CFMS 2.10

INSTALLATION MANUAL

Table of Contents

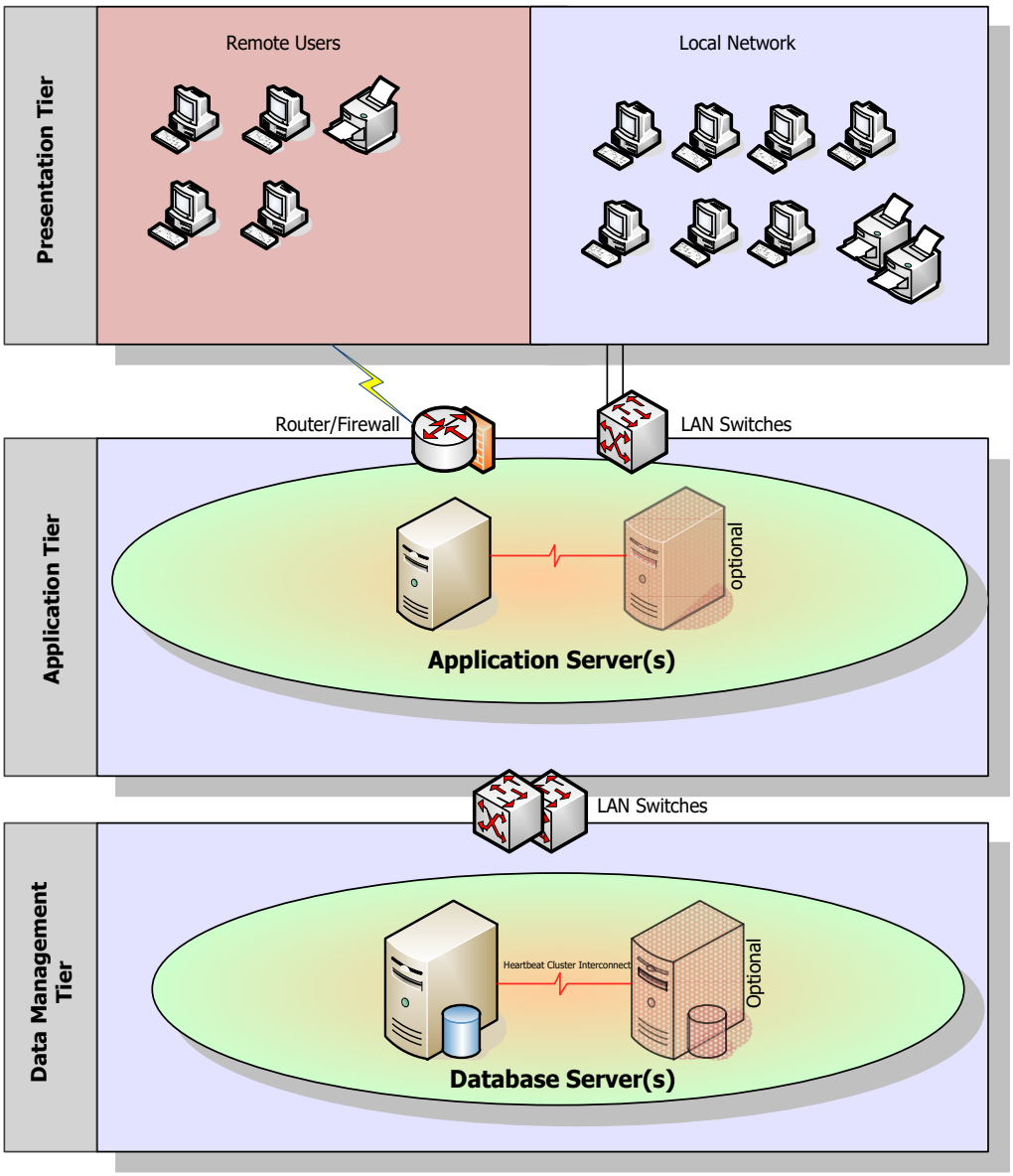
Architecture

Preface

This manual describes how to install, configure and maintain CFMS version 2.10. This document is intended for IT personnel installing or operating CFMS software and covers only these aspects.

CFMS Tiers

CFMS is a three-tier topology application.

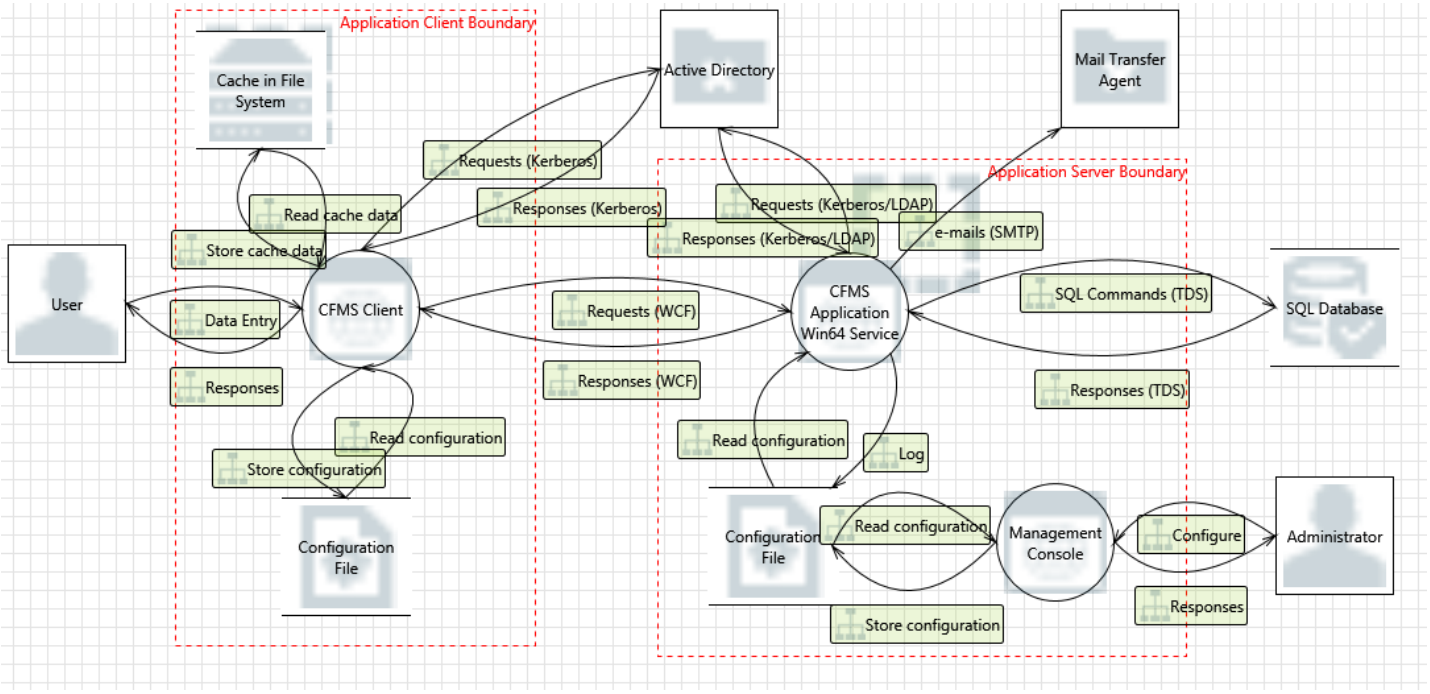


In the **Data Management Tier** the database server is Microsoft SQL Server 2016, 2017 or 2019. Data and backups can be encrypted and sensitive data can be masked.

In the **Application Tier** the application server runs as a Windows service listening for client requests to a TCP/IP socket (TCP port 9090 by default) and queries the database server using the TDS protocol (TCP port 1433 by default). Application server is stateless and can be clustered. Windows server built-in Network Load Balancer or another TCP/IP software or hardware load balancer can be used for load balancing the requests to multiple CFMS application servers and handle failover.

In the **Presentation Tier** the client workstations communicate with the application server. They don't need access to the database.

All communication between the client workstation and the application server and between the application server and the database server can be encrypted with transport level security (TLS).



CFMS Data Flow Diagram for Threat Modelling

Setup

Software and Hardware requirements

Minimum software requirements for **Database Server** are:

- 64 bit SQL Server 2016, 2017 or 2019 Standard or Enterprise Editions.
- 64 bit Windows Server 2012 R2, 2016 or 2019 Standard or Datacenter Editions.

Recommended hardware requirements for **Database Storage (50 users)** are:

- For directly attached storage we recommend NVMe SSD storage using M2, U2 or PCIe interfaces with RAID 1, 5, 6 or 10 and disk controller supporting TRIM. We do not recommend SAS or SATA interfaces, because SAS and SATA are designed for hard disks.
- For storage area network we recommend pure SSD storage interfacing with multipath 10Gbps iSCSI or Fibre Channel.
- Storage must be capable for 6000 IOPS with 4K block size, equally mixed read and write random IOPS with less than 10 msec latency.

  Note that 6000 IOPS for 4K block means throughput 23.44 MB/s but this performance does not guarantee that it is done in 10 msec, it gets the job done in 1 second. To ensure 10 msec latency you need a combination of more throughput and more I/O parallelism. An array of ten common SSD disks in RAID 10 can give these performance figures.

Recommended software and hardware requirements for **Database Server (50 users)** are:

- Windows Server 2016 or 2019 Standard Edition, 64 bit
- SQL Server 2017 or 2019 Standard Edition or Enterprise Edition, 64 bit
- Processor: 8 cores Xeon E5 or Xeon Gold
- Memory: 128 GB DDR4
- Network interface: 2x 10GBps

Minimum software and hardware requirements for **Application Server** are:

- 64 bit Windows Server 2012 R2, 2016 or 2019 Standard or Datacenter Editions
- .Net Framework 4.6.1
- Processor: 2 Cores Xeon E5 or E6 v1, v2, v3 or v4 Family 2Ghz
- Memory: 8 GB RAM
- Storage: 20 GB free space

Recommended software and hardware requirements for **two (2) redundant Application Servers (50 Users)** are 2x:

- Windows Server 2016 or 2019 Standard Edition, 64 bit
- .Net Framework 4.8
- Processor: 8 cores Xeon E5 or Xeon Gold
- Memory: 64 GB DDR4
- Storage: 200 GB
- Network interface: 2x 10GB NIC

For production environment we recommend setting up two application servers using windows NLB for failover and load balancing.

Minimum software and hardware requirements for **Client** are:

- 64 bit Windows 8.1 or Windows 10 or Windows Server 2012 or later using Remote Desktop Connection.
- .Net Framework 4.6.1
- Screen resolution: 1024 x 768 pixels
- Processor: 4 logical CPUs, Core i3 3GHz, Core i5/i7/i9 2.5 GHz (4th to 10th Generation)
- Memory: 4 GB DDR3
- Storage: 20 GB free space
- Video Card: 512 MB GDDR2

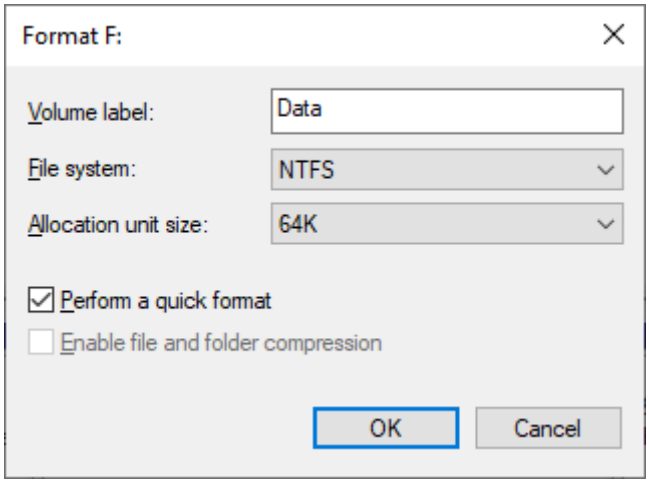
Recommended software and hardware requirements for **Client** are:

- 64 bit Windows 10 version 1809 or later

- .Net Framework 4.8
- Screen resolution: 1920 x 1080 pixels or bigger
- Processor: 4 physical CPUs, Core i5/i7/i9 2.5 GHz (6th to 10th Generation)
- Memory: 8 GB DDR4/LPDDR3
- Storage: 120 GB SSD
- Video Card: 1 GB GDDR3

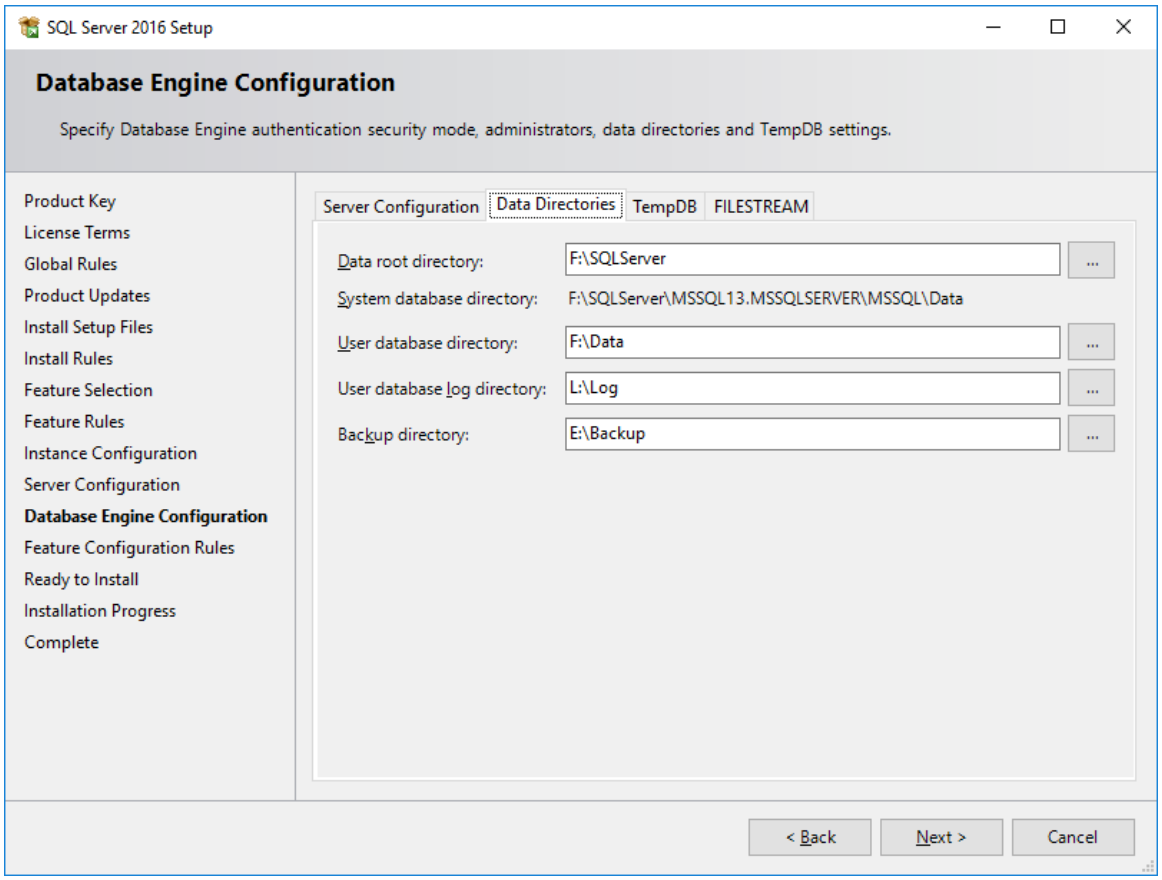
Database server setup

- SQL Server Standard or Enterprise Edition 2016 with SP2, 2017 and 2019 are required.
- The latest Service Pack for SQL Server 2016 and the latest Cumulative Update for your version must be installed.
- The database server installation, data, log, tempdb and backup volumes must have starting offset aligned to integer multiple of SSD block of pages. The smallest unit of an SSD is a page, usually 512KB = 4KB cell size x 128 pages. There is big performance hit if this offset partitioning parameter is not correct, and the configuration way is different than that of hard disks who require sector size (512 bytes) multiple. For SSD partitioned with diskpart the correct command for 512KB block of pages is CREATE PARTITION PRIMARY ALIGN=524288.
- If you are using an external storage consider having your TempDB at a local SSD disk. Since TempDB is recreated each time SQL Server you cannot have data loss. What you get is better performance and less traffic for your storage. If you cannot do that, then allocate the fastest disk on your storage for TempDB.



- The database server installation, data, log, tempdb and backup volumes must be formatted with NTFS file system and **64KB allocation unit size**. The system C: volume must be formatted with the default 4K allocation unit size.

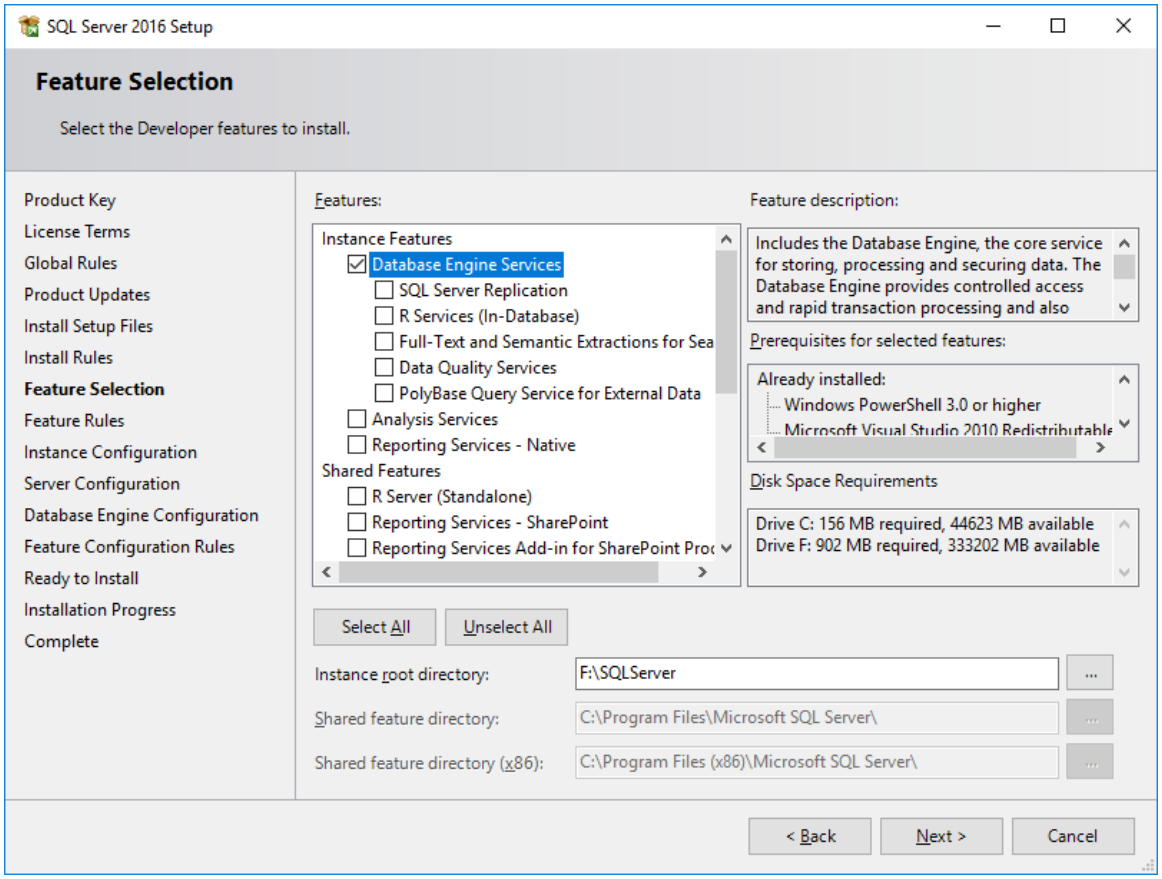
- You need 6 logical disks. C: for system, L: for logs, T: for tempdb, F: for SQL server installation (master and msdb), G: for data and H: for backup. Create folders for each logical disk L:\Logs, T:\TempDB, F:\SQLServer, G:\Data, H:\Backup. The folders are required.



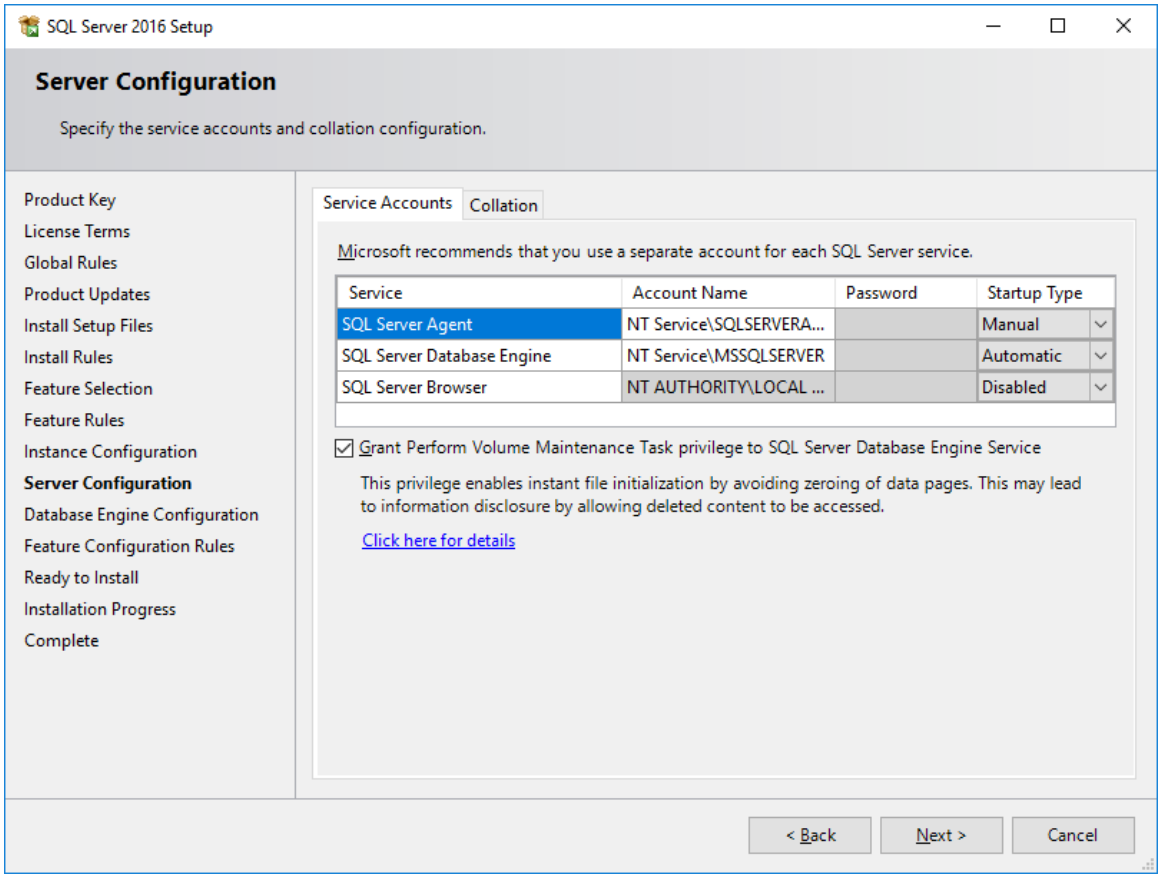
- Exclude the database folders from the antivirus. Please read Microsoft KB254649. You can exclude the database files extensions from MS antivirus with the following Powershell command:

```
Add-MpPreference -ExclusionExtension "*.mdf","*.ldf","*.ndf","*.bak"
```

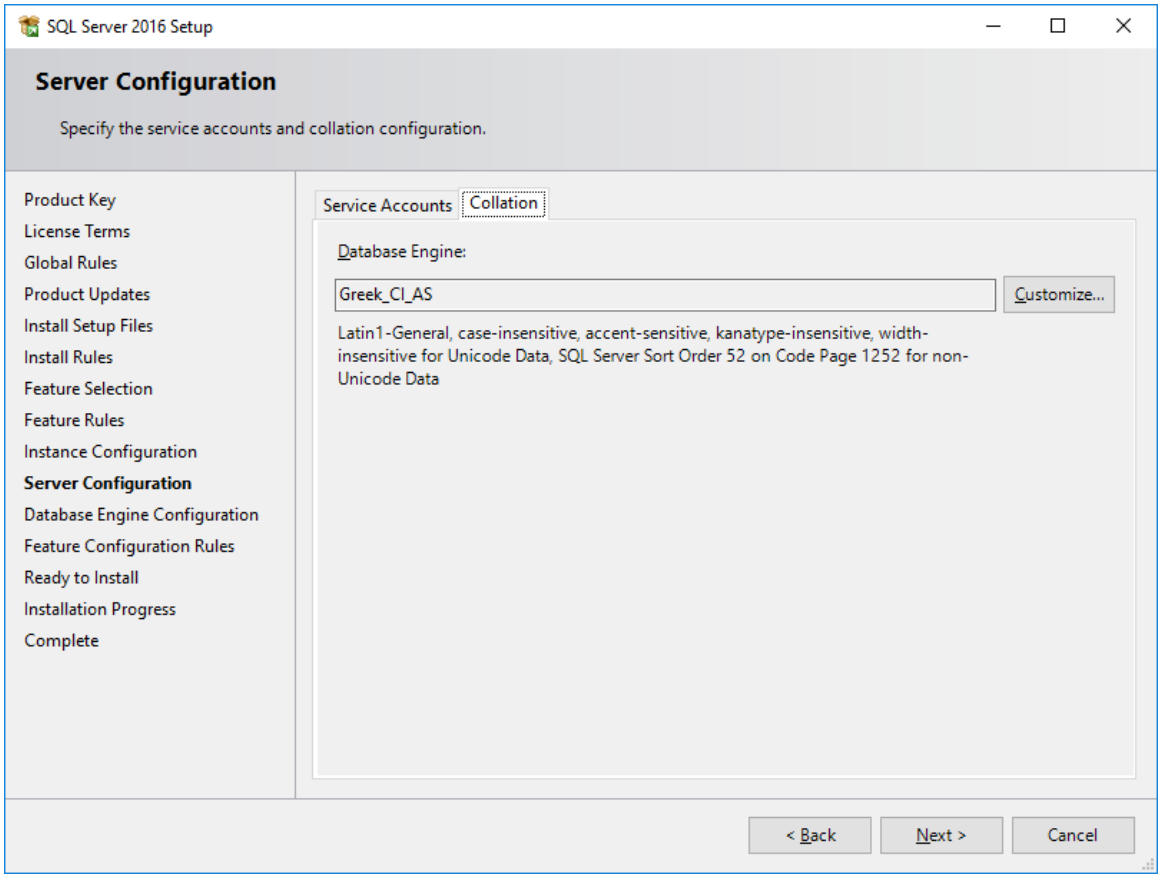
- Install **only** the "Database Engine Services", nothing else. You must **not** install the database server on the C: drive.



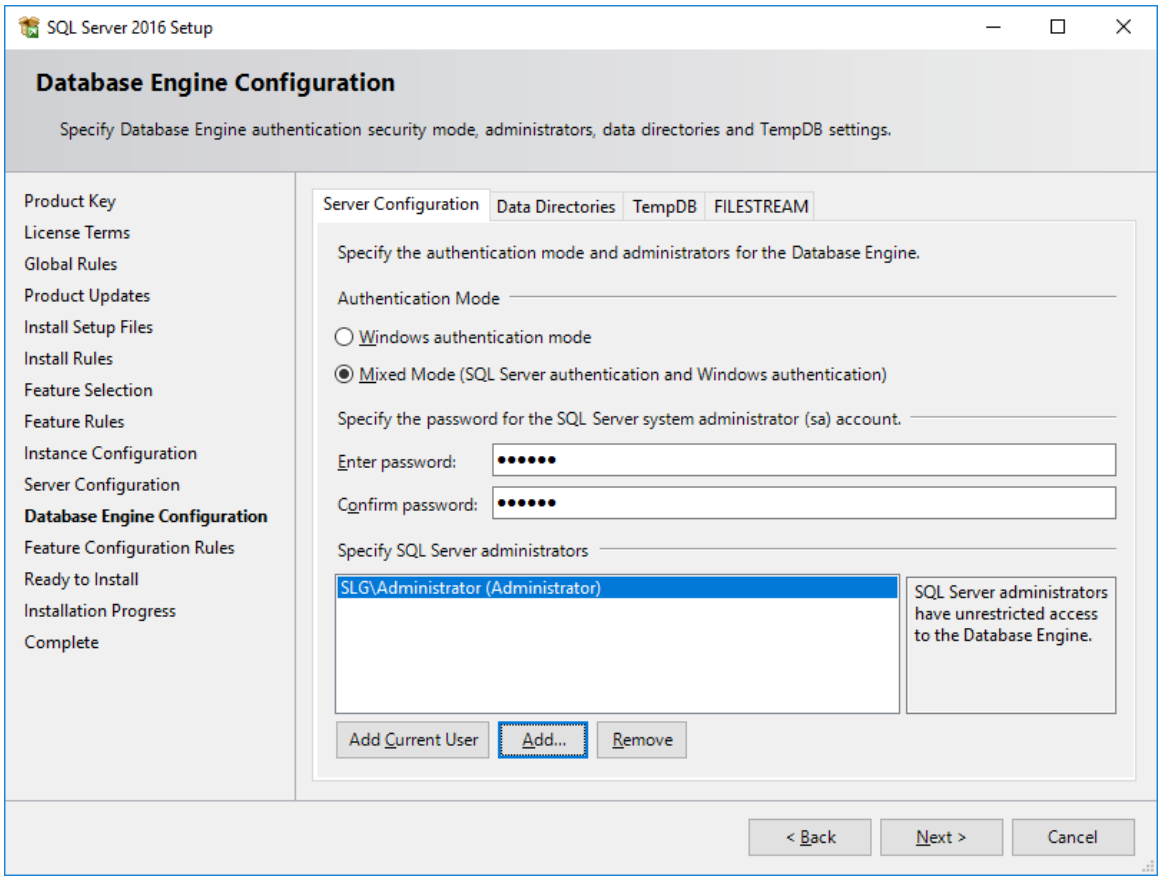
- Select the option "Grant Perform Volume Maintenance Task privilege to SQL Server Database Engine Service".



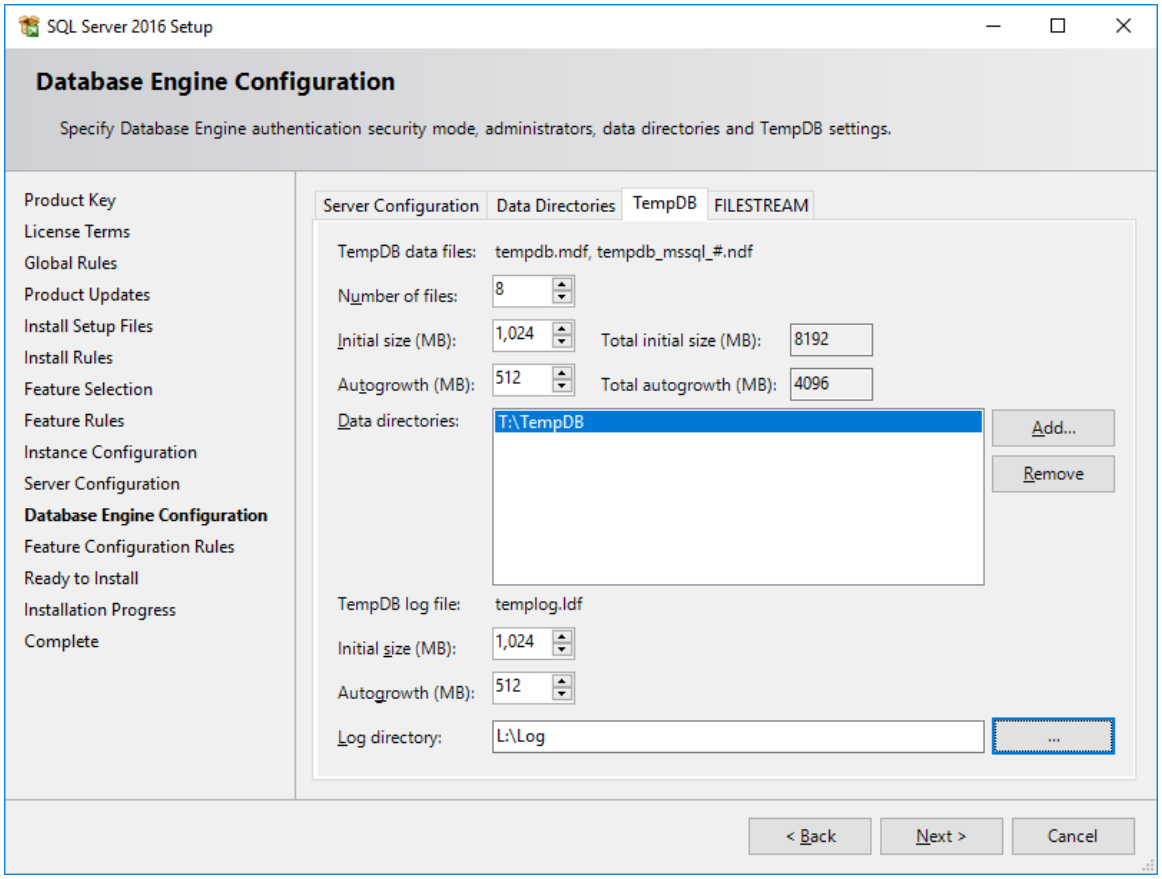
- In the "Collation" the SQL Server code page must be case insensitive (e.g. English_CI, Greek_CI_AI).



- You must select Mixed Mode to enable SQL server authentication.



- TempDB number of files must be between 2 and 8 depending on the number of cores and the CPU NUMA Cores. See Microsoft KB2154845. Set the Log directory of tempdb to the tempdb folder T:\TempDB.



- In the firewall open the TCP port 1433.
- Set the MAXDOP parameter between 2 and 8 according to KB2806535 (number of cores in a physical CPU NUMA node).
- Set the Cost Threshold for Parallelism to 50 as a good initial value, since the default value 5 is too small.
- Set the Max Memory according to KB2663912. Set it up to 90% of available memory and you must have at least 4GB free.
- In the database server we must disable screen saver and in the "Power Options" we select "High Performance" and not "Balanced".
- In the database server you must run as administrator the "OneStopScript.sql". The script enables the CLR integration option, creates a login named "clroot" for CFMS (the initial password is also "clroot" please change that), grants the "clroot" login rights to view the

database server state, load and trust the Singular Logic certificate when running code in the database.

The expected output of the script "OneStopScript.sql" follows.

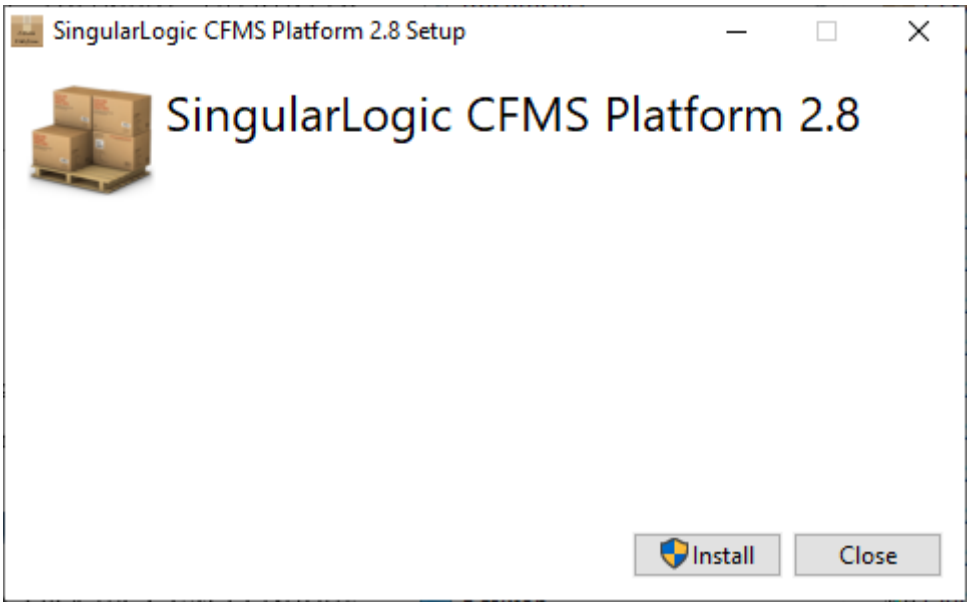
```
Server [CFMSDB] initialized for CFMS.

Database [CFMS] initialized for CFMS.

Optimizing database...

OK
```

The script requires that there are no other users connected in the same database. Please close any other sessions to this database and switch the output to the "Messages" tab. If you forget a tab connected to the database the script will wait forever to close it.

Application server setup

Double click the CFMS Platform Setup 2.10 or run:

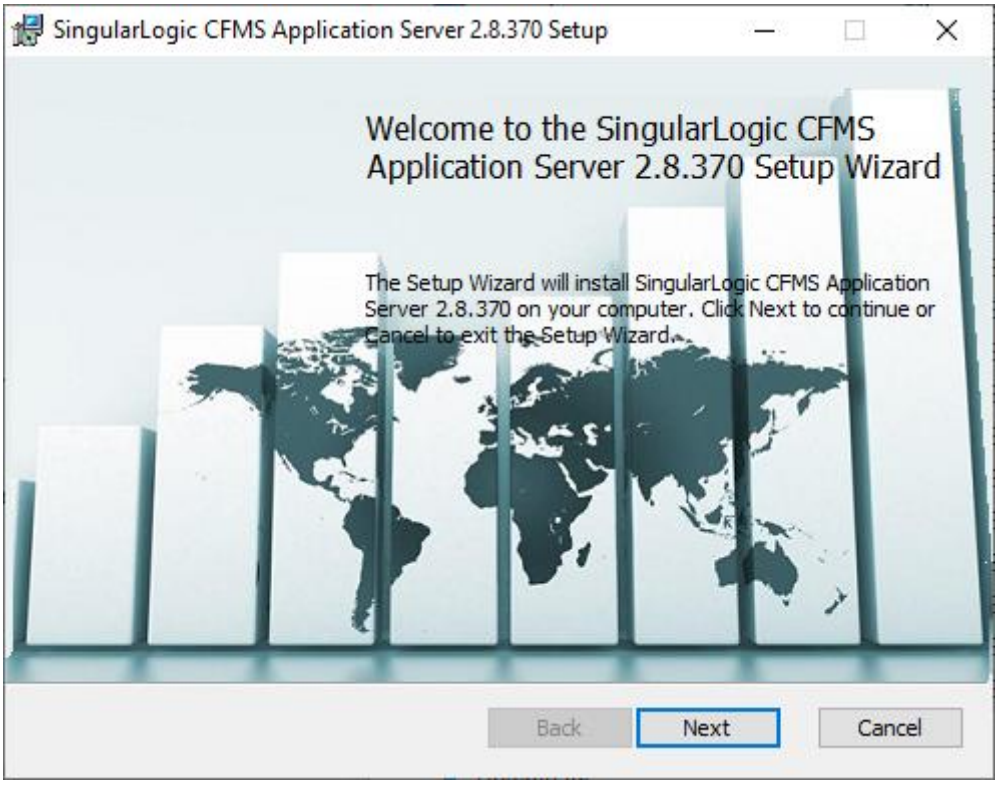
```
CFMSPlatformSetup-2.10_3.exe
```



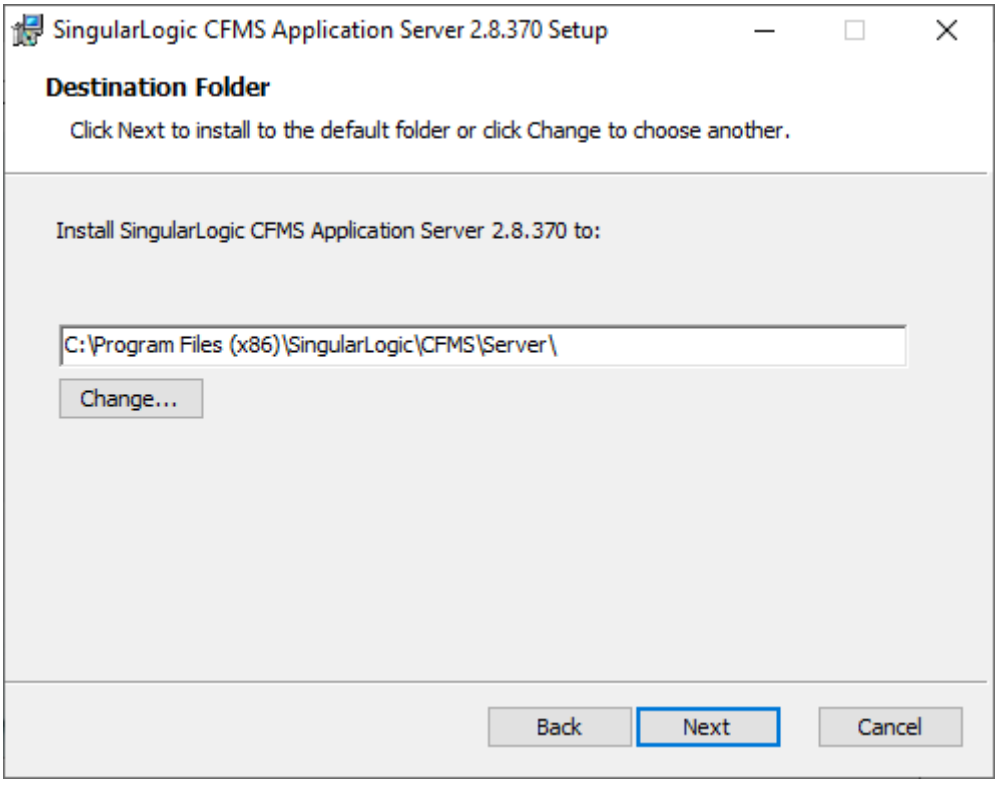
Press the Install button.

Double click the CFMS server Windows Installer Package or run:

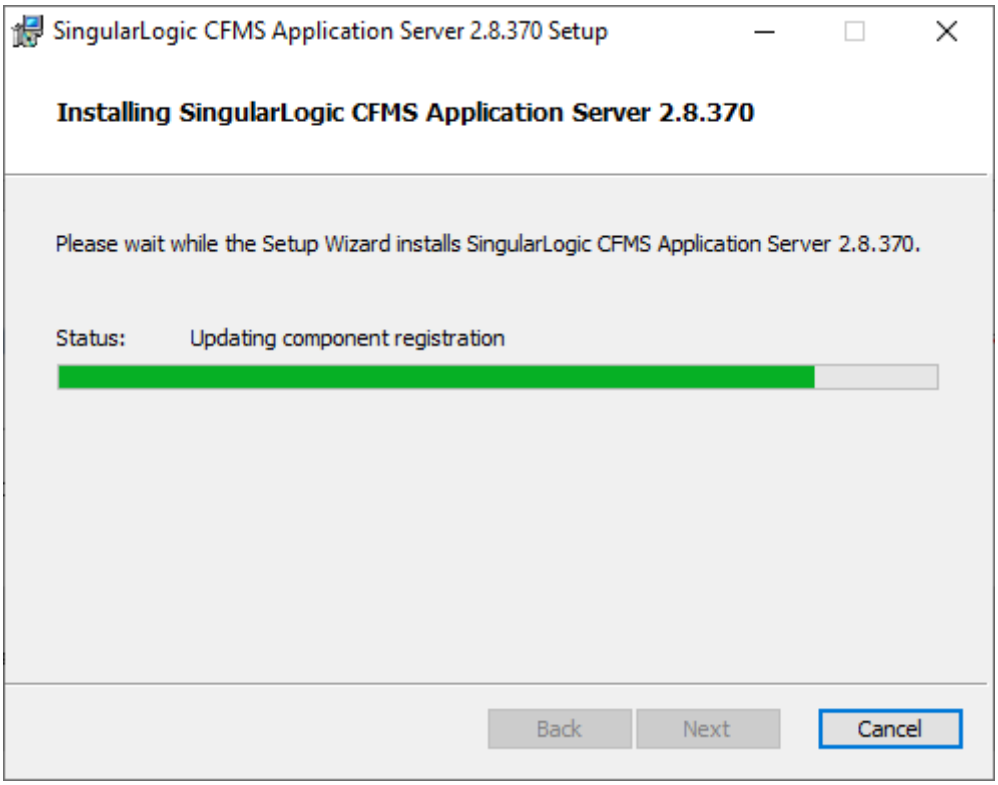
```
msiexec /i CFMS-2.10.410.2-server.msi
```

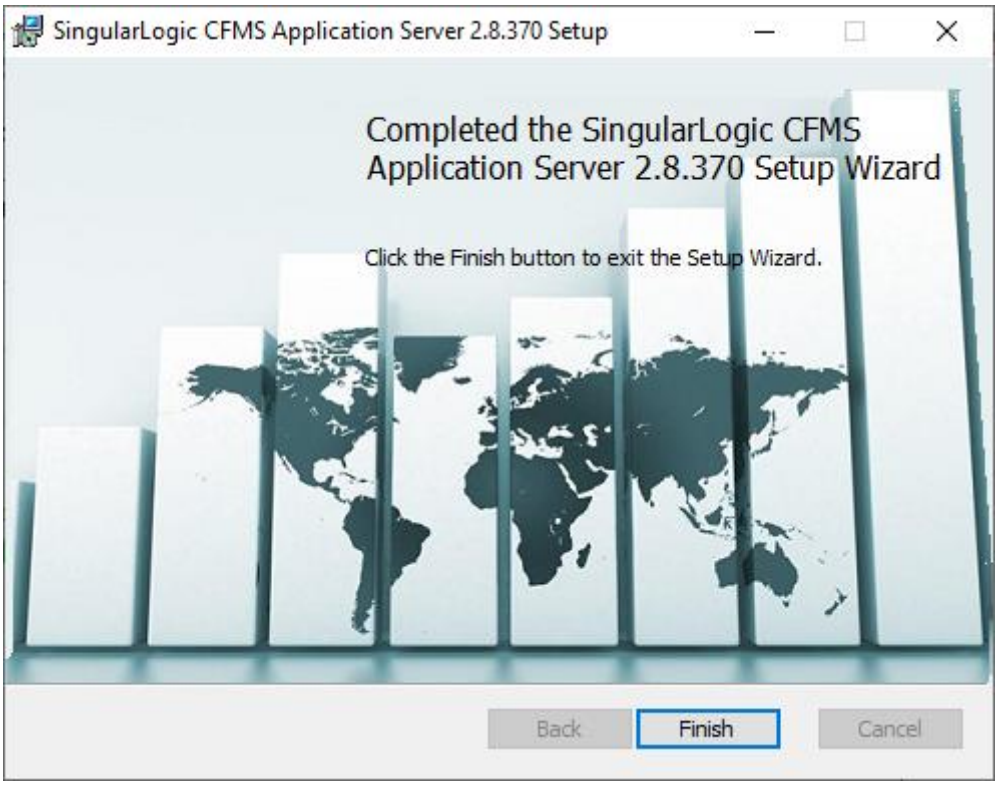


Press Next and select the Destination Folder.



Keep pressing the Next, Install and Finish buttons.



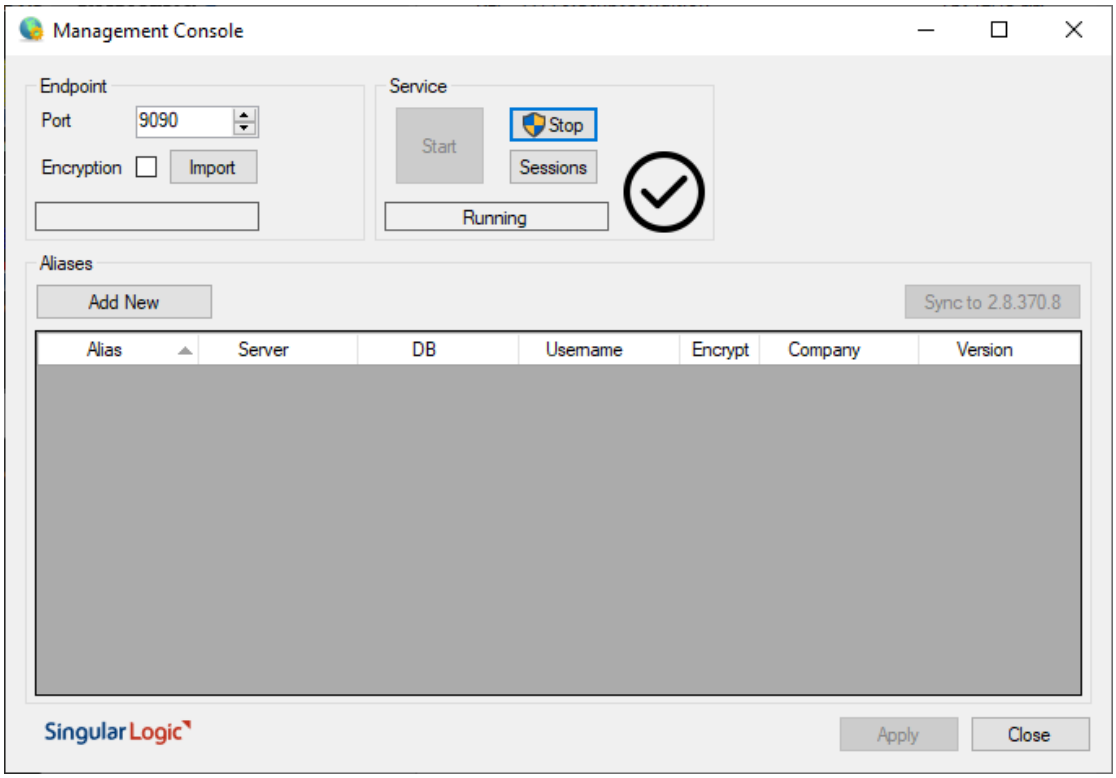


After that you need to run the CFMS Management Console:

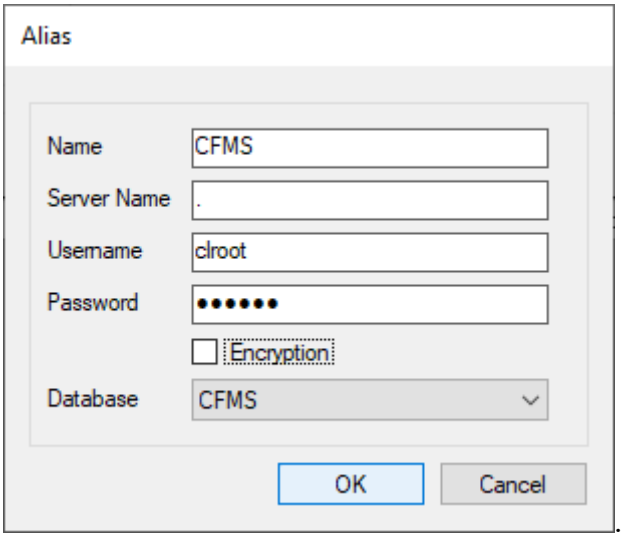
Double click on the CFMS "Management Console" icon on your Desktop.

Or run the following command:

```
%ProgramFiles%\SingularLogic\CFMS\Server\CFMS.ManagementConsole.exe
```



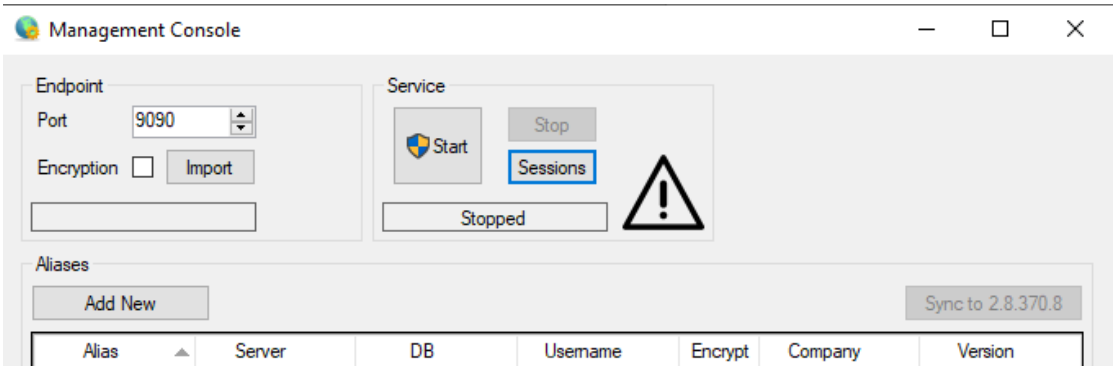
Press the "Add New" button to setup the database connection Alias



Fill the form with the database connection parameters, press the OK button to close the form. Press the Apply button to save the configuration.

You must use the "clroot" user created from OneStopScript (default password is "clroot"). The selected Alias Name must be given to clients.

From the Management console stop and restart the application server by pression the "Start" button.



You can also stop and start the application server using the local services in the control panel or the command:

```
net stop CFMS
net start CFMS
```

The status of the server is visible in the CFMS Management Console.

If you want to apply configuration changes and the service is running, stop it first using the "Stop" button in the CFMS Management Console and then press the "Start" button to restart.

Client setup

Double click the CFMS Platform Setup 2.10 or run:

```
CFMSPlatformSetup-2.10_3.exe
```

Click the Install button, wait for it to display "Installation Successfully Completed" and press the "Close" button.

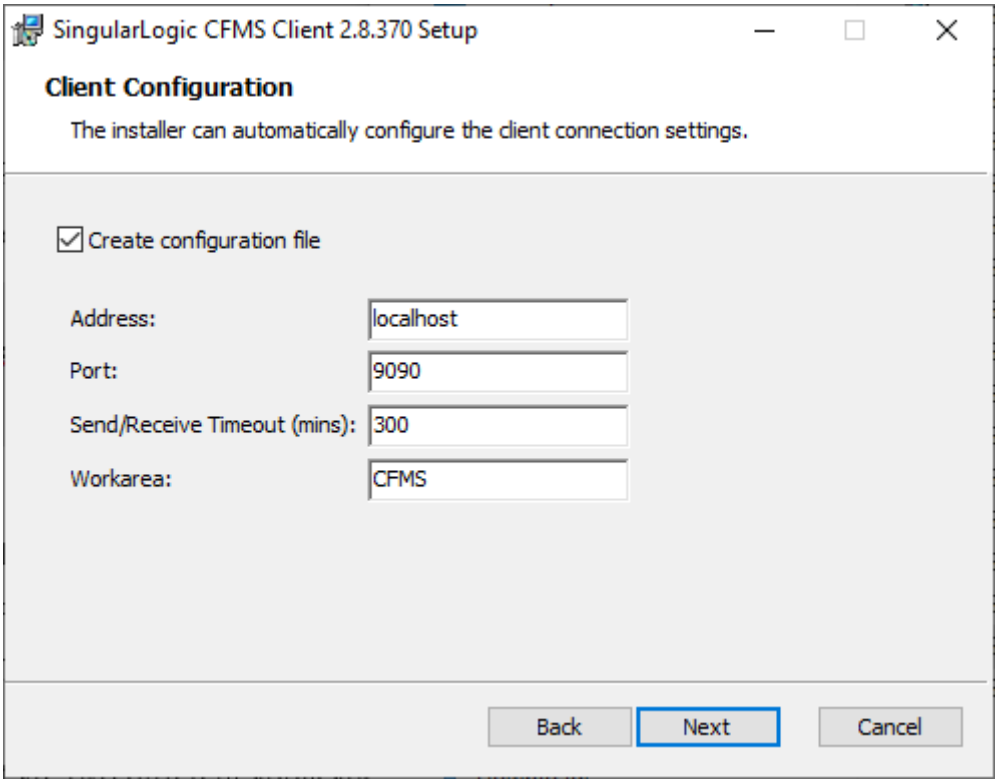
Double click the CFMS client Windows Installer Package or run:

```
msiexec /i CFMS-2.10.410.2-client.msi
```

Press Next. Select the destination folder where the application files will be installed. If another version of CFMS Client has already been installed in the same folder, it will be replaced with the version you are currently installing.

NOTE: If CFMS Server is also installed on the same machine, make sure not to select the same destination folder for CFMS Client. CFMS Server and CFMS Client are two different applications and they should not share the same installation folder.

Press the Next button to continue.



In the Client Configuration form:

If you chose to install CFMS Client in an empty folder, then check 'Create configuration file' and fill in the following information:

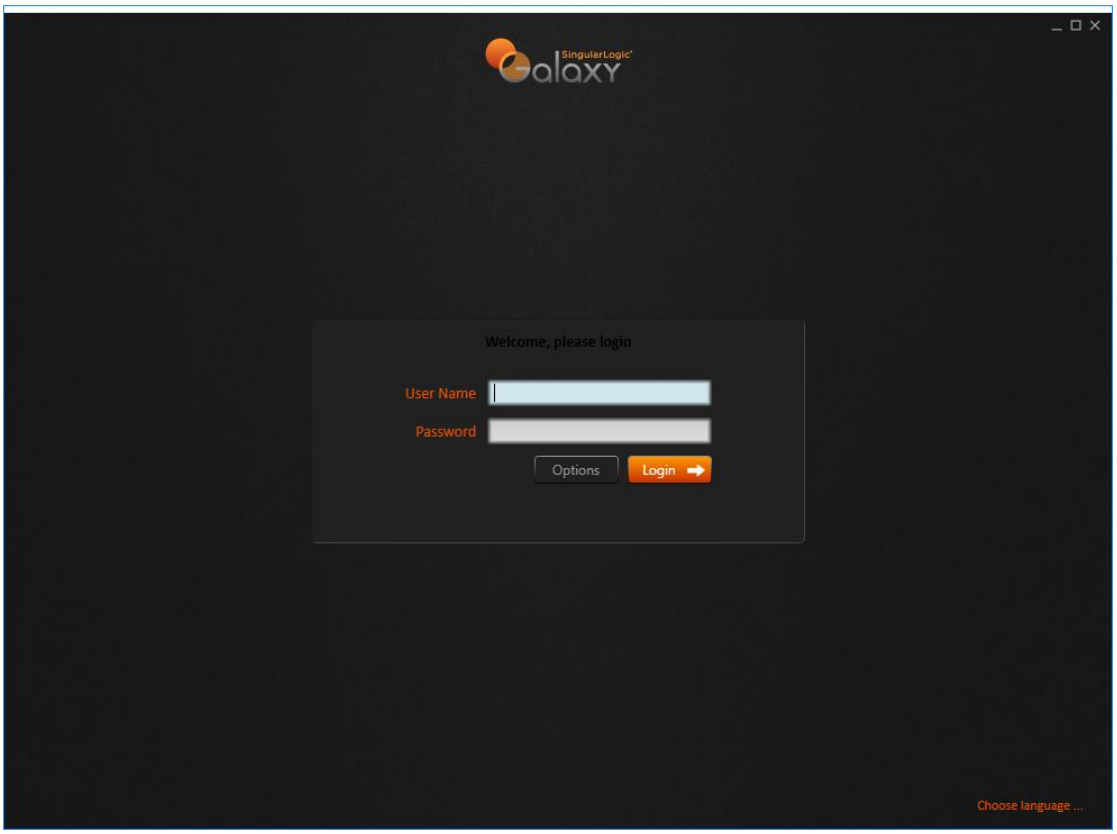
Address	The host name or IP address of the CFMS application server to use.
Port	The port on which the application server is listening to client connections. 9090 is the default port.
Send/Receive Timeout (mins)	Leave the default value (300).
Workarea	The database "alias name" exposed by the application server.

If you chose to install CFMS Client on top of a previous client installation, leave the 'Create configuration file' unchecked to keep the existing configuration.

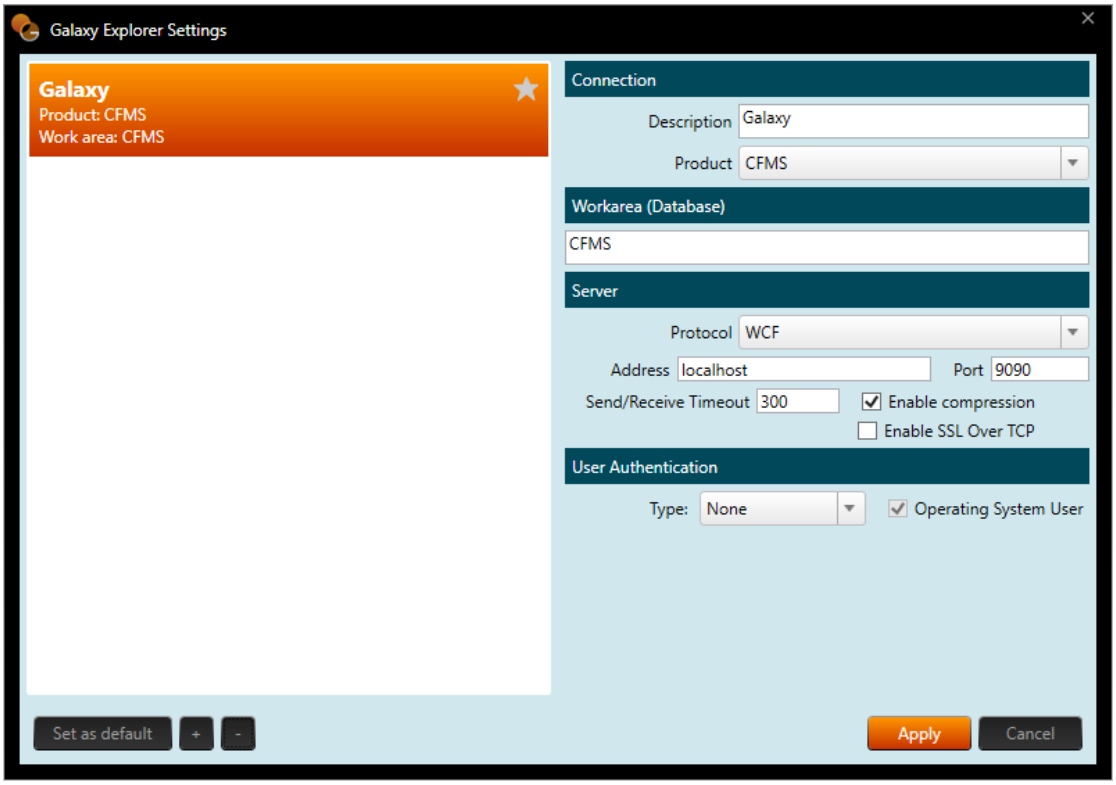
Keep pressing the Next and Finish button to continue.

Start CFMS Client (Galaxy Explorer) from the icon on your Desktop.

Press the Options button.



Review or change the configuration settings and press Apply.



The protocol settings must match the application server settings. In Address is the host name or IP address of the application server.

Press the "Set as default" button if a star is not visible in the alias list of the Galaxy Explorer Settings.

Batch Client Setup

You can automate the client upgrade using a batch file that runs a silent setup:

```
start /wait \\file-server\share\CFMSPlatformSetup-2.10_3.exe /quiet
/norestart

start /wait msiexec /i \\file-server\share\CFMS-2.10.410.2-client.msi
/quiet /norestart
```

Installer package parameters

The client msi installer package supports the following optional parameters:

- **HOSTADDRESS**

Is the application server IP network address or host name.

- **HOSTPORT**

The TCP port that the application server is listening for client requests. The default value is 9090.

- **SRTIMEOUT**

The Send/Receive Timeout in minutes.

- **WORKAREA**

The workarea (alias) of the application server.

- **AUTOCONFIG**

Set to 1 when upgrading an existing installation.

Set to 0 for installation in new directory.

- **INSTALLDIR**

The application installation folder. The default value is %ProgramFiles%\SingularLogic\CFMS\Client.

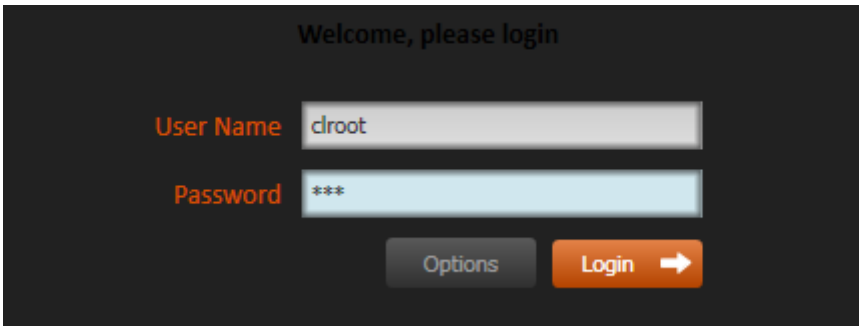
The server msi installer package supports the INSTALLDIR parameter only.

Create application database from scratch

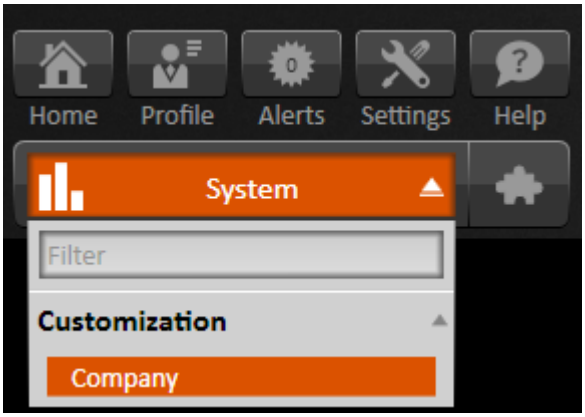
From CFMS Management Console create an alias to the new database, select it and press the "Sync" button.

Wait until the database schema creation is completed.

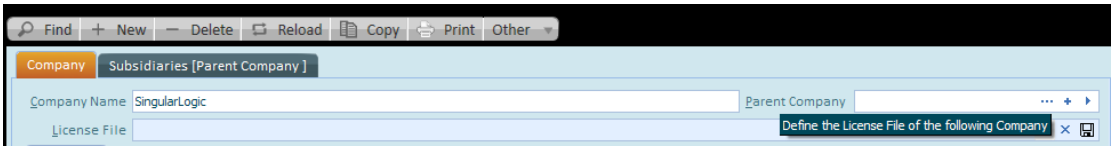
Star the CFML Client, for User Name enter "clroot", for Password enter "qaz". These are the default credentials after a database initialization.



From the system menu select the Company entry and press the "+ New" button.



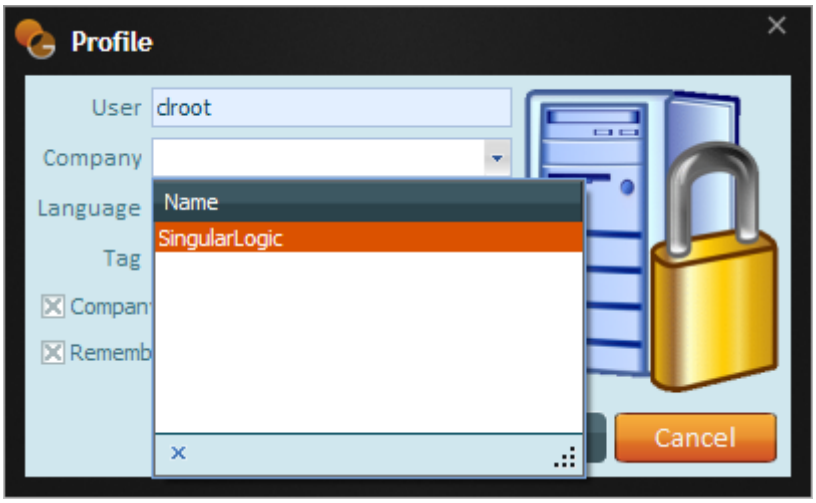
Enter a code for your company name with Latin characters without spaces, then press the "…" button and select the license file.



Press "Save Data".

Press the "Profile" button on the top after the "Home" button.

Select the created Company and press "OK".



The default user "clroot" is able to create users and grant permissions.

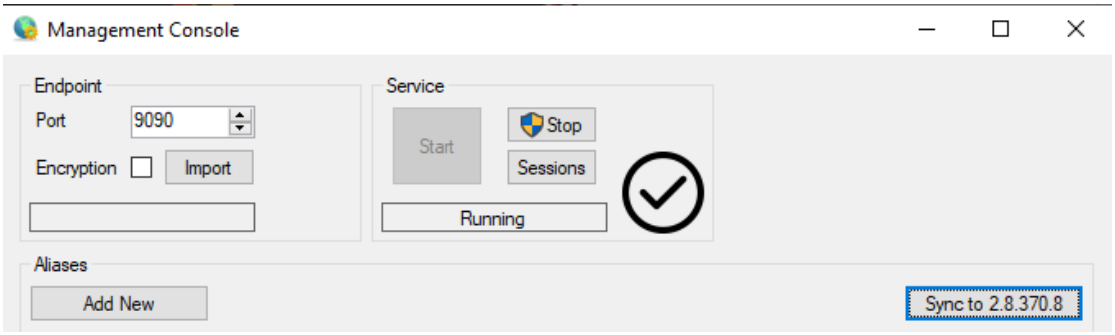
Application upgrade

Backup the database and stop the application server(s).

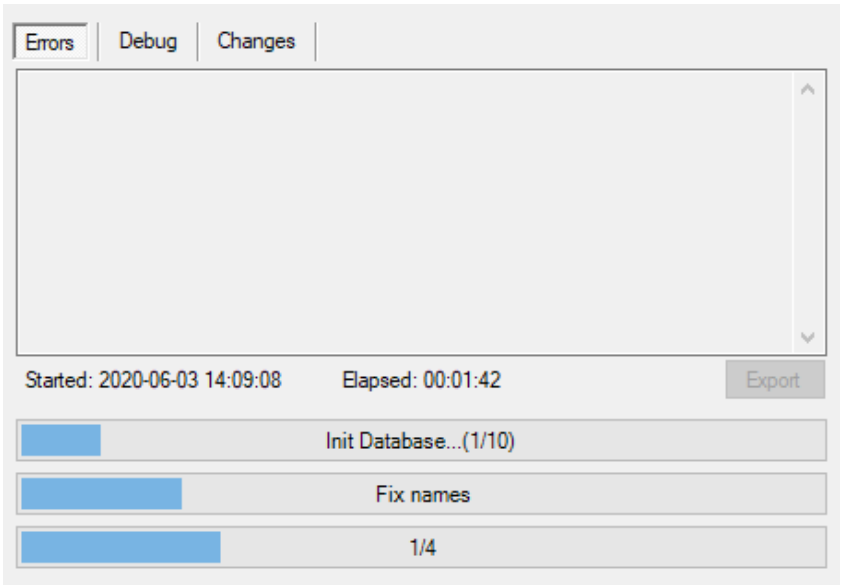
Deploy the new server version in the application servers.

From an application server, start the CFMS Management console.

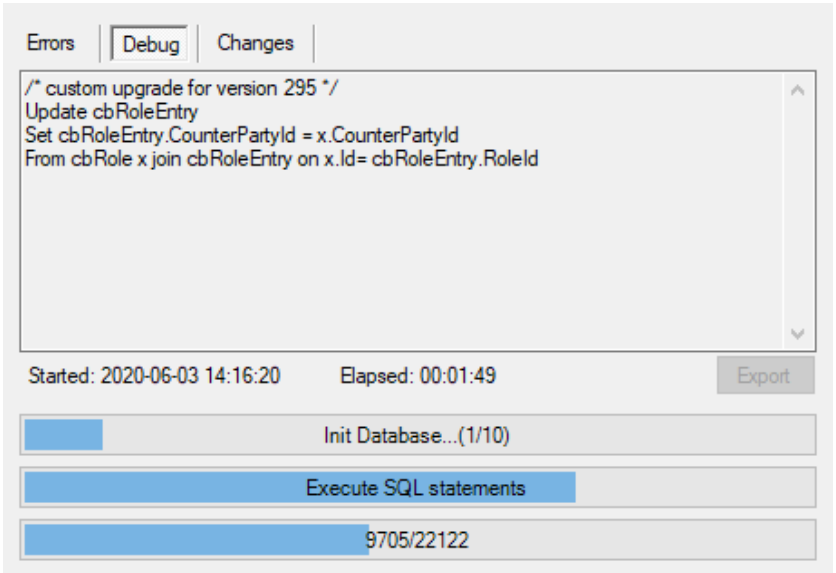
Select the alias of the database to upgrade and press the "Sync to 2.10.410.x" button.



Double click on the alias in the "Aliases" list to view the upgrade process.



The "Errors" tab displays errors that happened during the upgrade, you can press the "Export" button to save the error list.



The “Debug” tab displays the current executing command, you can press the “Export” button to save the top slowest upgrade SQL statements in the process.

The “Changes” tab displays changes in the database schema, you can press the “Export” button to save the schema changes.

Configuration

Encrypted Transport (TLS)

We highly recommend to encrypt the transport between clients and the application server and the transport between application server and database server. CFMS can use TLS protocol version 1.0, 1.1 or 1.2, the exact protocol is defined by the operating system. CFMS is using only the configured Schannel protocols and crypto algorithms specified in the Windows Registry.

You need a certificate authority that is trusted from the server and all clients; otherwise the X.509 certificate of the certificate authority that signs the server certificate must be imported in the Trusted Root Certification Authorities store of all the CFMS clients and servers machines.

Generate two PKCS #12 file (file extension is either .pfx or .p12) protected with a password that includes the generated private key and its X.509 certificate signed from your certificate authority. Both certificates require enhanced key usage to be "Server Authentication". The first certificate is for the application server and must be issued for the Subject of hostnames and/or IP address that clients are using to connect to the application server. The second certificate is for the database server and must be issued for the Subject of hostname and/or IP that the application server uses to connect to the database server.

Instructions for creating a Certificate Authority on Windows Server are on Microsoft Windows Server documentation.

Application Server Configuration

Run the CFMS.ManagementConsole.exe.

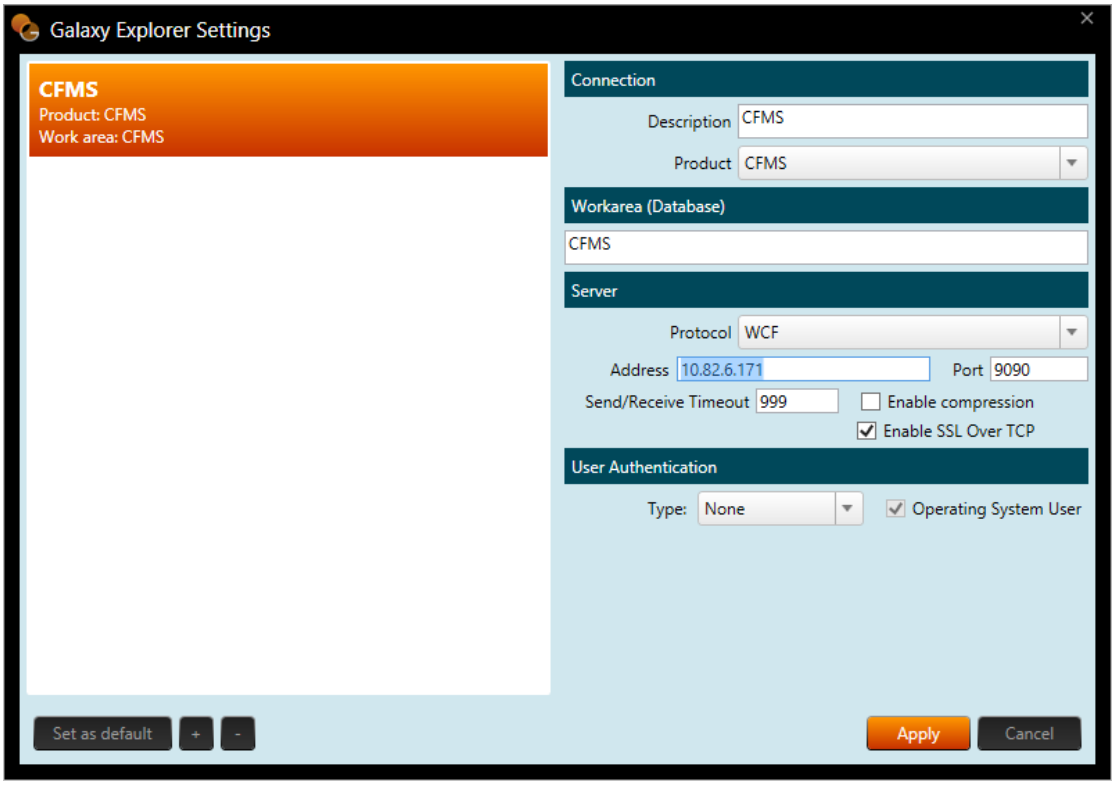
Press the "Import" button and select the application server certificate (PKCS #12 file with extension .pfx or .p12), enter the password used to protect the key. Ensure that the "Encryption" checkbox is active.

Double click on the database alias. Enable the "Encryption" checkbox.

Restart the application server.

Client Configuration

Run the CFMS client. Click "Options" and activate the checkbox *"Enable SSL Over TCP"* option.



SQL Server configuration

Complete configuration instructions are in the SQL Server documentation https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-ver15.

You must add the issued certificate into the computer account (and not the user account). SQL Server documentation describes a process that in the MMC console you can add the Computer Certificates snap-in and import the issued certificate into the computer account.

From the SQL Server Configuration Manager, select Properties from the Protocols for MSSSQLSERVER. On the Flags tab select Yes in the *"Force Encryption"*.

Encrypted Storage

Backup Encryption

It is recommended to encrypt your backups using at least the 128 bit AES cypher.

The backup encryption is asymmetric i.e. there is a different key for backup encryption and restore decryption.

Implementation instructions are in the SQL Server Documentation https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/backup-encryption?view=sql-server-ver15

Data Encryption

It is recommended to encrypt your data at rest. Transparent Data Encryption encrypts the sensitive data in the database and protect the keys that are used to encrypt the data with a certificate. This prevents anyone without the keys from using the data.

Implementation instructions are in the SQL Server Documentation https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15

Data Masking

Dynamic data masking helps prevent unauthorized access to sensitive data. Dynamic data masking can be configured on the database to hide sensitive data in the result sets of queries over designated database fields, while the data in the database is not changed.

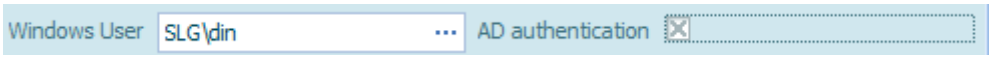
Implementation instructions are in the SQL Server Documentation https://docs.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-ver15

Authentication

Active directory integration

You can setup the CFMS users to login with their active directory credentials by activating the *"AD authentication"* and setting the *"Windows User"* in their principal entry.

You must set the Windows user in the DOMAIN\username format and activate the *"AD authentication"* in the user form.



The user enters for Login Name DOMAIN\username and for Password his active directory domain password.

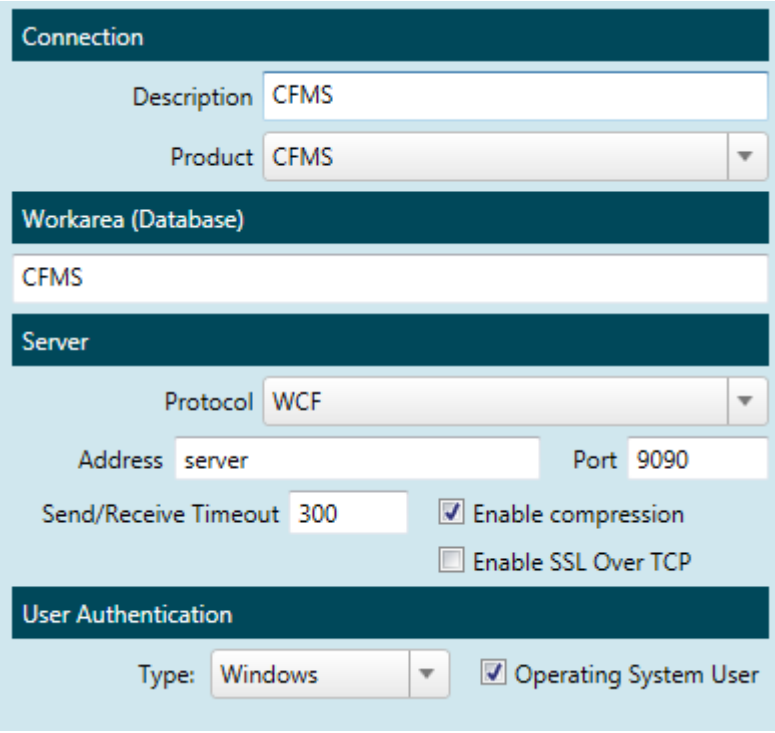
CFMS actually is using Kerberos to authenticate the user by connecting to its LDAP entry on the active directory.

Single Sign-On

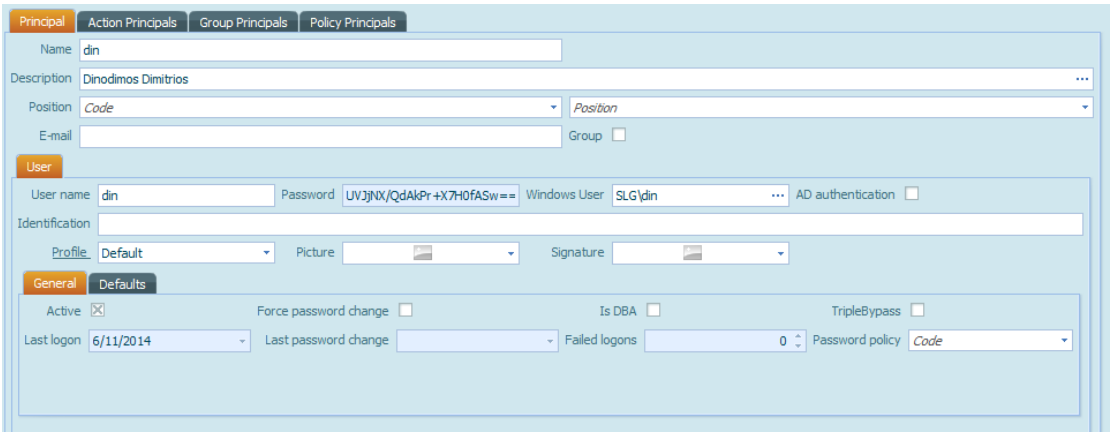
By joining all the client machines and the application servers to the same windows domain you can have single sign on. The end users from the client workstations provide their credentials when login-in to their computers; their credentials are transferred to the application server automatically when they don't provide credentials in the CFMS client login screen. (They must leave both the username and the password empty and press the login button).

To enable single sign-on follow these steps:

1. You must set the *"User Authentication"* type to *"Windows"* in the client connection options for all client workstations.



2. You must set the *"Windows User"* for each CFMS principal in the DOMAIN\username format.



Fail Over and Load Balancing

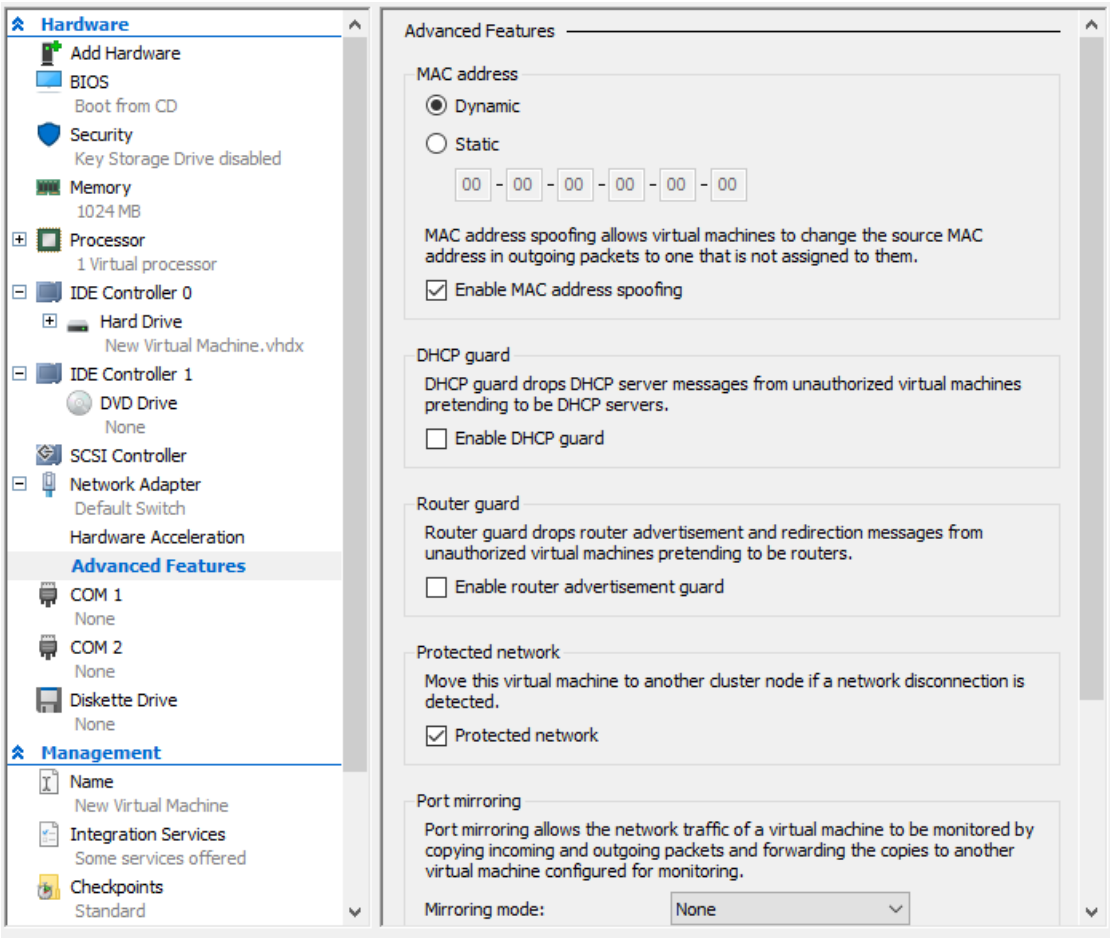
Requirements

Both application servers must be on the same subnet using a switch that supports IGMP.

Both application server and the virtual IP address must be static addresses in the same subnet.

Hyper-V

To run NLB on virtual machines under the Hyper-V hypervisor you must enable on all application servers the "Enable MAC address spoofing" in the Hardware / Network Adapter / Advanced Features.



Setup

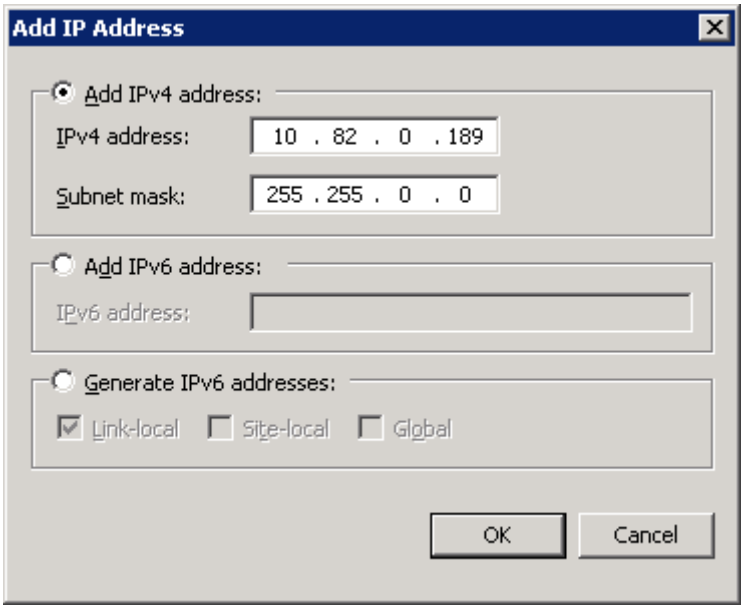
For out test setup we are using the following configuration:

- Network subnet 10.82.0.0/16 (i.e. old style mask 255.255.0.0)
- Cluster Virtual IP address 10.82.0.189
- CFMSAPP1 IP address 10.82.0.190
- CFMSAPP2 IP address 10.82.0.191

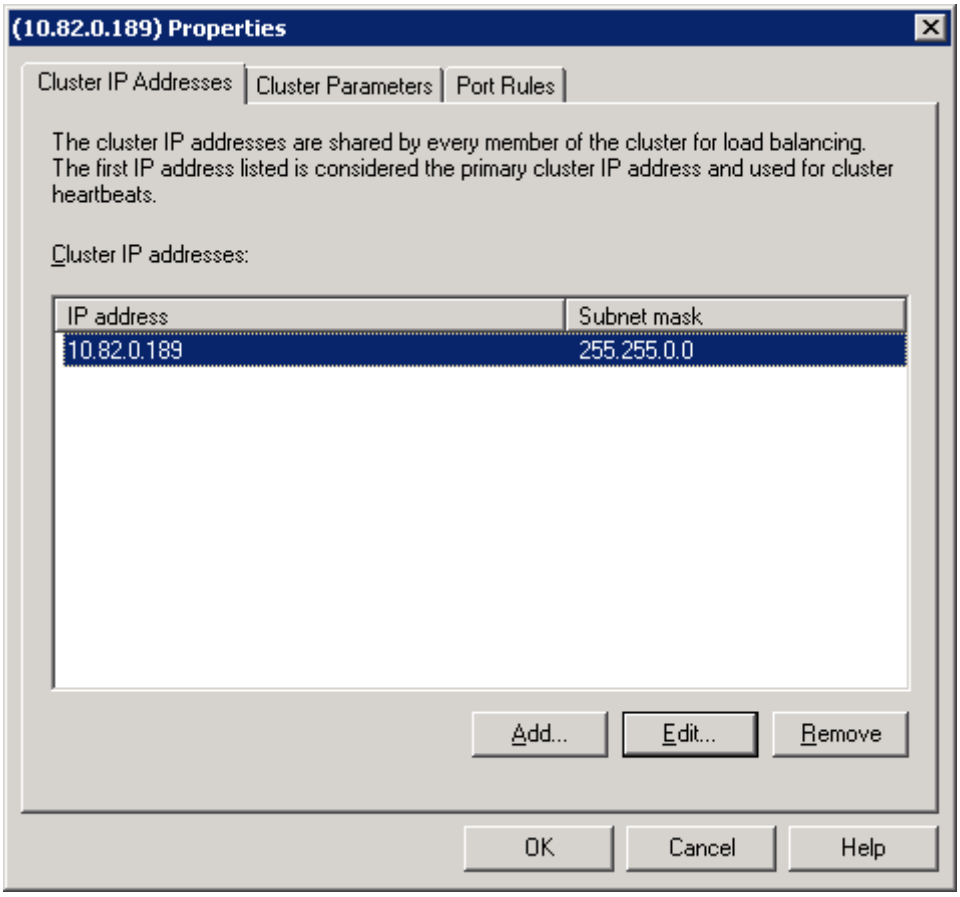
First of all configure the subnet and static address in the network adapter of the two application servers.

Start NLB manager to create the NLB cluster.

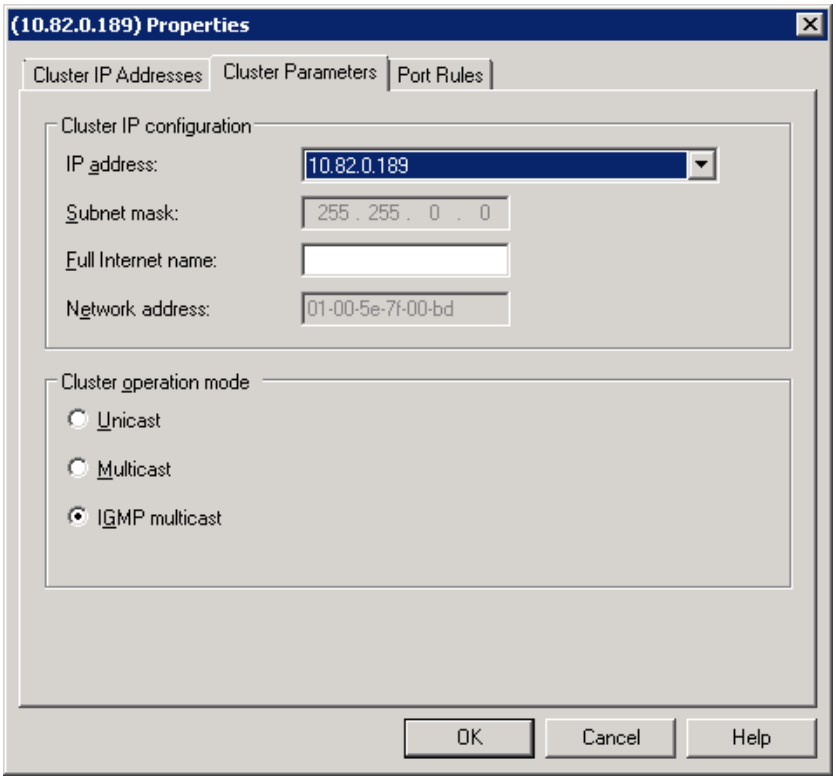
Add a cluster IP address using the virtual IP address and network subnet:



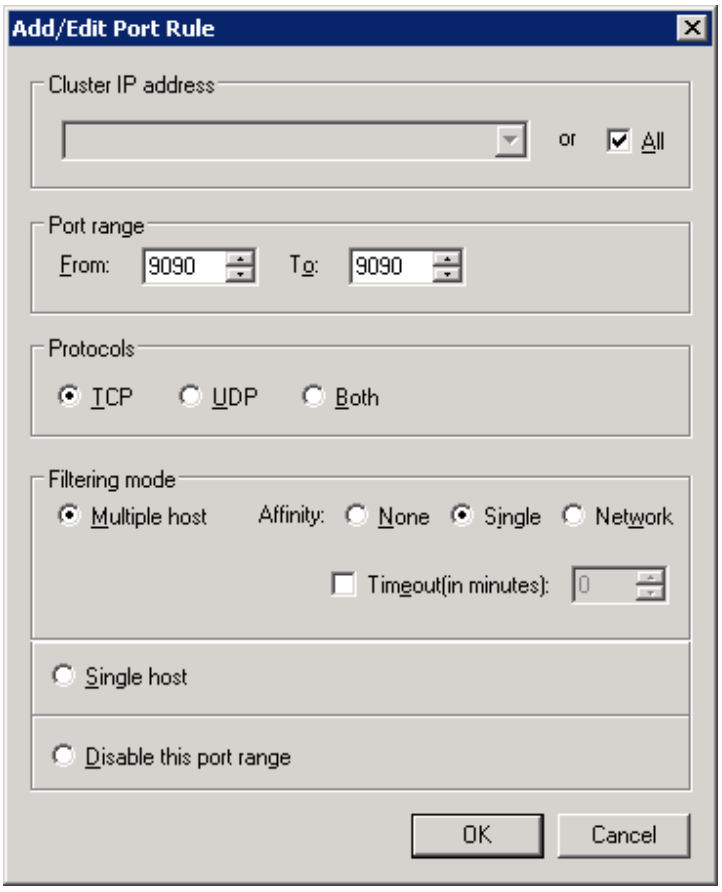
The cluster properties must be:



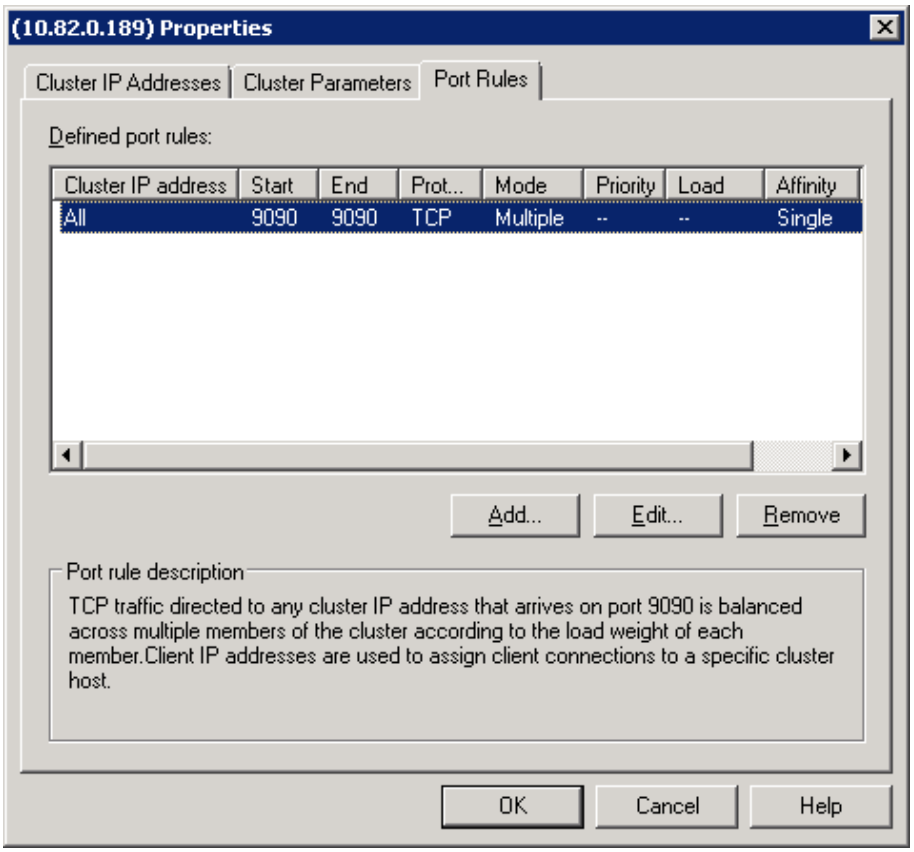
In the cluster parameters we set the Cluster operation mode to *"IGMP multicast"*.



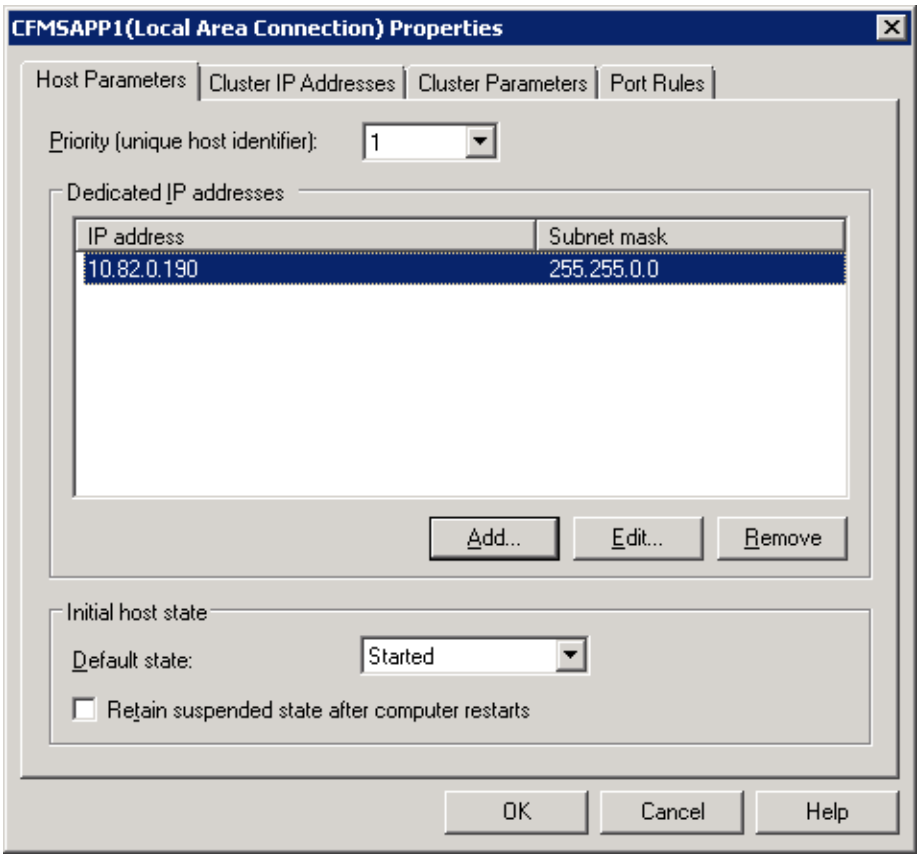
In *"Port Rules"* we press "Add…" and enter *"9090"* for the values of the port range, *"TCP"* for the protocols and filtering mode *"Multiple Host"* with *"Single"* affinity.



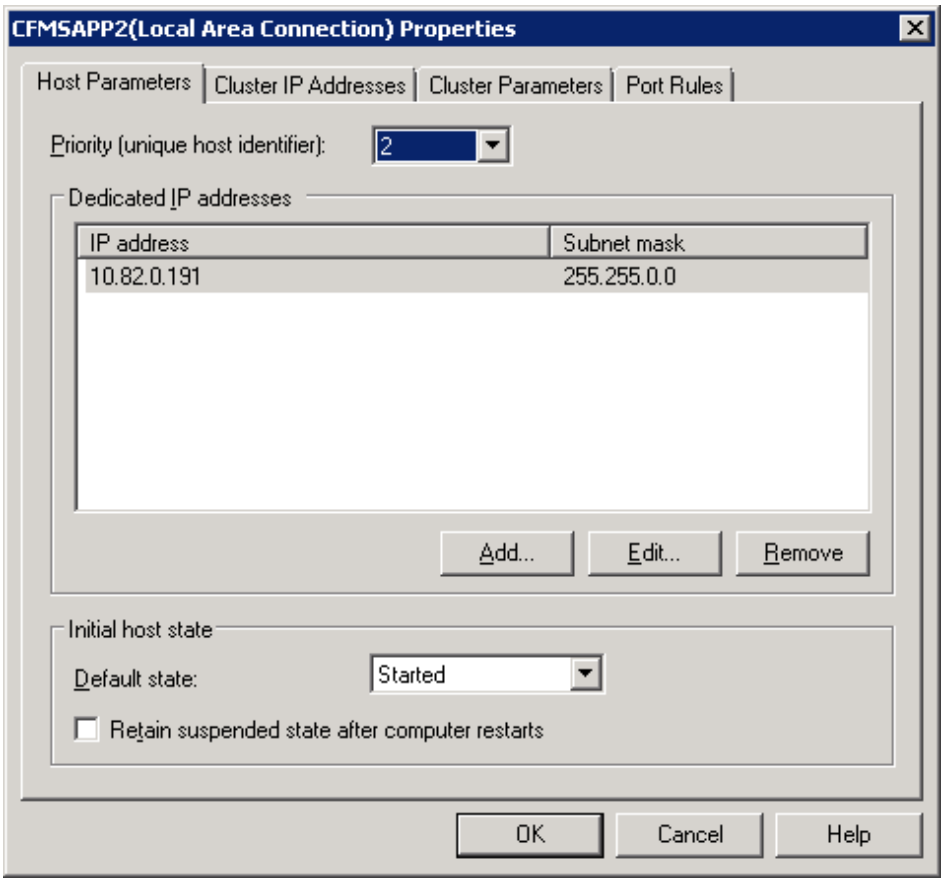
The status must be:



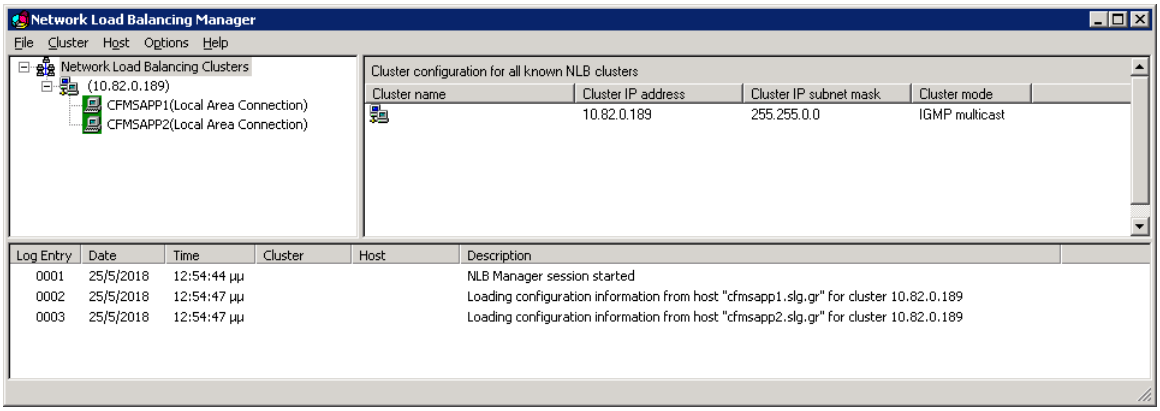
Add the first application server node.

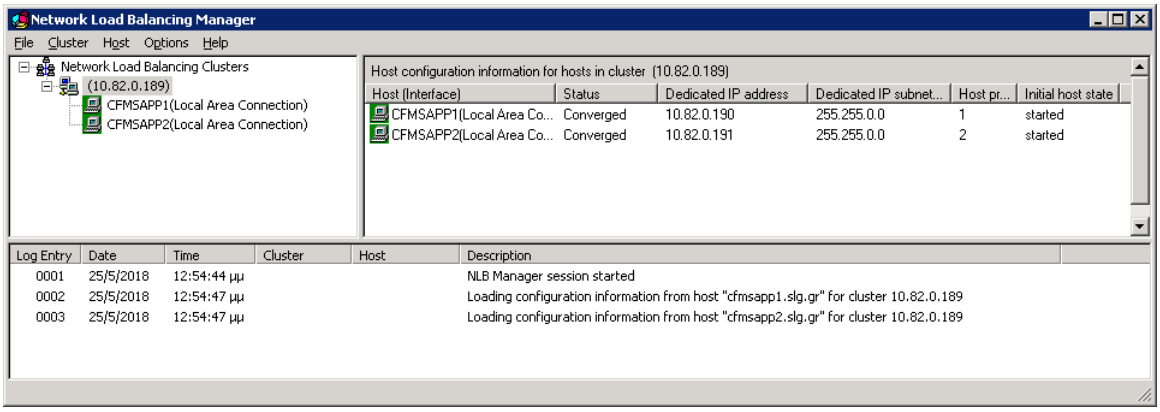


Then add the second application server node.



The status must be:





Testing Failover

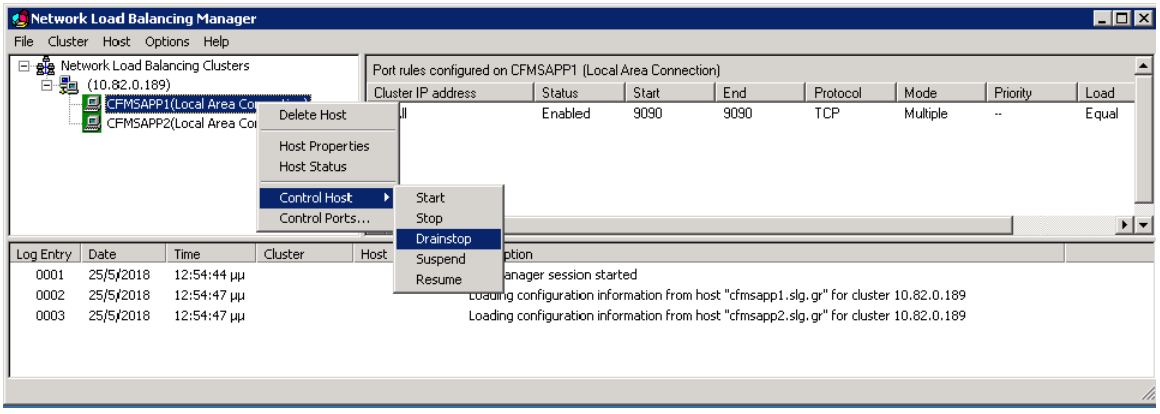
Runping can be used to test the connectivity between the client and the application server and the availability of the application server.

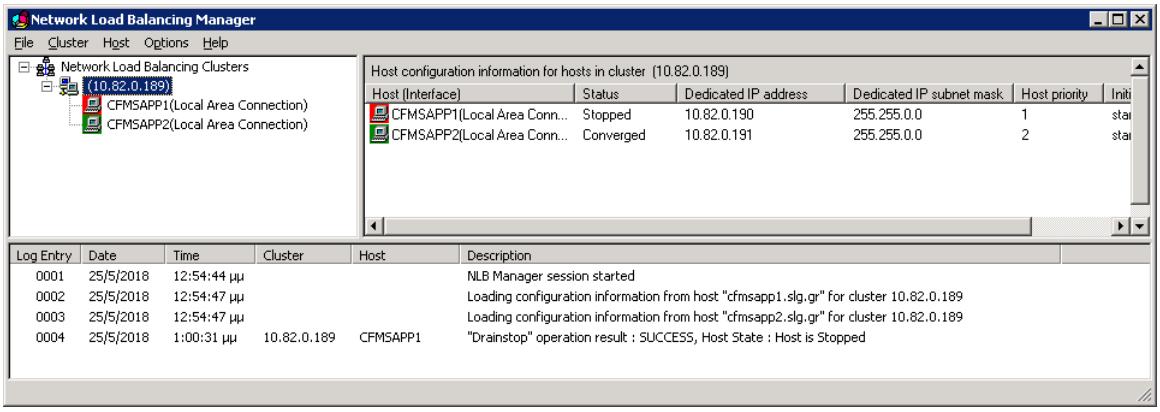
```
> runping -a CFMS -d CFMS -s APPLICATION_SERVER -p 9090 -n 2

Pinging APPLICATION-SERVER, port 9090:
Reply from APPLICATION-SERVER: Pong. Time elapsed: 00:00:00.0060651
Reply from APPLICATION-SERVER: Pong. Time elapsed: 00:00:00.0039898
```

Add "–w true" if windows authentication is enabled.

Runping is useful to test the failover because it displays the application server that sends the response.

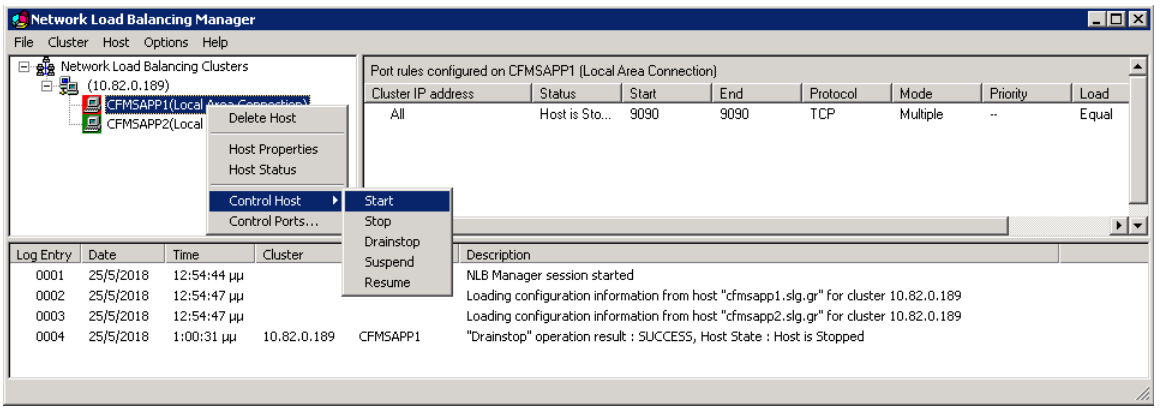
Start the runping, using as application server the cluster virtual address.

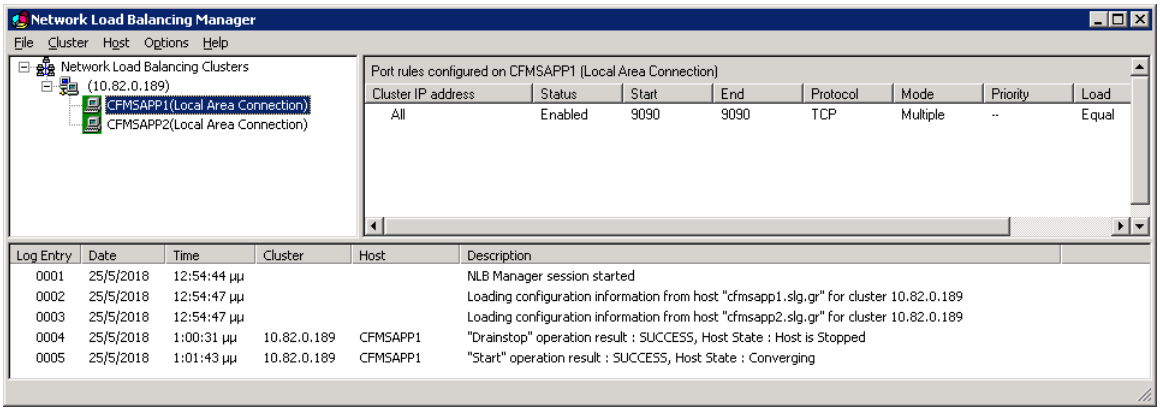




For the first host select "Drainstop". Notice how the balanced responses from the ping are going now to the second host only.

For the first host select "Start" to continue load balancing.





Maintenance

Database integrity checks

You must frequently (daily or weekly) run the DBCC CHECKDB SQL Server command for the CFMS and the system databases.

The full or differential backup command must always have the CHECKSUM option to verify the integrity of the data we backup.

Database maintenance

Run infrequently (monthly) the following T-SQL script (update_indexes.sql) to rebuild or reorganize the fragmented indexes in the database. The script is smart enough to not rebuild or reorganize not fragmented indexes.

```
declare @sql as nvarchar(max);

set @sql = '';



select @sql += 'ALTER INDEX ' + quotename(i.name) + ' ON ' +
quotename(o.name) +

    (case when s.avg_fragmentation_in_percent <= 30

        then ' REORGANIZE'

        else ' REBUILD WITH (FILLFACTOR = ' +

            (case when t.name = 'uniqueidentifier'

                then '90'

                else '100'

            end) + ', SORT_IN_TEMPDB = ON)'

    end) + ';' + char(13)

from sys.dm_db_index_physical_stats(0, null, null, null, null) as s

inner join sys.indexes as i on i.object_id = s.object_id and
i.index_id = s.index_id

inner join sys.index_columns as ic on ic.object_id = i.object_id

                                    and ic.index_id = i.index_id
```

```
                                    and ic.index_column_id = 1
inner join sys.objects as o on o.object_id = i.object_id
inner join sys.columns as oc on oc.object_id = o.object_id
                               and oc.column_id = ic.column_id
inner join sys.types as t on oc.user_type_id = t.user_type_id
where o.type = 'U'
and i.index_id <> 0
and s.avg_fragmentation_in_percent >= 10
and s.page_count >= 1000;


if @sql <> ''
    exec sp_executesql @sql;
```

Run frequently (daily) the following T-SQL script (update_statistics.sql) to update the database statistics:

```
set @sql = '';


select @sql += 'UPDATE STATISTICS ' + quotename(sch.name) + '.' +
quotename(obj.name) +
    ' (' + quotename(stat.name) + ') WITH FULLSCAN;' + char(13)
from sys.objects as obj
inner join sys.schemas as sch on sch.schema_id = obj.schema_id
inner join sys.stats as stat on stat.object_id = obj.object_id
cross apply sys.dm_db_stats_properties(stat.object_id, stat.stats_id)
as p
where obj.type in ('U','S')
and (p.modification_counter * p.modification_counter > 9 * p.rows
       or p.rows_sampled < 0.9 * p.rows);
```

```
if @sql <> ''
    exec sp_executesql @sql;
```

It is highly recommended to run this script and nothing else on production for updating your statistics. The script updates only the statistics of the tables that there are a lot of modifications. In a good maintained database it runs fast. In a unmaintained database it takes time.

Do **not** run sp_updatestats or the built in maintenance plan for update statistics because they use sample of data and can lead to wrong plans and big delays.

Database Backup

Select recovery model between Simple and Full

The recovery model controls transaction log maintenance on a database. The recovery model of database determines its backup and restore requirements.

Simple recovery model does not require log backups and automatically reclaims log space to keep space requirements small, essentially eliminating the need to manage the transaction log space. Operations that require transaction log backups are not supported by the simple recovery model (i.e. Log shipping, Always On mirroring, Point in time restores). Changes since the most recent backup are unprotected in simple recovery mode; in the event of a disaster, those changes must be redone.

Full recovery model requires log backups. Full recovery model can recover to an arbitrary point in time.

You must get frequently (at least daily) backup of the CFMS and the system databases.

To create a full database backup

- Connect to the database server with a user that have backup permissions (member of sysadmin, db_owner and db_backupoperator roles).
- Execute the BACKUP DATABASE statement to create the full database backup, specifying the name of the database to back up and the backup device where the full database backup is written.

Example:

```
BACKUP DATABASE CFMS TO DISK = 'C:\CFMSFull.bak' WITH COMPRESSION,
CHECKSUM
```

Every time you get a backup specify the COMPRESSION option and the CHECKSUM option. The COMPRESSION option speeds up backup and reduces the backup file size. The CHECKSUM option while reads data from the disk validates the checksum to detect disk failures.

To create a database log backup

- Connect to the database server with a user that have backup permissions (member of sysadmin, db_owner and db_backupoperator roles).
- Execute the BACKUP LOG statement to create a log backup and truncate the transaction log, specifying the name of the database to back up and the backup device where the transaction log backup is written.

Example:

```
BACKUP LOG CFMS TO DISK = 'C:\CFMSLog.trn'
```

Log backup is required in full recovery mode. We have to keep all the log backups after the full backup to be able to restore.

Operations

CFMS database restore

After restoring a CFMS database to your environment, you must link the database server clroot login with the database user clroot.

```
use CFMS;
alter user clroot with login = clroot, default_schema = dbo;
go
```

Best practice is to run the OneStopScript.sql on the restored database to ensure that the setup is perfect. The script links user with login and also either detects or fixes installation problems.

Restoring clroot user when you cannot access CFMS

The following SQL command enables the "clroot" user and sets its password to "qaz".

```
update syUser set
    Username = 'clroot',
    Password = 'ig+DaptWSZI1WrV5GCegMw==',
    Active = 1,
    FailedLogonCount = 0,
    ExpirationDate = null,
    LastLogonDate = null,
    LastPasswordChange = null
where Id = 'A34F4234-E9AD-43D5-BE4D-AAF53453B02D';
```

Running CFMS Batch jobs

Runbatch can be used to initiate batch jobs using the command line or the Windows Scheduler.

```
RunBatch ConnectionAlias "BatchCode" /D CFMS /C Company /L en
/U UserName /P Password /HA IPAddress /HP 9090
```

Run batch parameters are:

/D Domain	Set the domain name. Default = CFMS
/C Company	Set the company name. Default = (user default)
/L Language	Set the language code. Default = (user default)
/T Tag	Set the tag name. Default = (user default)
/R Restricted	Set the company restricted flag. Specify 1 or 0. Default = (user default)
/U UserName	The user name for logging in. If not provided, windows authentication will be employed.
/P Password	The password for logging in. If windows authentication is being used, it is not necessary.
/HA Address	The host address of the server to log in to. Default = 127.0.0.1
/HP HostPort	The host port of the server to log in to. Default = 9090
/W	Use Windows Authentication.
/WP	Watch path for files to load/import.
/RE	Confirm batch rerun.
/EN Entity	Set the parent entity of the operation. Default = System:syBatch
/OP Operation	Set the operation to run. Default = System:syRunBatch
/TE	Test-only mode. Skips the actual posting of the operation.
/S	Saves arguments to file and exits without executing. ConnectionAlias and BatchCode will be ignored. Password will be encrypted.

Run batch exit status possible values are:

0 – Success

1 – Invalid argument

2 – Login failed

3 – Entity not found

4 – Operation failed