



**DEPARTMENT OF VETERANS AFFAIRS  
VHA Innovation Program**

**Traumatic Brain Injury (TBI) Clinical Decision Support (CDS)  
Implementation  
(VA118-14-C-0015)**

**Comprehensive Server Manual**

**Deliverable CLIN 0002 AR**

**Installation Guide**

Intellica Corporation  
209 W. Poplar  
San Antonio, Texas 78212

05/28/2015



## Table of Contents

|  |    |
|--|----|
| Change Log .....   | 3  |
| Introduction .....   | 3  |
| Minimum System Requirements .....  | 3  |
| Web Server .....   | 3  |
| Database Server .....  | 3  |
| Web server .....   | 4  |
| Install Oracle Client .....  | 4  |
| Verify/Install Microsoft .NET Framework 4.5 .....                          | 5  |
| Create Application Pool .....  | 5  |
| Install the IQLogic component .....  | 7  |
| Install TBI Practitioner Portal .....                                      | 9  |
| Database Server .....  | 13 |
| Procedures for Backing-up Audit Table .....                                | 14 |
| CPRS Tools Menus Installation .....  | 14 |
| Connecting from TBI/CDS to MDWS is via SSL .....                           | 15 |
| Connectivity Troubleshooting .....   | 16 |
| Ping database host IP .....  | 16 |
| Ping database hostname .....   | 16 |
| Test if listener port is reachable/opened for your remote connection ..... | 16 |
| Test Oracle Net connectivity – tnsping .....                               | 17 |
| Test database connection .....   | 18 |
| Troubleshoot the client side .....   | 18 |
| Error codes for problems on the client side .....                          | 18 |
| tnsnames.ora file example .....  | 20 |
| Troubleshoot IIS7 errors .....   | 21 |
| Enable IIS7 detailed errors .....  | 21 |
| Deeper diagnostics with Failed Request Tracing .....                       | 27 |
| HTTP Detailed Errors in IIS 7.0 .....                                      | 30 |
| Client Errors .....  | 30 |
| Server Errors .....  | 30 |
| Examples of Sub-status Codes .....   | 30 |

## Change Log

| Date       | Version | Author           | Revision Description                       |
|------------|---------|------------------|--|
| 01/08/2015 | 1       | Craig Rebo       | Created                                    |
| 04/14/2015 | 2       | Craig Rebo       | Updated                                    |
| 05/27/2015 | 3       | Antonio Armelles | Updated                                    |
| 05/28/2015 | 4       | Adam Richter     | Revised                                    |
| 06/03/2015 | 5       | Craig Rebo       | Added procedure backing up the audit table |
| 06/04/2015 | 6       | Antonio Armelles | Added Minimum System Requirements          |

## Introduction

This document is the installation guide for the Traumatic Brain Injury (TBI) Clinical Decision Support (CDS) application. It is suggested that the application be installed on servers that have not been locked down to start with and that the servers then be locked down once everything is working correctly.

## Minimum System Requirements

### Web Server

**OS:** Windows Server 2008 or higher.

**Processor:** 2GHz x86 or x64 processor (better recommended)

**Memory:** 2 GB Ram.

**Hard Drive:** 8 GB available space

### Database Server

**OS:** Windows Server 2008 or higher with Oracle Database 11g Release 2.

**Processor:** 2GHz x86 or x64 processor (better recommended)

**Memory:** 4 GB Ram.

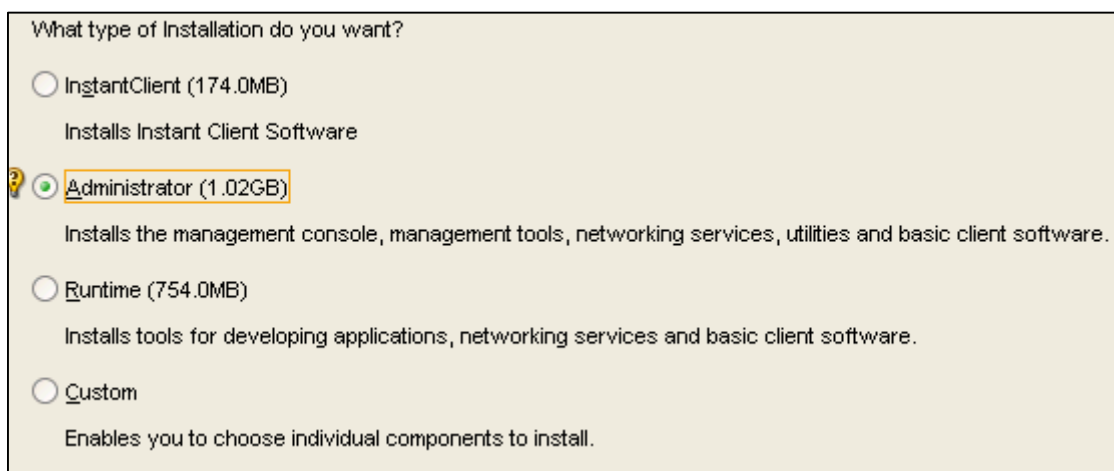
**Hard Drive:** 20 GB available space

## Web server

### Install Oracle Client

The TBI CDS application uses Oracle Client to communicate with the backend Oracle database. NOTE: You must install the 32-bit version of Oracle client or the assessment logic component will not work.

1. Install 32-bit Oracle client 11g or later using the ‘Administrator’ option and use default settings.



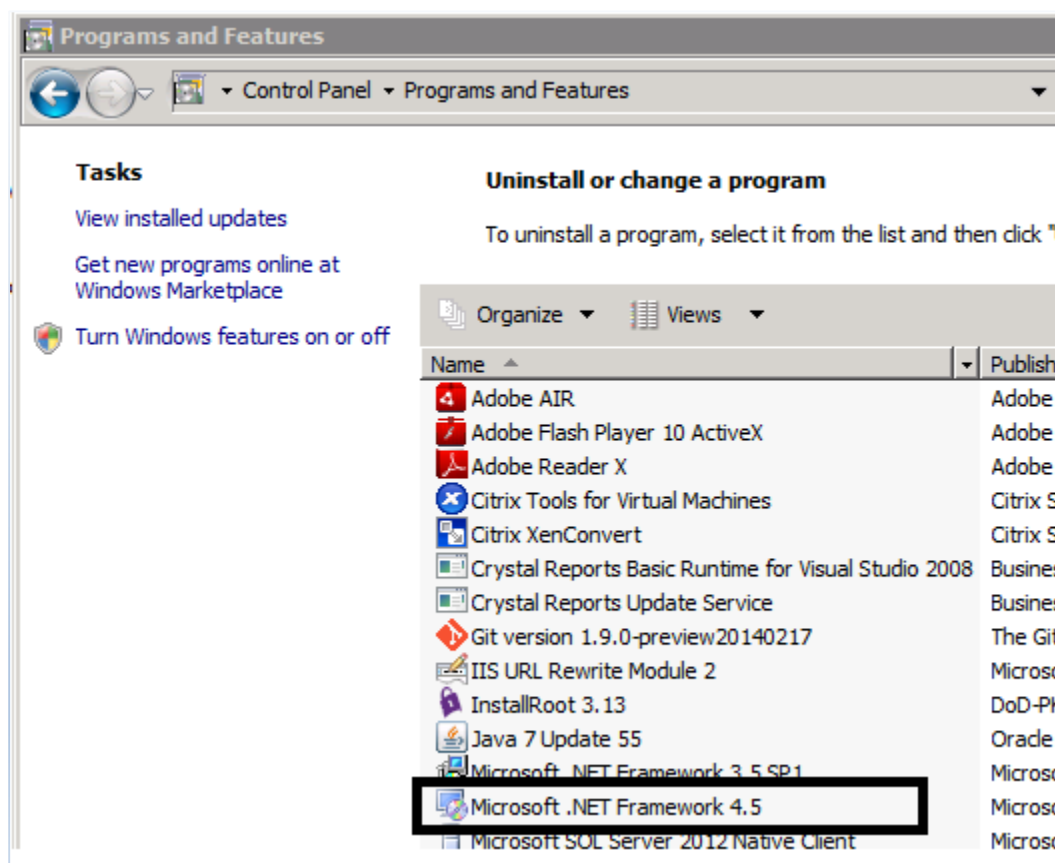
2. After the client is successfully installed, use Notepad to open the tnsnames.ora file.  
(Note: the file will be in a path similar to:  
C:\app\Administrator\product\11.1.0\client\_1\network\admin)
3. Add an entry to the tnsnames.ora file as follows, where DRDB\_TBICDS is the name of the entry, XXX.XX.X.XXX is the IP address of the database server and DRDB is the service name. Work with the DBA and follow local conventions when adding this entry.

```
DRDB_TBICDS =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = XXX.XX.X.XXX)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = DRDB)  
  )  
)
```

## Verify/Install Microsoft .NET Framework 4.5

TBI CDS requires Microsoft .NET Framework 4.5

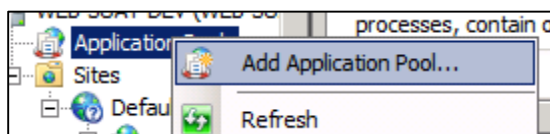
1. Verify .NET Framework 4.5 is installed.
  - a. Navigate to Programs and Features and ensure that Microsoft .NET Framework 4.5 is installed. If you do not see the framework installed, install it per local installation procedures.



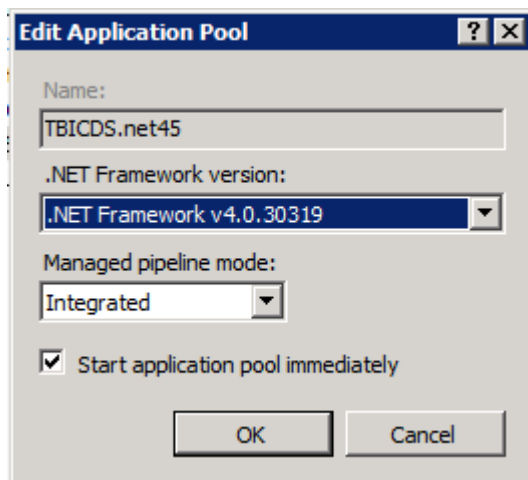
## Create Application Pool

Create a new application pool for the TBI CDS application.

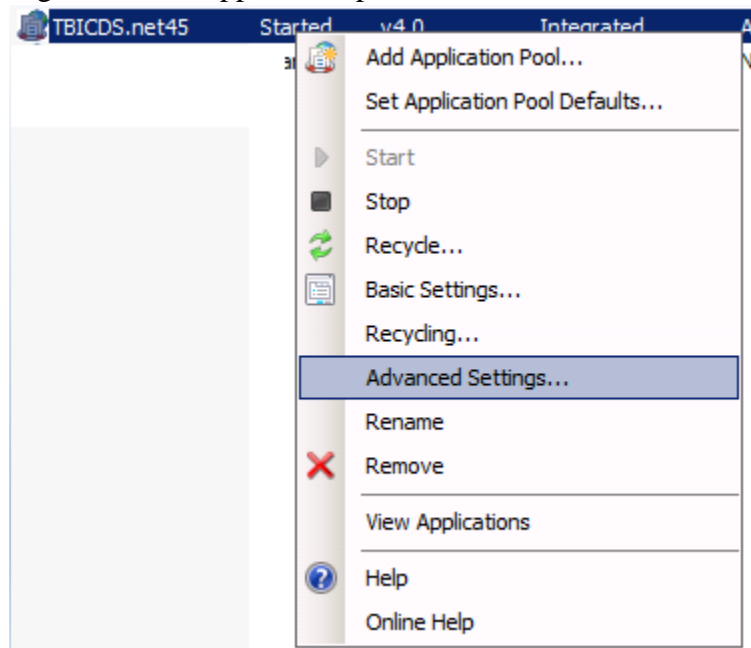
1. Open Internet Information Services (IIS) manager, right-click the Application Pools option and select 'Add Application Pool...'



2. Name the application pool TBICDS.net45 and choose the following options:



3. Right-click the application pool in the list and select the Advanced Settings option.



4. Update the settings for the application pool as follows. These settings can be adjusted later if needed. 'Enable 32-Bit Applications' must be set to 'True'.

| (General)                  |              |
|----------------------------|--------------|
| .NET Framework Version     | v4.0         |
| Enable 32-Bit Applications | True         |
| Managed Pipeline Mode      | Integrated   |
| Name                       | TBICDS.net45 |
| Queue Length               | 1000         |
| Start Automatically        | True         |

| <b>CPU</b>                 |            |
|----------------------------|------------|
| Limit                      | 0          |
| Limit Action               | NoAction   |
| Limit Interval (minutes)   | 5          |
| Processor Affinity Enabled | False      |
| Processor Affinity Mask    | 4294967295 |

| <b>Process Model</b>                 |                                |
|--------------------------------------|--------------------------------|
| Identity                             | <b>ApplicationPoolIdentity</b> |
| Idle Time-out (minutes)              | 20                             |
| Load User Profile                    | <b>False</b>                   |
| Maximum Worker Processes             | 1                              |
| Ping Enabled                         | <b>False</b>                   |
| Ping Maximum Response Time (seconds) | 90                             |
| Ping Period (seconds)                | 30                             |
| Shutdown Time Limit (seconds)        | 90                             |
| Startup Time Limit (seconds)         | 90                             |

| <b>Process Orphaning</b> |       |
|--------------------------|-------|
| Enabled                  | False |
| Executable               |       |
| Executable Parameters    |       |

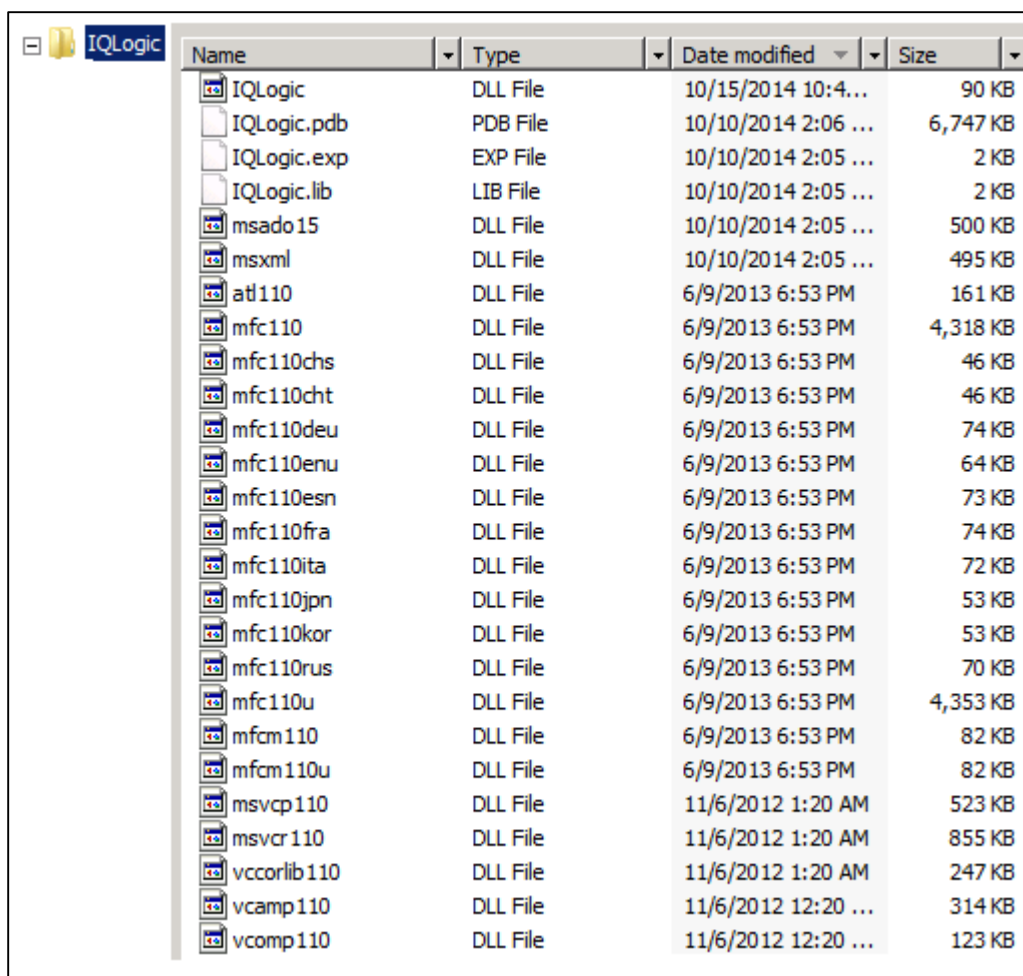
| <b>Rapid-Fail Protection</b>        |              |
|-------------------------------------|--------------|
| "Service Unavailable" Response Type | HttpLevel    |
| Enabled                             | <b>False</b> |
| Failure Interval (minutes)          | 5            |
| Maximum Failures                    | 5            |
| Shutdown Executable                 |              |
| Shutdown Executable Parameters      |              |

| <b>Recycling</b>                            |                         |
|---|-------------------------|
| Disable Overlapped Recycle                  | False                   |
| Disable Recycling for Configuration Changes | False                   |
| Generate Recycle Event Log Entry            |                         |
| Private Memory Limit (KB)                   | 0                       |
| Regular Time Interval (minutes)             | <b>90000</b>            |
| Request Limit                               | 0                       |
| Specific Times                              | <b>TimeSpan[] Array</b> |
| Virtual Memory Limit (KB)                   | 0                       |

## Install the IQLogic component

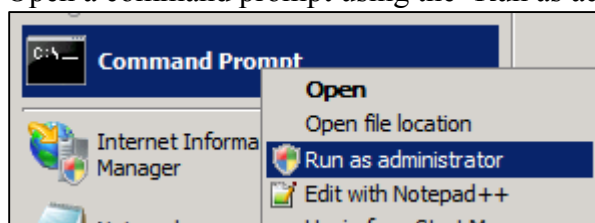
IQLogic is a Windows COM component used to run assessment logic.

1. Copy all IQLogic files to a directory on the web server.

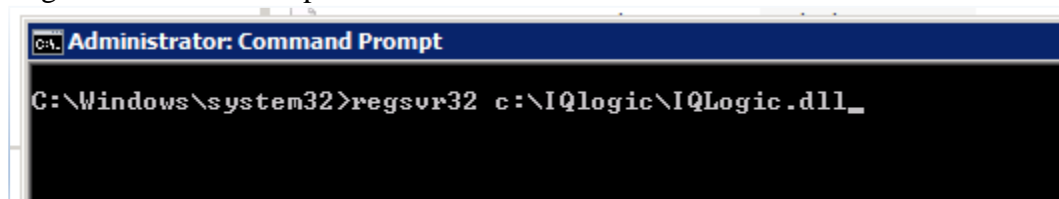


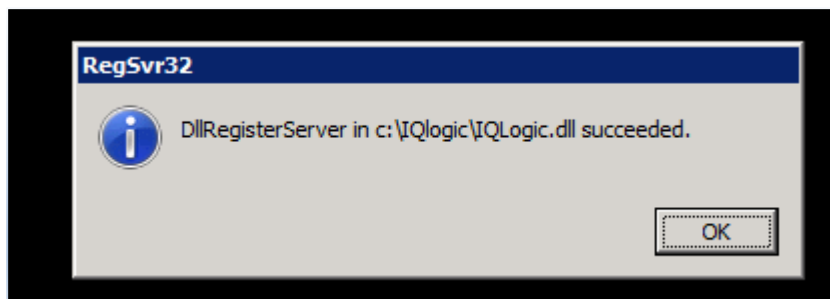
| Name        | Type     | Date modified       | Size     |
|-------------|----------|---------------------|----------|
| IQLogic     | DLL File | 10/15/2014 10:4...  | 90 KB    |
| IQLogic.pdb | PDB File | 10/10/2014 2:06 ... | 6,747 KB |
| IQLogic.exp | EXP File | 10/10/2014 2:05 ... | 2 KB     |
| IQLogic.lib | LIB File | 10/10/2014 2:05 ... | 2 KB     |
| msado15     | DLL File | 10/10/2014 2:05 ... | 500 KB   |
| msxml       | DLL File | 10/10/2014 2:05 ... | 495 KB   |
| atl110      | DLL File | 6/9/2013 6:53 PM    | 161 KB   |
| mfc110      | DLL File | 6/9/2013 6:53 PM    | 4,318 KB |
| mfc110chs   | DLL File | 6/9/2013 6:53 PM    | 46 KB    |
| mfc110cht   | DLL File | 6/9/2013 6:53 PM    | 46 KB    |
| mfc110deu   | DLL File | 6/9/2013 6:53 PM    | 74 KB    |
| mfc110enu   | DLL File | 6/9/2013 6:53 PM    | 64 KB    |
| mfc110esn   | DLL File | 6/9/2013 6:53 PM    | 73 KB    |
| mfc110fra   | DLL File | 6/9/2013 6:53 PM    | 74 KB    |
| mfc110ita   | DLL File | 6/9/2013 6:53 PM    | 72 KB    |
| mfc110jpn   | DLL File | 6/9/2013 6:53 PM    | 53 KB    |
| mfc110kor   | DLL File | 6/9/2013 6:53 PM    | 53 KB    |
| mfc110rus   | DLL File | 6/9/2013 6:53 PM    | 70 KB    |
| mfc110u     | DLL File | 6/9/2013 6:53 PM    | 4,353 KB |
| mfc110u     | DLL File | 6/9/2013 6:53 PM    | 82 KB    |
| mfc110u     | DLL File | 6/9/2013 6:53 PM    | 82 KB    |
| msvc110     | DLL File | 11/6/2012 1:20 AM   | 523 KB   |
| msvc110     | DLL File | 11/6/2012 1:20 AM   | 855 KB   |
| vccorlib110 | DLL File | 11/6/2012 1:20 AM   | 247 KB   |
| vcamp110    | DLL File | 11/6/2012 12:20 ... | 314 KB   |
| vcomp110    | DLL File | 11/6/2012 12:20 ... | 123 KB   |

2. Register IQLogic.dll using the regsvr32 command.
  - a. Open a command prompt using the 'Run as administrator' option.



- b. Register the COM component as follows:

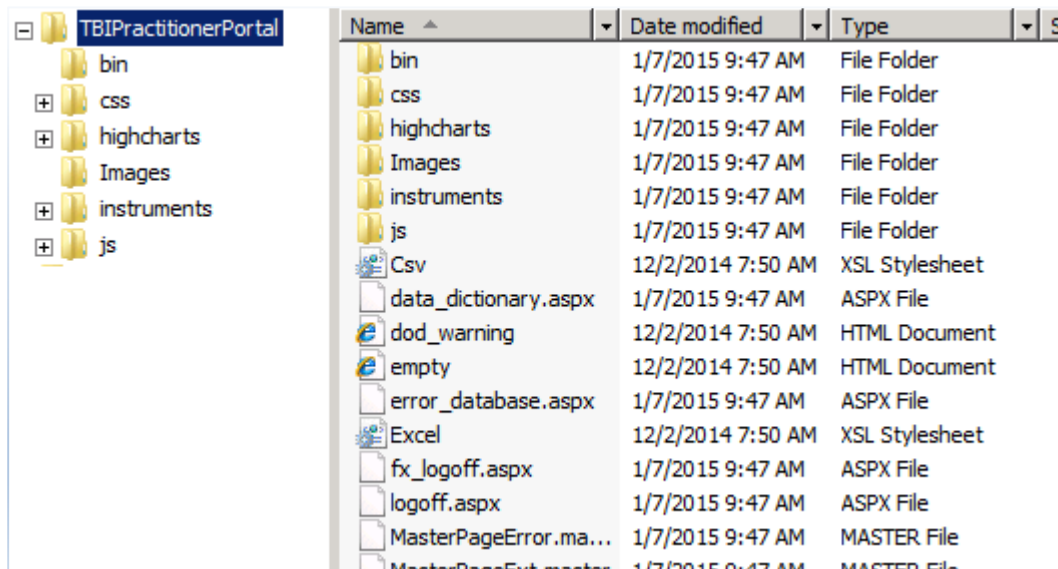




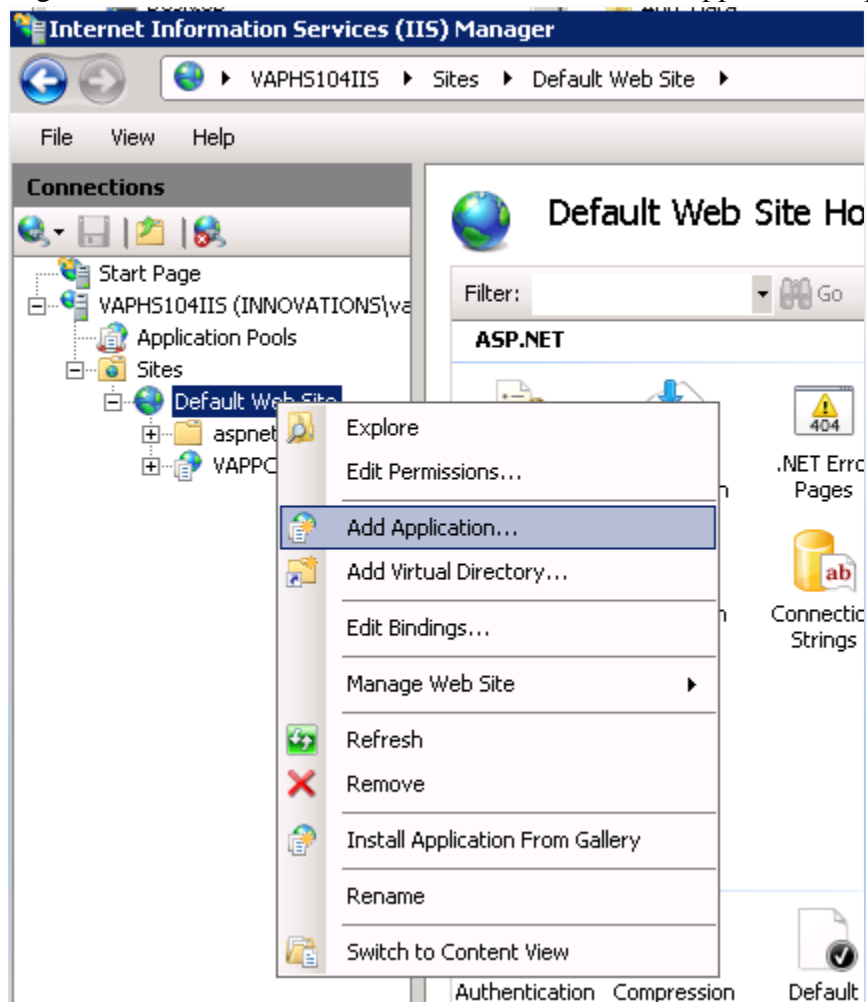
## Install TBI Practitioner Portal

TBI Practitioner Portal is the web application used by providers.

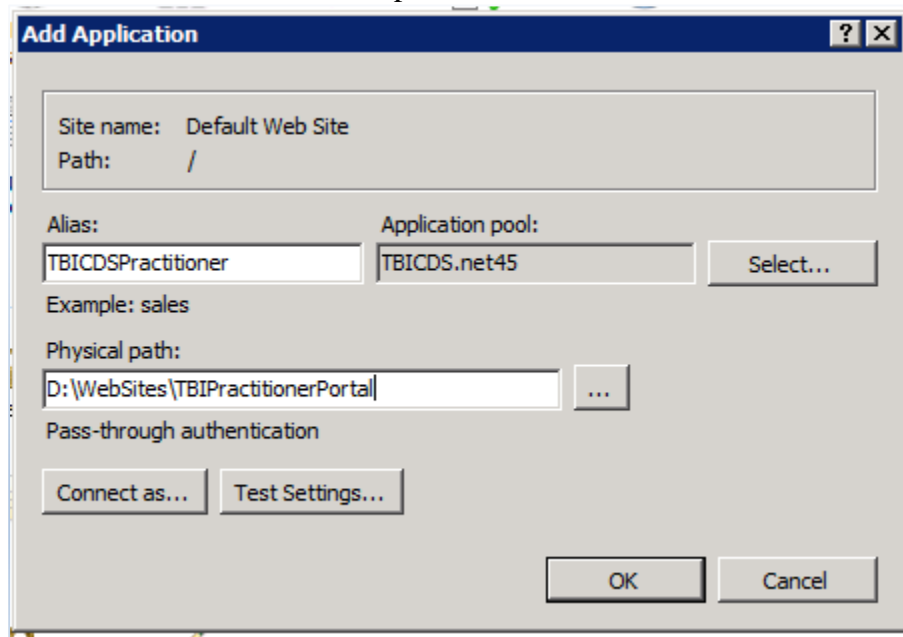
1. Copy all website files to a directory on the web server. The files should be copied to a drive other than the C: drive.



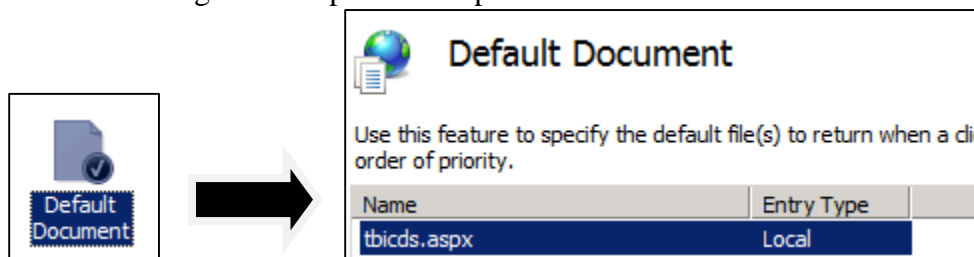
2. Open IIS manager and create a new web application as follows:
  - a. Right-click the default website and choose the ‘Add Application’ option.



- b. Name the application TBICDSPractitioner, select the TBICDS.net45 application pool created earlier and choose the path for the website.



- c. Set the default document for the website by double-clicking the default document icon and adding tbicds.aspx to the top of the list.



- d. Encrypt the connection information in the web.config file as follows:
- i. Modify the Web.Config file in the application directory with the connection information used to connect to the database. You may have to open Notepad as an administrator for the file to save.

```
<connectionStrings>  
  <add name="DBConnString" connectionString="Data Source=XXXX;User Id=XXXX;Password=XXXX;" />  
</connectionStrings>
```

ii. Connection information must be encrypted in the web.config file using the aspnet\_regiis tool as follows:

- 1) Change directory to:  
windows\microsoft.net\framework\v2.0.50727\
- 2) Run the following command:

```
aspnet_regiis -pef "connectionStrings" "[path to files]\TBICDSPractitioner" -  
prov "RsaProtectedConfigurationProvider"
```

e. Open the web.config file with Notepad and make sure the "connectionStrings" section is encrypted.

## Database Server

After Oracle Database is installed and configured, navigate to the database installation scripts and edit the run.bat file parameters as follows:

P1\_SYSDUSER – database user that has sysdba privileges

P2\_SYSPWD – password of the above system user

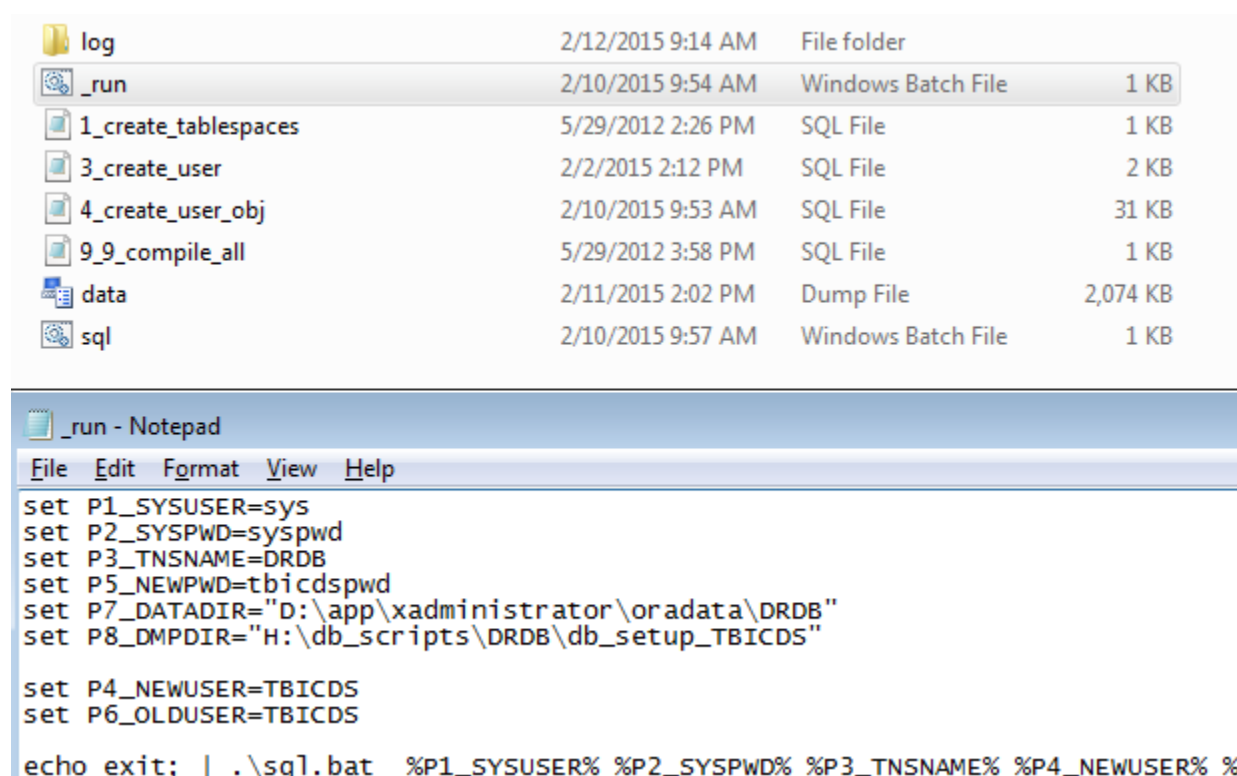
P3\_TNSNAME – target database TNS name

P5\_NEWPWD – password for the TBICDS user

P7\_DATADIR – Oracle data directory that will contain the data files

P8\_DMPDIR – path for the data dump file, which should be the same path as this batch file

Note: Do not edit P4 and P6 unless it is decided to use a different schema name.



Once the batch file is edited, save and execute it from the database server. The batch file will perform the following:

1. Create the table spaces,
2. Create the TBICDS schema/user,
3. Create all database objects,
4. Compile all objects, and
5. Write the output from the scripts to the log directory for review by the database administrator.

After running the scripts, edit the run.bat file and scrub the parameters or simply delete the files and empty the recycle bin.

We will provide similar scripts to update the database as needed for future software releases.

## Procedures for Backing-up Audit Table

We use an Oracle procedure named “PRC\_UTL\_BACKUP\_FX\_AUDIT”, that is scheduled via an Oracle job to run 1:00AM every day.

The parameters needed for the procedure are stored in “UTL\_PARAMETER” table.

Below are the current values.

| PARAMETER_NAME          | PARAMETER_VALUE         |
|-------------------------|-------------------------|
| FX_AUDIT_BACKUP_DIR     | F:\ora_backup\fx_audit\ |
| FX_AUDIT_RETENTION_DAYS | 3                       |

FX\_AUDIT\_BACKUP\_DIR is the directory where the audit data will be exported

FX\_AUDIT\_RETENTION\_DAYS is the number of days data you wish to keep in the fx\_audit table after an export.

So for example, a value of 3 means that after the complete FX\_AUDIT table is exported, All records are deleted with a time stamp < 3 days ago

## CPRS Tools Menus Installation

An entry for the website must be added to the tools menu of CPRS using the menus as follows.

```

CS      GUI Cover Sheet Display Parameters ...
HS      GUI Health Summary Types
TM      GUI Tool Menu Items
MP      GUI Parameters - Miscellaneous
UC      GUI Clear Size & Position Settings for User
RE      GUI Report Parameters ...
NV      GUI Non-VA Med Statements/Reasons
EX      GUI Expired Orders Search Hours
RM      GUI Remove Button Enabled
NON     GUI Remove Button Enabled for Non-OR Alerts
CLOZ    GUI Edit Inpatient Clozapine Message
COAG    GUI Anticoagulation Parameters ...
DEA     GUI ePCS Management Menu ...
EIE     GUI Mark Allergy Entered in Error

Select GUI Parameters <TEST ACCOUNT> Option:

```

CPRS GUI Tools Menu may be set for the following:

|     |          |     |                                 |
|-----|----------|-----|---------------------------------|
| 1   | User     | USR | [choose from NEW PERSON]        |
| 2   | Location | LOC | [choose from HOSPITAL LOCATION] |
| 2.5 | Service  | SRV | [choose from SERVICE/SECTION]   |
| 3   | Division | DIV | [choose from INSTITUTION]       |
| 4   | System   | SYS | [GOLD.VAINNOVATION.US]          |

Enter selection:

Website tools menu example:

Name=Command: TBI CDS

Portal=<https://65.36.42.218/TBIPractitionerPortal/TBICDSXPAT.aspx?p1=%DUZ&p2=%DFN&p3=%SRV&p4=%PORT&p5=%MREF>

## Connecting from TBI/CDS to MDWS is via SSL.

To change the MDWS site that TBI CDS points to, edit the web.config file and change the endpoint address as follows:

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="EmrSvcSoap" allowCookies="true"
        maxBufferPoolSize="524288"
        maxReceivedMessageSize="131072"
        bypassProxyOnLocal="false"
        hostNameComparisonMode="StrongWildcard"
        maxBufferSize="131072"
        transferMode="Buffered"
        useDefaultWebProxy="true" >
        <security mode="Transport" />
      </binding>
      <binding name="EmrSvcSoap1" allowCookies="true"
        maxBufferPoolSize="524288"
        maxReceivedMessageSize="131072"
        bypassProxyOnLocal="false"
        hostNameComparisonMode="StrongWildcard"
        maxBufferSize="131072"
        transferMode="Buffered"
        useDefaultWebProxy="true" />
    </basicHttpBinding>
  </bindings>
</system.serviceModel>
```

```

<customBinding>
  <binding name="EmrSvcSoap12">
    <textMessageEncoding messageVersion="Soap12" />
    <httpsTransport allowCookies="true"
      maxBufferPoolSize="524288"
      maxReceivedMessageSize="131072"
      authenticationScheme="Anonymous"
      bypassProxyOnLocal="false"
      hostNameComparisonMode="StrongWildcard"
      keepAliveEnabled="false"
      maxBufferSize="131072"
      proxyAuthenticationScheme="Anonymous"
      realm=""
      transferMode="Buffered"
      unsafeConnectionNtlmAuthentication="false"
      useDefaultWebProxy="true" />
  </binding>
</customBinding>
</bindings>
<client>
  <endpoint address="https://mdws.vaccloud.us/mdws2/EmrSvc.asmx"
    binding="basicHttpBinding" bindingConfiguration="EmrSvcSoap"
    contract="MDWSEmrSvc.EmrSvcSoap" name="EmrSvcSoap" />
  <endpoint address="https://mdws.vaccloud.us/mdws2/EmrSvc.asmx"
    binding="customBinding" bindingConfiguration="EmrSvcSoap12"
    contract="MDWSEmrSvc.EmrSvcSoap" name="EmrSvcSoap12" />
</client>
</system.serviceModel>

```

## Connectivity Troubleshooting

To troubleshoot & diagnose Oracle connectivity problems between the application and the database server, follow this checklist:

### Ping database host IP

```
ping 11.222.333.44
```

If it works, go to the next step.

If not, check the server availability.

### Ping database hostname

```
ping orcl.testsrv.com
```

If it works, go to the next step.

If not, something is wrong with DNS /ACTIVE directory => try using hosts file.

### Test if listener port is reachable/opened for your remote connection

To test, use the telnet utility that is available on Unix and can be enabled on Windows.

## 1. Linux example

```
$ telnet orcl.testsrv.com 1522
```

```
Trying 10.126.247.11...
```

```
Connected to orcl.testsrv.com (12.222.333.44).
```

```
Escape character is '^['.
```

```
get
```

```
Connection closed by foreign host.
```

## 2. Windows example

1) First enable telnet on Windows 7.

```
pkgmgr /iu:"TelnetClient"
```

2) Check the database port.

```
telnet testsrv.com 1523
```

```
=> no output in case of a port opened; in case of a failure - see below Oracle
```

```
error:
```

```
Could not open connection to the host, on port 1523: Connect failed
```

If the database listener port is not reachable, you may face a firewall issue. Here are two solutions to this problem.

## Test Oracle Net connectivity – tnsping

The Oracle Net Listener is the gateway to the Oracle instance for all nonlocal user connections. A single listener can service multiple database instances and thousands of client connections. tnsping is the Oracle Net equivalent of the TCP/IP ping utility. It offers a quick test to verify that the network path to a destination is good. The utility validates that the host name, port, and protocol reach a listener. It does not actually check whether the listener handles the service name or whether a database is up and running

```
tnsping orcl.testsrv.com:1521/orcl or
```

```
tnsping orcl
```

1) If it works, you will get the following message:

```
OK (10 msec)
```

2) In case of issues or errors, verify that the database listener is configured properly, and/or troubleshoot the client side (see the section below).

## Test database connection

1) With EZCONNECT bypassing tnsnames.ora (you can omit default port 1521):

```
sqlplus user@'//orcl.testsrv.com:1521/orcl'
```

```
sqlplus user@'//orcl.testsrv.com/orcl'
```

2) With TNS alias using tnsnames.ora file:

```
sqlplus user@orcl
```

## Troubleshoot the client side

Before trying to resolve a particular Oracle error on client side, perform the following on client side:

- Ensure your Oracle client is installed and configured properly,
- Identify your Oracle environment,
- Identify current ORACLE\_HOME,
- Identify a location of tnsname.ora file (if used),
- Verify that you have correctly entered the service name of the database that you want to reach, and
- If you are connecting from a login dialog box, verify that you are not placing an at symbol (@) before your connection service name.

You can use Oracle Universal Installer (OUI) and OS commands to achieve all above steps. For example, on Windows, the following sqlplus commands can be useful in identifying your Oracle environment:

```
sqlplus /nolog
```

```
@%ORACLE_HOME%
```

```
@%TNS_ADMIN%
```

## Error codes for problems on the client side

The following error codes are related to problems on the client side:

*ORA-12154: TNS:could not resolve the connect identifier specified*

### Cause and Action:

Usually this error indicates that a connect identifier / tns alias you use in your connection cannot be recognized or found somewhere. Cross check your tnsnames.ora if it exists there.

*ORA-12198: "TNS:could not find path to destination" and*

*ORA-12203: "TNS:unable to connect to destination"*

**Cause:** The client cannot find the desired database.

### Action:

1. Verify that the service name ADDRESS parameters in the connect descriptor of your TNSNAMES.ORA file are correct.
2. Verify that the listener on the remote node has started and is running. If not, start the listener by using the Listener Control utility.

*ORA-12533: "TNS:illegal ADDRESS parameters"*

**Cause:** The protocol-specific parameters in the ADDRESS section of the designated connect descriptor in your tnsnames.ora file are incorrect.

**Action:** For more information about protocol-specific keywords, refer to the Oracle operating system documentation for your platform.

*TNS-12541: TNS:no listener*

**Cause:** The listener on the remote node cannot be contacted.

**Action:** Verify that the listener on the remote node has been started. You can check its status with the STATUS command of the Listener Control utility and start it with the START command if necessary. Verify that the database listener is configured properly using the following commands:

tnslsnr status

tnslsnr status

tnslsnr services

...

## tnsnames.ora file example

```
ORCL1 =
```

```
(DESCRIPTION =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = orcl1.testsrv.com)(PORT = 1521))
```

```
(CONNECT_DATA =
```

```
(SERVER = DEDICATED)
```

```
(SERVICE_NAME = ORCL1)
```

```
)
```

```
)
```

```
ORCL2 =
```

```
(DESCRIPTION =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = orcl2.testsrv.com)(PORT = 1522))
```

```
(CONNECT_DATA = (SERVICE_NAME = ORCL2))
```

```
)
```

```
ORCL3 =
```

```
(DESCRIPTION =
```

```
(ADDRESS_LIST =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = orcl3.testsrv.com)(PORT = 1523))
```

```
)
```

```
(CONNECT_DATA = (SID = ORCL3))
```

```
)
```

## Troubleshoot IIS7 errors

### Enable IIS7 detailed errors

IIS7 introduces a new custom errors feature, which by default hides the error responses issued by the server to remote clients, replacing them with a basic error message. This is critical for security of your site, as errors frequently contain sensitive information that you don't want others to see. So, if you are requesting your site from another machine, you may still get a basic error that looks like this:

### Server Error

---

#### *HTTP Error 404 - File or directory not found.*

**Description:** The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.

---

**Server Version Information:** Internet Information Services 7.0.

You have two options here:

#### **1) Make the request locally from the server machine.**

By default, you will get the detailed error.

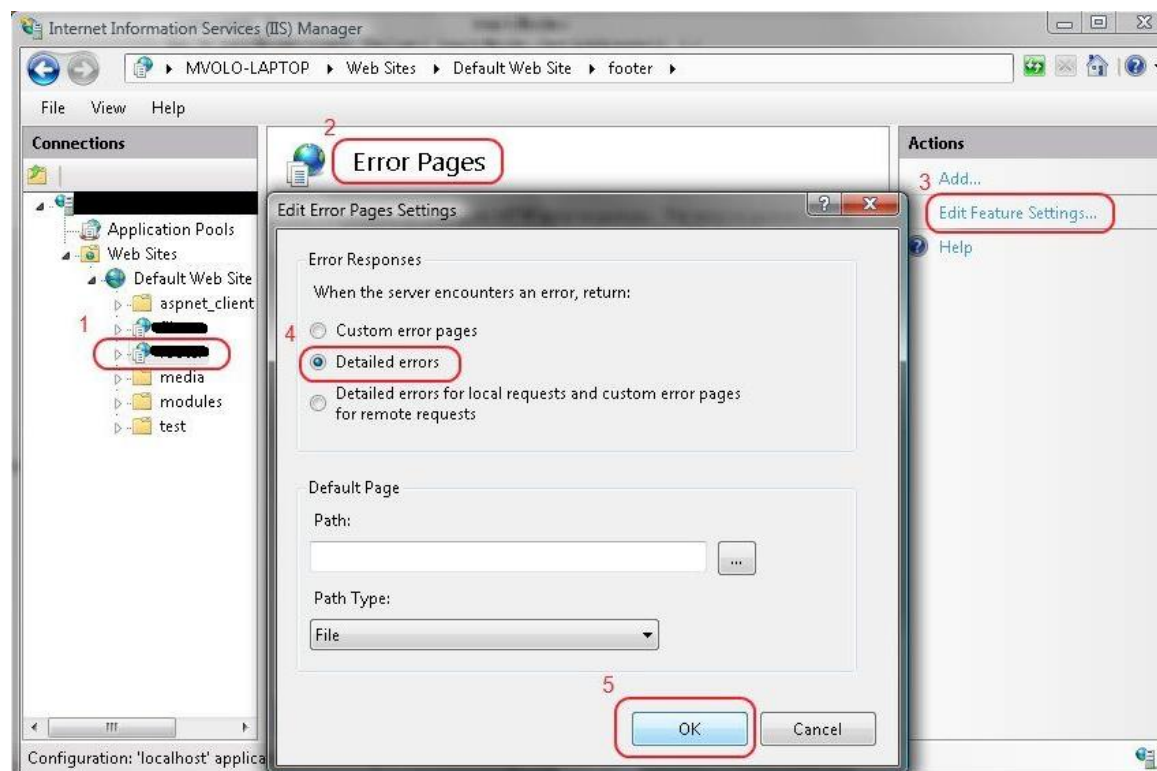
#### **2) Enable detailed errors for remote clients.**

First, if your error is an ASP.NET exception (you can tell if it says "Runtime Error" and has the framework version), please be aware that ASP.NET overrides the IIS custom error mechanism with its own implementation of custom errors, so you should turn the ASP.NET custom errors off to see detailed ASP.NET exceptions. You DO NOT have to configure IIS7 custom errors for ASP.NET exceptions (it would be silly to have to do it in two places). To turn off ASP.NET custom errors, place the following in your web.config:

```
<system.web>
  <customErrors mode="Off" />
</system.web>
```

If the error is not an ASP.NET error, turning off IIS7 custom errors will allow error responses from your application to be sent to remote clients without being censored by the IIS7's custom errors module.

You can do this from the IIS7 Admin tool by running “Start>Run>inetmgr.exe”, selecting your website/application/virtual directory in the left-hand tree view, clicking on the “Error Pages” icon, clicking “Edit Feature Settings” action, and then selecting “Detailed Errors”.



You can alternatively do this from an elevated command line prompt (Run as Administrator):

```
> %windir%\system32\inetsrv\appcmd.exe set config "Default Web Site/yourapp"  
/section:httpErrors /errorMode:Detailed
```

Where “Default Web Site” should be replaced with your site’s name if different, and “/yourapp” with the virtual path you’d like to enable detailed errors for.

NOTE: When you are done debugging, please return the settings back to custom errors for remote requests, or the security of your website may be compromised. Here is how to do it with AppCmd:

```
> %windir%\system32\inetsrv\appcmd.exe set config "Default Web Site/yourapp"  
/section:httpErrors /errorMode:DetailedLocalOnly
```

Now, you should be getting detailed errors back – for example, the error I was getting before

now looks like this (this is the Vista error page – Longhorn Server and Vista SP1 error pages will look much nicer).

## Server Error in Application "Default Web Site"

### HTTP Error 404.0 - Not Found

**Description:** The resource you are looking for has been removed, had its name changed, or is temporarily unavailable.

**Error Code:** 0x80070002

**Notification:** MapRequestHandler

**Module:** IIS Web Core

**Requested URL:** http://[REDACTED]

**Physical Path:** D:[REDACTED]

**Logon User:** Anonymous

**Logon Method:** Anonymous

**Handler:** StaticFile

**Most likely causes:**

- The directory or file specified does not exist on the Web server.
- The URL contains a typographical error.
- A custom filter or module, such as URLScan, restricts access to the file.

**What you can try:**

- Create the content on the Web server.
- Review the browser URL.
- Create a tracing rule to track failed requests for this HTTP status code and see which module is calling SetStatus. For more information about creating a tracing rule for failed requests, click [here](#).

**More Information...** This error means that the file or directory does not exist on the server. Create the file or directory and try the request again.

**Server Version Information:** Internet Information Services 7.0.

Notice that this error contains quite a bit of useful information:

- 1) The heading contains the substatus code, 404.0, which is an IIS specific code that further describes the problem. The majority of common errors have a unique <status\_code>.<substatus\_code> combination that you can easily Google for additional information.
- 2) The page indicates what module (IIS Web Core) and in what notification (MapRequestHandler) an error occurred. This information is critical whenever you are debugging server problems, especially on the IIS7+ world when most of them occur in one of the modules that take part in the processing of the request.
- 3) The page shows you some useful request information, such as the requested URL and the physical path to which it resolved. Most 404 errors will be resolved right here, by seeing that the request is being made to a wrong URL or resolving to a wrong physical path due to incorrect virtual directory mapping.

4) The “**most likely causes**” and “**what you can try**” sections offer the most likely explanation for the error and what you can do about it. They are dead-on for the majority of common IIS7 errors.

### 3) The 503 “Service Unavailable” error

Sometimes, you will get an error that looks like this:

## Service Unavailable

HTTP Error 503. The service is unavailable.

This failure is typically caused by a critical error during worker process initialization, or more likely an unhandled exception / access violation occurring during worker process startup. The most common instance of this is a module AVing inside its RegisterModule function during worker process initialization. It can also be caused by the process going away due to an unhandled exception / AV during the processing of the request that causes the process to be torn down.

After a certain number of failures, the application pool will trigger Rapid Fail Protection, a WAS feature designed to stop application pools with a persistent failure condition to avoid an endless loop of failing to start worker processes. At this point, all requests to applications within the stopped application pool will result in the 503 error, and the application pool will need to be re-started manually.

**> %systemroot%\windowssystem32\inetsrv\AppCmd.exe list apppools**

```

Administrator: D:\Windows\system32\cmd.exe

D:\Windows\System32\inetsrv>appcmd list wp

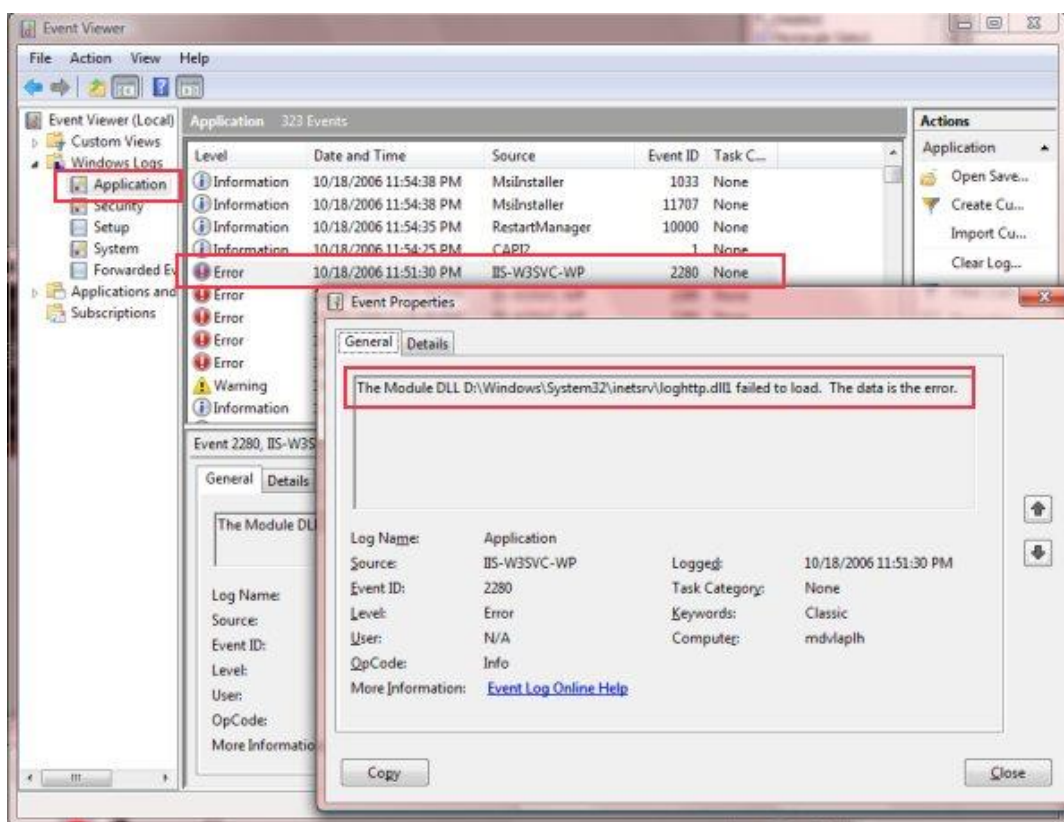
D:\Windows\System32\inetsrv>appcmd list apppools
APPPool "DefaultAppPool" <MgdVersion:v2.0,MgdMode:Integrated,state:Stopped>
APPPool "Classic .NET AppPool" <MgdVersion:v2.0,MgdMode:Classic,state:Started>

D:\Windows\System32\inetsrv>eventvwr

D:\Windows\System32\inetsrv>_

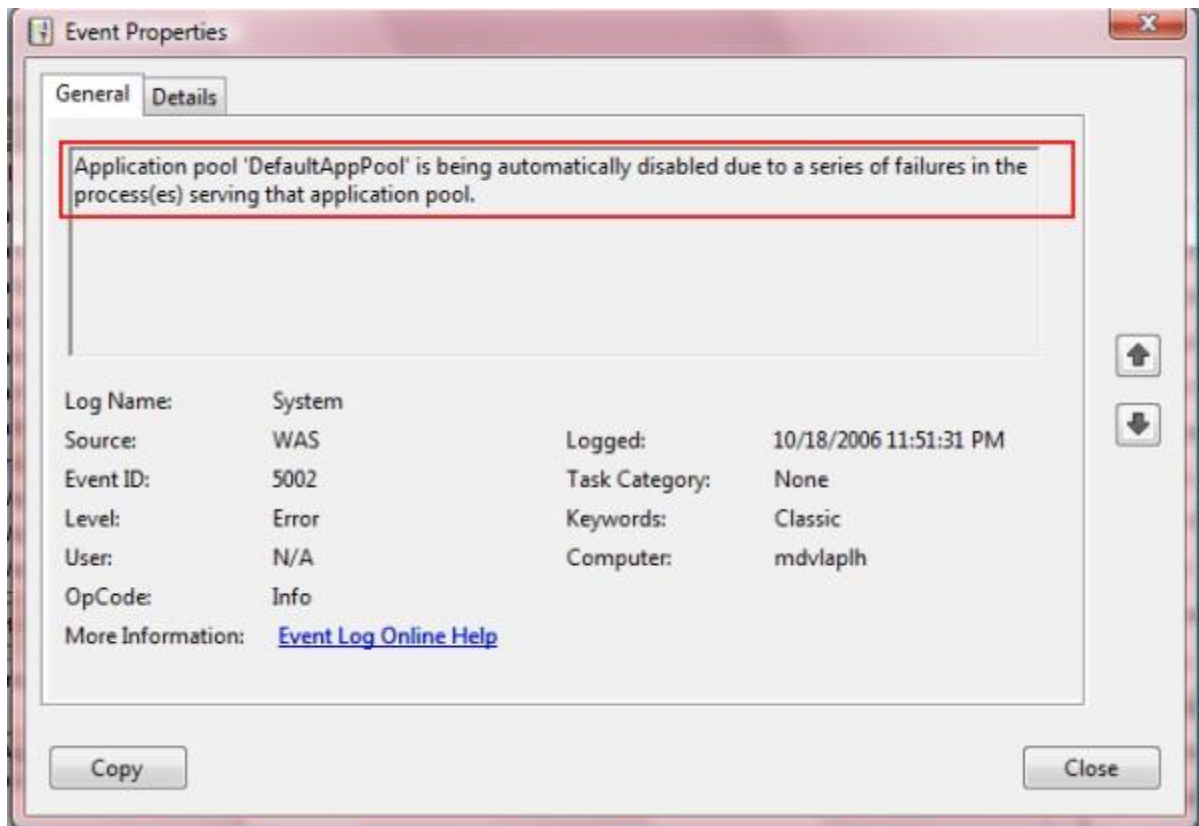
```

If your application pool is stopped, then we are on the right track. The next step is to check the event log for information about why your worker process could not be started.



In this case, the IIS worker process could not initialize because it failed to load a module DLL (the path of which I “accidentally” misspelled in my configuration).

If the error occurred before the worker process could be started, such as with WAS failing to create the process, the error will be in the “System” event log. In my case, the IIS worker process had the initial error, and WAS eventually disabled the application pool after triggering rapid fail protection, leaving the following in the “System” event log:

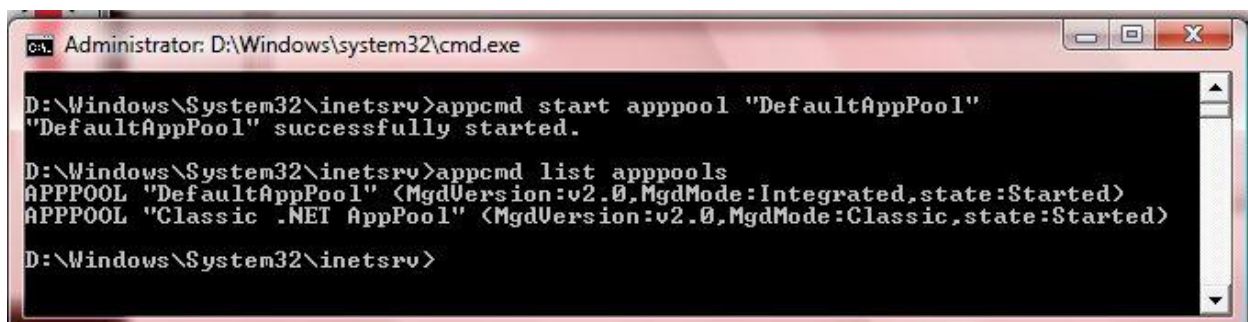


Now that the error is fixed, we can start the application pool and try the request again:

```
> %systemroot%\system32\inetsrv\appcmd.exe start apppool DefaultAppPool
```

(Replace **DefaultAppPool** with the name of your application pool.)

You should see that the application pool successfully started.



## Deeper diagnostics with Failed Request Tracing

If the error alone is not sufficient to diagnose the condition, or more information is needed to determine what lead up to the error occurring, or, there is no error (for example, request times out), you can take advantage of the wonderful IIS7 feature called “Failed Request Tracing”.

This feature can be used to generate a detailed trace of events leading up to the failure, much like ETW, but now with more/more useful events and significantly easier to turn on and use for troubleshooting.

More importantly, you can configure Failed Request Tracing to only trace requests that encounter a specific failure condition, such as a specific response status/substatus codes, an event of specific verbosity, or a timeout.

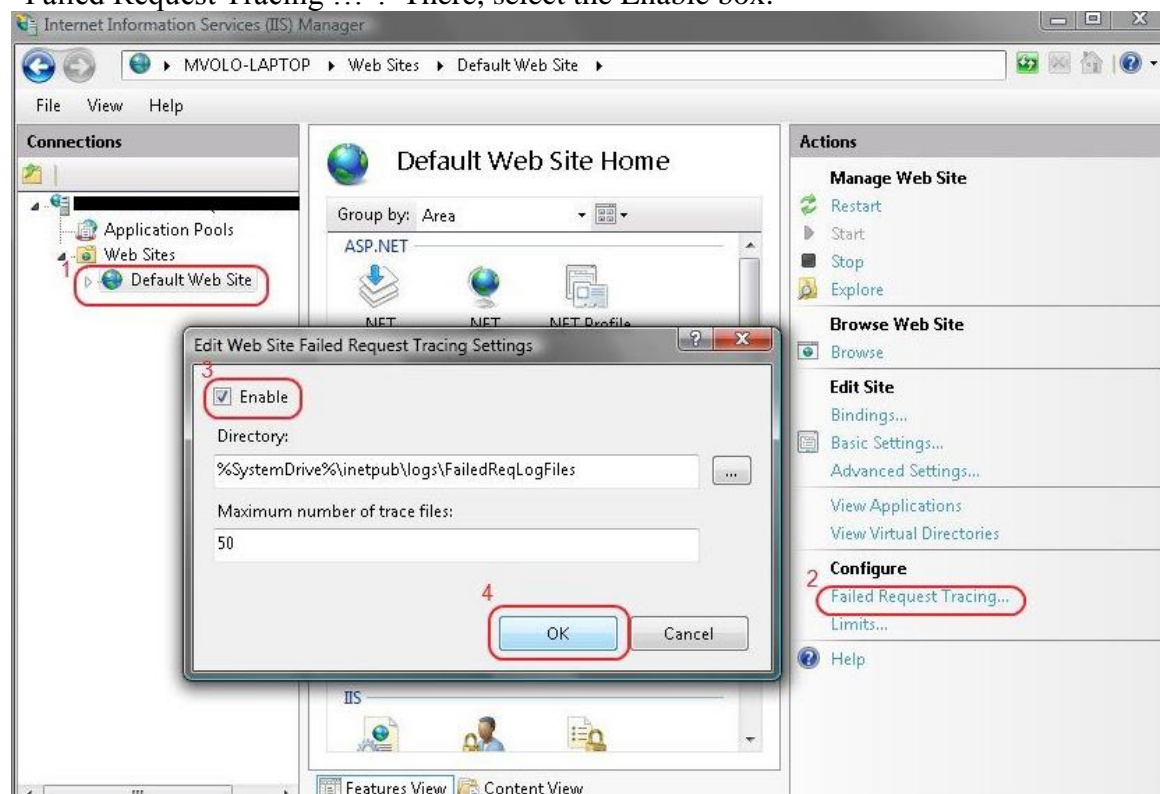
To do this, you are going to need to install it first:

On Windows Server 2008: Start>Run>**Server Manager, Roles, Web Server(IIS), Add Role Services**, check **Web Server Health and Diagnostics Tracing**

Then, to enable IIS to generate these logs, you need to do two things:

### 1) Enable Failed Request Tracing for the site you want to trace

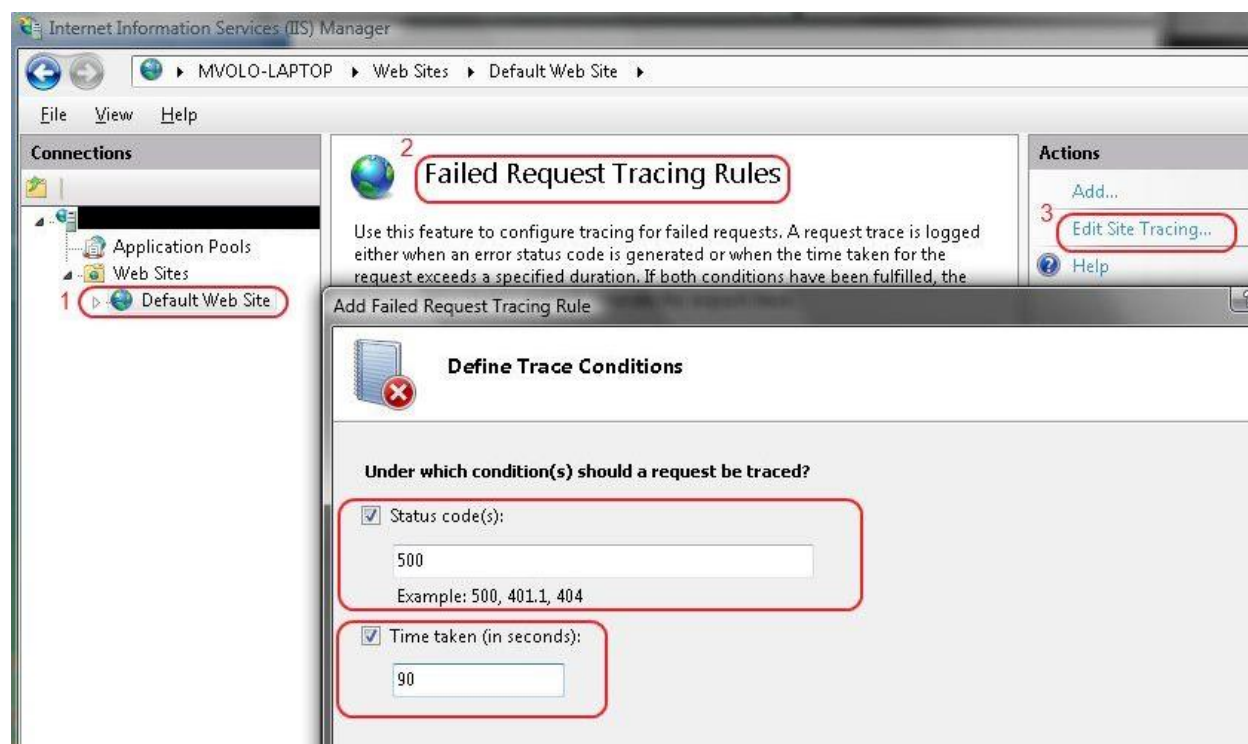
In InetMgr, select your site in the left-hand tree view, then under Actions on the right click “Failed Request Tracing ...”. There, select the Enable box.



### 2) Create a Failed Request Tracing rule with the events and failure definition of choice

In InetMgr, select the site/application/virtual directory that you’d like to trace in the tree view, click the “Failed Request Tracing Rules” icon, click the “Add...” action, and follow the wizard to create the rule.

You will be asked what content you’d like to trace (based on an extension), when to consider the request as failed (a list of response status codes, and in Longhorn Server, also event verbosity) or a timeout, and finally the events you’d like to trace. I recommend to leave all events selected by default, unless you are sure you know what you are looking for.



Now, when you make a request, if it meets the failure definition, the detailed trace will be logged as an XML file that you can inspect.

You can by default find this file in:

**%systemdrive%\inetpublogsFailedReqLogFilesW3SVC<ID>**

If you double-click this file, it will open with the provided XSL style sheet that shows the trace events. In Longhorn Server, this style sheet has gotten a major overhaul to highlight the important information better.

The Failed Request Tracing log provides verbose execution information for the request, that can be used in a number of ways to further diagnose more complex problems that may result from the operation of more than one module. For example, if a url rewriter module changes a url in the request, which later leads to the static file handler failing to find the file, the Set Url event can provide the needed information. The log also provides such key info as the times taken by each module to execute, the outcome of each of the built-in module's operation (often including the reason why something didn't work), and any times when a module fails a request by setting an error response status code.

Finally, one of the main strengths of Failed Request Tracing is that you do not have to be the client to receive the error information. As an administrator, you can configure it to monitor for errors and log the forensic trace when they occur, allowing you to just leave the feature on in production.

## HTTP Detailed Errors in IIS 7.0

### Client Errors

Status codes between 400 and 500 specify an error that the client made, e.g. bad syntax or a request to a resource that doesn't exist. You can try this by requesting a bogus URL from the web-site of your choice, for example:

`http://<IIS7Server>/this_resource_does_not_exist`. You get a "404 - File not found" error.

### Server Errors

Status codes starting with 500 are errors caused by the server. The most common causes for 500 errors on IIS systems are:

- An ASP or ASPX page that contains a syntax error
- The web server configuration or the application configuration cannot be read or is invalid
- The site is stopped

It is important to note that browsers like IE often replace errors returned from a web server with their own errors. This makes troubleshooting harder. In IE you can turn this feature off. Go to the "Tools" menu, select "Internet Options", click the "Advanced" tab and find the "Show friendly HTTP error messages" check box and uncheck it.

### Examples of Sub-status Codes

Many HTTP errors have a sub-status. The IIS default Custom Errors configuration does not differentiate based sub-status codes. It sends the same Custom Error page if you enter the wrong credentials (401.1), or if you get access denied based on invalid rights to access a file (401.3). You can see the different sub-status codes in the log files. Here is a list of the different 404 sub-status codes that IIS produces:

| Status | Description  |
|--------|--|
| 404.1  | Site could not be found  |
| 404.2  | Denied by Policy. The request ISAPI or CGI program is not allowed in the Restriction List.       |
| 404.3  | The static file handler did not have the file in its MimeMap and therefore rejected the request. |
| 404.4  | No handler was found to serve the request.   |
| 404.5  | The Request Filtering Module rejected an URL sequence in the request.                            |
| 404.6  | The Request Filtering Module denied the HTTP verb of the request.                                |

|        |  |
|--------|--|
| 404.7  | The Request Filtering module rejected the file extension of the request.                         |
| 404.8  | The Request Filtering module rejected a particular URL segment (characters between two slashes). |
| 404.9  | IIS rejected to serve a hidden file.   |
| 404.10 | The Request Filtering module rejected a header that was too long.                                |
| 404.11 | The Request Filtering module rejected a request that was double escaped.                         |
| 404.12 | The Request Filtering module rejected a request that contained high bit characters.              |
| 404.13 | The Request Filtering module rejected a request that was too long (request + entity body).       |
| 404.14 | The Request Filtering module rejected a request with a URL that was too long.                    |
| 404.15 | The Request Filtering module rejected a request with a too long query string.                    |

You can configure the `httpErrors` section to show a Custom Error for particular sub-status codes. If you add the following line to the `httpErrors` configuration section, IIS returns `404_3.htm` if a file with a file extension is requested that is not included in the IIS `MimeMap` (`<staticContent>` configuration section).

```
<error statusCode="404" subStatusCode="3" prefixLanguageFilePath="c:\inetpub\custerr"
path="404_3.htm" />
```