

Device Manager Server®

for M2M devices

(routers, metering modems)

Installation Manual

v2.3

2024-07-04

Document specifications

This document was made for the **Device Manager**[®] software and it contains detailed description of configuration and usage for the proper operation of the software.

Document category:	Installation Manual
Document subject:	Device Manager [®] Server
Author:	WM Systems LLC
Document version No.:	REV 2.3
Number of pages:	15
Device manager version:	v7.3
Last modified:	04.07.2024
Approval date:	04.07.2024

Table of contents

1. Introduction	4
2. Setup and Configuration	6
2.1. Prerequisites	6
2.2. System components.....	6
2.3. Startup	7
2.3.1 Install and configure the SQL Server.....	7
2.3.2. Configure the DataBrokerDaemon.....	7
2.3.4. Configure the DeviceManagerDaemon.....	10
2.3.4.1 Set the services.....	12
2.3.5. Network preparations.....	13
3. Support	14
3.1. Technical Support.....	14
3.2. GPL license.....	14
4. Legal notice	15

Chapter 1. Introduction

The Device Manager can be used for remote monitoring and central management of our industrial routers, data concentrators (M2M Router, M2M Industrial Router, M2M Router PRO4), DCUs, and smart metering modems (WM-Ex family, WM-i device).

A remote device management platform with continuous monitoring of devices, analytic capabilities, mass firmware updates, reconfiguration.

The software allows you to check the service KPIs of the devices (QoS, life signals), intervene and control the operation, running maintenance tasks on your devices.

It's a cost-effective way of continuous, online monitoring of your connected devices in remote locations.

By receiving info on the device's availability, the monitoring of life signals, and operation characteristics of onsite devices - owing to the analytics data derived from them - it continuously checks the operation values (signal strength of the cellular network, communication health, device performance).

With the usage of the application - as a service provider or maintenance company - you can manage the installation of new firmware releases for groups or devices, and distribute a basic configuration for a bunch of devices.

The Microsoft® Windows®-based application allows installing or replacing the firmware running on the device. In addition, you can install or replace certifications (CSR, CA certifications, etc.) for your devices.

You can configure the usage of the encrypted TLS protocol communication between the M2M device and the Device Manager® software.

You can also remotely control your devices (rebooting them or executing other tasks on the device).

The application enables the grouping, arrangement and management of devices in groups according to on-site installation or according to other logic. In this way, you can manage the installation of new firmware releases and the maintenance of devices individually or even per installation site.

Chapter 2. Setup and Configuration

2.1. Prerequisites

Max. 10.000 pieces of metering modems can be managed by a single Device Manager instance.

The usage of the Device Manager server application requires the following conditions:

Hardware environment:

- Physical or virtual environment supported
- 8 Core CPU
- 8 GB RAM (minimum) – 16 GB RAM (preferred), depending from amount of devices
- 1Gbit LAN connection
- 500 GB storage capacity (depends of the amount of devices)

Software environment:

- Windows Server 2019 or newer - Linux or Mac OS not supported
- MS SQL Express Edition 2019 (minimum) – MS SQL Standard 2019 or newer (preferred) - Other types of databases are not supported (Oracle, MongoDB, MySql)
- MS SQL Server Management Studio – for creating accounts and database and managing the database (eg.: backup or restore)

2.2. System elements

The Device Manager consists of three main software elements:

- *DataBrokerDaemon.exe* – communication platform between the database and the data collector service
- *DeviceManagerDaemon.exe* – collecting the data from the remote metering modems

DataBrokerDaemon

The device manager's data broker's main task is maintaining the database connection with the SQL server.

It provides a REST API interface to the Device Manager Daemon and the Device Manager Client applications.

In addition, it has a data synchronization feature, to keep all the running UIs synchronized with the database.

DeviceManagerDaemon

This is the device management service and business logic. It communicates with the Data Broker via a REST API, and with the M2M devices through WM Systems' proprietary device management protocol. The communication flows in a TCP socket, which can optionally be secured with an industry-standard TLSv1.2 transport layer security solution, based on mbedTLS (on the device side) and OpenSSL (on the server side).

2.3. Startup

2.3.1 Install and configure the SQL Server

If you need to install an SQL server, please visit the following website and select the preferred SQL product:

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

If you already have an SQL server installation, create a new database eg. DM7.3, and create the database user account with owner rights on that DM7.1 database. When you start the data broker for the first time, it will create all necessary tables and fields in the database. You don't need to create them manually.

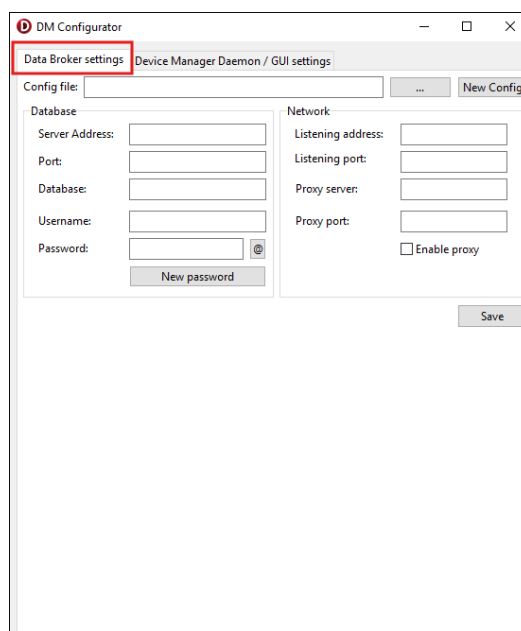
1. Create the root folder on the destination system' partition. eg. **C:\DMv7.3**

2. Unzip the Device Manager compressed software package into the folder.

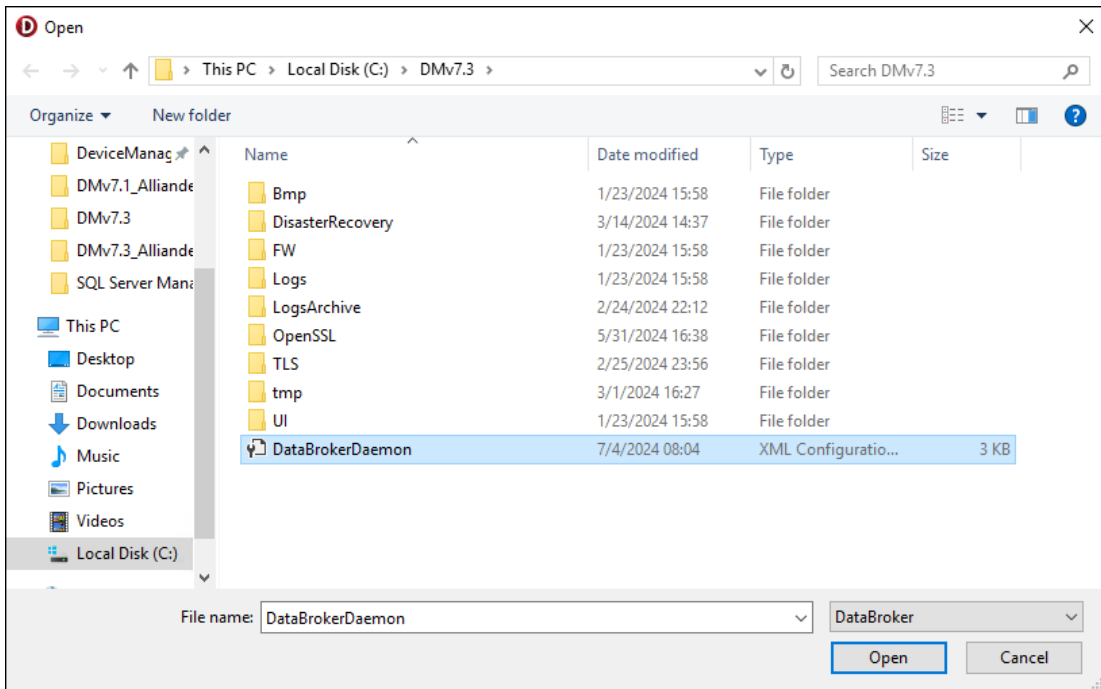
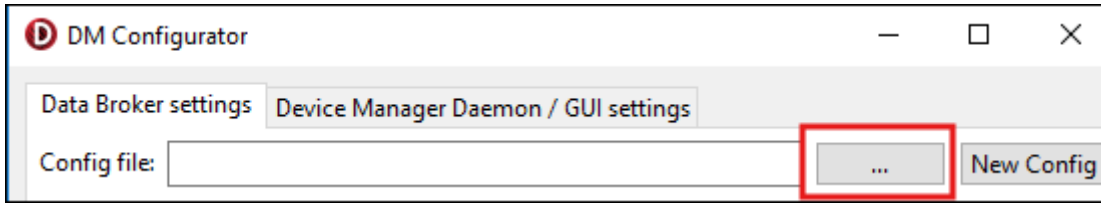
2.3.2. Configure the DataBrokerDaemon

1. Execute the **DMConfigurator.exe** file from the package.

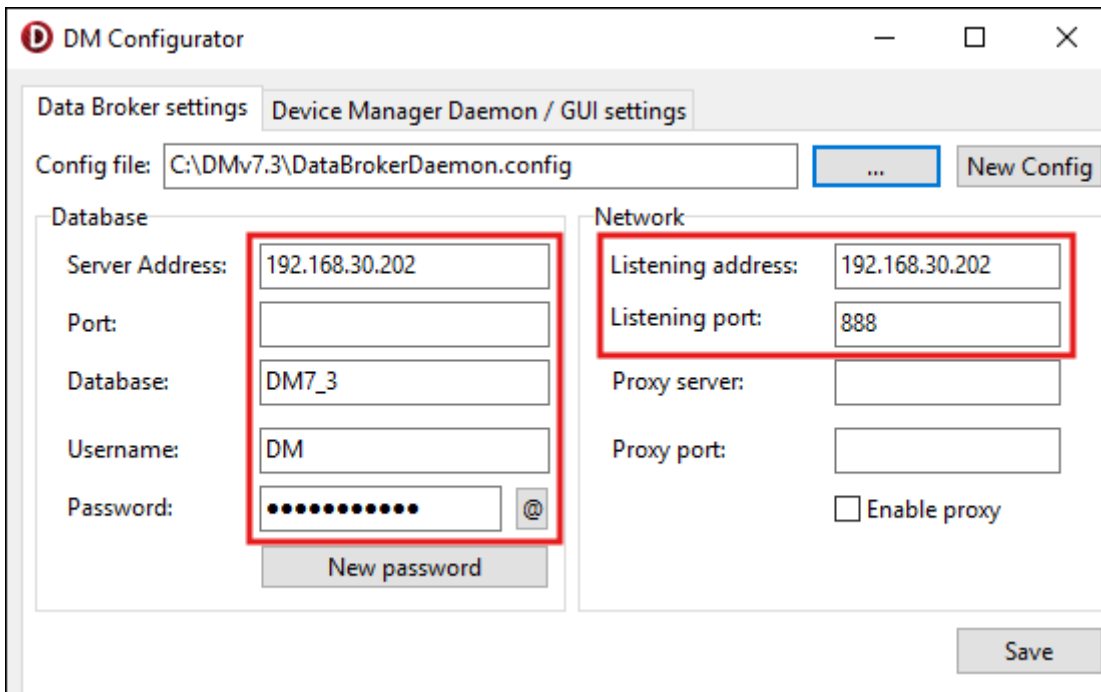
2. Select the **Data Broker Settings** ta.b



3. Browse the **DataBrokerDaemon.config** template file from the DM folder and press the **Open** button.



4. Set the SQL server parameters on the left marked part, and fill listening interface parameters on the right marked part. The **Listening address** is the server where the DM services are running. In the following sample, as you can see, and SQL server and the Data Broker service, which are deployed on the same server.

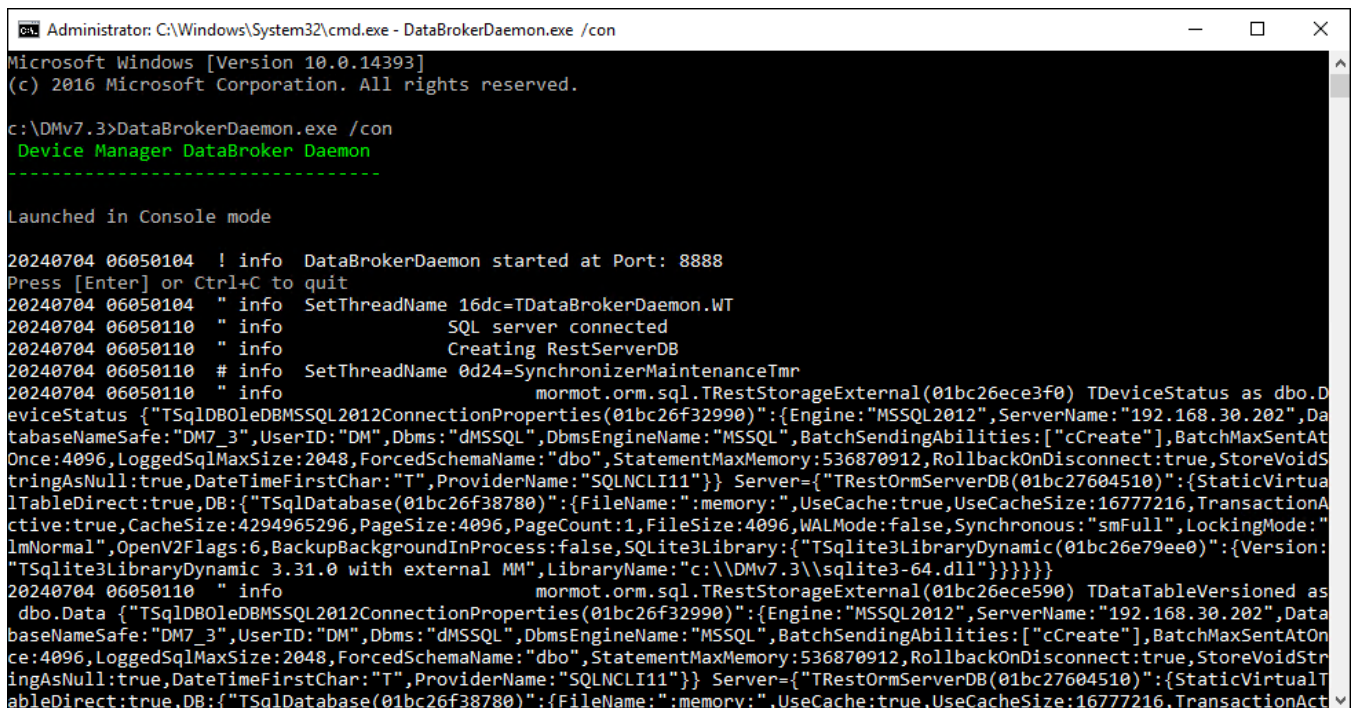


5. When settings are done, press the **Save** button.

6. After the modifications, execute the data broker with administrator privileges in console mode by using the following command in the command line:

```
c:\DMv7.3>DataBrokerDaemon.exe /con
```

7. If the parameters are fine, you will see this windows and message:



```
Administrator: C:\Windows\System32\cmd.exe - DataBrokerDaemon.exe /con
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\DMv7.3>DataBrokerDaemon.exe /con
Device Manager DataBroker Daemon
-----
Launched in Console mode

20240704 06050104 ! info  DataBrokerDaemon started at Port: 8888
Press [Enter] or Ctrl+C to quit
20240704 06050104 " info  SetThreadName 16dc=TDaemon.WT
20240704 06050110 " info  SQL server connected
20240704 06050110 " info  Creating RestServerDB
20240704 06050110 # info  SetThreadName 0d24=SynchronizerMaintenanceTmr
20240704 06050110 " info  mormot.orm.sql.TRestStorageExternal(01bc26ece3f0) TDeviceStatus as dbo.D
eviceStatus {"TSqlDBOLEDBMSSQL2012ConnectionProperties(01bc26f32990)":{Engine:"MSSQL2012",ServerName:"192.168.30.202",Da
atabaseNameSafe:"DM7_3",UserID:"DM",Dbms:"dMSSQL",DbmsEngineName:"MSSQL",BatchSendingAbilities:["cCreate"],BatchMaxSentAtOn
ce:4096,LoggedSqlMaxSize:2048,ForcedSchemaName:"dbo",StatementMaxMemory:536870912,RollbackOnDisconnect:true,StoreVoidStr
ingAsNull:true,DateTimeFirstChar:"T",ProviderName:"SQLNCLI11"}} Server={"TRestOrmServerDB(01bc27604510)":{StaticVirtualT
ableDirect:true,DB:{"TSqlDatabase(01bc26f38780)":{FileName:"memory:",UseCache:true,UseCacheSize:16777216,TransactionAct
ive:true,CacheSize:4294965296,PageSize:4096,PageCount:1,FileSize:4096,WALMode:false,Synchronous:"smFull",LockingMode:"
lmNormal",OpenV2Flags:6,BackupBackgroundInProcess:false,SQLite3Library:{"TSQLite3LibraryDynamic(01bc26e79ee0)":{Version:
"TSQLite3LibraryDynamic 3.31.0 with external MM",LibraryName:"c:\DMv7.3\sqlite3-64.dll"}}}}}
20240704 06050110 " info  mormot.orm.sql.TRestStorageExternal(01bc26ece590) TDataTableVersioned as
dbo.Data {"TSqlDBOLEDBMSSQL2012ConnectionProperties(01bc26f32990)":{Engine:"MSSQL2012",ServerName:"192.168.30.202",Data
baseNameSafe:"DM7_3",UserID:"DM",Dbms:"dMSSQL",DbmsEngineName:"MSSQL",BatchSendingAbilities:["cCreate"],BatchMaxSentAtOn
ce:4096,LoggedSqlMaxSize:2048,ForcedSchemaName:"dbo",StatementMaxMemory:536870912,RollbackOnDisconnect:true,StoreVoidStr
ingAsNull:true,DateTimeFirstChar:"T",ProviderName:"SQLNCLI11"}} Server={"TRestOrmServerDB(01bc27604510)":{StaticVirtualT
ableDirect:true,DB:{"TSqlDatabase(01bc26f38780)":{FileName:"memory:",UseCache:true,UseCacheSize:16777216,TransactionAct
```

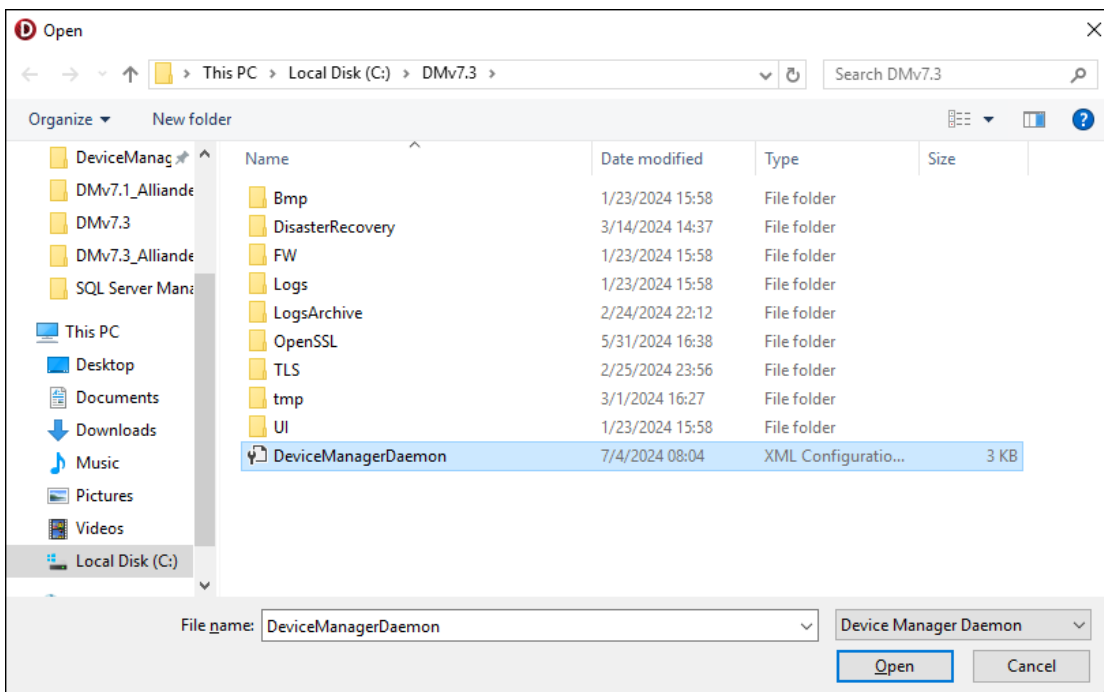
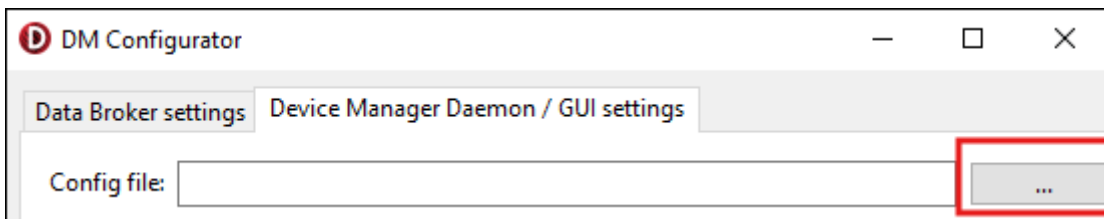
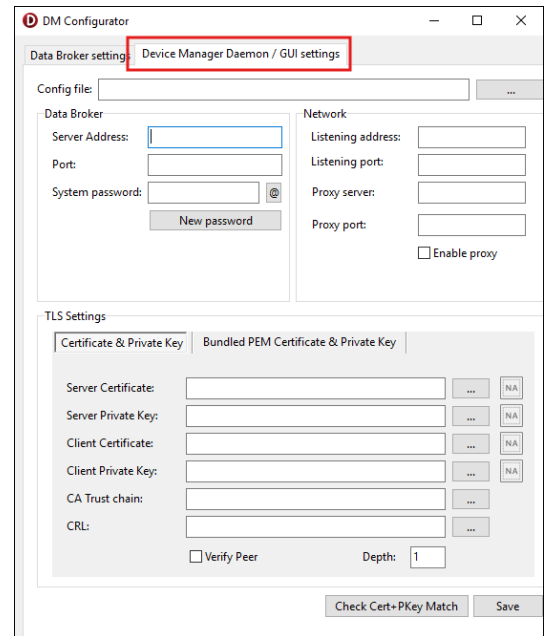
8. Now **Data Broker** is connected to the database server with the given credentials and create/modify the database structure automatically. At the first usage the database table structure creation will need more time.

IMPORTANT!

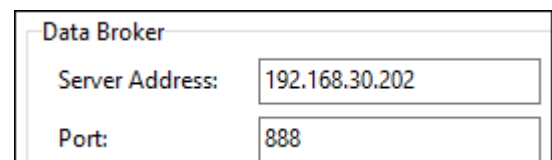
If you want to change the Device Manager Data Broker settings, stop and close the application. If you finished with the modification, execute the application with administrator privileges. In other cases, the application will overwrite the modified settings with the last good known/working settings!

2.3.4. Configure the DeviceManagerDaemon

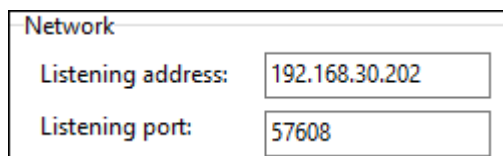
1. Execute the **DMConfigurator.exe** file. Then DM will be started and the following window will appear.
2. Select the **Device Manager Daemon / GUI Settings** tab.
3. Browse the **DataBrokerDaemon.config** template file from the program folder and press the **Open** button.



4. Set **Data Broker** parameters (Data **Broker** server IP, and **Listening port**).

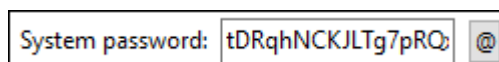


- Set **Network** parameters (**Listening address** and **Listening port**) for devices on the field at the server side.



A screenshot of a configuration window titled "Network". It contains two input fields: "Listening address" with the value "192.168.30.202" and "Listening port" with the value "57608".

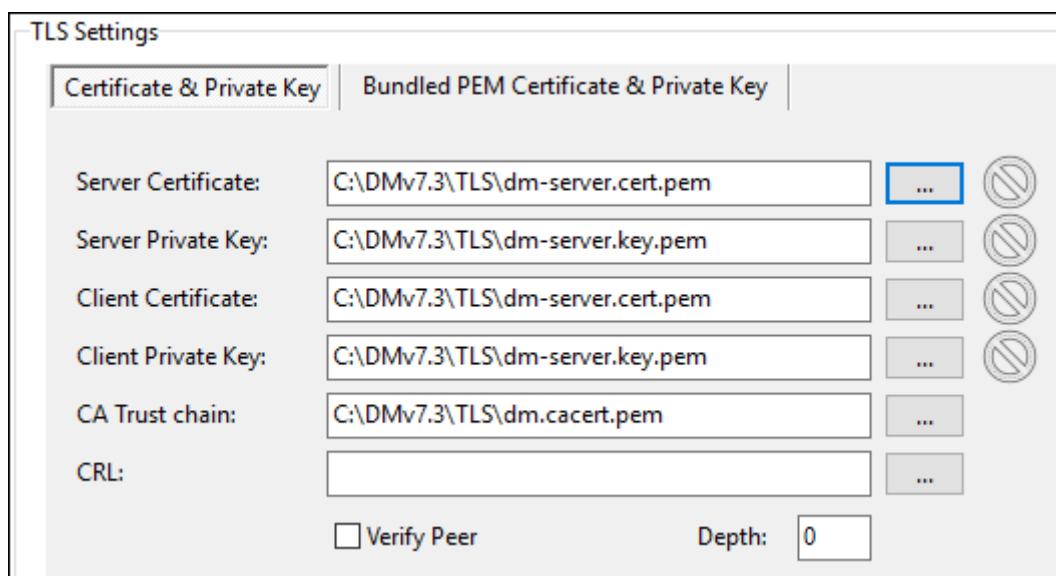
- Set the system password for client applications



A screenshot of a configuration window titled "System password". It contains an input field with the password "tDRqhNCKJLTg7pRQ" and a small icon to its right.

Note, if system password is not same in the **Device Manager Daemon** and the **Client applications** config file, then Client app will be not able to connect to the system.

- Locate and set the certificate files (with *.pem file extension) from the program directory.



A screenshot of a "TLS Settings" window. It has two tabs: "Certificate & Private Key" (selected) and "Bundled PEM Certificate & Private Key". The window contains several input fields for certificate and key files, each with a browse button (three dots) and a lock icon. The fields are: "Server Certificate" (C:\DMv7.3\TLS\dm-server.cert.pem), "Server Private Key" (C:\DMv7.3\TLS\dm-server.key.pem), "Client Certificate" (C:\DMv7.3\TLS\dm-server.cert.pem), "Client Private Key" (C:\DMv7.3\TLS\dm-server.key.pem), "CA Trust chain" (C:\DMv7.3\TLS\dm.cacert.pem), and "CRL" (empty). At the bottom, there is a checkbox for "Verify Peer" and a "Depth" field set to "0".

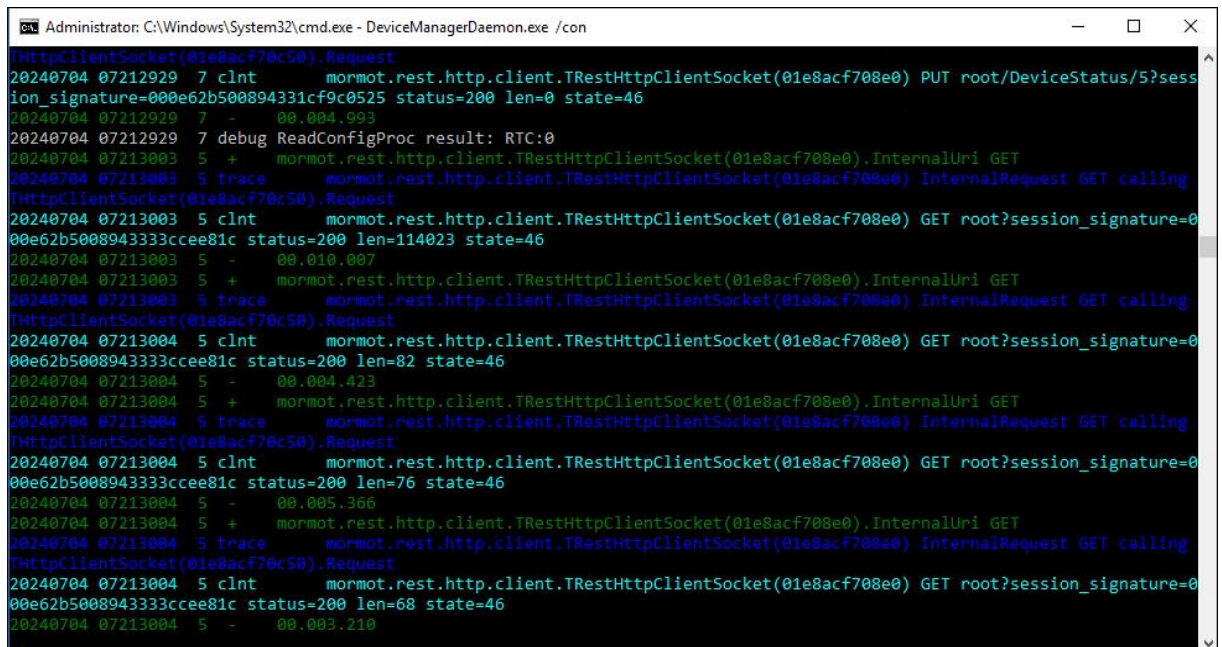
When the configuration is done, press the Save button,

- After modifications, execute the **DeviceManagerDaemon** with administrator credentials in console mode with the following command in command line:
c:\DMv7.3> DeviceManagerDaemon.exe /con

With this option you can check the connection between the **Data Broker** and the **Device Manager Daemon**.

If you receive error messages with **red** color, please check the settings again.

If parameter settings were correct, you will see a window like this:



```
Administrator: C:\Windows\System32\cmd.exe - DeviceManagerDaemon.exe /con
[HttpClientSocket(01e8acf70c50)].Request
20240704 07212929 7 cInt mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) PUT root/DeviceStatus/5?session_signature=000e62b500894333cf9c0525 status=200 len=0 state=46
20240704 07212929 7 - 00.004.993
20240704 07212929 7 debug ReadConfigProc result: RTC:0
20240704 07213003 5 + mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0).InternalUri GET
20240704 07213003 5 trace mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) InternalRequest GET calling
[HttpClientSocket(01e8acf70c50)].Request
20240704 07213003 5 cInt mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) GET root?session_signature=00e62b5008943333ccee81c status=200 len=114023 state=46
20240704 07213003 5 - 00.010.007
20240704 07213003 5 + mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0).InternalUri GET
20240704 07213003 5 trace mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) InternalRequest GET calling
[HttpClientSocket(01e8acf70c50)].Request
20240704 07213004 5 cInt mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) GET root?session_signature=00e62b5008943333ccee81c status=200 len=82 state=46
20240704 07213004 5 - 00.004.423
20240704 07213004 5 + mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0).InternalUri GET
20240704 07213004 5 trace mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) InternalRequest GET calling
[HttpClientSocket(01e8acf70c50)].Request
20240704 07213004 5 cInt mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) GET root?session_signature=00e62b5008943333ccee81c status=200 len=76 state=46
20240704 07213004 5 - 00.005.366
20240704 07213004 5 + mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0).InternalUri GET
20240704 07213004 5 trace mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) InternalRequest GET calling
[HttpClientSocket(01e8acf70c50)].Request
20240704 07213004 5 cInt mormot.rest.http.client.TRestHttpClientSocket(01e8acf708e0) GET root?session_signature=00e62b5008943333ccee81c status=200 len=68 state=46
20240704 07213004 5 - 00.003.210
```

9. Now **DeviceManagerDaemon** is connected to the **Data Broker** server and the system can receive data from devices.

IMPORTANT! If you want to change **DeviceManagerDaemon** settings, first stop the service. If you finished the modification start the service again.

In other case, the service will overwrite the modified settings with the last good known/working settings!

2.3.5 Set the applications as services

1. If you want to create a service, then open the command line and execute the following commands with administrator privileges:

DataBrokerDaemon.exe /install → this will install the **Data Broker** as service

DeviceManagerDaemon.exe /install → this will install the **Device Manager Daemon** as service

2. Before starting services, make sure that the previously started console windows are all closed.

3. Start the service from the services list (Windows+R button → enter: services.msc and push <ENTER> key.)

2.3.6 Network preparations

Open the appropriate ports on the **Device Manager Server** for the correct communication, such as:

- Server port for the incoming endpoint device communication (for routers, modems).
- Data Broker port for the client application communication.
- Enable the outgoing communication on your firewall from the server to the devices.

Chapter 3. Support

3.1. Technical Support

If you have any questions concerning the usage of the device, contact us through your personal and dedicated salesman.

Online product support can be required here at our website:

<https://www.m2mserver.com/en/support/>

The documentation and software release for this product can be accessed via the following link:

<https://www.m2mserver.com/en/product/device-manager/>

3.2. GPL license

The Device Manager software is not a free product. WM has the application's copyrights. The software is ruled by the GPL licensing terms.

The product uses the Synopse mORMot Framework component's source code, which is also licensed under GPL 3.0 licensing terms.



Chapter 4. Legal notice

©2024. WM Systems LLC.

The content of this documentation (all information, pictures, tests, descriptions, guides, logos) is under copyright protection. Copying, using, distributing and publishing is only permitted with the consent of WM Systems LLC., with clear indication of the source.

The pictures in the user guide are only for illustration purposes.

WM Systems LLC. does not acknowledge or accept responsibility for any mistakes in the information contained in the user guide.

The published information in this document is subject to *change without notice*.

All data contained in the user guide is for information purposes only. For further information, please, contact our colleagues.

Warning! Any errors occurring during the program update process may result in failure of the device.