Chemotherapy Order Management System (COMS)

Installation Guide

Version 2.2

September 2015

Department of Veterans Affairs

Revision History

Date	Revision	Description	Author
9/4/2015	2.2	Finalized Prototype Version	Sean Cassidy
7/8/2015	2.1	Updated Content Post Pilot Testing	Sean Cassidy
2/24/2015	2.0	Initial Prototype Version	Sean Cassidy
9/17/2012	1.2	Proof of Concept Acceptance Version	Sean Cassidy
9/15/2012	1.1	Revised Edition	Sean Cassidy
4/7/2012	1.0	Initial Version	Sean Cassidy

Table of Contents

List of Figures

1. Background and Introduction

The Veterans Health Administration (VHA) has one of the largest cancer populations in the country; it is also the fastest growing group of VHA patients. VHA provides oncology services at more than 100 different locations by integrated oncology care teams consisting of, but not limited to, a physician/provider, nurse, and pharmacist. Teams typically provide care on an outpatient basis, although some patients may require hospitalization and inpatient services.

A uniquely high-risk and high-complexity domain of health care, Oncology services support has not been effectively implemented within the existing VHA Electronic Health Record primarily due to the lack of functionality required for the specialty. VHA's oncology processes are a mix of paper-based and computer-based practices, presenting potential error, adverse events, and inefficiencies. This creates a clinical environment with minimal standardization and limited direct order entry of chemotherapy. For these reasons, the VHA Office of Health Information (OHI) Patient Safety Workgroup rated this issue as having a high level of patient safety risk. Accordingly, an initiative within VHA's Innovations Program sought to enhance the clinical environment and safety for oncology patients through development of the Chemotherapy Order Management System (COMS) application as part of VHA's Strategic Incubation.

The COMS application enhances the clinical environment and safety for oncology patients through development and implementation of an automated ordering and management process available within VHA's clinical practice setting. In either an outpatient or inpatient setting, the COMS application supports the unique needs of oncology healthcare teams with standardized capabilities to meet direct order entry, clinical documentation, and assessing the administration of chemotherapy. COMS provides interoperability with VHA's electronic health record, interfacing and interacting with existing applicable systems, modules, capabilities, and processes within Computerized Patient Record System (CPRS) graphical user interface and Veterans Health Information Systems and Technology Architecture (VistA) databases.

The web-based COMS application consists of an interface via Hypertext Precursor (PHP), Java Script, Node.js, Simple Object Access Protocol (SOAP), and Representational state transfer (REST) web services. The application provides five clinical modules – Chemotherapy Template Order Source, Order Entry Management, Treatment Documentation, Flow Sheet, and End of Treatment Summary – and miscellaneous functionality that collectively serve to deliver robust functionality to support users in executing their roles and responsibilities in various oncology care processes. Within current legacy system functionality, the COMS application provides provider order entry and promotes patient safety via read/write interoperability with the electronic health record. The application uniquely offers exportability of chemotherapy templates for national vetting and proliferation to facilitate VHA-wide standardization of chemotherapy regimens for oncology services and patients across the VA enterprise.

2. Orientation

This COMS Installation Guide is for use in conjunction with VHA's COMS application. It outlines steps for installation of the COMS application to enable use at VHA facilities. The

intended audience of this manual is Office of Information and Technology (OI&T) staff responsible for the proper installation of VHA clinical applications.
This guide provides an overall explanation of COMS from a deployment, installation, and production operations perspective with the assumption that the reader is familiar with the following:

- CPRS/VistA computing environment
- Internet and database server environments
- Microsoft operating environment

This guide does not detail user operation or technical aspects of the COMS application. Such topics are available in the COMS User Manual and COMS Technical Manual, respectively.

3. Deployment Overview

COMS is a thin client, web-based application that requires an application server with 4 GB of Random Access Memory (RAM) and hard drive space as follows:

- COMS Application: 200mb
- COMS Database (if local): 500mb free space after Structured Query Language (SQL) Server 2008/2012 installation
- COMS Database logs (if local), managed by an administrator, at least 300mb+

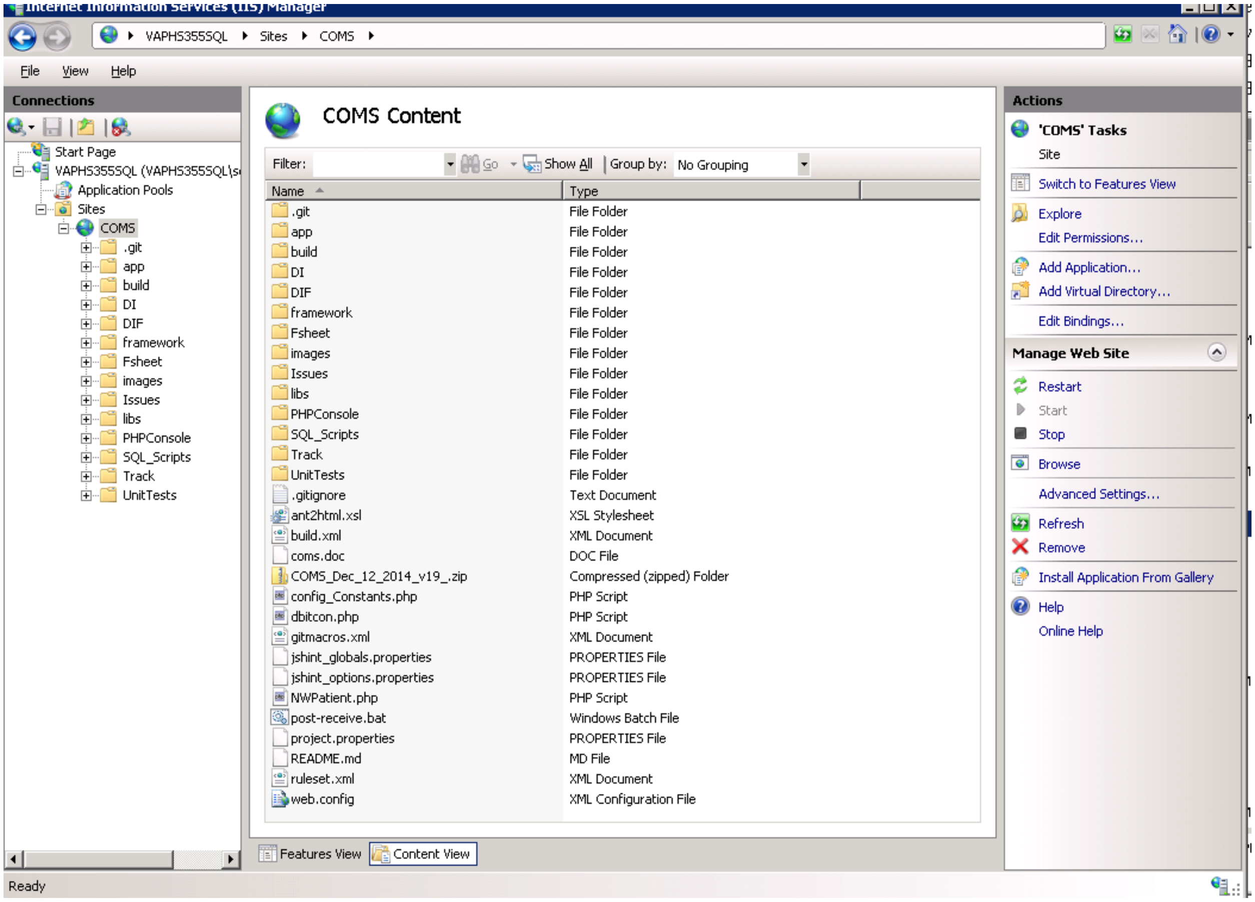
These resources may be provided as physical servers or through a virtual machine environment. In either scenario, the recommended resources above should be dedicated to COMS for optimal performance.

4. Prior to Installation

Although COMS is supported by only a limited number of system requirements, these must be verified and satisfied prior to installation to ensure full functionality of the application and its interoperability with the associated VistA instance.

4.1. Requirements

- Windows Server 2003 or greater
- PHP 5.3.28 or greater
- SQL Server 2008 or 2012 (optional for non-Broker Security Enhancement (BSE) installations)
- Internet Information Services (IIS) 7.0 or greater with the following specifications:
    - IIS should be relatively close to default settings
    - COMS may either be installed as a new web site or as a virtual directory of an existing web site as in **Figure 1** on the next page
    - COMS requires a COMS-dedicated application pool for the site using .NET Framework 2.0 at a minimum
    - Uniform Resource Locator (URL) Rewrite must be installed
    - Fast Common Gateway Interface (Fast-CGI) must be installed
    - PHP Manager
    - SQL PHP Driver must be installed.



Name	Type
.git	File Folder
app	File Folder
build	File Folder
DI	File Folder
DIF	File Folder
framework	File Folder
Fsheet	File Folder
images	File Folder
Issues	File Folder
libs	File Folder
PHPConsole	File Folder
SQL_Scripts	File Folder
Track	File Folder
UnitTests	File Folder
.gitignore	Text Document
ant2html.xsl	XSL Stylesheet
build.xml	XML Document
coms.doc	DOC File
COMS_Dec_12_2014_v19_.zip	Compressed (zipped) Folder
config_Constants.php	PHP Script
dbitcon.php	PHP Script
gitmacros.xml	XML Document
jshint_globals.properties	PROPERTIES File
jshint_options.properties	PROPERTIES File
NWPatient.php	PHP Script
post-receive.bat	Windows Batch File
project.properties	PROPERTIES File
README.md	MD File
ruleset.xml	XML Document
web.config	XML Configuration File

Figure 1: COMS Root Directory of IIS

4.2. Configuring the COMS Database on SQL Server

The following steps are required to configure the COMS database on the SQL Server:

1. Identify your SQL server and execute the following scripts
   a. 01_COMS_Database_Create.sql – Ensure to set the database name throughout the script
   b. 02_COMS_Database_Objects.sql
2. The SQL scripts are within the COMS code under SQL_Scripts
3. Create a local SQL account, coms_db_user
4. Grant Read Data and Write Data privileges to the coms_db_user for the new database
5. Ensure the account information is entered properly in the COMS\app\Config_Constants.php file

Enter VistA Instance location
Open SQL Server and execute the following insert statement on the database created above:

```
INSERT INTO [COMS_Database_Name].[dbo].[Globals]
           ([domain]
           ,[sitelist]
           ,[vista]
           ,[port])
```

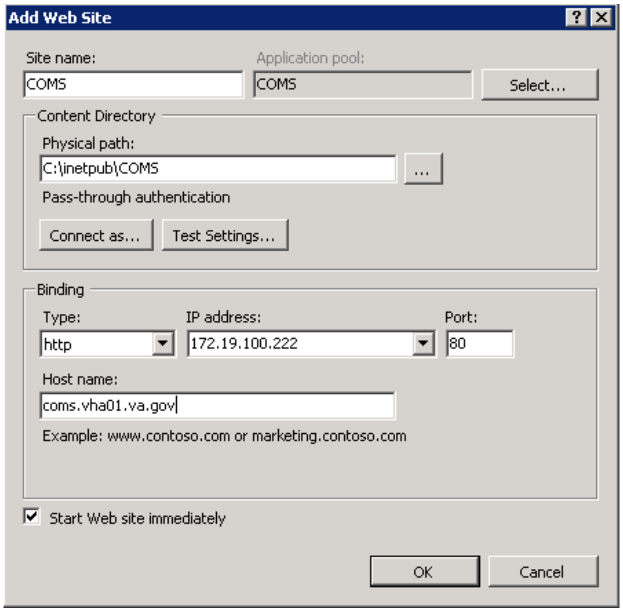
```
      VALUES
            ('URL Domain of COMS',
            ,'Local Site Number',
            ,'IP Address of VistA Instance',
            ,'Port Number of VistA Instance')
GO
```

4.3. Configuring IIS

4.3.1. Create COMS Site

Perform the following steps to create the COMS site while configuring the IIS:

1. Create a new web site in IIS and set the name to COMS.
2. Then set the path to your chosen web directory. The example in **Figure 2** uses the default setting of C:\inetpub\COMS.
3. Bind COMS to the Internet Protocol (IP) Address of the web server and set the host name to the web address of the COMS application. The example in Figure 2 uses the host name of coms.vha01.va.gov. This name must be set as an A Record in your Domain Name System (DNS) server.



Add Web Site

Site name: COMS

Application pool: COMS

Content Directory

Physical path: C:\inetpub\COMS

Pass-through authentication

Binding

Type: http

IP address: 172.19.100.222

Port: 80

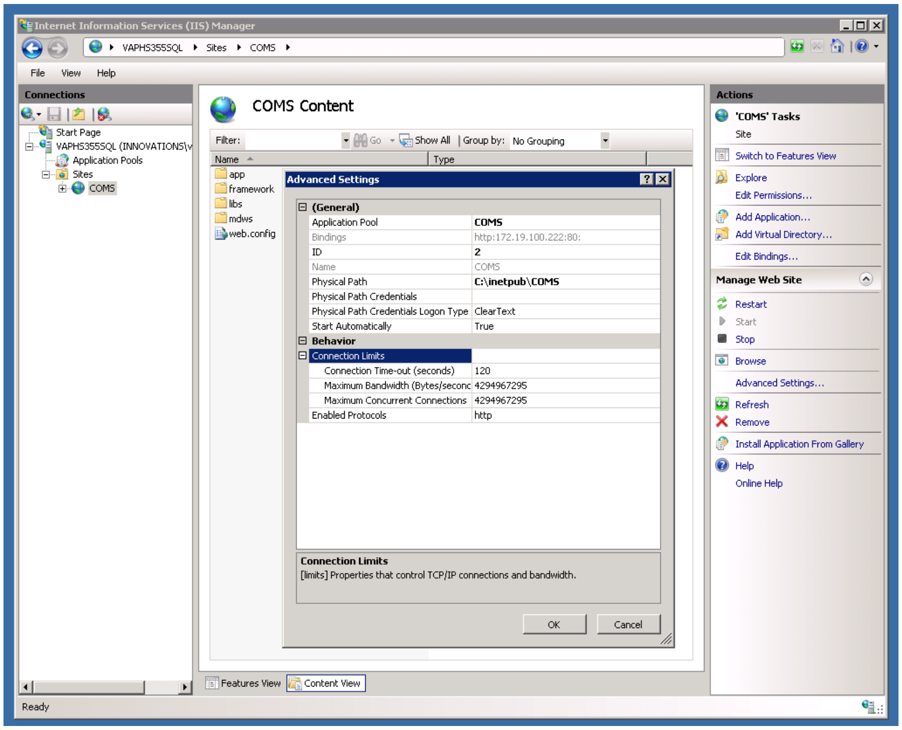
Host name: coms.vha01.va.gov

Example: www.contoso.com or marketing.contoso.com

☑ Start Web site immediately

Figure 2: Adding COMS Site to IIS

4. Verify your settings by right clicking on the site name in IIS and selecting Advanced Settings, as shown in **Figure 3** on the next page.



Advanced Settings

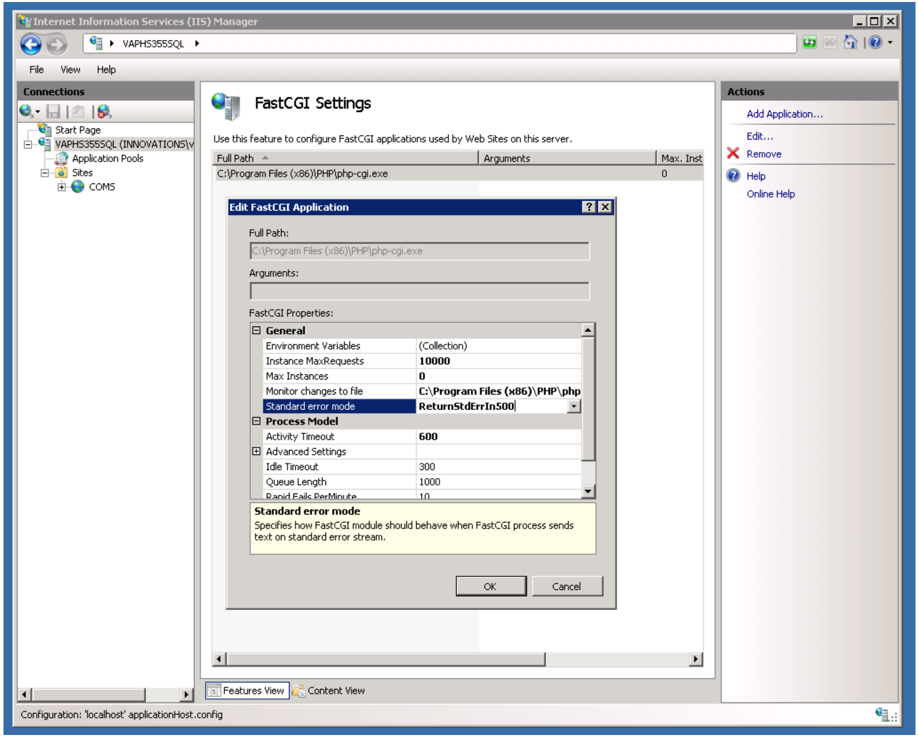
(General)	
Application Pool	COMS
Bindings	http:172.19.100.222:80:
ID	2
Name	COMS
Physical Path	C:\inetpub\COMS
Physical Path Credentials	
Physical Path Credentials Logon Type	ClearText
Start Automatically	True
Behavior	
Connection Limits	
Connection Time-out (seconds)	120
Maximum Bandwidth (Bytes/seconc	4294967295
Maximum Concurrent Connections	4294967295
Enabled Protocols	http

Connection Limits
[limits] Properties that control TCP/IP connections and bandwidth.

Figure 3: COMS Content Advanced Settings for IIS

4.3.2. Configuring Fast-CGI Settings for IIS

To configure Fast-CGI settings for IIS, ensure the full path points to the php-cgi.exe file in your PHP directory. In **Figure 4**, an example is shown for C:\Program Files (x86)\PHP.



Use this feature to configure FastCGI applications used by Web Sites on this server.

Full Path	Arguments	Max. Inst
C:\Program Files (x86)\PHP\php-cgi.exe		0

Edit FastCGI Application

Full Path:
C:\Program Files (x86)\PHP\php-cgi.exe

Arguments:

FastCGI Properties:

General	
Environment Variables	(Collection)
Instance MaxRequests	10000
Max Instances	0
Monitor changes to file	C:\Program Files (x86)\PHP\php
Standard error mode	ReturnStdErrIn500
Process Model	
Activity Timeout	600
Advanced Settings	
Idle Timeout	300
Queue Length	1000
Rapid Fails PerMinute	10

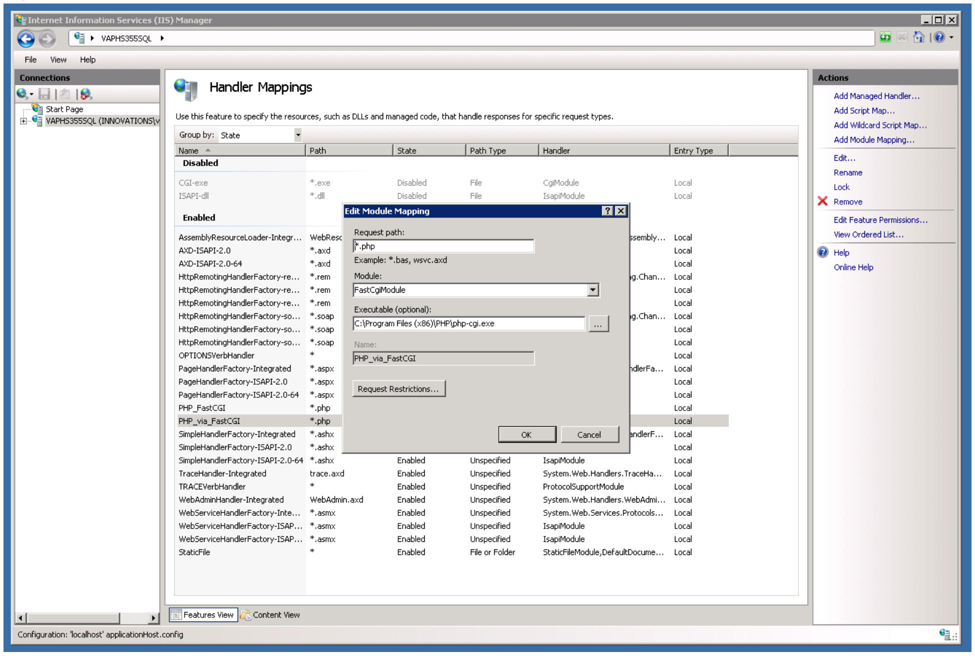
Standard error mode
Specifies how FastCGI module should behave when FastCGI process sends text on standard error stream.

Configuration: 'localhost' applicationHost.config

Figure 4: Fast-CGI Settings for IIS

4.3.3. Edit Handler Mappings for PHP

Verify the IIS Handler Mappings are set properly in the PHP directory for the php-cgi.exe file, as shown in **Figure 5**.



Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Name	Path	State	Path Type	Handler	Entry Type
Disabled					
CGI-exe	*.exe	Disabled	File	CgiModule	Local
ISAPI-dll	*.dll	Disabled	File	IsapiModule	Local
Enabled					
AssemblyResourceLoader-Integr...	WebResc			ssembly...	Local
AXD-ISAPI-2.0	*.axd				Local
AXD-ISAPI-2.0-64	*.axd				Local
HttpRemotingHandlerFactory-re...	*.rem			g.Chan...	Local
HttpRemotingHandlerFactory-re...	*.rem				Local
HttpRemotingHandlerFactory-re...	*.rem				Local
HttpRemotingHandlerFactory-so...	*.soap			g.Chan...	Local
HttpRemotingHandlerFactory-so...	*.soap				Local
HttpRemotingHandlerFactory-so...	*.soap				Local
OPTIONSVerbHandler	*				Local
PageHandlerFactory-Integrated	*.aspx			ndlerFa...	Local
PageHandlerFactory-ISAPI-2.0	*.aspx				Local
PageHandlerFactory-ISAPI-2.0-64	*.aspx				Local
PHP_FastCGI	*.php				Local
PHP_via_FastCGI	*.php				Local
SimpleHandlerFactory-Integrated	*.ashx			andlerF...	Local
SimpleHandlerFactory-ISAPI-2.0	*.ashx				Local
SimpleHandlerFactory-ISAPI-2.0-64	*.ashx	Enabled	Unspecified	IsapiModule	Local
TraceHandler-Integrated	trace.axd	Enabled	Unspecified	System.Web.Handlers.TraceHa...	Local
TRACEVerbHandler	*	Enabled	Unspecified	ProtocolSupportModule	Local
WebAdminHandler-Integrated	WebAdmin.axd	Enabled	Unspecified	System.Web.Handlers.WebAdmi...	Local
WebServiceHandlerFactory-Inte...	*.asmx	Enabled	Unspecified	System.Web.Services.Protocols...	Local
WebServiceHandlerFactory-ISAP...	*.asmx	Enabled	Unspecified	IsapiModule	Local
WebServiceHandlerFactory-ISAP...	*.asmx	Enabled	Unspecified	IsapiModule	Local
StaticFile	*	Enabled	File or Folder	StaticFileModule,DefaultDocume...	Local

Edit Module Mapping

Request path: *.php

Example: *.bas, wsvc.axd

Module: FastCgiModule

Executable (optional): C:\Program Files (x86)\PHP\php-cgi.exe

Name: PHP_via_FastCGI

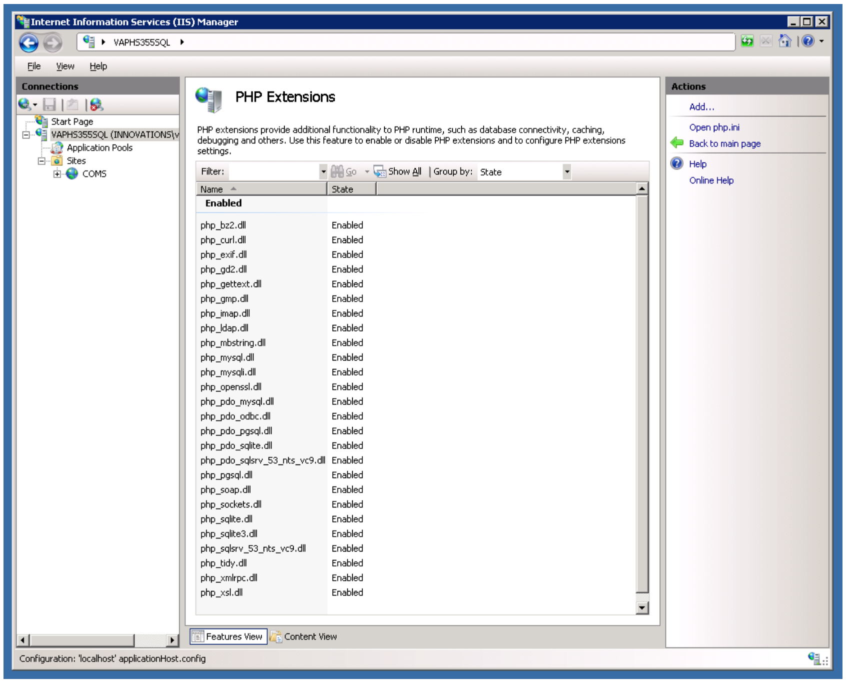
Configuration: 'localhost' applicationHost.config

Figure 5: IIS Handler Mappings for IIS

4.3.4. Enabled PHP File Extensions

The following dynamic link library (dll) files are the Enabled PHP File Extensions on the server, as shown in **Figure 6** on the next page:

php_bz2	php_curl	php.exif
php_qd2	php_gettext	php_gmp
php_imap	php_Idap	php_mbstring
php_mysql	php_mysqli	php_openssl
php_pdo_mysql	php_pdo_odbc	php_pdo_pgsql
php_pdo_sqlite	php_pdo_sqlsrv_53_nts_vc9	php_sgsql
php_soap	php_sockets	php_sqlite
php_sqlite3	php_ sqlsrv_53_nts_vc9	php_tidy
php_xmlrpc	php_xsl	



PHP Extensions

PHP extensions provide additional functionality to PHP runtime, such as database connectivity, caching, debugging and others. Use this feature to enable or disable PHP extensions and to configure PHP extensions settings.

Group by: State

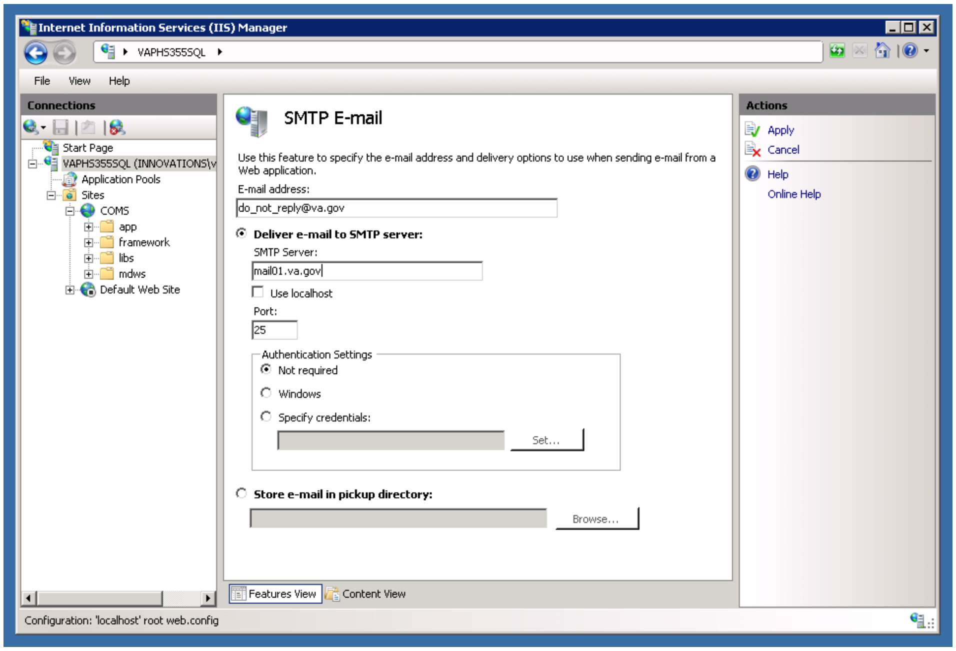
Name	State
Enabled	
php_bz2.dll	Enabled
php_curl.dll	Enabled
php_exif.dll	Enabled
php_gd2.dll	Enabled
php_gettext.dll	Enabled
php_gmp.dll	Enabled
php_imap.dll	Enabled
php_ldap.dll	Enabled
php_mbstring.dll	Enabled
php_mysql.dll	Enabled
php_mysqli.dll	Enabled
php_openssl.dll	Enabled
php_pdo_mysql.dll	Enabled
php_pdo_odbc.dll	Enabled
php_pdo_pgsql.dll	Enabled
php_pdo_sqlite.dll	Enabled
php_pdo_sqlsrv_53_nts_vc9.dll	Enabled
php_pgsql.dll	Enabled
php_soap.dll	Enabled
php_sockets.dll	Enabled
php_sqlite.dll	Enabled
php_sqlite3.dll	Enabled
php_sqlsrv_53_nts_vc9.dll	Enabled
php_tidy.dll	Enabled
php_xmlrpc.dll	Enabled
php_xsl.dll	Enabled

Configuration: 'localhost' applicationHost.config

Figure 6: PHP File Extensions for IIS

4.3.5. Configure SMTP Settings

Set the Simple Mail Transfer Protocol (SMTP) server settings of your local SMTP server. In **Figure 7**, the example is SMTP Server: mail01.va.gov.



SMTP E-mail

Use this feature to specify the e-mail address and delivery options to use when sending e-mail from a Web application.

E-mail address: do_not_reply@va.gov

● **Deliver e-mail to SMTP server:**

SMTP Server: mail01.va.gov

□ Use localhost

Port: 25

Authentication Settings

● Not required

○ Windows

○ Specify credentials:

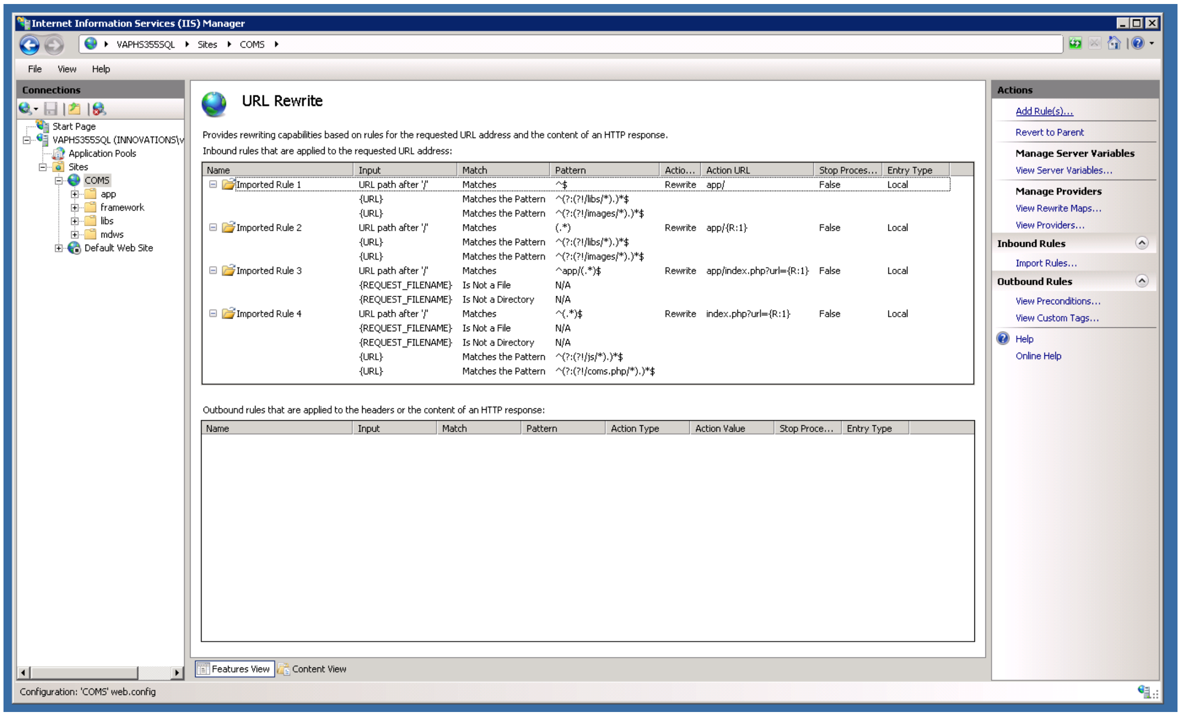
○ **Store e-mail in pickup directory:**

Configuration: 'localhost' root web.config

Figure 7: SMTP Settings of IIS

4.3.6. Configure URL Rewrite

The URL Rewrite module should contain the settings shown in **Figure 8** after extracting the COMS application to this web site.



Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

Name	Input	Match	Pattern	Actio...	Action URL	Stop Proces...	Entry Type
Imported Rule 1	URL path after '/'	Matches	^$	Rewrite	app/	False	Local
	{URL}	Matches the Pattern	^(?:(?!/libs/*).)*$				
	{URL}	Matches the Pattern	^(?:(?!/images/*).)*$				
Imported Rule 2	URL path after '/'	Matches	(.*)	Rewrite	app/{R:1}	False	Local
	{URL}	Matches the Pattern	^(?:(?!/libs/*).)*$				
	{URL}	Matches the Pattern	^(?:(?!/images/*).)*$				
Imported Rule 3	URL path after '/'	Matches	^app/(.*)$	Rewrite	app/index.php?url={R:1}	False	Local
	{REQUEST_FILENAME}	Is Not a File	N/A				
	{REQUEST_FILENAME}	Is Not a Directory	N/A				
Imported Rule 4	URL path after '/'	Matches	^(.*)$	Rewrite	index.php?url={R:1}	False	Local
	{REQUEST_FILENAME}	Is Not a File	N/A				
	{REQUEST_FILENAME}	Is Not a Directory	N/A				
	{URL}	Matches the Pattern	^(?:(?!/js/*).)*$				
	{URL}	Matches the Pattern	^(?:(?!/coms.php/*).)*$				

Outbound rules that are applied to the headers or the content of an HTTP response:

Name	Input	Match	Pattern	Action Type	Action Value	Stop Proce...	Entry Type

Configuration: 'COMS' web.config

Figure 8: URL Rewrite Module

4.4. Configuring PHP

Visit http://php.iis.net/ and follow directions for downloading and installing the latest PHP version or manually identify and download the latest PHP version from http://www.php.net for windows platform and follow the steps for extraction and installation.

After installation of PHP, modify the php.ini at C:\Program Files\PHP\php.ini to the following settings:

- extension=php_pdo_sqlsrv_53_nts_vc9.dll
- extension=php_sqlsrv_53_nts_vc9.dll
- date.timezone=America/New_York – or your time zone if not EST
- cgi.force_redirect = 0
- fastcgi.impersonate = 1
- cgi.fix_pathinfo=1

Verify PHP is installed properly by opening IIS PHP Manager and selecting the link checkinfo(). Confirm sqlsrv as a module in the output.

4.5. Loading the SQL PHP Driver for IIS

To ensure functionality of the COMS application, the SQL PHP driver must be properly loaded and configured.

4.5.1. Loading the Driver (SQLSRV)

Users may download the SQLSRV driver from the Microsoft Download Center or by using the Web Platform Installer. Included in the download are several .dll files, each with a name that indicates procedural or PDO extension (i.e. sqlsrv or pdo), compatibility with PHP 5.3 or PHP 5.2 (i.e. 53 or 52), thread-safe or non-thread-safe (i.e. ts or nts), and the compiler used to compile the extension (i.e. VC6 or VC9). For example, the **php_sqlsrv_53_nts_vc9.dll** file is the procedural extension; it is compatible with PHP 5.3, non-thread-safe, and compiled with Visual C++ 9 (vc9) compiler. Note that the recommended method to run PHP with Internet Information Services is to use the Fast-CGI module and a non-thread-safe version of PHP (and therefore a non-thread-safe version of the SQLSRV driver). Selection of a vc6 or vc9 version of the driver will depend on the compiler used to compile your version of PHP. For more information about which .dll file you should use, see System Requirements.

Loading the SQLSRV driver is similar to loading any PHP extension:

1. Place the driver file in your PHP extension directory.
2. Modify the php.ini file to include the driver. For example:

   extension=php_sqlsrv_53_nts_vc9.dll

   See **Figure 9** (SQLSRV) below for more detail.
3. Restart the Web server.

For more information, see Loading the Driver in the product documentation.

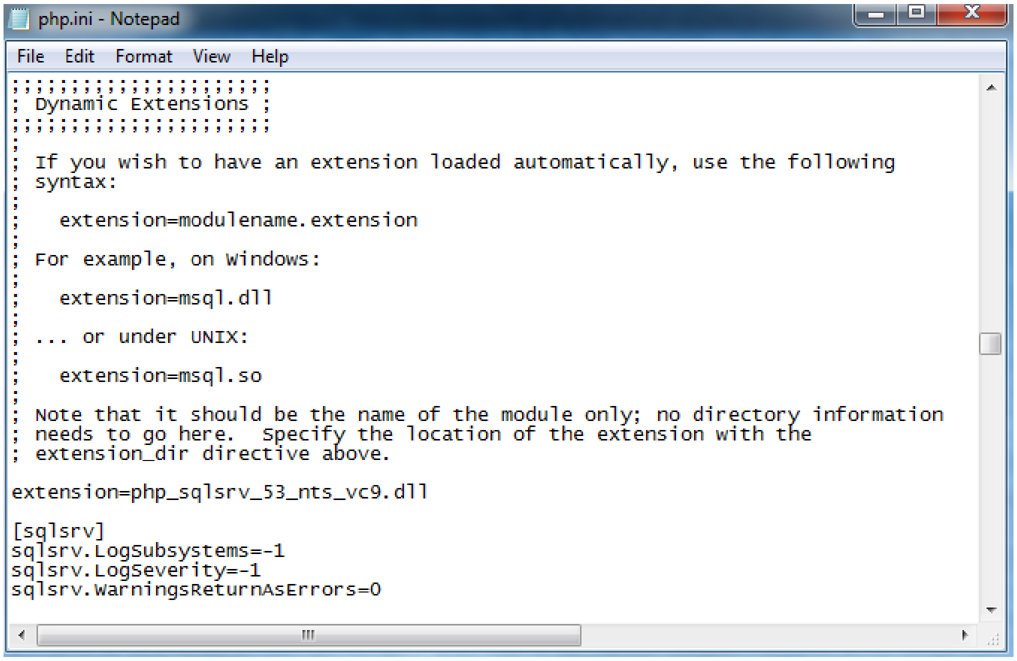
4.5.2. Configuring the Driver (SQLSRV)

The SQLSRV driver has three configuration options as follows:

- LogSubsystems
- Use this option to turn the logging of subsystems on or off. The default setting is SQLSRV_LOG_SYSTEM_OFF (i.e. logging turned off).
- LogSeverity
- Use this option to specify severities to log after logging is turned on. With the default setting SQLSRV_LOG_SEVERITY_ERROR, only errors are logged when logging is turned on.
- WarningsReturnAsErrors
- By default, the SQLSRV driver handles warnings generated by sqlsrv functions as errors. Use the WarningsReturnAsErrors option to change this behavior. The default setting for this option is true (1) with warnings handled as errors. Note: There are exceptions to this rule. For example, the warning generated by changing the database context is never categorized and handled as an error.

For more information about these options and settings, see Configuring the Driver in the product documentation.

Configuration options can be set in the php.ini file or in a PHP script with the sqlsrv_configure function. **Figure 9** shows the Dynamic Extensions section of the php.ini file modified to load the driver, log activity on all subsystems, log all activity (errors, warnings, and notices), and turn off the **WarningsReturnAsErrors** behavior.



```
;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;
;
; If you wish to have an extension loaded automatically, use the following
; syntax:
;
;   extension=modulename.extension
;
; For example, on Windows:
;
;   extension=msql.dll
;
; ... or under UNIX:
;
;   extension=msql.so
;
; Note that it should be the name of the module only; no directory information
; needs to go here.  Specify the location of the extension with the
; extension_dir directive above.

extension=php_sqlsrv_53_nts_vc9.dll

[sqlsrv]
sqlsrv.LogSubsystems=-1
sqlsrv.LogSeverity=-1
sqlsrv.WarningsReturnAsErrors=0
```

Figure 9: Dynamic Extensions Section of the php.ini. for SQLSRV

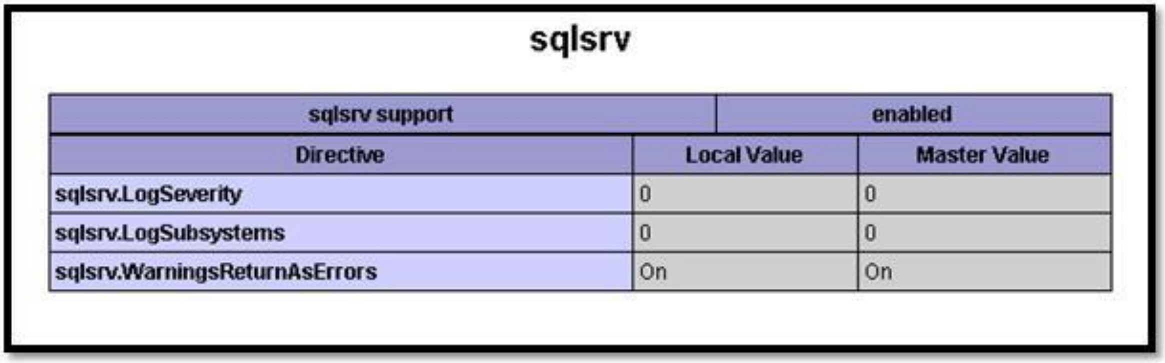
For more information about how to change the default settings, see Logging Activity and How to: Configure Error and Warning Handling Using the SQLSRV Extension in the product documentation.

To ensure the driver is loaded and to verify configuration settings, run a script that calls the **phpinfo()** function. To do this, process the following steps:

1. Open a text file and copy the following code into the text file:

   ```
   <?php phpinfo(); ?>
   ```

2. Save the file as info.php in your Web server's root directory
3. Open an internet browser and go to http://localhost/info.php
4. Scroll down the resulting page to find the **sqlsrv** section of the phpinfo() page, as shown in **Figure 10**. This output confirms the driver is loaded and the configuration settings are set to default values.



sqlsrv

sqlsrv support	enabled	
Directive	**Local Value**	**Master Value**
sqlsrv.LogSeverity	0	0
sqlsrv.LogSubsystems	0	0
sqlsrv.WarningsReturnAsErrors	On	On

Figure 10: SQLSRV Section of the phpinfo() Page

4.6. Node.JS Installation

To ensure functionality of the COMS application, Node.js must be properly loaded.

1. Install Node.js for windows from www.nodejs.com
2. Go to C:\inetpub\COMS\NodeJS\coms-api in a command prompt window with Administrative privileges and type in **npm install**
3. Then type **application\app.js**

4.7. Web References

The following web references support information provided in this COMS Installation Guide:

1. http://social.technet.microsoft.com/wiki/contents/articles/accessing-sql-server-databases-from-php.aspx
2. http://www.php.net
3. www.nodejs.com

5. Installation Instructions

After preparing the local environment for receipt of COMS, installation of the application is a straightforward process.

5.1. Obtaining the Software

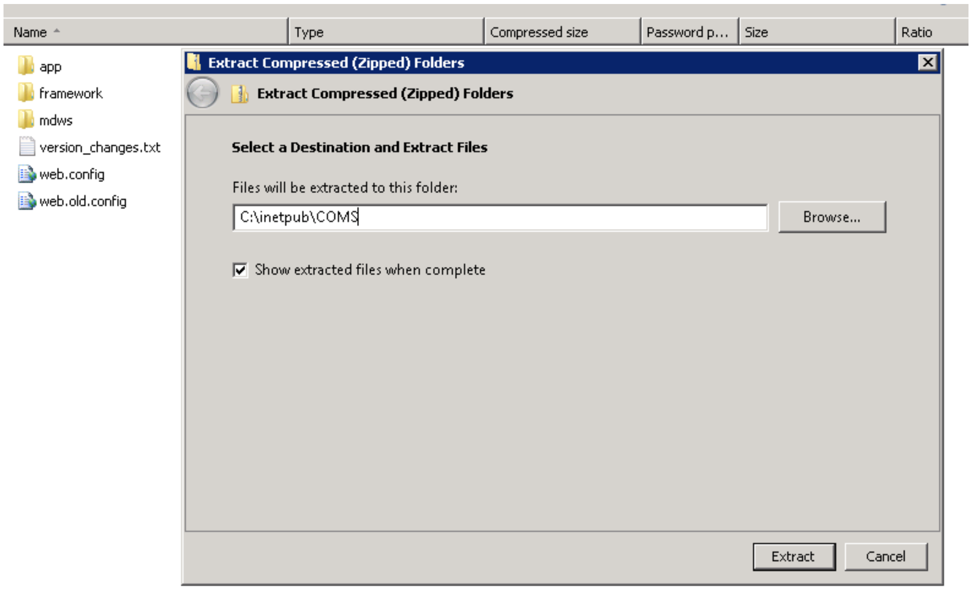
Use your preferred File Transfer Protocol (FTP) client to download the COMS.zip installation file from https:/coms.vacloud.us/download

Username: **anonymous**
Password: no password

Save the installation file to the server with the requirements specified on page 2 of this Installation Guide.

5.2. Extracting COMS Web Site Files

Copy the COMS.zip file to the server and extract all the contents to the COMS web site created for IIS and depicted in Figure 2 of this Installation Guide. **Figure 11** shows the example C:\inetpub\COMS.



Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\inetpub\COMS

☑ Show extracted files when complete

Figure 11: SMTP Settings of IIS

5.3. Installing COMS

To begin the COMS installation process, double-click the file and extract the files to the directory created for the COMS web site on the web server. If the COMS application is the default site for IIS, the web.config file should look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
   <httpErrors errorMode="Detailed" />
    <rewrite>
      <rules>
        <rule name="Imported Rule 1">
          <match url="^$" ignoreCase="false" />
          <conditions logicalGrouping="MatchAll">
            <add input="{URL}" pattern="^(?:(?!/libs/*).)*$" ignoreCase="false" />
            <add input="{URL}" pattern="^(?:(?!/images/*).)*$" ignoreCase="false" />
          </conditions>
          <action type="Rewrite" url="app/" />
        </rule>
        <rule name="Imported Rule 2">
          <match url="(.*)" ignoreCase="false" />
          <conditions logicalGrouping="MatchAll">
            <add input="{URL}" pattern="^(?:(?!/libs/*).)*$" ignoreCase="false" />
            <add input="{URL}" pattern="^(?:(?!/images/*).)*$" ignoreCase="false" />
          </conditions>
          <action type="Rewrite" url="app/{R:1}" />
        </rule>
        <rule name="Imported Rule 3" enabled="true">
          <match url="^app/(.*)$" ignoreCase="false" />
          <conditions logicalGrouping="MatchAll">
            <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
            <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
          </conditions>
          <action type="Rewrite" url="app/index.php?url={R:1}" appendQueryString="true" />
        </rule>
        <rule name="Imported Rule 4">
          <match url="^(.*)$" ignoreCase="false" />
          <conditions logicalGrouping="MatchAll">
            <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
            <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
            <add input="{URL}" pattern="^(?:(?!/js/*).)*$" ignoreCase="false" />
            <add input="{URL}" pattern="^(?:(?!/coms.php/*).)*$" ignoreCase="false" />
          </conditions>
          <action type="Rewrite" url="index.php?url={R:1}" appendQueryString="false" />
        </rule>
      </rules>
    </rewrite>
    <handlers>
      <remove name="PHP_via_FastCGI" />
      <add name="PHP_via_FastCGI" path="*.php" verb="*" modules="FastCgiModule"
scriptProcessor="C:\Program Files (x86)\PHP\php-cgi.exe" resourceType="Either"
requireAccess="Script" />
```

```
    </handlers>
      </system.webServer>

    </configuration>
```

Ensure the *scriptProcessor* path of the *PHP_via_Fast-CGI* handler is set to the correct path of your PHP installation of the *php-cgi.exe* file. **Note**: If this path is changed due to a new installation of PHP, the path must be updated in the *web.config* file and in *IIS Handler Mappings*.

After the files are extracted and installed on web server and the SQL Server Database is created, change the configuration file of the application to point to the Database Server, Database and the login required for the COMS application. An example of the config.php file located at *C:\inetpub\COMS\app/config_constants.php* is located below.

```
<?php

/** Configuration Variables **/

define(‘COMS’, true);

if (defined(‘COMS’)) {
	define('DB_NAME', 'COMS');
	define('DB_TYPE', 'sqlsrv');
	define('DB_HOST', "VA\SQLServer");
	define('DB_USER', 'va_coms_db_user');
	define('DB_PASSWORD', 'COMSpassword1');
}
```

Note: The config.php file does not end with ?> as typical due to the framework code that is used to support the COMS application.

6. Troubleshooting and Uninstalling COMS

In general, to troubleshoot any problem, check the following sources:
Browse to the local web services and make sure the Web Service Definition Language (WSDL) displays. Run the connection test page.

6.1. Troubleshooting

Consider the following steps when troubleshooting the COMS application:

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

6.2. Uninstalling COMS

To uninstall the COMS application, perform the following steps:

1. Open IIS and delete the web site.
2. Open Windows Explore and delete the files installed at the physical location on the server.
3. Open the SQL Server and delete the database and its logs.