

11.2.8.a.2 EDRS Setup for Single Instance of ExamDeveloper



Configure IIS Web Server



Set up SearchService service (EDSS) for single instance of ExamDeveloper



If you do not plan to use EDSS, you can skip to setting up ExamService.



Set up ExamService for single instance of ExamDeveloper



ExamDeveloper setup home page

ExamDeveloper Report Service (EDRS) is a micro service that handles the generation of various reports such as the Custom Reports.

This page covers the **EDRS Configuration for Single or First Instance of ExamDeveloper**. If you are running multiple instances of ExamDeveloper, you can go to the EDRS installation and configuration for multiple instances of ExamDeveloper in **EDRS Setup for Multiple Instances of ExamDeveloper**.



The following is applicable only for ExamDeveloper version 7.1601.0.0 and above.



It is important to ensure that the following is installed and configured before you proceed with the EDRS installation and configuration, so that it works as expected:

- Microsoft Office 2010 (64-bit) or above, or Microsoft Access Database Engine 2010 Redistributable <http://www.microsoft.com/en-in/download/details.aspx?id=13255> (Download and install the file named *AccessDatabaseEngine_x64.exe*).
- Double click the following files found in the `<SVNFolder>\External\Import-Export\` folder. These files install the required registry entries.
JetDriver_ExcelImportSetting_64bitOffice.reg
JetDriver_ExcelImportSetting_64bitOffice_14.0.reg
- For the ED Application Pool in IIS, ensure that **Integrated** is selected for the **Managed pipeline** mode drop-down and **Enable 32-Bit Applications** setting is set to **False** under the **Advanced Settings** from the context menu as mentioned in **step 2 of IIS Web Server Configuration**.
- Reporting Services need to be set up and ready as detailed in **Reporting Services Setup**.

EDRS Installation and Configuration



It is important to ensure that all the values in the config files such as service name, URLs, username, and password are case-sensitive. ExamDeveloper may not run as expected if care is not taken to ensure that case sensitivity is maintained.

To install the ExamDeveloper Report Service (EDRS) for the ExamDeveloper application, please perform the following steps:

1. Create a folder where you want to install EDRS, such as `C:\EDRS\ExamDeveloper\` (henceforth known as `<EDRSinstallDir>`) and copy all the contents of `<SVNFolder>\EDRS\` to this folder.
2. Update the configuration files of EDRS located in `C:\EDRS\ExamDeveloper\configs\ (<EDRSinstallDir>\configs)` - `connectionStrings.config`, `appSettings.config`, `massTransit.config`, `httpClients.config`, and `log4net.config` - with Notepad or a suitable text file editor.
 - a. Update the `connectionStrings.config` of EDRS for the values of the "connectionString" keys "Data Source", "Initial Catalog", "User ID", and "Password" as below:



Do not modify the value of "name" from "examDeveloper" to any other value as this will result in EDRS not being able to run.

```
<add name="examDeveloper" connectionString="Data Source=WIN-VSGF1NBSVPB;Initial Catalog=ExamDeveloper;Persist Security Info=True;User ID=ExamDev;Password=P@s$w0rd"/>
```

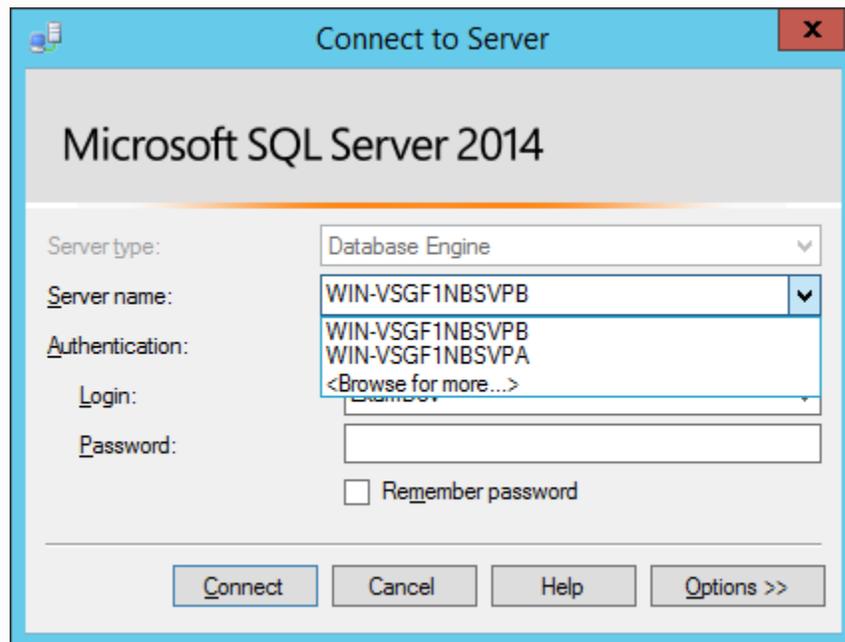
where,

- *Data Source* = Name of SQL Server instance. You can identify the name of the SQL Server instance being used for ExamDeveloper as detailed in **SQL Server Database Setup**.

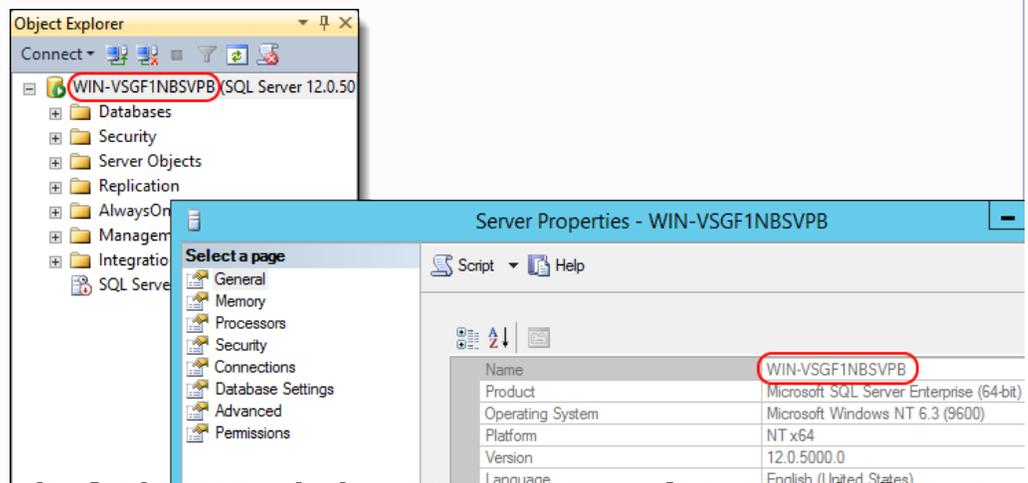


▼ To learn how to identify the SQL Server instance when there are more than one instances, click here to expand...

If you have more than one instances of SQL Servers, then you must select the one you need to connect to use with ExamDeveloper as shown in the Connection window below.



You can later identify the name of the SQL Server instance by right clicking the Server node in the SQL Server, clicking **Properties**, then selecting **General** page in the left pane of the **Server Properties** page, and observing the **Name** in the right pane as shown below.



- *Initial Catalog* = ExamDeveloper (which is the **ExamDeveloper Database**).
- *User ID* = Username of the SQL Server User who owns the database. This was created during **SQL Server Database Setup** under **SQL Server user management**. In our example, this is "ExamDev".
- *Password* = Password of the SQL Server who owns the database. This was created during **SQL Server Database Setup** under **SQL Server user management**. In our example, this is "P@s\$w0rd".

This connection string needs to match the application's connection string from the ExamDeveloper *connectionStrings.config* as shown in **step 9 of Configure ExamDeveloper**.

b. Update the key values in the *appSettings.config* of EDRS as below:

- Update the *appSettings.config* for the keys related to the Report Server in the ExamDeveloper *appSettings.config* as shown in **step 6 of Configure ExamDeveloper**. This includes values for "reportServerUrl", "reportServerPath", "reportServerUserName", "reportServerPassword", and "reportServerDomainName".
 - *reportServerUrl* = Report Server Web Service URL shown on the **Web Service URL** tab of the **Reporting Services Configuration Manager**.
 - ▼ [Where is the Report Server Web Service URL? Click to learn...](#)

i The Report Server Web Service URL can be found out by clicking the **Web Service URL** link on the left as shown below. If the URL is too long, some part of the URL may not be visible. In such a case, you can hover the mouse cursor over the URL and note it down from the tool tip.

- *reportServerPath* = the folder name and path of the reporting services application folder created in **step 7 of Setting up Reporting Services application folder**, which is /ExamDeveloper/ in our example.
- *reportServerUserName* = the administrative username used to log into the *Report Manager*, in **step 8 of Setting up Reporting Services application folder**.
- *reportServerPassword* = the password for the administrative username used to log into the *Report Manager*, in **step 8 of Setting up Reporting Services application folder**.
- *reportServerDomainName* = domain property specifies the domain or realm to which the user name belongs. Typically, this is the host computer name where the reporting services runs or the user domain for the user configured.

ii. Modify

```
<add key="serviceDisplayName" value="Unique EDRS DisplayName"/>
```

to

```
<add key="serviceDisplayName" value="UniqueEDRSServiceDisplayName"/>
```

where *UniqueEDRSServiceDisplayName* is the unique display name given to the EDRS service. This name needs to be unique per install and must not contain spaces.

iii. Modify

```
<add key="serviceName" value="Unique EDRS Name"/>
```

to

```
<add key="serviceName" value="UniqueEDRSServiceName"/>
```

where *UniqueEDRSServiceName* is the unique name given to the EDRS service. This name needs to be unique per install and must not contain spaces.

iv. Modify

```
<add key="outputDirectory" value=""/>
```

to

```
<add key="outputDirectory" value="C:\ExamDeveloperData\EDRSOutput"/>
```

It must be ensured that the value of the "outputDirectory" key specified in the *appSettings.config* must match the value for the "EdrsOutputDirectory" specified in the ExamDeveloper *appSettings.config* (that will be using this instance of EDRS) as in [step 7 of Configure ExamDeveloper](#). This is the location where reports generated by EDRS are stored.



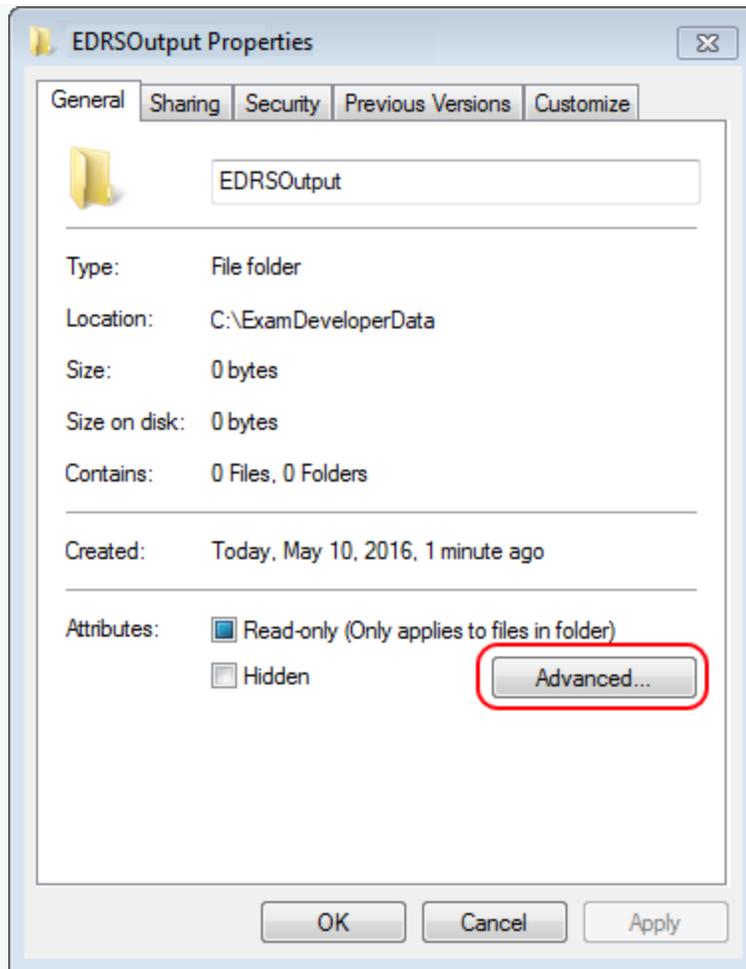
Both "LOCAL SERVICE" and "NETWORK SERVICE" users must have access to this folder, which can be achieved in a way similar to how it is detailed in [Grant Write Access for the Application Data Folder to Network User](#).



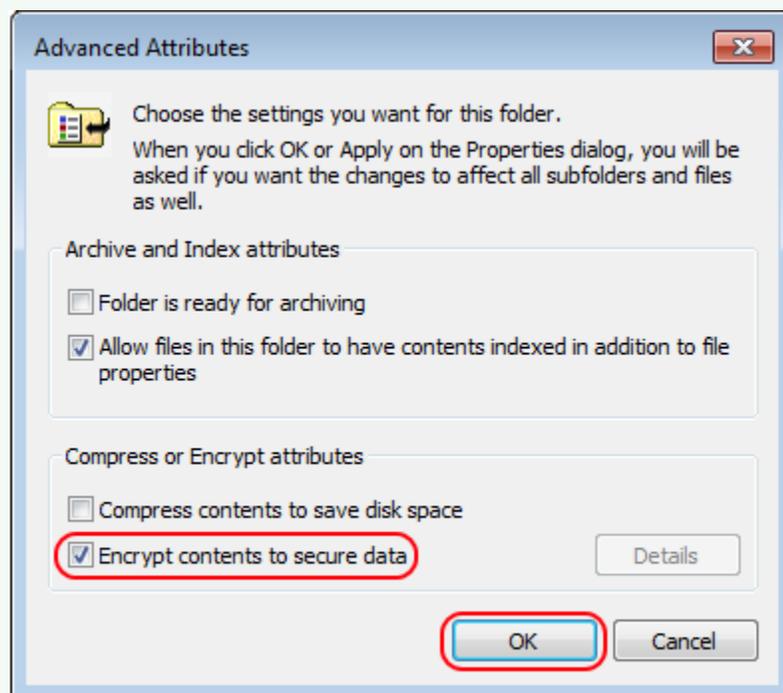
It is prudent, in the interest of data security, to encrypt the folder where EDRS stores its output. This can be done as follows:

▼ [How to encrypt the folder where EDRS stores its output? Click to expand...](#)

1. Open **File Explorer** in Windows and right-click the folder specified in the "outputDirectory" key of the EDRS *appSettings.config* file and click **Properties**.
2. In the folder **Properties** dialog, click the **Advanced** button.



3. In the **Advanced Attributes** dialog box that appears, select the **Encrypt contents to secure data** option and click **OK**.



4. Click **OK** to close the folder Properties window.

The current contents of the selected folder are now encrypted and any files or folders that are placed into it in the future will also be encrypted.



Step 2.b.v is applicable only for **ExamDeveloper version 7.1604.0.0 and above**.

- v. Ensure that the values of the keys "basicAuthUserName" and "basicAuthPassword" match the values from ExamDeveloper *httpClient.config* of the keys "userName" and "password" respectively for the line with httpClientConfig name "EDRS" as in **step 11 of Configure ExamDeveloper**.

```
<add key="basicAuthUserName" value="examDeveloper"/>
<add key="basicAuthPassword" value="Dummy_Password"/>
```

- vi. If a line with the key "useSsl" exists, set the value to "true" as follows:

```
<add key="useSsl" value="true"/>
```

- vii. If a line with the key "relativeUri" exists, set the value to "UniqueEDRSServiceName" as follows:

```
<add key="relativeUri" value="UniqueEDRSServiceName"/>
```

where *UniqueEDRSServiceName* is the name of the EDRS service.

- c. Update the *massTransit.config* of EDRS as below:

- i. Update the *massTransit/host/rabbit@hostAddress* attribute to append the name of the new Virtual Host created in **step 1 of Create a new Virtual Host** and the values for username and password to access that Virtual Host from **Create a RabbitMQ user for EDRS**. e.g., if the default value is *rabbitmq://hostname*, and your new Virtual Host name is *ExamDeveloper*, while the username is "ExamDeveloper" and password is "Password", then the line should look like below:

```
<rabbit hostAddress="rabbitmq://localhost/ExamDeveloper" username="ExamDeveloper" password="Password"/>
```



If RabbitMQ is installed on a separate server, then you must substitute "localhost" with the hostname of that server. If *RabbitMQ-URL* is the server hostname where RabbitMQ is installed, then the line becomes as below:

```
<rabbit hostAddress="rabbitmq://RabbitMQ-URL/ExamDeveloper" username="ExamDeveloper" password="Password"/>
```



It is extremely important that the same RabbitMQ Virtual Host name should not be shared between different instances of ExamDeveloper, as this can result in issues such as data loss and security breach.

It is important to ensure that the RabbitMQ Virtual Host name, username, and password are case-sensitive. You will encounter an error while trying to log into ExamDeveloper if care is not taken to ensure that both are specified exactly as they are in the **Create and configure RabbitMQ Virtual Host** and in the **Create a RabbitMQ user for EDRS** section of the **RabbitMQ Setup and Configuration** page.

- ii. You must ensure that the address specified in the *massTransit.config* of EDRS matches the address specified in the *massTransit.config* of ExamDeveloper (that will be using this instance of EDRS) as stated in **step 10 of Configure ExamDeveloper**.

e.g., if your Virtual Host is named *ExamDeveloper*, the *hostAddress* attribute value would become *rabbitmq://localhost/ExamDeveloper* username="examDeveloper" password="Password". In this case, the username "ExamDeveloper" and password "Password" are for the **RabbitMQ user** accessing the Virtual Host "ExamDeveloper".

```
<rabbit hostAddress="rabbitmq://localhost/ExamDeveloper" username="ExamDeveloper" password="Password"/>
```

If RabbitMQ is installed on a separate server, then you must substitute "localhost" with the hostname of that server. If Ra

bbitMQ-URL is the server hostname where RabbitMQ is installed, then the line becomes as below:

```
<rabbit hostAddress="rabbitmq://RabbitMQ-URL/ExamDeveloper" username="ExamDeveloper"
password="Password"/>
```



Step 2.d is applicable only for **ExamDeveloper version 7.1604.0.0 and above**.

- d. Update the *httpClients.config* of EDRS for the line with the "name" as "ExamDev" entry as shown below:

```
<httpClientConfig name="ExamDev" url="https://ED-HostName/ExamDeveloper" userName="examDeveloper" password="Dummy_Password"/>
```

The values of the keys "username" and "password" have to be same as the values from the ExamDeveloper *appSettings.config* of the keys "ApiAuthenticationUserName" and "ApiAuthenticationPassword" respectively as mentioned in [step 8 of Configure ExamDeveloper](#).

The value of "url" is the server hostname *ED-HostName* where ExamDeveloper is installed, with the name of the ExamDeveloper application *ExamDeveloper* appended to it.

For example, if the ExamDeveloper application URL is *https://examdev.pearsonvue.com/ExamDeveloper*, then the line appears as below:

```
<httpClientConfig name="ExamDev" url="https://examdev.pearsonvue.com/ExamDeveloper" userName="examDeveloper"
password="Dummy_Password"/>
```

It is important to ensure that "https" and not "http" is used in the "url" value.



The port number need not be specified here and is deemed to be "443", which is determined from the usage of "https" in the "url" value.

In the unlikely scenario where a different port number has been specified for the ExamDeveloper application in IIS, you will need to specify that port number over here and assuming the port number is "11401" the line will appear as below:

```
<httpClientConfig name="ExamDev" url="https://ED-HostName/ExamDeveloper:11401" userName="examDeveloper"
password="Dummy_Password" />
```

Or if the ExamDeveloper URL is as mentioned in the example, then the line will appear as below:

```
<httpClientConfig name="ExamDev" url="https://examdev.pearsonvue.com/ExamDeveloper:11401"
userName="examDeveloper" password="Dummy_Password" />
```



The below modifications of EDRS *httpClients.config* are applicable only for **ExamDeveloper version 7.1701.0.0 and above**.

- i. For the line with the "name" as "UserService" entry, the values of the keys "userName" and "password" have to be same as the values from the User service *appSettings.config* of the keys "basicAuthUserName" and "basicAuthPassword" respectively as mentioned in [step 4 of User service setup](#).

```
<httpClientConfig name="UserService" url="https://App-HostName/UniqueUserServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

The value of "url" is the server hostname *App-HostName* where the User service is installed, with the User service name appended to it, which is "UniqueUserServiceName" in our example.

For example, if the server hostname where the User service is installed is *examdevapp.pearsonvue.com*, then the line will appear as below:

```
<httpClientConfig name="UserService" url="https://examdevapp.pearsonvue.com/UniqueUserServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

It is important to ensure that "https" and not "http" is used in the "url" value.

- ii. For the line with the "name" as "ProjectService" entry, the values of the keys "userName" and "password" have to be same as the values from the Project service *appSettings.config* of the keys "basicAuthUserName" and "basicAuthPassword" respectively as mentioned in **step 4 of Project service setup**.

```
<httpClientConfig name="ProjectService" url="https://App-HostName/UniqueProjectServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

The value of "url" is the server hostname *App-HostName* where the Project service is installed, with the Project service name appended to it, which is "UniqueProjectServiceName" in our example.

For example, if the server hostname where the Project service is installed is *examdevapp.pearsonvue.com*, then the line will appear as below:

```
<httpClientConfig name="ProjectService" url="https://examdevapp.pearsonvue.com/UniqueProjectServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

It is important to ensure that "https" and not "http" is used in the "url" value.

- iii. For the line with the "name" as "ItemBankService" entry, the values of the keys "userName" and "password" have to be same as the values from the ItemBank service *appSettings.config* of the keys "basicAuthUserName" and "basicAuthPassword" respectively as mentioned in **step 4 of ItemBank service setup**.

```
<httpClientConfig name="ItemBankService" url="https://App-HostName/UniqueIBServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

The value of "url" is the server hostname *App-HostName* where the ItemBank service is installed, with the ItemBank service name appended to it, which is "UniqueIBServiceName" in our example.

For example, if the application hostname is *examdevapp.pearsonvue.com*, then the line will appear as below:

```
<httpClientConfig name="ItemBankService" url="https://examdevapp.pearsonvue.com/UniqueIBServiceName"
userName="examDeveloper" password="Dummy_Password"/>
```

It is important to ensure that "https" and not "http" is used in the "url" value.

- iv. Add the following line below `<httpClientConfig name="ItemBankService".../>` (if not already present):

```
<httpClientConfig name="AssignmentService" url="https://localhost/assignment" userName="examDeveloper"
password="Dummy_Password"/>
```

- e. Update the values in the `<log4net>` section of the *log4net.config* file as shown below to indicate the path where the log file for EDRS is to be located.

```
<log4net>
  <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
    <file value="C:\Program Files\Pearson\logs\EDRS\ReportService.log" />
```

By default, the log file *ReportService.log* will save to the EDRS folder (*C:\EDRS\ExamDeveloper*) in our example). You can specify a different path to save it by modifying the value to something like "*C:\Program Files\Pearson\logs\EDRS\ReportService.log*" as shown above.



The "NETWORK SERVICE" user must have access to this folder, which can be achieved in a way similar to how it is detailed [here](#).

- f. The modified areas of the EDRS configuration files appear similar to the ones shown in the below screenshots.
 - i. **appSettings.config**:

```

<appSettings>
  <add key="reportServerUrl" value="http://WIN-VSGF1NBSVPB:80/ReportServer"/>
  <add key="reportServerPath" value="/ExamDeveloper"/>
  <add key="reportServerUserName" value="edReporting"/>
  <add key="reportServerPassword" value="Password123"/>
  <add key="reportServerDomainName" value="WIN- VSGF1NBSVPB"/>
  <add key="serviceDescription" value="ExamDeveloperReportServiceDescription"/>
  <add key="serviceDisplayName" value="UniqueEDRSDisplayName"/>
  <add key="serviceName" value="UniqueEDRSServiceName"/>
  <add key="outputDirectory" value="C:\ExamDeveloperData\EDRSOutput\"/>
  <add key="showSql" value="false"/>
  <add key="basicAuthUserName" value="examDeveloper"/>
  <add key="basicAuthPassword" value="Dummy_Password"/>
  <add key="useSsl" value="true"/>
  <add key="relativeUri" value="UniqueEDRSServiceName"/>
  <!-- For new relic -->
</appSettings>

```

 Please ensure that the user, EDRS uses to connect to SSRS ("edReporting" in our example), has been granted at least the **Browser** and **Content Manager** roles as explained in **Reporting Services Setup - SSRS User Permissions**. EDRS will not start if this user does not have this permission.

ii. **connectionStrings.config:**

```

<connectionStrings>
  <add name="examDeveloper" connectionString="Data Source=WIN-VSGF1NBSVPB;Initial Catalog=ExamDeveloper;Persist Security Info=True;User ID=ExamDev;Password=P@s$w0rd"/>
</connectionStrings>

```

iii. **massTransit.config:**

```

<massTransit>
  <endpoint concurrencyLimit="5" name="ExamDev.Reports">
    <retryPolicy>
      <incremental retryLimit="2" initialInterval="00:00:01" intervalDelta="00:00:01"/>
      <!--<exponential retryLimit="10" minInterval="00:00:01" maxInterval="00:05:00" intervalDelta="00:00:01" /> -->
    </retryPolicy>
  </endpoint>
  <host>
    <rabbit hostAddress="rabbitmq://localhost/ExamDeveloper" username="ExamDeveloper" password="Password"/>
  </host>
</massTransit>

```

iv. **httpClient.config** (this config file is available only for **ExamDeveloper** version **7.1604.0.0** and above):

```

<httpClientConfigs>
  <httpClientConfig name="ExamDev" url="https://ED-HostName/ExamDeveloper" userName="examDeveloper" password="Dummy_Password"/>
  <httpClientConfig name="UserService" url="https://App-HostName/UniqueUserService" userName="examDeveloper" password="Dummy_Password"/>
  <httpClientConfig name="ProjectService" url="https://App-HostName/UniqueProjectServiceName" userName="examDeveloper" password="Dummy_Password"/>
  <httpClientConfig name="ItemBankService" url="https://App-HostName/UniqueIBServiceName" userName="examDeveloper" password="Dummy_Password"/>
  <httpClientConfig name="AssignmentService" url="https://localhost/assignment" userName="examDeveloper" password="Dummy_Password"/>
</httpClientConfigs>

```

v. **log4net.config:**

```

<log4net>
  <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
    <file value="C:\Program Files\Pearson\logs\EDRS\ReportService.log"/>
  </appender>
</log4net>

```

3. Open an Administrator command prompt and run the following command:

```
netsh http add urlacl url=https://+:443/UniqueEDRSServiceName user=Everyone
```

where *UniqueEDRSServiceName* is the name of the EDRS service.

 If you get an error stating "Url reservation add failed" on running the above command, please ignore it as it means that this has already been taken care of.

4. Install the EDRS service:

To do this, open an Administrator command prompt (click **Start**, type "*command*" in the search box, right-click *Command Prompt* in the search results that appear, and click **Run as administrator** from the context menu), navigate to *<EDRSinstallDir>*, and run the following command:

```
edrs.exe install --sudo --localservice
```

Here, *edrs.exe* is the name of the EDRS service executable file. This file may be available with any other name in the format *[Application Name].exe*.

This installs the EDRS Windows service with the display name *UniqueEDRSDisplayName* set to start with Windows, but it doesn't start running right away.

5. Start the EDRS service:

After installing the EDRS service, open the Administrator command prompt, navigate to *<EDRSinstallDir>*, and run the following command:

```
edrs.exe start
```



Ensure that the EDRS service is running before proceeding any further

Ensure that the EDRS service is running in the following way:

1. Run "services.msc" (click **Start** **Run** or **Windows + R** keys together, type "services.msc" in the Run dialog box, and click **OK**). Ensure that the **Status** for the EDRS service instance shows as **Started** and the **Startup Type** is **Automatic**.
2. If you don't see as mentioned in **step 1**, then right click the EDRS service name in the list, which should appear as *UniqueEDRSServiceDisplayName*, and click **Properties** to open its properties window.
3. Select **Automatic** from the **Startup type** drop-down and click the **Start** button under **Service status**.
4. This starts the EDRS service and sets it to start automatically at Windows startup.

▼ If you need to stop or uninstall the EDRS service, click here to learn how...

Stop the EDRS service

1. Open the Administrator command prompt and navigate to *<EDRSinstallDir>*.
2. To stop the installed service, run the following command:

```
edrs.exe stop
```

Uninstall the EDRS service

1. Open the Administrator command prompt and navigate to *<EDRSinstallDir>*.
2. First stop the service by running the following command:

```
edrs.exe stop
```

3. To uninstall the new instance of the service, run command:

```
edrs.exe uninstall --sudo --localservice
```



Please visit [EDRS Configuration Settings](#) to learn more about the various settings mentioned above.

[Return to the top of the page.](#)



[Set up SearchService service \(EDSS\) for single instance of ExamDeveloper](#)



If you do not plan to use EDSS, you can skip to setting up ExamService.



[Set up ExamService for single instance of ExamDeveloper](#)



Configure IIS Web Server



ExamDeveloper setup home page