

Install ABBYY FineReader Server 14 inside Docker container

1. Please check the system requirements for Docker, ABBYY FineReader Server, and Windows before you start.
2. You must use ABBYY FineReader Server 14 R2 U2 or later.
3. You must use Windows Server 2019 1809+ for the host OS.
4. The build number of the host OS (**all** the 4 portions of the build number, including the very last portion, e.g. 10.0.17763.1098) must be identical to that of your container. Otherwise, some applications (32-bit applications in particular) may fail to start with errors like the following:
 - Entrypoint not found
 - The Windows Installer Service could not be accessed

The build number of the host OS may change as a result of an update by system administrators

The list of available Microsoft images can be found here: <https://mcr.microsoft.com/v2/windows/servercore/tags/list>

5. If the host is a virtual machine, make sure that your network is configured correctly. For example, if you are using Hyper-V, be sure to hook up the host to an **External** switch. Additionally, make sure that all the required ports are open for the stations, the server, the WEB API, etc.
6. Install Docker Desktop. When installing, select the "Use Windows Containers" option.
7. As the Docker Desktop GUI is very basic, use the Docker command line instead.
8. If ABBYY FineReader Server is installed on the host, be sure to disable the FineReader Server service.
9. Virtualization support must be enabled in the host's BIOS.
For Intel machines, enable "Intel Virtualization Technology."
For AMD machines, enable "Secure Virtual Machines."
10. In the Docker settings, enable the use of Windows containers (see <https://docs.docker.com/docker-for-windows/#switch-between-windows-and-linux-containers>).

Building images

1. Copy the contents of this folder into a folder on the host (e.g. `C:\docker`).
2. Create `C:\docker\sources\FRS14 x64` and copy the FineReader Server distribution files into FRS14 x64. You should get the following structure inside FRS14 x64:

```
External Components
  DotNet
  SqlLocalDB
  VC_Redist
  VC_Redist_9
  ABBYY FineReader Server 14.0 x64.msi
```

The FineReader Server distribution files can be obtained by running the FineReader Server self-extracting distribution executable and extracting its files to the default location (`C:\Temp\ABBYY FineReader Server 14.0.X.YYY`). After the extraction

is complete, the main installer will be launched. You can cancel it as there is no need to install anything on the host machine, or you can install just the Remote Management Console and use it later to connect to the server inside the container.

3. Open all the `*.Dockerfile` files and change the build number to that of the host OS. If there is no corresponding image on the Microsoft servers, you can try using an earlier build, but that's touch-and-go.
4. Start `all.build.ps1`, `server.build.ps1`, or `station.build.ps1` to build an image.
 - `all.build.ps1` - Both the server and the station components will be placed into the container, with the station already connected to the server.
 - `server.build.ps1` - Only the server component will be placed into the container. Any stations will have to be connected to this server manually (to do this, you will need install the matching version of the Remote Administration Console on the host).
 - `station.build.ps1` - Only the station component will be placed into the container. This station will mostly behave as a regular non-containerized station.
Note: You can modify `station.run.ps1` to automatically register the station with a server having a specified location.
5. If something goes wrong during the installation, review the logs to identify the error.

Running containers

1. Once you have built the image, open `all.run.ps1` or `server.run.ps1` and provide your license token file (renamed to `token.ABBYY.token`). To find out how to activate your online license on multiple machines/containers with a license token, see this Help page: <https://help.abbyy.com/en-us/finereaderserver/14/help/cloudlicense>
2. Start `all.run.ps1`, `server.run.ps1` or `station.run.ps1`. The containers will run in background mode and will appear in the Docker Desktop Dashboard. You can also review the start log in the Docker Desktop Dashboard.
3. If no connection is established, start `*.run.ps1 -Interactive` to get the interactive console inside the container. Then try and fix the issue from that console.
4. If the interactive console won't start, try `*.run.ps1 -Recovery` to get the interactive console inside the container without any additional actions on your part. Then try and fix the issue from that console.
5. Each time you start `*.run.ps1`, the current container is deleted and a new one is created in its original post-build state. If you simply need to pause and re-start an existing container, use the Docker Desktop Dashboard.
6. To double-check, use a different machine with only the Remote Administration Console installed. In the examples below, `frs-ws-2019` is the name of the machine where the containers are running, and `frs-test` is the name of the test machine with only the Remote Administration Console installed.
7. On the `frs-test` machine, open the Remote Administration Console, add the `frs-ws-2019` server in the console, and configure the necessary settings.
8. If a server and a station are in two different containers, you can connect the station to the server. Even if both containers are on the same machine, use the name of the machine, not localhost, when setting up a connection.
9. On the `frs-test` machine, open this URL in a browser: `http://frs-ws-2019:8080/FineReaderServer14`
10. You should get a functioning WebConverter, with its server running from one container, its station running from another container, and both accessible over the network.

Adding into a container a volume associated with a directory on the host machine

To add into your container a volume associated with a directory on the host machine, start your container with the following flag:

```
-v <hostshare>:<containershare>
```

For example:

```
docker run --name $name`
--publish 3990:3990 -v g:\c:\in`
--publish 3991:3991 --publish 3992:3992 --publish 8081:8081`
$mode $image -ContainerHost $containerHost -Server -Station @switches
```

Mapping a remote share for subsequent use by a container

See <https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/persistent-storage>

This will only work with Windows Server 1709+.

To map a remote share, run this code in PowerShell:

```
$creds = Get-Credential  
New-SmbGlobalMapping -RemotePath \\contosofileserver\share1 -Credential $creds -LocalPath G:
```

where `-RemotePath` is the path to the remote share you want to map, and `-LocalPath` is the path where to add the share on the host.

Troubleshooting

Most issues can be fixed by restarting Docker, restarting the daemon, or building your container anew.

Unable to start container within PowerShell ISE**

<https://github.com/docker/for-win/issues/223>

Solution: Use regular PowerShell instead of PowerShell ISE.

When starting a container, the following error occurs:

```
docker : Error response from daemon: Ports are not available:  
listen tcp 0.0.0.0:8080: bind: An attempt was made to access a socket in a way forbidden by its access permissions.
```

This means that the port used for forwarding traffic is not available.

Solution: Specify a different port in the container's Dockerfile.

When starting a container, the following error occurs:

```
error during connect: Get http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.35/info:  
open ../pipe/docker_engine: The system cannot find the file specified.  
In the default daemon configuration on Windows, the docker client must be run elevated to connect.  
This error may also indicate that the docker daemon is not running.
```

<https://github.com/docker/for-win/issues/1825>

Problem: Typically, this error indicates that the daemon is not running or there are not enough permissions to start the daemon.

Solution: In an elevated PowerShell session, run

```
cd "C:\Program Files\Docker\Docker"  
./DockerCli.exe -SwitchDaemon
```

Failed to create endpoint on network nat: hnsCall failed in Win32: The process cannot access the file

<https://stackoverflow.com/questions/53836103/failed-to-create-endpoint-on-network-nat-hnscall-failed-in-win32-the-process-c>

Solution: In an elevated PowerShell session, run

```
Stop-Service docker
Stop-service hns
Start-service hns
Start-Service docker
docker network prune
```

Problem When processing EDOC documents this error occurs: Conversion of file 'filename' failed. Exit code: 4 (DDCModule: Unable to find the specified file.).

Solution Stop services of ABBYY FineReader Server Manager and all Processing Stations.

Open the Configuration.xml on Processing Stations machines (Default path is: C:\ProgramData\ABBYY FineReader Server 14.0\Configuration.xml).

Change for Processing Stations, which will be used for EDOC Processing, CanProcessorUseLocalServerFiles from true to false.

```
<OCRStations>
  <OCRStation ID="{GUID}" ...CanProcessorUseLocalServerFiles="true" ...>
    ...
  </OCRStation>
</OCRStations>
```

Start all services back again.