Post configuration Steps for SAS Cost and Profitability Management 8.3 for TLSv1.2 enabled SQL Server Database (All versions)

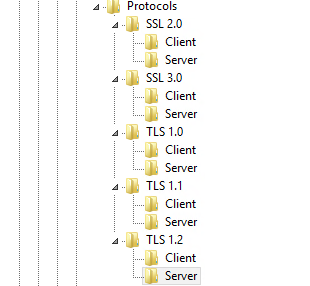
1. Open Windows Registry, and verify if all the protocols (SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2) under Computer> HKEY\_LOCAL\_MACHINE> SYSTEM> CurrentControlSet> Control> SecurityProviders> SCHANNEL> Protocols are available .If not please run the Attached REGISTRY Script.



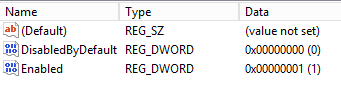
It should be done on all the CPM Server including the Client machines. **PLEASE MAKE SURE TO TAKE A BACKUP OF THE REGISTRY BEFORE MAKING ANY CHANGES**

Following screenshot gives you the details of the reg entries.

Two keys named “Client” and “Server” should be made under each of the protocol having two DWORDs as “DisabledByDefault” and “Enabled”.



And only the key TLS 1.2> Client and TLS 1.2> Server should have value as DisabledByDefault=0 and Enabled=1, rest all should have value have DisabledByDefault=1 and Enabled=0

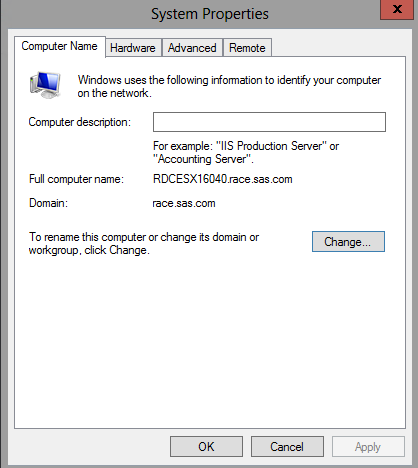


1. Verify Fully Qualified Domain name (FQDN) Requirement:

Verify that your full computer name is FQDN (for example, rdcesx16040.race.sas.com). If not, add the domain name (for example, race.com) to the computer name.

Follow these steps:

* Access the properties panel of your computer (for example, right-click the Computer icon on your desktop and select Properties).
* Click Advanced system settings in the left pane.
* Click the Computer Name tab.
* Click Change.
* Click More.
* Enter your domain name in the Primary DNS suffix of this computer field.
* Click OK and restart the computer.
* Verify that your full computer name is now FQDN.



1. Verify and Apply patches for Microsoft SQL Server according to the version installed (if required)

<https://support.microsoft.com/en-in/help/3135244/tls-1-2-support-for-microsoft-sql-server>

1. Install IIS on the database server if it has not been installed already. The link below provided should guide you for the installation of IIS.

<https://docs.microsoft.com/en-us/iis/install/installing-iis-85/installing-iis-85-on-windows-server-2012-r2>

<https://docs.microsoft.com/en-us/iis/install/installing-iis-7/installing-iis-7-and-above-on-windows-server-2008-or-windows-server-2008-r2>

1. Import the Certificate to Database Server:

* Click Start, Run, and enter inetmgr to open IIS.
* Click ***<server\_name>***.
* Locate and double-click **Server Certificates**.
* In the right pane under Actions, Click on Create Self -Signed Certificate…
* Specify a friendly name as machine name e.g.: rdcesx16040.race.sas.com.
* Select certificate store as Personal.
* Click OK.

1. Grant SQL Server Rights to use the Certificate:

You must provide the SQL Server rights to use the certificate. You can use SQL Server Configuration Manager and Microsoft Management Console to perform this task.

Follow these steps:

* Open SQL Server Configuration Manager.
* Locate and select SQL Server Services in the left pane.
* Select your SQL Server instance in the right pane.
* Right-click SQL Server instance (e.g.: SQL Server (MSSQLSERVER)) and select Properties.
* Copy the account name entry present in the Account Name field.
* Open the Microsoft Management Console (MMC).
* Click File, Add/Remove Snap-in.
* Click Certificates.
* Click Add.
* Select Computer account.
* Click Next.
* Select the local computer option.
* Click Finish.
* Click OK.
* Locate and select the certificate.
* Right-click the certificate, select All Tasks, Manage Private Keys from the context menu.
* Add the copied account name.
* Grant the Read access to the account name.

1. Enable Encryption on Database Server:

* Open SQL Server Configuration Manager.
* Locate and expand SQL Server Network Configuration.
* Right-click on Protocols for <SQL\_Server> and select Properties from the context menu. Click the Certificate tab.
* Select the required certificate from the Certificate drop-down list.
* Click the Flags tab.
* Select Yes for the Forced Encryption option.
* Click OK.
* Restart the SQL Server service.

1. Export the Certificate from Database Server:

(For self-signed certificates) Export the self-signed certificate from the database server so that the CPM client can use it. The CPM Server (client) must trust the certificate that is available on the database server.

Follow these steps:

* Launch Microsoft Management Console (MMC).
* Click File, Add/Remove Snap-in.
* Click Certificates.
* Click Add.
* Select Computer account.
* Click Next.
* Select the local computer option.
* Click Finish.
* Click OK.
* Locate the certificate.
* Right-click the certificate and select All Tasks, Export from the context menu
* Click Next on the Certificate Export Wizard.
* Follow the required selections for Base-64 encoded X.509 (.CER) and specify the location where you want to save the exported file. The location must be accessible to the CPM Server.
* The self-signed certificate is successfully exported
* Now add the certificate to SAS trust store through SAS Deployment Manager and in case of multimachine, Certificate should be added in Compute and Middle Tier.

*Restart SQL server and SSAS (if present) services on Database machine.*

1. Configuration changes in the File System:

***Changes in settings.xml on your Compute tier:***

* File location: $DRIVE:\SAS Config Folder\Lev1\SASApp\ABMServer\config\settings.xml
* Add the below tag above </settings>

*<!--*

*Specify if the Database is TLS 1.2 Enabled for SQL Server database. 1 if Yes 0 if No*

*-->*

*<TLSEnabled>1</TLSEnabled>*

* If Database is TLS enable then the value will be one, otherwise 0.

***Changes in server.xml on your Middle tier:***

* File location $DRIVE:\SAS Config Folder \Lev1\Web\WebAppServer\SASServer9\_1\conf\server.xml
* Add the tag “encrypt=true;TrustServerCertificate=false;” in url field of SQL server resource tag
* It should look like below:

*<Resource auth="Container" driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver" factory="com.sas.vfabrictcsvr.atomikos.BeanFactory" maxPoolSize="100" minPoolSize="10" name="sas/jdbc/abmDS" password="${pw.sas.jdbc.abmDS}" testQuery="SELECT 1" type="com.atomikos.jdbc.nonxa.AtomikosNonXADataSourceBean" uniqueResourceName="sas/jdbc/abmDS" url="jdbc:sqlserver://rdcesx16040.race.sas.com:1433;DatabaseName=cpmdb;encrypt=true;* TrustServerCertificate=false;*" user="cpmuser"/>*

*Restart “SASServer9\_1 webapp server” and “Cost and Profitability Service” (if present) services.*

1. SQL Native Client Driver updates on Compute Tier and Middle Tier:

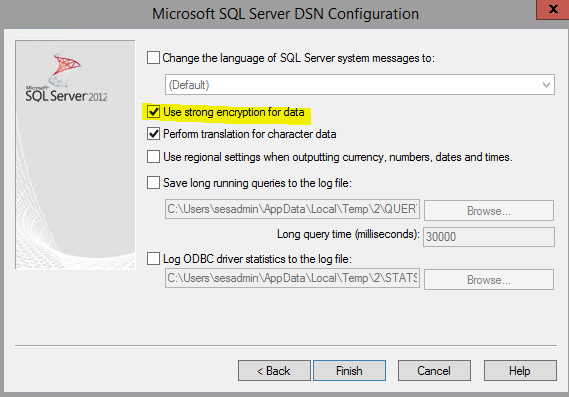
Update the **SQL Server Native Client Driver** on Compute Tier.

Follow the Link below for the updated driver that supports TLSv1.2:

<https://www.microsoft.com/en-us/download/details.aspx?id=56730>

1. Enable the check for Encryption in DSN configuration:

Select the check box for “Use strong encryption for data” in DSN configuration which is used by the application.



1. Update sqljdbc.jar:

Update the sqljdbc.jar to sqljdbc41.jar as it is needed for supporting TLSv1.2



*Restart Object Spawner service.*

1. Add a property to SAS Management Console:

* Launch SAS Management Console and login with an Unrestricted ID e.g.: sasadm@saspw account.
* Add the below mentioned property to Application Management >> Configuration Manager >> Cost and Profitability Mgmt 8.3 >> Properties >> Advanced Tab

Property Name: data.abmserver.db.host.tls.enabled

Property Value: true