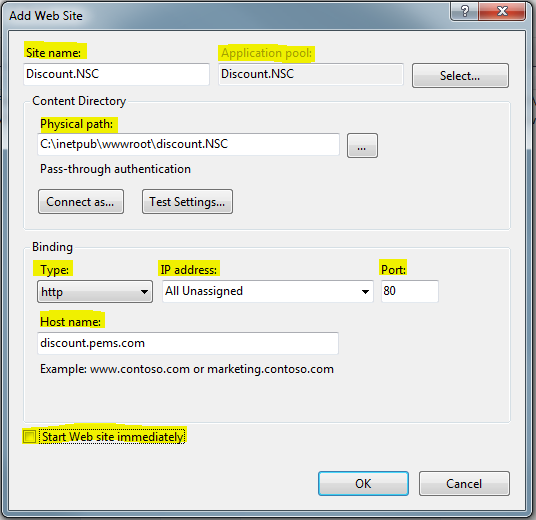


Figure 24 - IIS Discount Site Details

Enter the following fields:

**Site name** – Name as it will appear in IIS. Recommend calling site “PEMS.Discount”.

**Application pool** – IIS will automatically assign a new Application Pool. Note the name for later use.

**Physical path** – Physical path on local machine where web site files will be located. You will be publishing the PEMS web site files to the Physical path later in this installation. This path will be referenced as **[DISCOUNT\_PHYSICAL\_PATH]** in this document**.**

**Host name** – Optional host name. Not required if this site will handle all web traffic on this instance of IIS. The Host name will be the root web site name that will be published to the customers. This host will be determined by Duncan and appropriately registered. This host name will be referenced by **[WEB\_HOST\_NAME]** in this document.

**Binding: Type** – Leave as http.

**Binding: IP address** – Leave as ‘All Unassigned’ unless server has been configured with multiple NIC cards (IP addresses). Verify this setting with the network administrator.

**Binding: Port** – Leave this set at 80 unless requested by network administrator to use other port. A different port may be assigned based on other network attributes and security settings.

**Start Web site immediately** – Uncheck this. Site will be started later in the install process.

Click ‘OK’ to apply settings and create Discount website.

### Create HTTPS Binding

Edit Discount site bindings to enable HTTPS. It is assumed that the Certificate has been obtained and installed on the IIS Server.

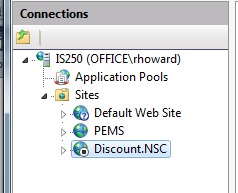


Figure 25 Discount Site Bindings

Right-click on the Discount site (name was set as *Site name:* in previous step) and select “Edit Bindings…”. This will bring up the following dialog.

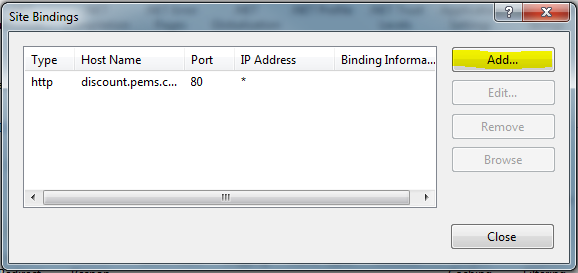


Figure 26 Discount Edit Site Bindings

Click the “Add…” button to bring up the following dialog.

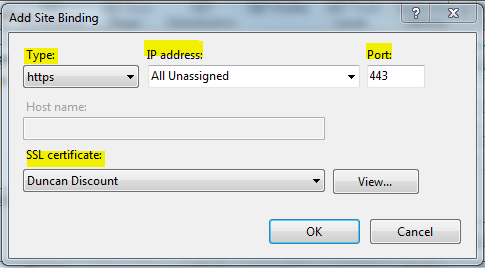


Figure 27 Discount Edit Site Bindings HTTPS

Enter the following fields:

**Type** – Select “https”.

**IP address** – Leave as ‘All Unassigned’ unless server has been configured with multiple NIC cards (IP addresses). Verify this setting with the network administrator.

**Port** – Leave this set at 443 unless requested by network administrator to use other port. A different port may be assigned based on other network attributes and security settings.

**SSL certificate** – Select the appropriate certificate that was pre-installed on IIS. The dialog pictured above has a certificate that is for demonstration purposes and is not the certificate that will be selected.

Click “OK” to assign the new binding. Click “Close” on the next dialog.

### Configure App Pool

Configure the website App Pool via IIS Manager.

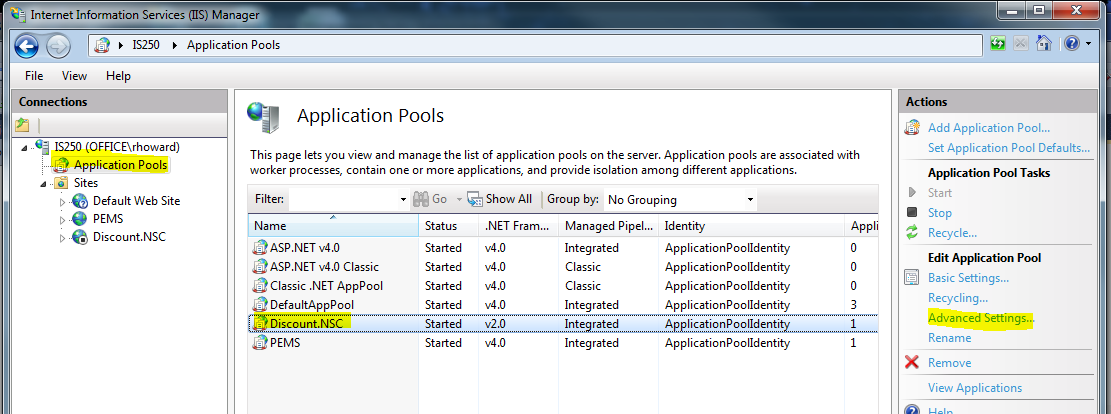


Figure 28 Navigate to Discount App Pool

Click on **Application Pool** under Connections on left side of IIS Manager. Next, click on the appropriate app pool as noted from the creation of the web site. In this example it is ‘Discount.NSC’. Finally, click on **Advanced Settings…**  on the right hand Actions column. This will bring up the following dialog.

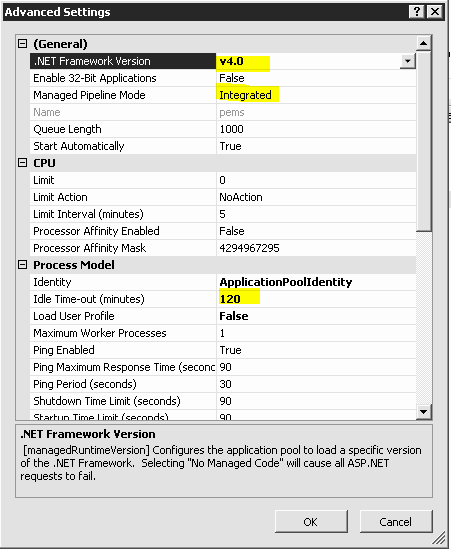


Figure 29 - IIS Discount .NET Version and Process Model

Edit the application pool for the web site. Under Advanced Settings for the application app pool, set the following:

.**NET Framework Version** – v4.0

**Managed Pipeline Mode** – Integrated

**Idle Time-out (minutes)** – 120

These basic settings will create and enable the Discount Site.

## Publish and Deploy Discount Source

If the Discount Website was provided as an installation package, you can skip this section. This section will create the package of files required to create the Discount Website.

Create a Discount website publish package with Visual Studio 2012 Website Publish facility. Obtain the latest configuration-managed version of the **Duncan.Discount.Web** project and open it in Microsoft Visual Studio 2012. Right-click on the project and select **Publish…**  The following dialog will open.

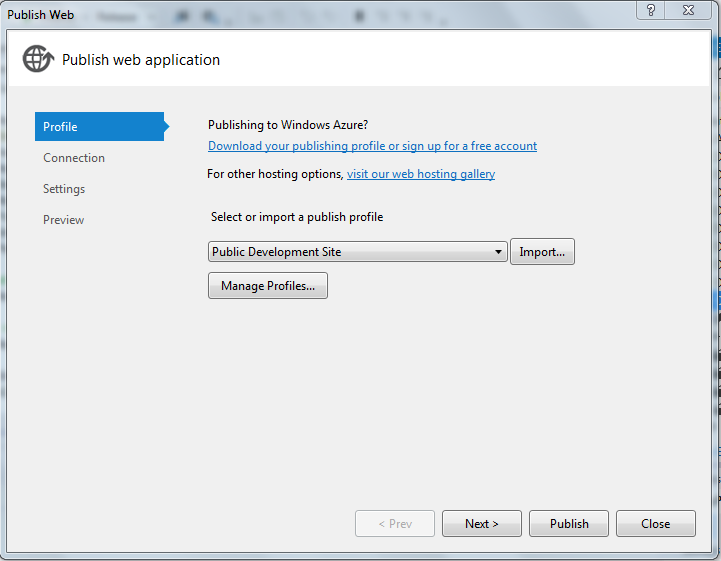


Figure 30 - Discount Publish Step 1

Select an existing publish profile or create a new profile. Click ‘Next >’.

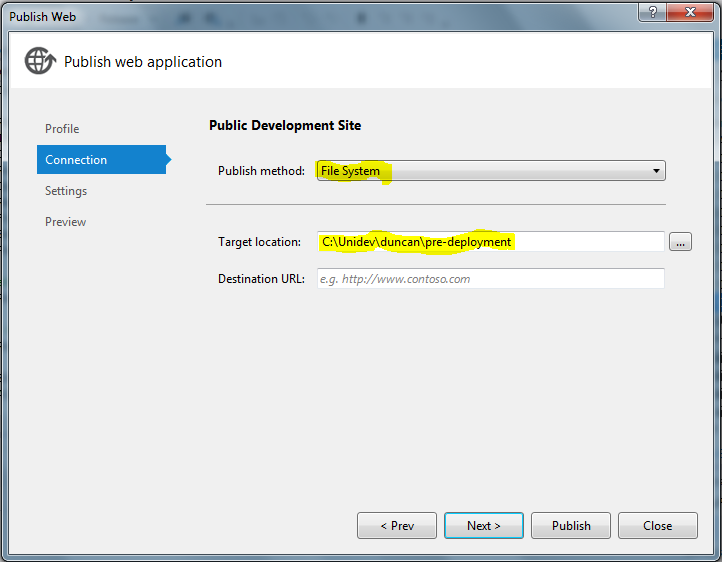


Figure 31 - Discount Publish Step 2

Select **Publish method** of ‘File System’. Select a **Target location** on your local computer. Destination URL is not required. Click ‘Next >’.

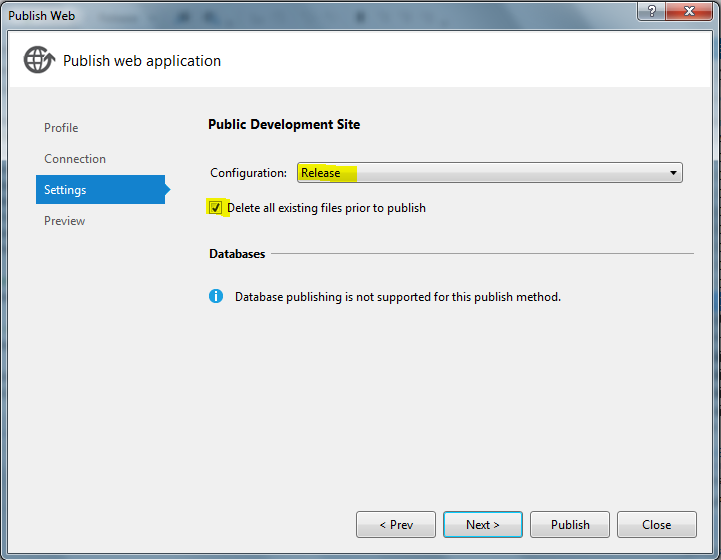


Figure 32 - Discount Publish Step 3

Select the **Configuration** that will be published. Generally you should be selecting ‘Release’. It is recommended to check **Delete all existing files prior to publish** to create a clean release. Click ‘Next >’.

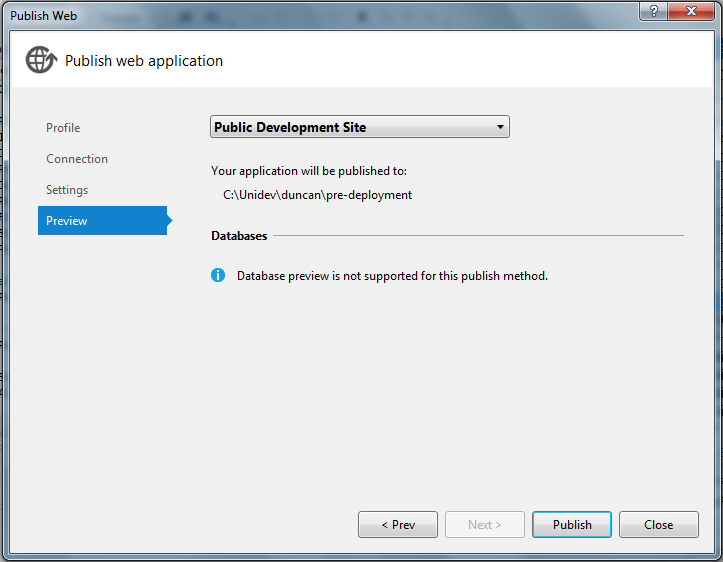
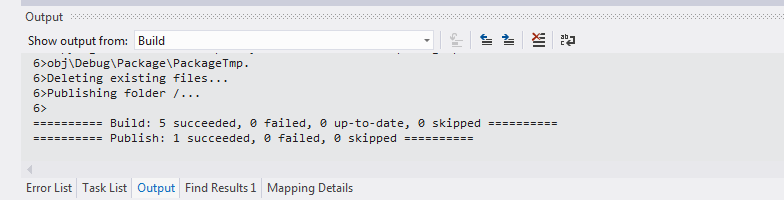


Figure 33 - Discount Publish Step 4

Review your settings then click ‘Publish’. This will publish all of the files required to create an instance of Discount on an IIS instance. Visual Studio will indicate progress and result of publish under the ‘Output’ tab at the bottom.



## Deploy PEMS Website Source

If Discount Website files were provided as a package, navigate to file location and copy all files and folders.

If Discount Website has just been published as indicated above, navigate to **Target location** for the published files and copy all files and directories from this directory.

Paste the copied files into the **Physical path ([DISCOUNT\_PHYSICAL\_PATH])** location configured in IIS above. If this is an initial installation copy all of the files including web.config, connectionStrings.config, and mailSettings.config.

### Deploy Additional Lib Directory

There is an additional directory that is not created as part of the project “Publish” process. This directory contains older version(s) of .dll files that are required by third-party libraries such as Izenda.

Copy the folder **INSTALL\_ROOT\lib** and all of its contents into the **Physical path ([DISCOUNT\_PHYSICAL\_PATH])**. This will create a folder at the root of the PEMS web application called “lib” with copies of any older .dll versions needed by PEMS.

It is important that the relative directory structure stay intact since there is an entry in the *web.config* for PEMS that references files in this location.

# Configuration Files – PEMS

Copies of all the configuration files used for the PEMS Website can be found under **INSTALL\_ROOT\ConfigFiles\PEMS.** It is recommended that you make backup copies of the configuration files before you edit them.

The following files will be used to configure the PEMS Website.

* mailSettings.config – Configures e-mail connectivity for site.
* connectionStrings.config – Configures database connection strings for site.
* NLog.config – Logging configuration.
* web.config – Website and application configuration.

## mailSettings.config

Setting used to send mail on behalf of the PEMS System. All PEMS-generated e-mail (I.E. Forgot password) will be sent from these settings and credentials.

<?xml version="1.0"?>

<smtp from="[MAIL\_FROM\_ADDRESS]" >

<network host="[MAIL\_HOST]" port="[MAIL\_PORT]" defaultCredentials="false" userName="[MAIL\_USER]" password="[MAIL\_USER]" />

</smtp>

[MAIL\_FROM\_ADDRESS] – Address of the PEMS administrative e-mail source. Formatted as ‘user@host’

[MAIL\_HOST] – Enter the host name of the SMTP mail server. For instance “mail.pemsportal.com”. Must be resolvable and exist.

[MAIL\_PORT] – Port to connect to on mail host. Normally “25”.

[MAIL\_USER], [MAIL\_PASSWORD] – Assumes that mail can only be sent by a particular user set up to manage mail being sent from this instance of PEMS.

## connectionStrings.config

Database connection strings for PEMS System. All connection strings are defined in this file and imported to the web.config file.

There are four types of connection strings defined in this file. There are the connection strings to connect to the RBAC database. There are connection strings to connect to the various instances of the PEMS databases. There are connection strings to connect to the various instances of the Maintenance databases. Lastly there are connection strings to connect to instances of the Reporting databases.

### RBAC Connections

There are two connection strings to the RBAC database: Duncan.Membership.Connector and PEMRBACEntities

**Duncan.Membership.Connector** – Connection string used by Membership system for user login, lost password token and other functions of authentication. This connection string is also used by NetSqlAzMan for connection to the authorization tables. These values will be set to connect to database created above.

Edit the following placeholders:

**[RBACSERVERNAME]** – The database server and optionally the instance. This should be set to the value indicated by **[SQL\_SERVER]** in Section 2.1 above.

**[RBACDATABASENAME]** – Name of database as defined earlier in this installation process.

**[RBACUSERNAME]** – Database user name for RBAC database. This should be set to value indicated by **[SQL\_USER\_NAME]** in Section 2.1 above.

**[RBACPASSWORD]** – Database password for RBAC database. This should be set to value indicated by **[SQL\_USER\_PASSWORD]** in Section 2.1 above.

**PEMRBACEntities** – Connection string used for the EntityFramework access to common PEMS database resources such as customer profiles and user profiles. This connection is used during nominal operation of PEMS. These values will be set to connect to database created above.

Edit the following placeholders:

**[RBACSERVERNAME]** – The database server and optionally the instance. This should be set to the value indicated by **[SQL\_SERVER]** in Section 2.1 above.

**[RBACDATABASENAME]** – Name of database as defined earlier in this installation process.

**[RBACUSERNAME]** – Database user name for RBAC database. This should be set to value indicated by **[SQL\_USER\_NAME]** in Section 2.1 above.

**[RBACPASSWORD]** – Database password for RBAC database. This should be set to value indicated by **[SQL\_USER\_PASSWORD]** in Section 2.1 above.

Note: Duncan.Membership.Connector and PEMRBACEntities must point to same database instance.

### PEMS (RipNet) Connections

There should be n entries for connection to PEMS. Each entry to connect to PEMS must be patterned after the “appSettings” key *pems.database.cnx\_pattern* in web.config. This entry gives the model that is used to find connections to PEMS instances. Each entry in connectionStrings.config must be based on the *pems.database.cnx\_pattern* with numbers 1 to n appended to it. These numeric values must be contiguous and start at 1. For example, if the value of *pems.database.cnx\_pattern* was “PEMS.CNX.” and there are three instances of the PEMS database then there should be three connection strings named “PEMS.CNX.1”, “PEMS.CNX.2” and “PEMS.CNX.3”. Each entry should point to a different instance of PEMS.

These connection string names are used to determine the location of each customer’s PEMS data and will be assigned to a customer during the customer creation process.

Note: The creation and initialization of PEMS database is not included in this document.

Edit the following placeholders:

**[PEMSSERVERNAME]** – The name, and, optionally, instance of the PEMS database server.

**[PEMSDATABASENAME]** – The name of the PEMS database instance.

**[PEMSUSERNAME]** – User name with rights to the PEMS database instance.

**[PEMSPASSWORD]** – Password for PEMS database user.

Do this for each separate PEMS.CNX.# that is required.

### Reports Connection

There should be n entries for connection to the reporting database instances. Each entry to connect to Reports must be patterned after the “appSettings” key *reporting.database.cnx\_pattern* in web.config. This entry gives the model that is used to find connections to PEMS instances. Each entry in connectionStrings.config must be based on the *reporting.database.cnx\_pattern* with numbers 1 to n appended to it. These numeric values must be contiguous and start at 1. For example, if the value of *reporting.database.cnx\_pattern* was “REPORTING.CNX.” and there are three instances of the PEMS database then there should be three connection strings named “REPORTING.CNX.1”, “REPORTING.CNX.2” and “REPORTING.CNX.3”. Each entry should point to a different instance of the Reporting database. Generally there should be a one-to-one match with the PEMS (RipNet) connections.

These connection string names are used to determine the location of each customer’s Reporting data and will be assigned to a customer during the customer creation process. If there is not a separate Reports database instance then point to the same database instance as the associated PEMS (RipNet) connection.

Note: The creation and initialization of Reporting database is not included in this document.

Edit the following placeholders:

**[REPORTINGSERVERNAME]** – The name, and, optionally, instance of the Reports database server.

**[REPORTINGDATABASENAME]** – The name of the Reports database instance.

**[REPORTINGUSERNAME]** – User name with rights to the Reports database instance.

**[REPORTINGPASSWORD]** – Password for Reports database user.

Do this for each separate REPORTING.CNX.# that is required.

### Maintenance Connection

There should be n entries for connection to the maintenance database instances. Each entry to connect to Maintenance must be patterned after the “appSettings” key *maint.database.cnx\_pattern* in web.config. This entry gives the model that is used to find connections to Maintenance instances. Each entry in connectionStrings.config must be based on the *maint.database.cnx\_pattern* with numbers 1 to n appended to it. These numeric values must be contiguous and start at 1. For example, if the value of *maint.database.cnx\_pattern* was “MAINT.CNX.” and there are three instances of the PEMS database then there should be three connection strings named “MAINT.CNX.1”, “MAINT.CNX.2” and “MAINT.CNX.3”. Each entry should point to a different instance of the Maintenance database.

These connection string names are used to determine the location of each maintenance group’s Maintenance data and will be assigned to a group during the group creation process.

Note: The creation and initialization of Maintenance database is not included in this document.

Edit the following placeholders:

**[MAINTENANCESERVERNAME]** – The name, and, optionally, instance of the Maintenance database server.

**[MAINTENANCEDATABASENAME]** – The name of the Maintenance database instance.

**[MAINTENANCEUSERNAME]** – User name with rights to the Maintenance database instance.

**[MAINTENANCEPASSWORD]** – Password for Maintenance database user.

Do this for each separate MAINT.CNX.# that is required.

## NLog.config

This configuration file is used to set the logging values for NLog logging facility. There are two sections of interest in the config file, targets and rules. NLog is provided with initial settings that should be suitable for initial production.

It is recommend that you review <http://nlog-project.org/> for more details on NLog.

### Targets

<targets async="true" >

<target xsi:type="File" name="PEMS\_file"

fileName="${basedir}/App\_Data/logs/PEMS\_${shortdate}.log"

layout="${longdate} | ${level} | ${callsite} | ${message}" />

</targets>

fileName – Path and file name template. Log files generated by NLog logging will be written to this location and named according to template. The logs are initially configured to be written to the website-relative ~/App\_Data/logs directory and named PEMS\_ YYYY-MM-DD.log.

${basedir} The root of the web application (website)

${shortdate} The date when file was created in form of YYYY-MM-DD.

layout – Format of data written by log calls. See <https://github.com/nlog/NLog/wiki/Configuration-file#wiki-layouts-and-layout-renderers> for details.

See <https://github.com/nlog/NLog/wiki/Configuration-file#wiki-targets> for more details on targets.

### Rules

<rules>

<!-- add your logging rules here -->

<logger name="\*" minlevel="Error" writeTo="PEMS\_file" />

</rules>

See https://github.com/nlog/NLog/wiki/Configuration-file#wiki-rules for more details on targets.

## web.config

The web.config has relatively few settings that require modification for operation. The config file comes from the publish operation with the majority of settings defaulted to usable values.

### <appSettings> section

***RouteDebugger:Enabled*** - Debug setting to show which MVC route is being applied to the page. This should normally be set to “false”.

***elmah*…** - Elmah logging settings. See <http://code.google.com/p/elmah/> for more details.

***rbac.menu.template.auth*** – Name of XML template file that is used to configure RBAC (NetSqlAzMan) system during creation of a normal customer (city). This is presently set to "Duncan.Auth.Template.xml". Do not change unless you have strong understanding of the configuration of RBAC via XML.

***rbac.menu.template.maint***– Name of XML template file that is used to configure RBAC (NetSqlAzMan) system during creation of a normal maintenance group. This is presently set to "Duncan.Maint.Template.xml ". Do not change unless you have strong understanding of the configuration of RBAC via XML.

**rbac.menu.template.dir** – Website relative location of the *rbac.menu.template.auth* and *rbac.menu.template.maint* files. PEMS will look in this location when creating either a customer or a maintenace group for the appropriate template file. Website must have read access to this directory. Present value is "~/App\_Data/rbac\_configs/".

***rbac.menu.template.upload*** – Temporary working directory used by the RBAC (NetSqlAzMan) configuration process. The website must have full access to this location (read/write/delete). Presently set to "~/App\_Data/uploads/".

***pems.asset.samples*** – Directory where examples of CSV file for bulk asset imports and instruction templates describing bulk asset import files are stored. The website must have read access to this directory. Presently set to "~/App\_Data/assets/".

***pems.asset.upload*** – Working directory for bulk asset imports process. The website must have full access to this location (read/write/delete). Presently set to "~/App\_Data/uploads/".

***pems.logging.access.pages*** – Setting to indicate if system is to log MVC page access. If “true” then log each PEMS page access. This will log access only if a valid “Area” and “City” are present in the base PemsController. Generally this should be set to “false”.

***pems.logging.access.ajax*** – Setting to indicate if system is to log AJAX calls to MVC-based controller actions. If “true” then log each PEMS AJAX access. This will log access only if a “Area” and “City” are blank in the base PemsController. Generally this should be set to “false”.

***pems.logging.access.rights.allowed, pems.logging.access.rights.denied, pems.logging.access.rights.undefined*** – Setting used in conjunction with *pems.logging.access.pages* and *pems.logging.access.ajax* to indicate if calls that are return certain access rights by the underlying authorization system are logged. Generally these will be set to “true” and controlled by higher level logging settings. These setting are used predominately in debugging and monitoring of the authorization settings.

***pems.logging.log\_attempts*** – Enables logging of all login attempts. Used to monitor system for excessive login attempts or issues where user believes they are logging in with the correct user name/password but are having difficulty. Generally this should be set to “false”.

***DaysPWValidFor*** – Used to globally set how long a user password is valid for before the user will be required to change their password. Setting is in days. Presently set to "180".

***pems.default.support\_number*** – Telephone number presented to anonymous page users if support is needed. Set this to the appropriate value for Duncan support for PEMS.

***pems.database.cnx\_pattern*** – String defining the template model for PEMS (RipNet) connection string names used in the connectionStrings.config file. Presently set to "PEMS.CNX.". Recommend leaving this setting as-is.

***pems.database.default\_cnx*** – Indicates the default PEMS (RipNet) connection string. This is used when creating customer. This value MUST exist in connectionStrings.config. Presently set to "PEMS.CNX.1".

***maint.database.cnx\_pattern*** – String defining the template model for Maintenance connection string names used in the connectionStrings.config file. Presently set to "MAINT.CNX.". Recommend leaving this setting as-is.

***maint.database.default\_cnx*** – Indicates the default Maintenance connection string. This is used when creating customer. This value MUST exist in connectionStrings.config. Presently set to "MAINT.CNX.1".

***reporting.database.cnx\_pattern*** – String defining the template model for Reporting connection string names used in the connectionStrings.config file. Presently set to " REPORTING.CNX.". Recommend leaving this setting as-is.

***reporting.database.default\_cnx*** – Indicates the default Reporting connection string. This value MUST exist in connectionStrings.config. Presently set to " REPORTING.CNX.1".

***Izenda.LicKey*** – Indicates the Izenda license key for the internal version of the reporting viewer. This setting is presently not being used as iFrames are being used to view reports.

***Izenda.ReportPath*** – Website-relative path to reporting resources and the reports. This setting is presently not being used as iFrames are being used to view reports. Presently set to "~/Reporting/Reports"

***Izenda.ReportViewer*** – Setting to indicate the type of Izenda report viewer that is being used. Presently Izenda reports may be viewed either by an internal MVC version or via iFrames. Depending on licensing models, select “iframes” if connecting to a Izenda reporting server or select “internal” if IIS app pool is constrained to number of cores licensed for Izenda. Valid values are: “iframes” and “internal”.

***pems.security.timeout*** – Sets the number of seconds that a session remains idle before it is automatically logged out. If there is no interaction with the client session within the configured time span, a dialog will be presented to user to continue the session or to close the session.

***pems.security.timeout.warning*** – Sets the number of seconds that the logout warning will be displayed for user before session times out and the user is automatically logged out. It is defaulted to 30 seconds if value is not configured.

***pems.security.polling*** – Sets the number of seconds between polling calls from the client session to the server. This is used to keep the server session active. This value must be less than ***pems.security.timeout***. It is defaulted to 300 seconds if value is not configured. Do not set this value below 60 seconds since it will create excess network traffic.

***pems.roles.required.#*** – Where # is equal 1 to n contiguous. Sets required authorization items for roles being defined for customers. Is applicable to all matching authorization settings. The value is a delimited pair of strings matching the authorization group parent heading and the associated child authorization text. These strings are discoverable and correspond to the authtext entries in the xml file referenced by ***rbac.menu.template.auth.*** Do not change unless you are familiar with the authorization settings of NetSqlAzMan.

### Elmah Settings

In order to enable and configure Elmah logging the following entries must be made in web.config. See <http://elmah.googlecode.com/svn/tags/REL-1.2/samples/web.config> for more information. The sections of the web.config that concern Elmah assume that you are using IIS 7 or newer.

In <system.webServer> section:

<system.webServer>

<handlers>

…

<add name="Elmah" verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />

…

</handlers>

<modules>

…

<add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah" preCondition="managedHandler" />

…

</modules>

In <system.web> section:

<system.web>

<httpModules>

…

<add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah" />

<add name="ErrorMail" type="Elmah.ErrorMailModule, Elmah" />

…

</httpModules>

<httpHandlers>

…

<add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />

…

</httpHandlers>

In <elmah> section:

<elmah>

<security allowRemoteAccess="[REMOTE\_ACCESS]" />

<errorLog type="Elmah.XmlFileErrorLog, Elmah" logPath="[LOG\_PATH]" />

<errorMail

from="[MAIL\_FROM\_ADDRESS]"

to="[MAIL\_TO\_ADDRESS]"

subject="[MAIL\_SUBJECT\_LINE]"

priority="Low"

async="true"

smtpPort="[MAIL\_PORT]"

smtpServer="[MAIL\_HOST]"

useSsl="false"

userName="[MAIL\_USER]"

password="[MAIL\_PASSWORD]"

/>

</elmah>

Set the following:

[REMOTE\_ACCESS] – Set to 1 to enable remote access to the Elmah log page.

[LOG\_PATH] – Website-relative path to folder where Elmah will store exception logs. Presently set to “~/App\_Data”. Web server process must have “write” rights to path.

[MAIL\_FROM\_ADDRESS] – Address of the PEMS exception log e-mail source. Formatted as ‘user@host’

[MAIL\_TO\_ADDRESS] – Address of the recipients of the PEMS exception logs. Separate addresses with a “;” for multiple recipients. Formatted as ‘user@host’

[MAIL\_HOST] – Enter the host name of the mail server. For instance “mail.pemsportal.com”. Must be resolvable and exist.

[MAIL\_PORT] – Port to connect to on mail host. Normally “25”.

[MAIL\_USER], [MAIL\_PASSWORD] – Assumes that mail can only be sent by a particular user set up to manage mail being sent from this instance of PEMS.

## Publishing PEMS Config Files

Once all configuration files have been edited, copy the files from their location, **INSTALL\_ROOT\ConfigFiles\PEMS**, to the root of the new instance of PEMS Website, **[PHYSICAL\_PATH]** as referenced in Section 6.1.1. Overwrite the existing files.

The configuration of PEMS Website is now complete. The following files have been edited and published.

* mailSettings.config – Configures e-mail connectivity for site.
* connectionStrings.config – Configures database connection strings for site.
* NLog.config – Logging configuration.
* web.config – Website and application configuration.

# Configuration Files - Discount

## mailSettings.config

Setting used to send mail on behalf of the Discount System. All Discount-generated e-mail (I.E. Forgot password) will be sent from these settings and credentials.

<?xml version="1.0"?>

<smtp from="[MAIL\_FROM\_ADDRESS]" >

<network host="[MAIL\_HOST]" port="[MAIL\_PORT]" defaultCredentials="false" userName="[MAIL\_USER]" password="[MAIL\_USER]" />

</smtp>

[MAIL\_FROM\_ADDRESS] – Address of the Discount administrative e-mail source. Formatted as ‘user@host’

[MAIL\_HOST] – Enter the host name of the mail server. For instance “mail.pemsportal.com”. Must be resolvable and exist.

[MAIL\_PORT] – Port to connect to on mail host. Normally “25”.

[MAIL\_USER], [MAIL\_PASSWORD] – Assumes that mail can only be sent by a particular user set up to manage mail being sent from this instance of Discount.

## connectionStrings.config

Database connection strings for Discount System. All connection strings are defined in this file and imported to the web.config file.

There are two types of connection strings defined in this file. There are the connection strings to connect to the RBAC database. There are connection strings to connect to the appropriate instance of the PEMS database.

### RBAC Connection

There is only one connection string to the RBAC database: **RbacEntities.**

**RbacEntities** – Connection string used for the EntityFramework access to common PEMS database resources such as customer profiles and user profiles. This connection is used during nominal operation of Discount.

Edit the following placeholders:

**[RBACSERVERNAME]** – The database server and optionally the instance. This should be set to the value indicated by **[SQL\_SERVER]** in Section 2.1 above.

**[RBACDATABASENAME]** – Name of database as defined earlier in this installation process.

**[RBACUSERNAME]** – Database user name for RBAC database. This should be set to value indicated by **[SQL\_USER\_NAME]** in Section 2.1 above.

**[RBACPASSWORD]** – Database password for RBAC database. This should be set to value indicated by **[SQL\_USER\_PASSWORD]** in Section 2.1 above.

Note: **Membership.Connector** and **RbacEntities** DO NOT point to same database instance.

### PEMS (RipNet) Connections

There are two connection strings for Discount to the customer PEMS (RipNet) database instance: **Entities** and **Membership.Connector.**

These connection strings MUST point to the proper PEMS (RipNet) instance for the customer supported by this instance of Discount website.

Note: The creation and initialization of PEMS database is not included in this document.

**Entities** – Connection string used by Discount for accessing discount plans and other customer-specific data.

Edit the following placeholders:

**[PEMSSERVERNAME]** – The name, and, optionally, instance of the PEMS database server.

**[PEMSDATABASENAME]** – The name of the PEMS database instance.

**[PEMSUSERNAME]** – User name with rights to the PEMS database instance.

**[PEMSPASSWORD]** – Password for PEMS database user.

**Membership.Connector** – Connection string used by Membership system for user login, lost password token and other functions of authentication.

Edit the following placeholders:

**[RBACSERVERNAME]** – The database server and optionally the instance. This should be set to the value indicated by **[SQL\_SERVER]** in Section 2.1 above.

**[RBACDATABASENAME]** – Name of database as defined earlier in this installation process.

**[RBACUSERNAME]** – Database user name for RBAC database. This should be set to value indicated by **[SQL\_USER\_NAME]** in Section 2.1 above.

**[RBACPASSWORD]** – Database password for RBAC database. This should be set to value indicated by **[SQL\_USER\_PASSWORD]** in Section 2.1 above.

## web.config

The web.config has relatively few settings that require modification for operation. The config file comes from the publish operation with the majority of settings defaulted to usable values. The following sections describe various web.config settings.

### <appSettings> section

***CustomerId*** – Id of customer that is served by this instance of the Discount site. This customer id MUST be homed on the PEMS (RipNet) database instance pointed to by connection strings **Entities** and **Membership.Connector** above.

***Password.ValidFor*** - Used to globally set how long a user password is valid for before the user will be required to change their password. Setting is in days. Presently set to “90”.

***SupportNumber*** - The anonymous support number for the system. used when a user is having problems logging in.

***Logo*** – Website-relative path and file for the customer logo. Each different customer will have its own logo. The logo is a banner that is 1000x140 pixels.

***ForgotPassword.From*** - The email that the forgot password form will use in the "From" field

***ForgotPassword.Sender*** - The email that the forgot password form will use in the "Sender" field.

***ForgotPassword.Subject*** - the text that the forgot password form will use in the "Subject" field.

***ApplicationSubmission.From*** - The email that the application submission form will use in the "From" field.

***ApplicationSubmission.Sender*** - The email that the application submission form will use in the "Sender" field.

***ApplicationSubmission.Subject*** - The text that the application submission form will use in the "Subject" field.

## Publishing Discount Config Files

Once all configuration files have been edited, copy the files from their location, **INSTALL\_ROOT\ConfigFiles\Discount**, to the root of the new instance of Discount Website, **[PHYSICAL\_PATH]** as referenced in Section 7.1.1. Overwrite the existing files.

The configuration of Discount Website is now complete. The following files have been edited and published.

* mailSettings.config – Configures e-mail connectivity for site.
* connectionStrings.config – Configures database connection strings for site.
* web.config – Website and application configuration.

# Starting the Websites

Both PEMS Website and Discount Website are ready to be started at this point.

In the IIS Management Studio, right click the new instance of PEMS Website, named in Section 6.1.1 and select ‘Manage Web Site’ -> ‘Start’.

In the IIS Management Studio, right click the new instance of Discount Website, named in Section 7.1.1 and select ‘Manage Web Site’ -> ‘Start’.

Verify that both web sites indicate that they are running.

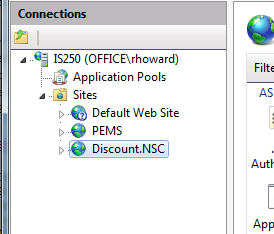


Figure 34 Verify both sites running

# Testing PEMS Website

## Initial Access

At this point the basic PEMS Website should be up and running. Open a browser and navigate to **[WEB\_HOST\_NAME]** as configured in Section 6.1. You should see the following web page displayed.

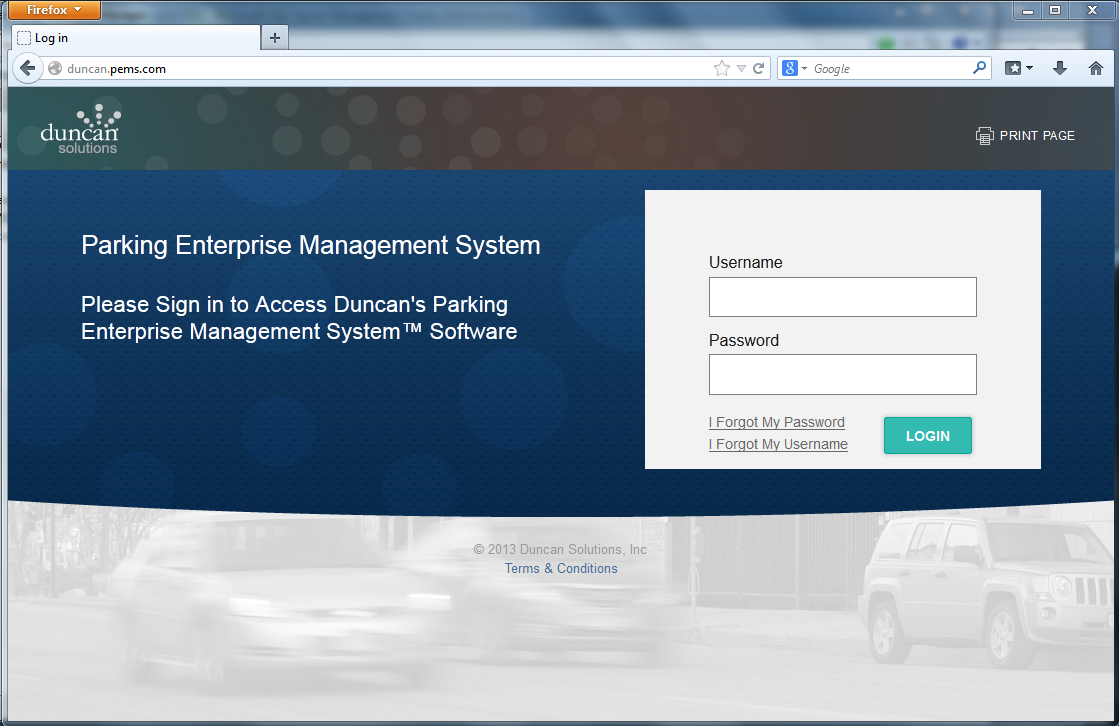


Figure 35 Inital PEMS Website Test

### Troubleshooting

The initial presentation of the web site can fail for many reasons. Most common reasons are errors in the config files and database connectivity issues. The following are typical failures encountered.

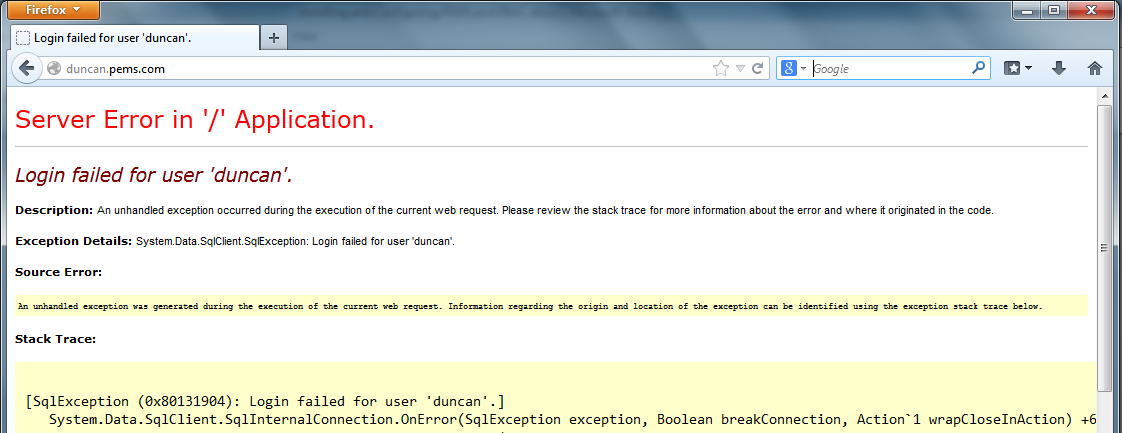


Figure 36 Error in SQL User Name or Password - Duncan.Membership.Connector

Bad SQL Login – Error in *connectionStrings.config* in the **Duncan.Membership.Connector** connection string. SQL user name or password mistyped. Check RBAC database SQL user name and password and correct in *connectionStrings.config* and save. Retest starting at Section 11.1.

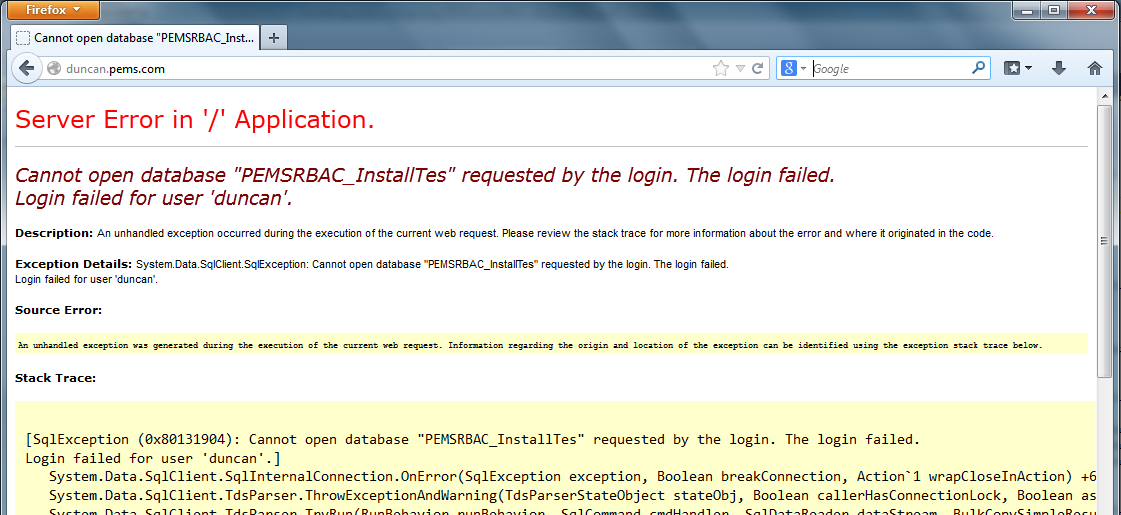


Figure 37 Configuration Error in SQL Initial Catalog Name - Duncan.Membership.Connector

Bad SQL Login – Error in *connectionStrings.config* in the **Duncan.Membership.Connector** connection string. SQL Initial Catalog name in error. Check RBAC database name and correct in *connectionStrings.config* and save. Retest starting at Section 11.1.

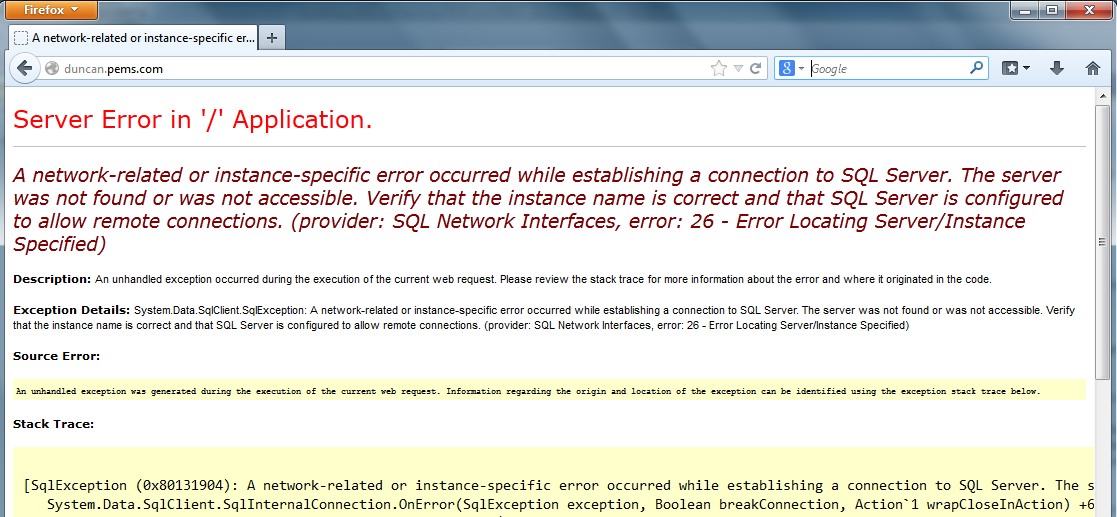


Figure 38 Configuration Error in SQL Server Name - Duncan.Membership.Connector

Bad SQL Server Name – Error in *connectionStrings.config* in the **Duncan.Membership.Connector** connection string. SQL server name mistyped. Check RBAC database SQL Server name and correct in *connectionStrings.config* and save. Retest starting at Section 11.1.

## Initial Login

Test the administrative login. Enter the **Username** of ‘Admin’ and the **Password** of ‘password’. If you entered a different password in Section 5.1, enter that password.

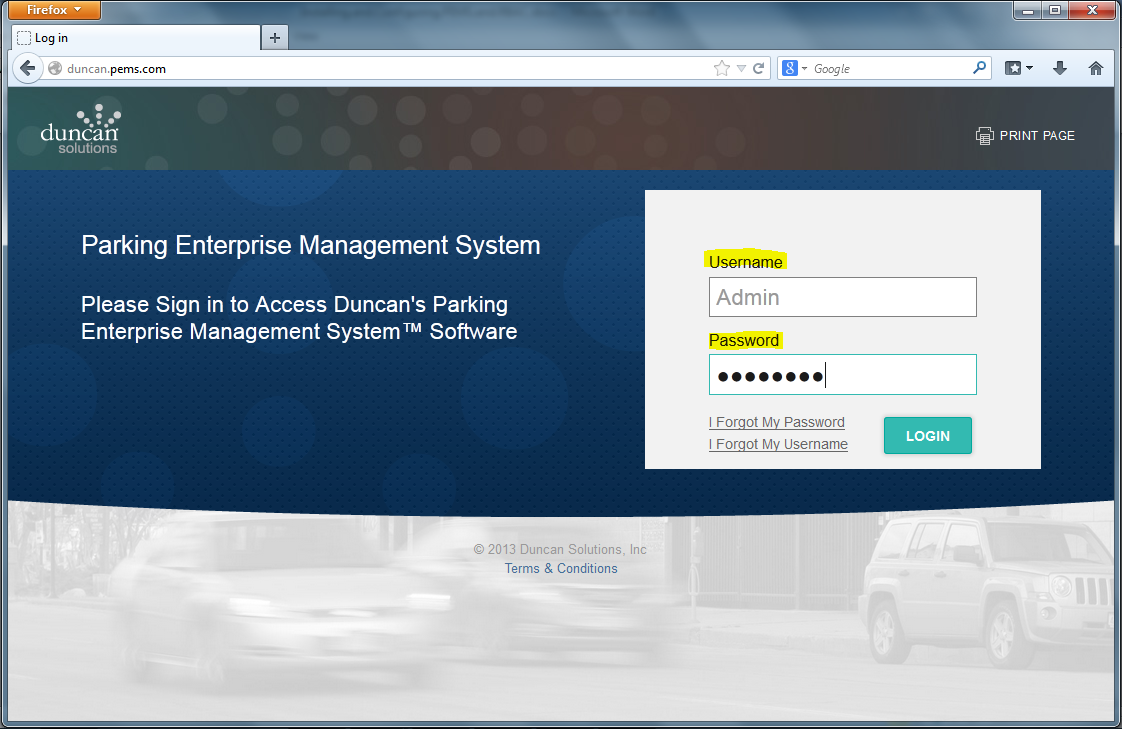


Figure 39 Testing Admin login

On successful login you will be taken to the following page where you will be required to change your password.

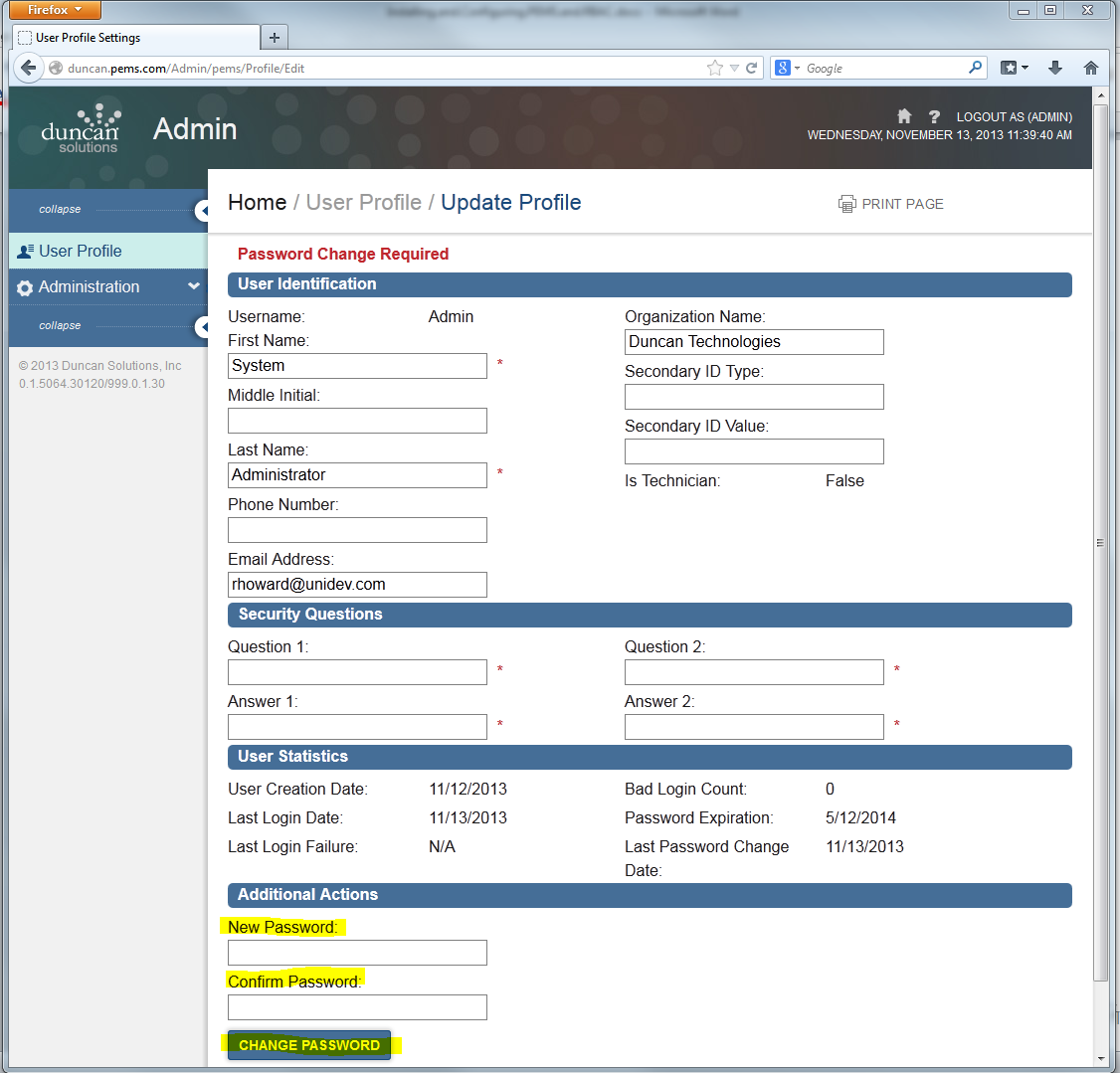


Figure 40 Testing Admin login - Required Password Change

Scroll to the bottom of the page, if required, and enter **New Password** and reenter in **Confirm Password.** The password must contain a minimum of one lower case character, one upper case character, one number and one special character. The password must be between 7 and 8 characters long.

Press the ‘**CHANGE PASSWORD’** button when done. If successful, you will be taken back to the login screen. Log back in using your new password.

The basic functionality of the PEMS Website and the database connections to the RBAC database have now been tested.

### Troubleshooting

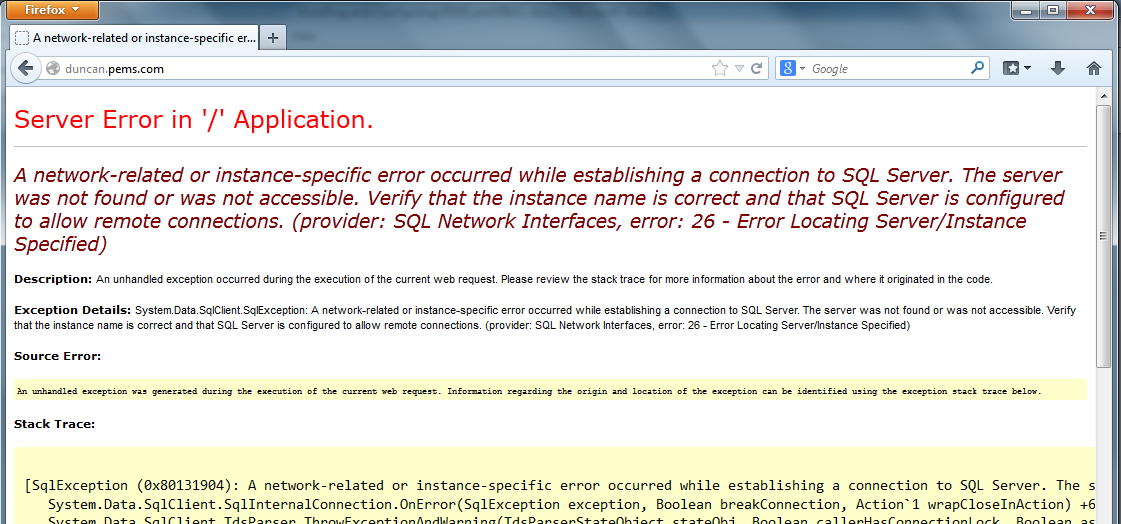


Figure 41 Configuration Error in SQL Server Name – PEMRBACEntities

Bad SQL Server Name – Error in *connectionStrings.config* in the **PEMRBACEntities** connection string. SQL server name mistyped. Check RBAC database SQL Server name and correct in *connectionStrings.config* and save. Retest starting at Section 11.1.

Similar error screens after attempt at login may indicate other errors in the **PEMRBACEntities** connection string.

## Creating First PEMS Customer

# Testing Discount Website

## Initial Access

At this point the basic Discount Website should be up and running. Open a browser and navigate to **[DISCOUNT\_WEB\_HOST\_NAME]** as configured in Section 7.1. You should see the following web page displayed.

In this example the Discount site is for North Sydney Council.

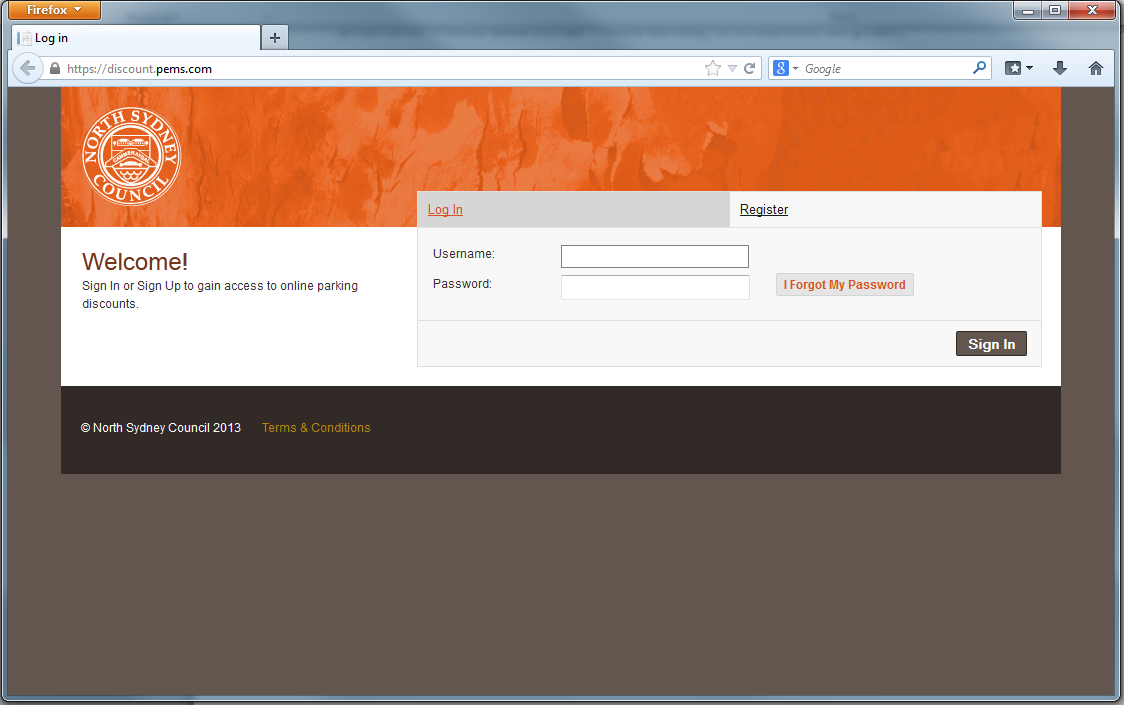


Figure 42 Initial Discount Website Test

### Troubleshooting

The initial presentation of the web site can fail for many reasons. Most common reasons are errors in the config files and database connectivity issues. The following are typical failures encountered.

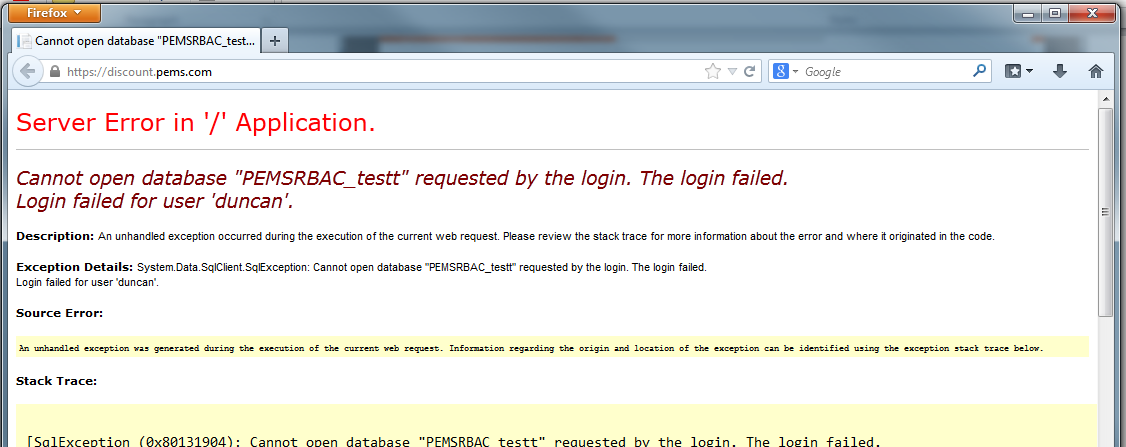


Figure 43 Configuration Error in SQL Initial Catalog Name – RbacEntities

Bad SQL Server Name – Error in *connectionStrings.config* in the **RbacEntities** connection string. SQL server name mistyped. Check RBAC database SQL Server name and correct in *connectionStrings.config* and save. Retest starting at Section 12.1.

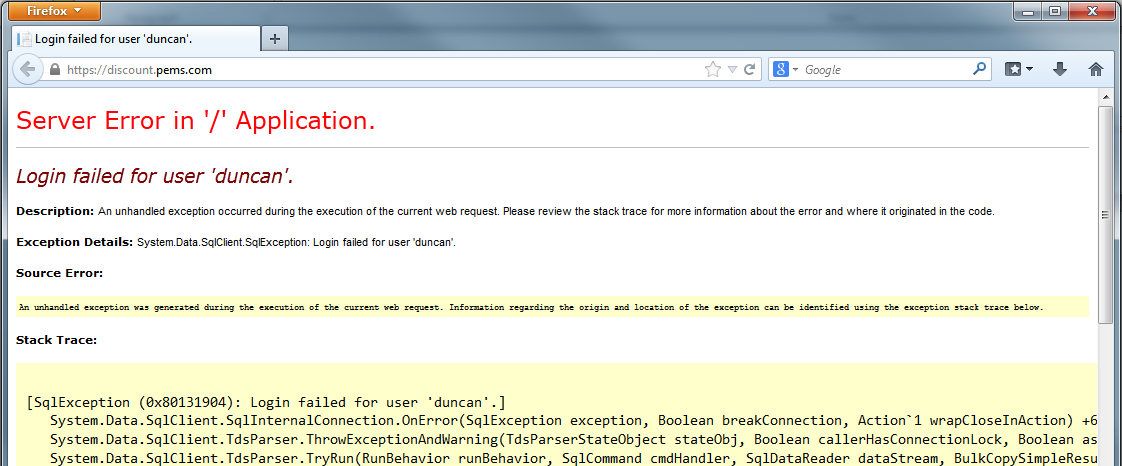


Figure 44 Configuration Error in SQL User Name or Password – RbacEntities

Bad SQL Login – Error in *connectionStrings.config* in the **RbacEntities** connection string. SQL user name or password mistyped. Check RBAC database SQL user name and password and correct in *connectionStrings.config* and save. Retest starting at Section 12.1.

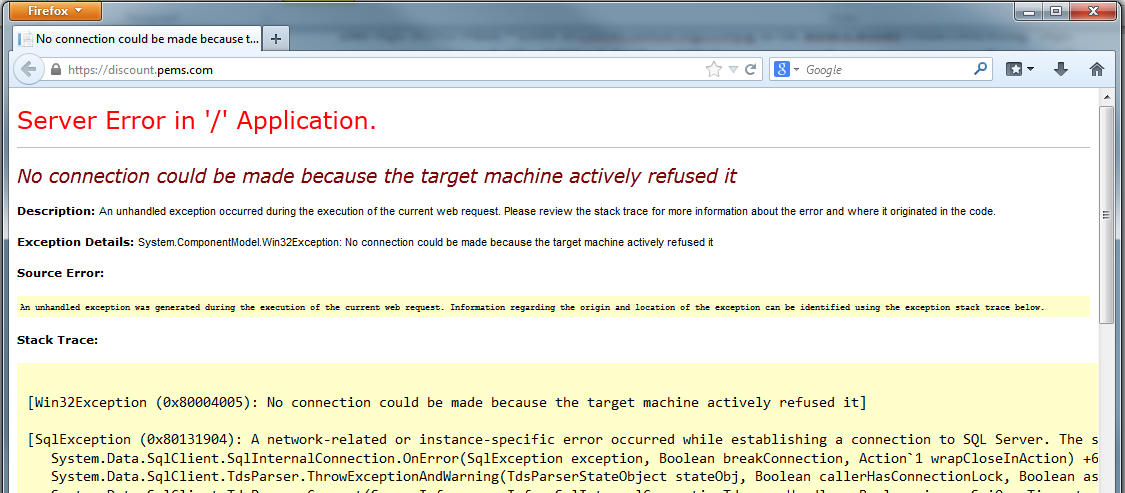


Figure 45 Configuration Error in SQL Data Source – RbacEntities

Bad SQL Server Name – Error in *connectionStrings.config* in the **RbacEntities** connection string. SQL server name mistyped. Check RBAC database SQL Server name and correct in *connectionStrings.config* and save. Retest starting at Section 12.1.

## Test Login

An additional test of the config files can be performed by logging into the Discount site with a test user and password.

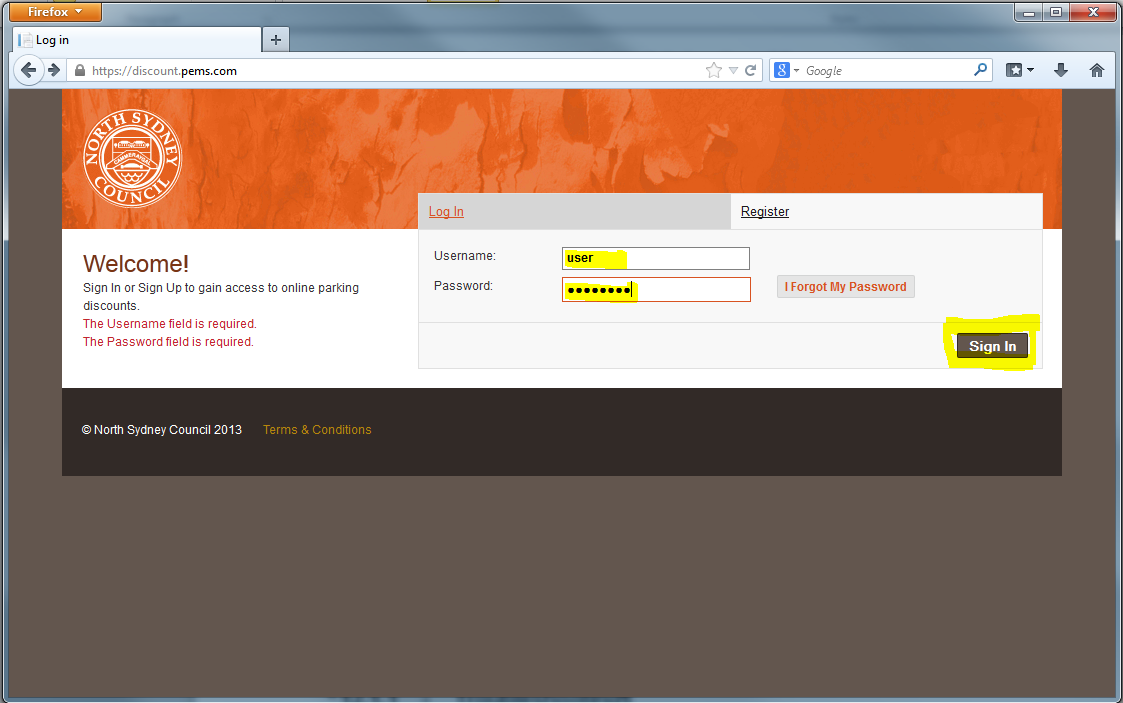


Figure 46 Discount Website Login Test

Enter a *Username* of ‘user’ with a *Password* of ‘password’. This is not a valid user but it will test the **MembershipConnection** connection string. Click “Sign In”. You should see the following displayed on the left side of the page. This verifies the connection is valid.

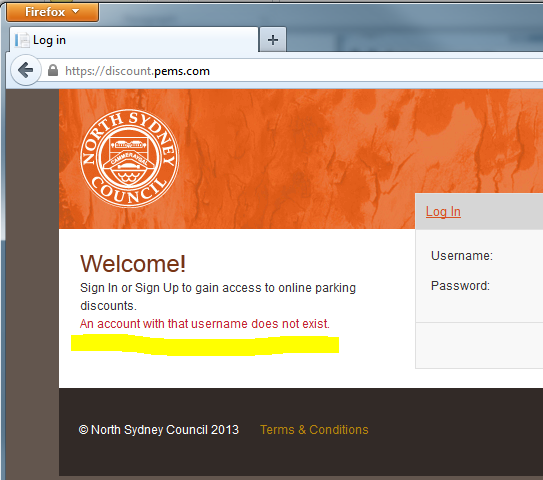


Figure 47 Discount Website Login Test Result

### Troubleshooting

The primary reason for a failure of the login test is an error in the **Membership.Connector** connection string. The following is the typical failure you will encounter if the connection string has mistakes or the system cannot contact the PEMS database instance.

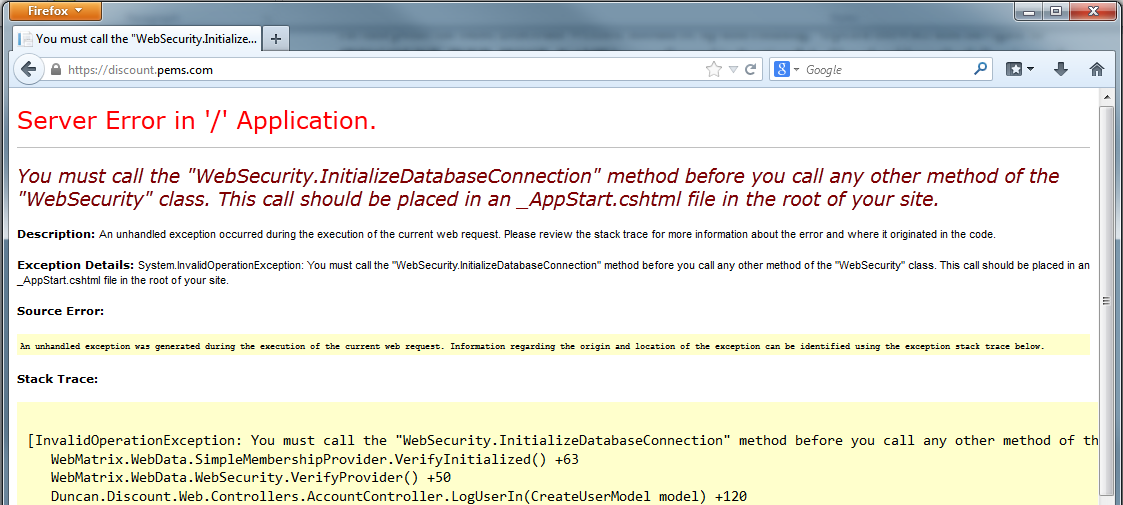


Figure 48 Configuration Error in Membership.Connector Connection String

Error in *connectionStrings.config* in the **Membership.Connector** connection string. Check SQL server name, user, password and initial catalog entries. Retest starting at Section 12.2.

# Conclusion

PEMS and Discount websites are ready for use at this point. The various connections to the databases have been tested as have been the initial database data inserts.

Failures at this point will be esoteric and will require assistance of the application development staff, database administrators or other it specialists.