

# Running FactoryStudio in Docker

## System Requirements

To make use of this feature, the following system requirements need to be matched.

- FactoryStudio version *fs-9.1.12*.
- Install [Docker](#).
- Windows 10 64-bit: Pro, Enterprise, or Education (Build 17134 or later). For Windows 10 Home, see [Install Docker Desktop on Windows Home](#).
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  - 64-bit processor with [Second Level Address Translation \(SLAT\)](#).
  - 4GB system RAM.
  - BIOS-level hardware virtualization support must be enabled in the BIOS. For more information, see [Virtualization](#).

## Configuration Procedure

### Copying Folders and Runtime Files

After installing Docker and *fs-9.1.12*, copy the files and subfolders from “.. *fs-9.1*” to the Docker’s temporary folder, as in the image.

Nome	Data de modificação	Tipo	Tamanho
Bin	17/03/2021 10:24	Pasta de arquivos	
Dockerfile	17/03/2021 10:08	Arquivo	1 KB

In the image, the **Bin** is the temporary folder.

Not all files and subfolders are required, only the ones used in Runtime. The subfolders for *DBProviders*, *Extensions*, *IoT* and *Templates* are not used in Runtime and do not need to be copied.

The **HTML5** subfolder is only necessary if Web Clients will access pages from runtime. **DevExpress** and **Sync- Fusion** files are only necessary if Smart Clients will access displays from runtime.

Also, only the protocol drivers used by the project need to be copied (Ex: *MODBUS.xml* and *T.ProtocolDriver.MODBUS.dll*). The size of the image depends on which files will be used by the project.

The Project file (*.tproj*) and all other necessary files (e.g.: vc redistrib, etc) should also be copied to the temporary folder.

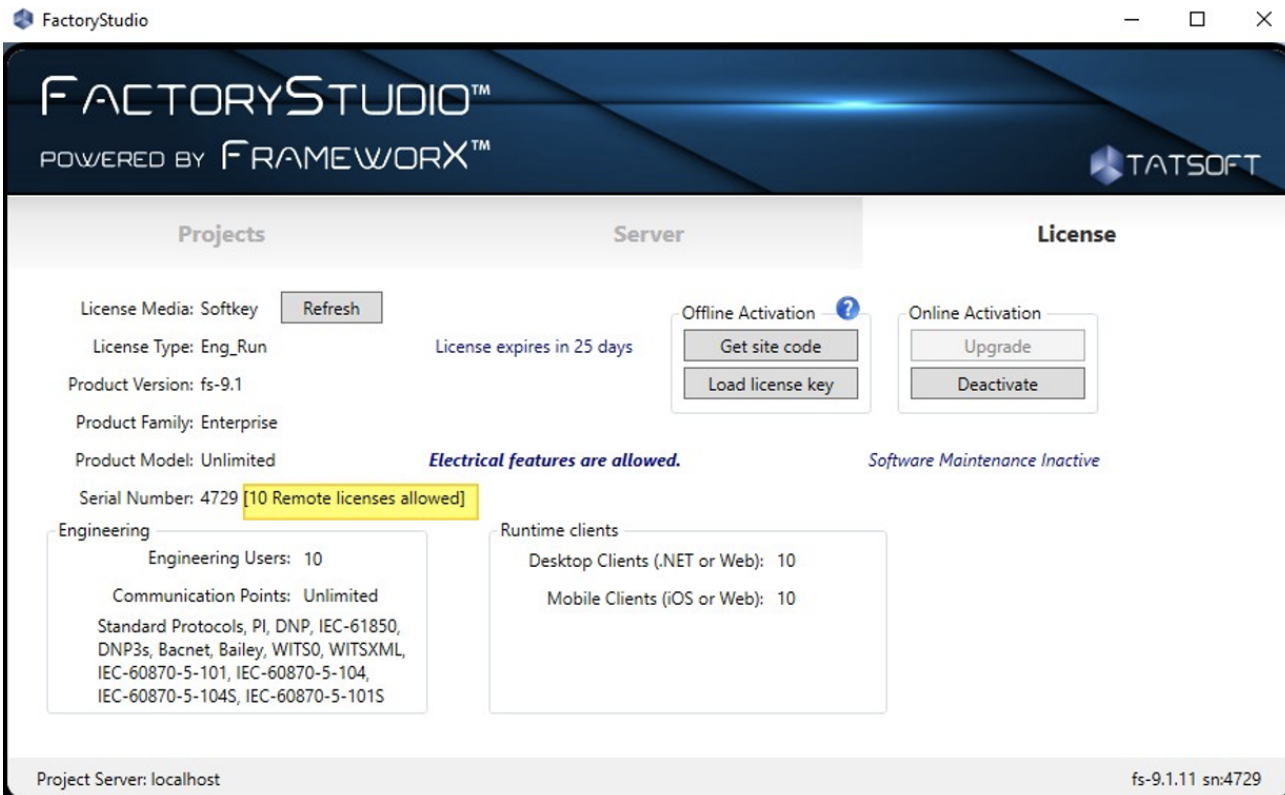
### Copying License and Argument Files

Create a **RemoteLicenseService.config** file and copy it to the temporary folder. The syntax is:

```
RemoteServer=<IpAddress>:<PortNumber>  
// IP:Port where the licensing server is Running RemoteServer=192.168.1.1:3100
```



The licensing server needs to have a license containing a *Remote Licenses* feature.



Create an **Arguments.config** file and copy it to the temporary folder. The syntax is:

```
/project:C:\App\<DockerTest>.tproj // Project file name (mandatory)
/docker // (mandatory)
```



This file contains the parameters that will be passed to the FactoryStudio modules (*TServer* and *TRun-Module*). These parameters are the same ones used by the TStartup program.

## Creating Docker File

Create a **Dockerfile** based on existent image containing **.NET Framework v4.6.2** or higher, (necessary by FactoryStudio). Example:

```
FROM mcr.microsoft.com/dotnet/framework/runtime WORKDIR /app
COPY bin .
RUN C:\app\vc_redist.x64.exe /quiet /install RUN C:\app\vc_redist.x86.exe /quiet /install
ENTRYPOINT [ "TServer.exe", "/args:Arguments.config" ]
```

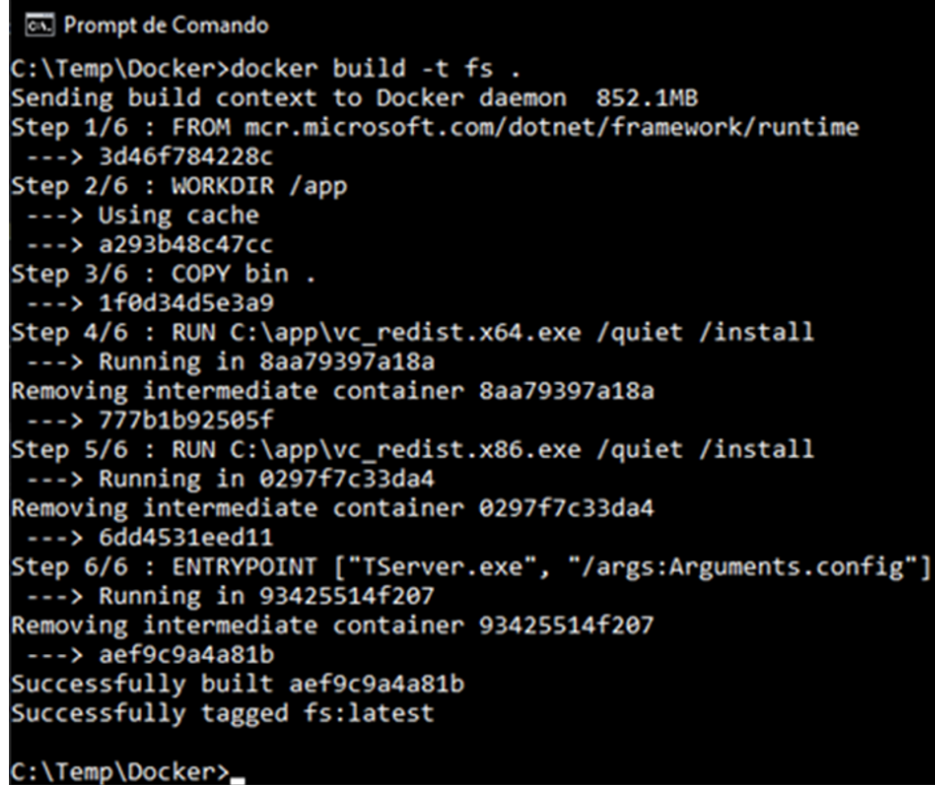
Notes on this procedure:

- Image from "<https://dotnet.microsoft.com/en-us/>" is used as initial image.
- In this example, all files from the temporary folder (Bin) will be copied to the "App" folder (on image).
- 3 VC redistrib files will be installed via the RUN command.

4 TServer will be executed using the parameters defined in the Arguments.config file.

To build the image, execute the following via the **CMD** command:

```
docker build -t fs . // dot (.) IS IMPORTANT!!!
```



The screenshot shows a Windows Command Prompt window titled "Prompt de Comando". The user is in the directory "C:\Temp\Docker" and has executed the command "docker build -t fs .". The output shows the build process: sending the build context (852.1MB), pulling the base image "mcr.microsoft.com/dotnet/framework/runtime:3d46f784228c", and then building the image in six steps. Step 1 is FROM, Step 2 is WORKDIR, Step 3 is COPY, Step 4 is RUN (installing VC redist x64), Step 5 is RUN (installing VC redist x86), and Step 6 is ENTRYPOINT. The build is successful and the image is tagged "fs:latest".

```
C:\Temp\Docker>docker build -t fs .
Sending build context to Docker daemon 852.1MB
Step 1/6 : FROM mcr.microsoft.com/dotnet/framework/runtime
---> 3d46f784228c
Step 2/6 : WORKDIR /app
---> Using cache
---> a293b48c47cc
Step 3/6 : COPY bin .
---> 1f0d34d5e3a9
Step 4/6 : RUN C:\app\vc_redist.x64.exe /quiet /install
---> Running in 8aa79397a18a
Removing intermediate container 8aa79397a18a
---> 777b1b92505f
Step 5/6 : RUN C:\app\vc_redist.x86.exe /quiet /install
---> Running in 0297f7c33da4
Removing intermediate container 0297f7c33da4
---> 6dd4531eed11
Step 6/6 : ENTRYPOINT ["TServer.exe", "/args:Arguments.config"]
---> Running in 93425514f207
Removing intermediate container 93425514f207
---> aef9c9a4a81b
Successfully built aef9c9a4a81b
Successfully tagged fs:latest
C:\Temp\Docker>
```

To run the image, execute the following via the *CMD* command:

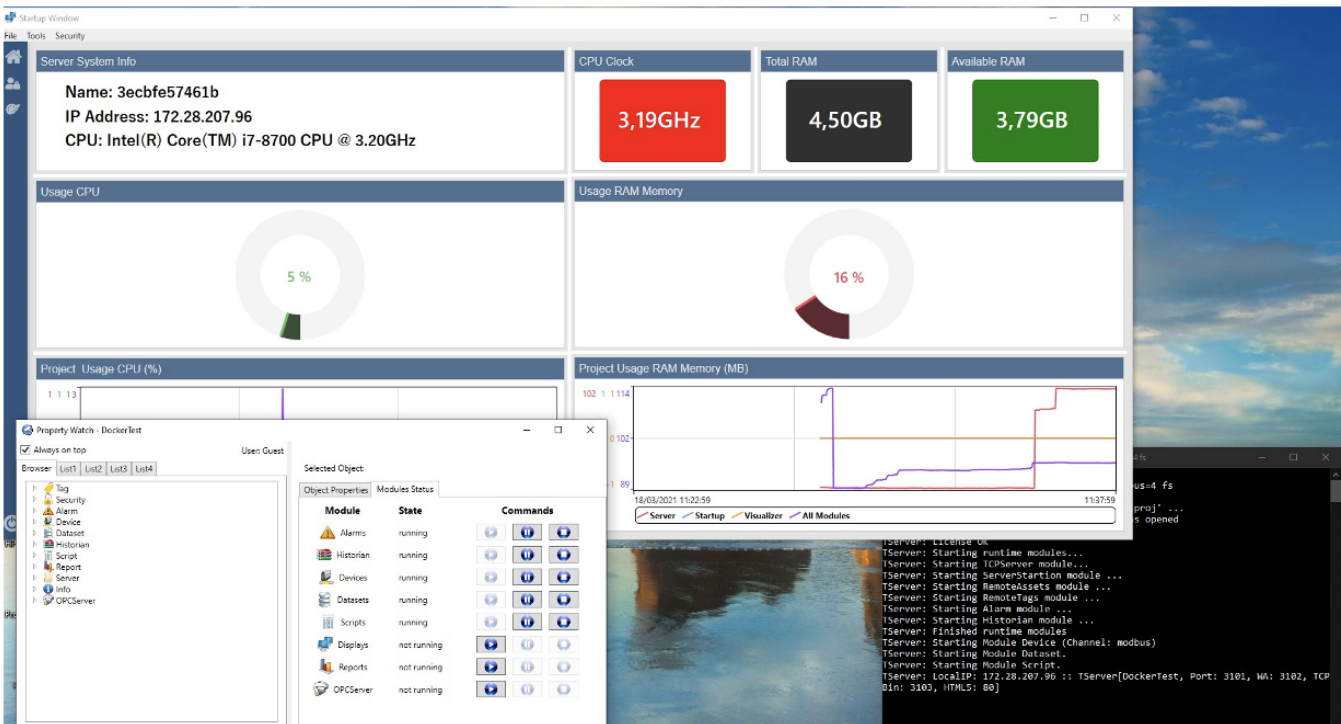
```
docker run --memory=4096m --cpus=4 fs
// Min Recommended 4 GB Mem and 4 CPU cores
```


```
C:\Temp\> docker run --memory=4096m --cpus=4 fs

C:\Temp\> docker run --memory=4096m --cpus=4 fs
TServer: Loading project...
TServer: Opening project 'C:\App\Dockertest.tproj' ...
TServer: Project 'C:\App\Dockertest.tproj' was opened
TServer: Checking license ...
TServer: License OK
TServer: Starting runtime modules...
TServer: Starting TCPServer module...
TServer: Starting ServerStartion module ...
TServer: Starting RemoteAssets module ...
TServer: Starting RemoteTags module ...
TServer: Starting Alarm module ...
TServer: Starting Historian module ...
TServer: Finished runtime modules
TServer: Starting Module Device (Channel: modbus)
TServer: Starting Module Dataset.
TServer: Starting Module Script.
TServer: LocalIP: 172.28.207.96 :: TServer[Dockertest, Port: 3101, WA: 3102, TCP
Bin: 3103, HTML5: 80]
```



LocalIP is the IP of the Docker image (fs), and it can be used to access the runtime via external clients (Smart and Web) and debugging tools.





Containers / Apps

Images

Images on disk

3 images

Total size: 12.94 GB

IN USE

UNUSED

Clean up...

LOCAL

REMOTE REPOSITORIES

Search

Sort by

		TAG	IMAGE ID	CREATED	SIZE
fs	IN USE	latest	f2e94edaddde	about 6 hours ago	6.37 GB
mcr.microsoft.com/dotnet...		latest	3d46f784228c	9 days ago	6.12 GB
mcr.microsoft.com/dotnet...	IN USE	4.8	5a71f579c3d9	11 months ago	6.57 GB

Connect to Remote Content

Not connected

Store things remotely

Collaborate with your team

Backup your content remotely

It's free

Sign in