ABBYY®

ABBYY® FineReader® Engine 12

System Administrator's Guide

Table of Contents

Introduction

This document is designed to help you install ABBYY FineReader Engine 12 library in various situations and for different purposes.

The first part provides step-by-step descriptions of basic installation scenarios. The second part contains more detailed reference for license management.

You may find this guide not only in the distribution package but also online.

Choosing your installation type

Why are you installing the library?

If you are going to develop a new application that uses the library, turn to the Developer installation section.

If you have developed an application that uses the library, you need a **Runtime installation** on the end user computer.

Runtime installation

Decide if you need a **standalone** or **network** installation:

- The **standalone** installation is intended for when your application is running on just one local computer; the license information will be kept on the same computer.
- The **network** installation allows you to store the licenses on a remote server and distribute them between the workstations as needed.

The type of the Runtime license required depends on this choice: a Standalone Runtime License for a standalone installation, and a Network Runtime License for a network installation.

You also need to determine if you are going to use the built-in ABBYY command-line installer (**automatic installation**) or copy the files manually/write your own script to do it (**manual installation**). Manual installation mode can be useful if your application requires a specific functionality subset, and you want to save disk space by omitting all unnecessary resource files.

For both Standalone and Network Runtime Licenses, three types of protection are available: **online protection** — connection with ABBYY Online licensing service over the Internet for license usage, **software protection** — a serial number requiring activation, **hardware protection** — a USB dongle with the license information. There are minor differences in the installation instructions, and each scenario is described for these protection types.

Choose your type of Runtime installation from the table below.

Installation Type	**Mode of Library Installation**	
	Automatic	**Manual***
Standalone	1. Online protection 2. Software protection	1. Online protection 2. Software protection

	3. Hardware protection	
Network	1. Online protection 2. Software protection 3. Hardware protection	1. Online protection 2. Software protection

** **Note:** The Hardware key drivers may be installed only during automatic library installation.*

ABBYY FineReader Engine in a native environment

You may run ABBYY FineReader Engine inside:

- two separate Docker containers: the first one with FineReader Engine, and the second one with the Licensing Service. See details in Running ABBYY FineReader Engine 12 inside a Docker container.
- Azure Services using projects and Azure Storage for processing your files. See details in Running ABBYY FineReader Engine in Azure Services.

Basic Installation Scenarios

In all the scenarios described below except the developer installation, you need first to install your application on the workstation, then install the Licensing Service, and finally, set up the Runtime License which permits you to use the library.

Developer installation

In this scenario, you are about to develop a new application that uses ABBYY FineReader Engine library, and are installing the library on your developer's computer.

You will need:

- your Customer Project ID
- if you are using a Standalone License: the serial number of your Developer License
- if you are using a Network License: your network license server connection details
- if you are using an Online License: it is recommended to install the license token file with the library, although it is not required

Run the **installDev32.exe** (for both 32-bit and 64-bit versions of Windows) or **installDev64.exe** (for 64-bit version of Windows) file from the distribution package and follow the installation wizard's instructions:

1. Read the Software Developer License Agreement and confirm your possession of a Developer License.
2. Select the features you need to install. See the description of each feature under its name.
3. Enter your Customer Project ID and select the folder where the library must be installed. If you are using an Online License, specify the full path and the password to the license token file.
4. On the next screen:
   - If you are using a Standalone License: skip this step.
   - If you are using a Network License: select "Yes, I want to connect to Network License Server" and enter the name or address of the license server and the type of connection protocol. See License Server Administrator's Guide for details about setting up the license server.
5. Wait for installation to complete.
   - If you are using a Standalone License, activate your license. Click on the **Activate License...** button and follow the wizard instructions. If your computer has access to the Internet it only takes a few moments. For other activation options (e.g., if your computer is not connected to the Internet) see License activation.
   - If you are using a Network License, the installation is now complete. License information will be received from the network license server which you have specified earlier.
   - If you are using an Online License, you need to allow connections to ***.abbyy.com** on port **443** (HTTPS) and check that the GoDaddy root certificate is installed in the local machine version of the Trusted Root Certification Authorities certificate store on your computer (see the detailed information about the certificate on the GoDaddy website).

If you wish to additionally install a 32-bit or 64-bit version of ABBYY FineReader Engine library, run the **installDev32.exe** or **installDev64.exe** file after installing the first version. The installer will automatically choose the installation directory for the second version. Changing of a previously installed version is performed separately for both 32 and 64-bit versions.

Runtime installation

In this scenario, you are about to develop an application using the ABBYY FineReader Engine functions inside, and want to distribute this application on your workstations.

When applying scenario of Runtime installation, consider its peculiarities:

- We recommend implementing the Network scenario, which allows separate use of the Licensing Service and the ABBYY FineReader Engine library, thus ensures continuous operation of the ABBYY FineReader Engine components. As an example of how this scenario can be applied in Docker containers, see Running ABBYY FineReader Engine 12 inside a Docker container.
- When installing the ABBYY FineReader Engine library, the registry keys are created in **HKEY_LOCAL_MACHINE** and **HKEY_LOCAL_USER** subtrees.
- Protection.Developer.dll and folders with help, samples, and Code Samples Library are not installed during Runtime installation (see details in ABBYY FineReader Engine Distribution Kit section in Developer's Help).
- Installing the latest version of ABBYY FineReader Engine over the previous version is not supported. In this case, you have to reinstall ABBYY FineReader Engine.

Automatic standalone runtime installation

This section provides guides on installing your application on a local workstation in a non-interactive mode using the command line.

Online protection

In this scenario, you are using the command line installer to deploy your application to a cloud instance, private virtual machine, or a local workstation. You will need:

- an Online License token file (named like SWRTXXXXXXXXXXXXXXXXXXXX.ABBYY.ActivationToken).

Note that Internet connection is required when the application is running.

Installing the library on the host:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from installation package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.

Note: *For additional command-line options, see Command line installation options.*

The command line for your installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder"
```

This command line will install the library into **C:\MyFolder** in silent mode, without a progress bar.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the workstation before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Activating the license:

An Online License does not need activation: license information is received at the application run-time from ABBYY Online licensing services. To authorize, the application sends a license token file and a license password. You will need to place the token file where it is expected by the application. This location must be specified by the developer (default is %ProgramData%\ABBYY\SDK\12\Licenses).
At run-time, the application needs to connect to ABBYY Online licensing services periodically to receive license information and send usage statistics. No private information is exchanged with ABBYY servers.

- Verify that you allow connections to ***.abbyy.com** on port **443** (HTTPS).
  If the connection is lost, the application will stop functioning after a certain reconnection timeout is exceeded. The synchronization period and the reconnection timeout are specific parameters of each Online License.

Check that the GoDaddy root certificate is installed in the local machine version of the Trusted Root Certification Authorities certificate store. See the detailed information about the certificate on the GoDaddy website.

Note that it is not possible to use two or more Online Licenses simultaneously on the same host. When the application is running and the connection to ABBYY licensing services is working, you can view details of the currently active Online License using the License Manager Utility.

Software protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a local workstation via the command line, without any interaction with the user. You will need:

- the serial number of a Standalone Runtime License.

Installing the library on the workstation:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.

Note: *For additional command-line options, see Command line installation options.*

The command line for your installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder"
```

This command line will install the library into **C:\MyFolder** in silent mode, without a progress bar.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the workstation before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Activating the license:

After installing the library, you can find the License Manager Utility (**LicenseManager.exe**) in the **Bin** (for 32-bit operating system) or **Bin64** (for 64-bit operating system) folder of your installation.

If your workstation has access to the Internet, run **LicenseManager.exe** with the following options to activate your license:

- **/SilentActivation**
- **/SN:<*serial number*>**
  The serial number of your Standalone Runtime License.
  Specify the serial number in one of the following formats: XXXX-XXXX-XXXX-XXXX-XXXX-XXXX or XXXXXXXXXXXXXXXXXXXXXXXX.

The command line for your license activation can look like this:

```
LicenseManager.exe /SilentActivation /SN:XXXX-XXXX-XXXX-XXXX-XXXX-XXXX
```

For other activation options (e.g., if your workstation does not have access to the Internet) see License activation.

Activating the license manually via the GUI is also possible. See License Manager Utility.

Hardware protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a local workstation via the command line, without any interaction with the user. You will need:

- a USB dongle containing the parameters of your hardware protection key.

Installing the library on the workstation:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.
- **WIBUDR=Yes**
  This option installs the Wibu CodeMeter hardware key drivers.

Note: *For additional command-line options, see Command line installation options.*

The command line for your installation can look like this:

```
installRnt64.exe /passive INSTALLDIR="C:\MyFolder" WIBUDR="YES"
```

This command line will install the library into **C:\MyFolder** in silent mode. A progress bar will be displayed.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the workstation before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Connecting the hardware protection key:

Connect the hardware connection key to the USB port of the workstation. License activation is not required. Now your application can use ABBYY FineReader Engine library while the key is connected, and you can view the license details in the License Manager.

Manual standalone runtime installation

This section provides guides on manually installing your application on a local workstation.

Note that manual standalone installation is possible only if you are using Software or Online protection. Using a Wibu CodeMeter hardware key requires key drivers which are installed during the automatic runtime installation.

Online protection

In this scenario, you deploy your application to a cloud instance, private virtual machine, or a local workstation, using an installer of your choice, a shell script, or manually copying the files. You will need:

- an Online License token file (named like SWRTXXXXXXXXXXXXXXXXXXXX.ABBYY.ActivationToken).

Note that Internet connection is required when the application is running.

We recommend creating ABBYY FineReader Engine package (let us call it a **Package**) which is a folder with the files necessary for ABBYY FineReader Engine functioning and licensing.

Note: *To simplify creating a custom package, unpack the files using the /extract option for the installRnt**.exe or installLS.exe in the command line from the distribution package to a certain folder without installation. Inside it, there will be the following folders:*

1. *for installRnt**.exe:*
   - *FineReader Engine - with the ABBYY FineReader Engine files*
   - *Inc - with the FREngine.tlb (for registering of the FREngine.dll)*
   - *Licensing - with the Licensing Service files*

Use these folders and the instruction below to create your Library package.

2. *for installLS.exe:*
   - *Licensing - with the the Licensing Service files*
   - *License Server - with the License Server files*

Use these folders and the instruction below to create your License Server package.

Adding to the Package on your computer:

1. Library package. It is a folder with the ABBYY FineReader Engine files listed in ABBYY FineReader Engine Distribution Kit section in Developer's Help. The list will include:
   - Files marked as "mandatory" in ABBYY FineReader Engine Distribution Kit. They are system modules and main recognition databases.
   - Recognition databases for handprinted text, if you want to recognize handprinted text.
   - Resource files for interface languages that will be used in your application.
   - Dictionary support files for recognition languages that your application will support. If the recognition languages include languages with the Latin alphabet, make sure that you select the Univers.amd and Univers.amm files.
   - Scanning modules, scanning-specific resources and Twain modules if your application will perform scanning via the ABBYY FineReader Engine interface.
   - If your application uses the OfficeConverters module (opens input digital documents with the same methods that open the images), copy the files for internal office converter.
   - Visual Components modules and corresponding specific resources, if your application uses ABBYY FineReader Engine Visual Components.
     You can also create the list of files automatically, with the help of the **FREngineDistribution.csv** file and the unpacked FineReader Engine folder. See Developer's Help for instructions.
2. License Server package. It consists of two folders with the Licensing Service and the License Manager utility files:
   - **Licensing Service** folder with all files copied from the unpacked Licensing folder.
   - **License Manager** (**License Manager64**) folder with the following files copied from the unpacked License Server folder: AbbyyZlib.dll, Awl.dll, concrt140.dll, FineFormats.dll, FineNet.dll, FineObj.dll,

LicenseManager.exe, LicenseManager12.chm, LicensingSchema.xe, msvcp140.dll, ProductLicensingSchema.xe, Protection.dll, ProtectionRes0.dll, ProtectionResShared.dll, ProtectionUI.dll, vccorlib140.dll, vcruntime140.dll.
Use ABBYY FineReader Engine and License Server from the same package. Otherwise, compatibility is not guaranteed.

3. Other software components:
   - KB2999226 update or the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912. If you are not installing the Windows Update package, you can install the redistributable from the **External Components\VC_Redist** folder in the distribution package:
     - For 32-bit Windows, use **vc_redist.x86.exe**.
     - For 64-bit Windows, use both **vc_redist.x86.exe** and **vc_redist.x64.exe**.
   - .NET Framework 4.5 or above if you intend to use the OfficeConverters module.

Copying the Package on the workstation:

Important! *The easiest way to make sure all resource files can be located by the program is to maintain the same folder structure and include in your distribution package the SharedFiles.ini file you will find in the **Bin** (or **Bin64**) folder. If you prefer not to use this file, instead of maintaining the folder structure do the following:*

- *Copy the **Data\Resource** folder with all subfolders into **Bin** (**Bin64**) folder.*
- *Copy contents of the **Data\ExtendedDictionaries** folder directly into **Bin** (**Bin64**) folder.*
- *Copy other files contained in the **Data** folder into **Bin** (**Bin64**) folder.*

Installing the necessary software components:

1. the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.
2. .NET Framework if it is needed.

You need to copy the folder with the ABBYY FineReader Engine files from the Library Package to the directory on the workstation you expected the library to be.

Registering the dynamic libraries:

1. If your application loads the **Engine** object by means of the **IEngineLoader** interface, register **FREngine.dll** using the following command line:

```
regsvr32 /s /n /i:"<path to the Inc folder>" "<path to FREngine.dll>"
```

2. If your application uses Visual Components, register **VisualComponentsX.dll** using the following command line:

```
regsvr32 /s "<path to VisualComponentsX.dll>"
```

3. If your application uses OfficeConverters, register the .NET assembly **DcAooConverter.dll** using the following command line:

```
regasm /nologo /tlb /codebase "<path to DcAooConverter.dll>"
```

Important! *In case you have problems when using COM loaders, working with OfficeConverters or Visual Components, please check the paths to these components in the registry and, if there are errors, repeat the procedure of registration.*

Creating the folders which will be used by ABBYY FineReader Engine:

In the list below, the default values for these folders are denoted:

- folder %ProgramData%\ABBYY\SDK\12\FineReader Engine (full control permission is required)
- folder %ProgramData%\ABBYY\SDK\12\Licenses (full control permission is required for license server, optional for workstations)

Note: *To specify the license data folder during installation, see an example in Working with the LicensingSettings.xml File. You may also use the* ***InitializeEngine*** *function or method of the* ***IEngineLoader*** *interface.*

Installing the License Server:

Important! *Administrator access rights are necessary for the installation.*

1. Copy the folders from the License Server Package to the directory on the workstation you expected the License Server to be.
   Note: *We recommend you organizing ABBYY FineReader Engine files in a folder structure, easy to find and use. See the example below:*
   - *<YourApplicationFolder>/ABBYY FineReader Engine Library.*
   - *<YourApplicationFolder>/License Server/Licensing Service.*
   - *<YourApplicationFolder>/License Server/License Manager (License Manager64).*
2. Create or choose a folder in which your application searches for the license token file, and place the token file there. You may choose the %ProgramData%\ABBYY\SDK\12\Licenses, which is the default behavior, or another folder used for storing the licensing data. Everyone must have full control permissions on these folders.
3. Create a **LicensingSettings.xml** file (see Working with the LicensingSettings.xml File for details) with the following contents:

```
<LocalLicenseServer>
  <ConnectionProtocol ProtocolType="<Protocol type>" />
</LocalLicenseServer>
```

   Specify the OnlineLicensing tag only in case of using an Online License with a proxy server (in other cases do not specify this tag), and leave other settings default (do not include them in your file). The XML schema to which it must conform can be found in the **LicensingSettings.xsd** file, which is located in the **Inc** folder of your developer installation or the **CADF\Inc** folder in the distribution package.
4. Copy the configured **LicensingSettings.xml** file into the following folders:
   - **Licensing Service**.
   - **License Manager** (**License Manager64**).
   - folder with ABBYY FineReader Engine files next to FREngine.dll.
5. Run **LicensingService.exe** with the "/install" parameter:

```
LicensingService.exe /install
```

Activating the license:

An Online License does not need activation: license information is received at the application run-time from ABBYY Online licensing services. To authorize, the application sends a license token file and a license password. You will need to place the token file where it is expected by the application. This location must be specified by the developer (default is %ProgramData%\ABBYY\SDK\12\Licenses).
At run-time, the application needs to connect to ABBYY Online licensing services periodically to receive license information and send usage statistics. No private information is exchanged with ABBYY servers.

- Verify that you allow connections to ***.abbyy.com** on port **443** (HTTPS).
  If the connection is lost, the application will stop functioning after a certain reconnection timeout is exceeded. The synchronization period and the reconnection timeout are specific parameters of each Online License.

Check that the GoDaddy root certificate is installed in the local machine version of the Trusted Root Certification Authorities certificate store. See the detailed information about the certificate on the GoDaddy

website.
Note that it is not possible to use two or more Online Licenses simultaneously on the same host. When the application is running and the connection to ABBYY licensing services is working, you can view details of the currently active Online License using the License Manager Utility.

Copy the license token to a default folder on each workstation or to a folder, the path to which is used by your application in the **InitializeEngine** function.

Software protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a local workstation manually. You will need:

- the serial number of a Standalone Runtime License.

We recommend creating ABBYY FineReader Engine package (let us call it a **Package**) which is a folder with the files necessary for ABBYY FineReader Engine functioning and licensing.

Note: *To simplify creating a custom package, unpack the files using the /extract option for the installRnt**.exe or installLS.exe in the command line from the distribution package to a certain folder without installation. Inside it, there will be the following folders:*

1. *for installRnt**.exe:*
   - *FineReader Engine - with the ABBYY FineReader Engine files*
   - *Inc - with the FREngine.tlb (for registering of the FREngine.dll)*
   - *Licensing - with the the Licensing Service files*

Use these folders and the instruction below to create your Library package.

2. *for installLS.exe:*
   - *Licensing - with the the Licensing Service files*
   - *License Server - with the License Server files*

Use these folders and the instruction below to create your License Server package.

Adding to the Package on your computer:

1. Library package. It is a folder with the ABBYY FineReader Engine files listed in ABBYY FineReader Engine Distribution Kit section in Developer's Help. The list will include:
   - Files marked as "mandatory" in ABBYY FineReader Engine Distribution Kit. They are system modules and main recognition databases.
   - Recognition databases for handprinted text, if you want to recognize handprinted text.
   - Resource files for interface languages that will be used in your application.
   - Dictionary support files for recognition languages that your application will support. If the recognition languages include languages with the Latin alphabet, make sure that you select the Univers.amd and Univers.amm files.
   - Scanning modules, scanning-specific resources and Twain modules if your application will perform scanning via the ABBYY FineReader Engine interface.
   - If your application uses the OfficeConverters module (opens input digital documents with the same methods that open the images), copy the files for internal office converter.
   - Visual Components modules and corresponding specific resources, if your application uses ABBYY FineReader Engine Visual Components.
     You can also create the list of files automatically, with the help of the **FREngineDistribution.csv** file and the unpacked FineReader Engine folder. See Developer's Help for instructions.
2. License Server package. It consists of two folders with the Licensing Service and the License Manager utility files:
   - **Licensing Service** folder with all files copied from the unpacked Licensing folder.
   - **License Manager** (**License Manager64**) folder with the following files copied from the unpacked License Server folder: AbbyyZlib.dll, Awl.dll, concrt140.dll, FineFormats.dll, FineNet.dll, FineObj.dll, LicenseManager.exe, LicenseManager12.chm, LicensingSchema.xe, msvcp140.dll,

ProductLicensingSchema.xe, Protection.dll, ProtectionRes0.dll, ProtectionResShared.dll, ProtectionUI.dll, vccorlib140.dll, vcruntime140.dll.
Use ABBYY FineReader Engine and License Server from the same package. Otherwise, compatibility is not guaranteed.

3. Other software components:
   - KB2999226 update or the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912. If you are not installing the Windows Update package, you can install the redistributable from the **External Components\VC_Redist** folder in the distribution package:
     - For 32-bit Windows, use **vc_redist.x86.exe**.
     - For 64-bit Windows, use both **vc_redist.x86.exe** and **vc_redist.x64.exe**.
   - .NET Framework 4.5 or above if you intend to use the OfficeConverters module.

Copying the Package on the workstation:

Important! *The easiest way to make sure all resource files can be located by the program is to maintain the same folder structure and include in your distribution package the SharedFiles.ini file you will find in the **Bin** (or **Bin64**) folder. If you prefer not to use this file, instead of maintaining the folder structure do the following:*

- *Copy the **Data\Resource** folder with all subfolders into **Bin** (**Bin64**) folder.*
- *Copy contents of the **Data\ExtendedDictionaries** folder directly into **Bin** (**Bin64**) folder.*
- *Copy other files contained in the **Data** folder into **Bin** (**Bin64**) folder.*

Installing the necessary software components:

1. the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.
2. .NET Framework if it is needed.

You need to copy the folder with the ABBYY FineReader Engine files from the Library Package to the directory on the workstation you expected the library to be.

Registering the dynamic libraries you will need:

1. If your application loads the **Engine** object by means of the **IEngineLoader** interface, register **FREngine.dll** using the following command line:

```
regsvr32 /s /n /i:"<path to the Inc folder>" "<path to FREngine.dll>"
```

2. If your application uses Visual Components, register **VisualComponentsX.dll** using the following command line:

```
regsvr32 /s "<path to VisualComponentsX.dll>"
```

3. If your application uses OfficeConverters, register the .NET assembly **DcAooConverter.dll** using the following command line:

```
regasm /nologo /tlb /codebase "<path to DcAooConverter.dll>"
```

Important! *In case you have problems when using COM loaders, working with OfficeConverters or Visual Components, please check the paths to these components in the registry and, if there are errors, repeat the procedure of registration.*

Creating the folders which will be used by ABBYY FineReader Engine:

In the list below, the default values for these folders are denoted:

- folder %ProgramData%\ABBYY\SDK\12\FineReader Engine (full control permission is required)
- folder %ProgramData%\ABBYY\SDK\12\Licenses (full control permission is required for license server, optional for workstations)

Note: *To specify the license data folder during installation, see an example in Working with the LicensingSettings.xml File. You may also use the* ***InitializeEngine*** *function or method of the* ***IEngineLoader*** *interface.*

Installing the License Server:

Important! *Administrator access rights are necessary for the installation.*

1. Copy the folders from the License Server Package to the directory on the workstation you expected the License Server to be.
   Note: *We recommend you organizing ABBYY FineReader Engine files in a folder structure, easy to find and use. See the example below:*
   - *<YourApplicationFolder>/ABBYY FineReader Engine Library.*
   - *<YourApplicationFolder>/License Server/Licensing Service.*
   - *<YourApplicationFolder>/License Server/License Manager (License Manager64).*
2. Create a **LicensingSettings.xml** file (see Working with the LicensingSettings.xml File for details) and specify settings in the LocalLicenseServer tag if necessary. The XML schema to which it must conform can be found in the **LicensingSettings.xsd** file, which is located in the **Inc** folder of your developer installation or the **CADF\Inc** folder in the distribution package.
3. Copy the configured **LicensingSettings.xml** file into the following folders:
   - **Licensing Service**.
   - **License Manager** (**License Manager64**).
   - folder with ABBYY FineReader Engine files next to FREngine.dll.
4. Run **LicensingService.exe** with the "/install" parameter:

```
LicensingService.exe /install
```

Activating the license:

Run **LicenseManager.exe**. Click on the **Activate License...** button and follow the wizard instructions to activate your license. If your computer has access to the Internet you only need to enter the serial number and wait a few moments. For details about the License Manager, see License Manager Utility.

You can also activate the license through the command line. For details, see Working with the License Manager from the command line.

Automatic network runtime installation

This section provides guides on installing your application on a network server and workstations connected to this server using the command line installer.

Online protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a server and workstations connected to this server through a network, via the command line, without any interaction with the user. You will need:

- an Online License token file (named like SWRTXXXXXXXXXXXXXXXXXXX.ABBYY.ActivationToken).

Note that Internet connection is required when the application is running.

First, you need to install the library on the License server — a computer that will manage and distribute licenses among workstations in a network. After that, install the library on the workstations, set up the connection to the license server, and copy the license token to each workstation.

Installing the library and the Licensing Service on the license server:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.

The command line for your server installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder"
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, taking the current computer for the license server.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the license server before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Note: *If your license server will not function as one of the workstations, you only need to install the Licensing Service on it. Run the* ***installLS.exe*** *file from the* ***License Server*** *folder in the distribution package and follow the wizard instructions:*

1) *On the first screen select the path to the folder where the Licensing Service will be installed and the connection protocol it will use. Click* ***Next****.*
2) *On the second screen review your settings and click* ***Install****.*

See License Server Administrator's Guide for more details.

Activating the license on the license server:

An Online License does not need activation: license information is received at the application run-time from ABBYY Online licensing services. To authorize, the application sends a license token file and a license password. You will need to place the token file where it is expected by the application. This location must be specified by the developer (default is %ProgramData%\ABBYY\SDK\12\Licenses).
At run-time, the application needs to connect to ABBYY Online licensing services periodically to receive license information and send usage statistics. No private information is exchanged with ABBYY servers.

- Verify that you allow connections to ***.abbyy.com** on port **443** (HTTPS).
  If the connection is lost, the application will stop functioning after a certain reconnection timeout is exceeded. The synchronization period and the reconnection timeout are specific parameters of each Online License.

Check that the GoDaddy root certificate is installed in the local machine version of the Trusted Root Certification Authorities certificate store. This certificate should be on the license server with the Licensing Service installed. See the detailed information about the certificate on the GoDaddy website.

Note that it is not possible to use two or more Online Licenses simultaneously on the same host. When the application is running and the connection to ABBYY licensing services is working, you can view details of the currently active Online License using the License Manager Utility.

Installing the library on the workstations:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)

- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.
- **LICENSESRV=No**
  Indicates that the Licensing Service need not be installed.
- **SERVERNAME="<*the DNS name or IP address*>"**
  The DNS name or IP address of the license server you set up in the previous step.
  Note: *When specifying SERVERNAME, LICENSESRV is automatically set to No.*
- **LICENSEDATADIR="<*destination path*>"** (optional)
  The path to a new folder where the auxiliary information about licensing will be stored. By default, it is the %ProgramData%\ABBYY\SDK\12\Licenses or, if the **installRnt**.exe** runs with the **LICENSEDATADIR** option, LICENSEDATADIR\ABBYY_SDK_12_Licenses folder.

The command line for your workstation installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder" LICENSESRV="No"
SERVERNAME="XXX.XXX.XXX.XX"?
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, and configure the workstation to receive licenses from the license server with the IP-address XXX.XXX.XXX.XX

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on all workstations before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Copy the license token to a default folder on each workstation or to a folder, the path to which is used by your application in the **InitializeEngine** function.

The license use will still be managed by the license server, and only the server needs an Internet connection. Your workstations require only the connection to the license server.

Software protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a server and workstations connected to this server through a network, via the command line, without any interaction with the user. You will need:

- the serial number of a Network Runtime License.

First, you need to install the library and activate the license on a license server — a computer that will manage and distribute licenses among workstations in a network. After that, install the library on the workstations. Activation on workstations is not required.

Installing the library and the Licensing Service on the license server:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.

The command line for your server installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder"
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, taking the current computer for the license server.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the license server before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Note: *If your license server will not function as one of the workstations, you only need to install the Licensing Service on it. Run the* ***installLS.exe*** *file from the* ***License Server*** *folder in the distribution package and follow the wizard instructions:*

1) *On the first screen select the path to the folder where the Licensing Service will be installed and the connection protocol it will use. Click* ***Next****.*
2) *On the second screen review your settings and click* ***Install****.*

See License Server Administrator's Guide for more details.

Activating the license on the license server:

After installing the library, you can find the License Manager Utility (**LicenseManager.exe**) in the **Bin** (for 32-bit operating system) or **Bin64** (for 64-bit operating system) folder of your installation.

If the network has access to the Internet, run **LicenseManager.exe** with the following options to activate your license:

- **/SilentActivation**
- **/SN:<*serial number*>**
  The serial number of your Standalone Runtime License.
  Specify the serial number in one of the following formats: XXXX-XXXX-XXXX-XXXX-XXXX-XXXX or XXXXXXXXXXXXXXXXXXXXXXXX.

The command line for your license activation can look like this:

```
LicenseManager.exe /SilentActivation /SN:XXXX-XXXX-XXXX-XXXX-XXXX-XXXX
```

For other activation options (e.g., if your network does not have access to the Internet) see License Activation.

Activating the license manually via the GUI is also possible. See License Manager Utility.

Installing the library on the workstations:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.
- **LICENSESRV=No**
  Indicates that the Licensing Service need not be installed.
- **SERVERNAME="<*the DNS name or IP address*>"**
  The DNS name or IP address of the license server you set up in the previous step.
  Note: *When specifying SERVERNAME, LICENSESRV is automatically set to No.*
- **LICENSEDATADIR="<*destination path*>"** (optional)
  The path to a new folder where the auxiliary information about licensing will be stored. By default, it is the %ProgramData%\ABBYY\SDK\12\Licenses folder.

The command line for your workstation installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder" LICENSESRV="No"
SERVERNAME="XXX.XXX.XXX.XX"
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, and configure the workstation to receive licenses from the license server with the IP-address XXX.XXX.XXX.XX

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on all workstations before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Note: *You also have the option of setting up a two-server configuration which will allow one of the license servers to act as a backup to another in case it goes offline. For details, see Setting up a redundant license server configuration.*

Hardware protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a server and workstations connected to this server through a network, via the command line, without any interaction with the user. You will need:

- a USB dongle containing the parameters of your hardware protection key.

First, you need to install the library on a license server — a computer that will manage and distribute licenses among workstations in a network. After that, install the library on the workstations.

Installing the library and the Licensing Service on the license server:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.
- **WIBUDR=Yes**
  This option installs the Wibu CodeMeter hardware key drivers.

Note: *For additional command-line options, see Command line installation options.*

The command line for your server installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder" WIBUDR="Yes"
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, and taking the current computer for the license server.

Important! *If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on the license server before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

Note: *If your license server will not function as one of the workstations, you only need to install the Licensing Service on it. Run the **installLS.exe** file from the **License Server** folder in the distribution package and follow the wizard instructions:*

1) *On the first screen select the path to the folder where the Licensing Service will be installed and the connection protocol it will use. Click **Next**.*
2) *On the second screen review your settings and click **Install**.*

In this case, you will have to install the Wibu CodeMeter key drivers yourself, as automatic installation is not provided in the Licensing Service installer. See License Server Administrator's Guide for more details.

Connecting the hardware protection key:

Connect the hardware protection key to the USB port of the server. License activation is not required.

Installing the library on the workstations:

Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library will be installed.
- **LICENSESRV=No**
  Indicates that the Licensing Service need not be installed.
- **SERVERNAME="<*the DNS name or IP address*>"**
  The DNS name or IP address of the license server you set up in the previous step.
  ***Note:** When specifying SERVERNAME, LICENSESRV is automatically set to No.*
- **LICENSEDATADIR="<*destination path*>"** (optional)
  The path to a new folder where the auxiliary information about licensing will be stored. By default, it is the %ProgramData%\ABBYY\SDK\12\Licenses folder.

The command line for your workstation installation can look like this:

```
installRnt64.exe /quiet INSTALLDIR="C:\MyFolder" LICENSESRV="No"
SERVERNAME="XXX.XXX.XXX.XX"
```

This command line will install the library into C:\MyFolder in silent mode, without a progress bar, and configure the workstation to receive licenses from the license server with the IP-address XXX.XXX.XXX.XX

***Important!** If you don't use "RegisterCOM=No" in your command line, please check the correctness of the path to FREngine.dll in the registry on all workstations before launching your application. Consider the same check for the OfficeConverters and Visual Components modules in case of their installation. If there are errors in the paths, repeat the procedure of registration.*

***Note:** You also have the option of setting up a two-server configuration which will allow one of the license servers to act as a backup to another in case it goes offline. For details, see Setting up a redundant license server configuration.*

Manual network runtime installation

This section provides guides on manually installing your application on a network server and workstations connected to this server.

Note that manual network installation is possible only if you are using Software or Online protection. Using a Wibu CodeMeter hardware key requires key drivers which are installed during the automatic network runtime installation.

Online protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a server and workstations connected to this server through a network manually. You will need:

- an Online License token file (named like SWRTXXXXXXXXXXXXXXXXXXX.ABBYY.ActivationToken).

Note that the License server requires an Internet connection when the application is running.

To implement this scenario, you need to install the Licensing Service on the license server — a computer that will manage and distribute licenses among workstations in a network. After that, install the library on the workstations, set up the connection to the license server, and copy the license token to each workstation.

Note: *To simplify creating a custom package, unpack the files using the /extract option for the installRnt**.exe or installLS.exe in the command line from the distribution package to a certain folder without installation. Inside it, there will be the following folders:*

1. *for installRnt**.exe:*
   - *FineReader Engine - with the ABBYY FineReader Engine files*
   - *Inc - with the FREngine.tlb (for registering of the FREngine.dll)*
   - *Licensing - with the the Licensing Service files*

Use these folders and the instruction below to create your Library package.

2. *for installLS.exe:*
   - *Licensing - with the the Licensing Service files*
   - *License Server - with the License Server files*

Use these folders and the instruction below to create your License Server package.

Creating the ABBYY FineReader Engine packages on your computer:

1. Library package. It is a folder with the ABBYY FineReader Engine files listed in ABBYY FineReader Engine Distribution Kit section in Developer's Help. The list will include:
   - Files marked as "mandatory" in ABBYY FineReader Engine Distribution Kit. They are system modules and main recognition databases.
   - Recognition databases for handprinted text, if you want to recognize handprinted text.
   - Resource files for interface languages that will be used in your application.
   - Dictionary support files for recognition languages that your application will support. If the recognition languages include languages with the Latin alphabet, make sure that you select the Univers.amd and Univers.amm files.
   - Scanning modules, scanning-specific resources and Twain modules if your application will perform scanning via the ABBYY FineReader Engine interface.
   - If your application uses the OfficeConverters module (opens input digital documents with the same methods that open the images), copy the files for internal office converter.
   - Visual Components modules and corresponding specific resources, if your application uses ABBYY FineReader Engine Visual Components.

     You can also create the list of files automatically, with the help of the **FREngineDistribution.csv** file and the unpacked FineReader Engine folder. See Developer's Help for instructions.
2. License Server package. It consists of two folders with the Licensing Service and the License Manager utility files:
   - **Licensing Service** folder with all files copied from the unpacked Licensing folder.
   - **License Manager** (**License Manager64**) folder with the following files copied from the unpacked License Server folder: AbbyyZlib.dll, Awl.dll, concrt140.dll, FineFormats.dll, FineNet.dll, FineObj.dll, LicenseManager.exe, LicenseManager12.chm, LicensingSchema.xe, msvcp140.dll, ProductLicensingSchema.xe, Protection.dll, ProtectionRes0.dll, ProtectionResShared.dll, ProtectionUI.dll, vccorlib140.dll, vcruntime140.dll.

     Use ABBYY FineReader Engine and License Server from the same package. Otherwise, compatibility is not guaranteed.
3. Other software components:

- KB2999226 update or the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912. If you are not installing the Windows Update package, you can install the redistributable from the **External Components\VC_Redist** folder in the distribution package:
  - For 32-bit Windows, use **vc_redist.x86.exe**.
  - For 64-bit Windows, use both **vc_redist.x86.exe** and **vc_redist.x64.exe**.
- .NET Framework 4.5 or above if you intend to use the OfficeConverters module.

Installing the Licensing Service on the license server:

Important! *Administrator access rights are necessary for the installation.*

1. Copy the License Server package and the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912 on the license server.
2. Install the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.
3. Copy the folders from the License Server package to the directory on the license server you expected the Licensing Service to be.
   Note: *We recommend you organizing ABBYY FineReader Engine files in a folder structure, easy to find and use. See the example below:*
   - *<YourApplicationFolder>/License Server/Licensing Service.*
   - *<YourApplicationFolder>/License Server/License Manager (License Manager64).*
4. Create a folder for storing the licensing data (%ProgramData%\ABBYY\SDK\12\Licenses folder by default). Every user running your application must have full control permissions on this folder.
5. Create a **LicensingSettings.xml** file. The XML schema to which it must conform can be found in the **LicensingSettings.xsd** file, which is located in the **Inc** folder of your developer installation or the **CADF\Inc** folder in the distribution package.In this file you need to set up the type of connection protocol your license server will use. The supported types are LocalInterprocessCommunication, NamedPipes, TCP/IP, and Sockets:

```
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="[your protocol type]"/>
</LocalLicenseServer>
```

   By default, the 3022 TCP/IP port is used. If this port is closed on your server, specify another open port:

```
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="TCP/IP" EndPointName="1025"/>
</LocalLicenseServer>
```

6. Copy the configured **LicensingSettings.xml** file into the following folders:
   - **Licensing Service**.
   - **License Manager** (**License Manager64**).
7. Run **LicensingService.exe** with the "/install" parameter:

```
LicensingService.exe /install
```

Activating the license on the license server:

An Online License does not need activation: license information is received at the application run-time from ABBYY Online licensing services. To authorize, the application sends a license token file and a license password. You will need to place the token file where it is expected by the application on all workstations. This location must be specified by the developer (default is %ProgramData%\ABBYY\SDK\12\Licenses). At run-time, the application needs to connect to ABBYY Online licensing services periodically to receive license information and send usage statistics. No private information is exchanged with ABBYY servers.

- Verify that you allow connections to ***.abbyy.com** on port **443** (HTTPS).
  If the connection is lost, the application will stop functioning after a certain reconnection timeout is exceeded. The synchronization period and the reconnection timeout are specific parameters of each Online License.

Check that the GoDaddy root certificate is installed in the local machine version of the Trusted Root Certification Authorities certificate store. This certificate should be on the license server with the Licensing Service installed. See the detailed information about the certificate on the GoDaddy website.

Note that it is not possible to use two or more Online Licenses simultaneously on the same host. When the application is running and the connection to ABBYY licensing services is working, you can view details of the currently active Online License using the License Manager Utility.

Installing the ABBYY FineReader Engine library on the workstations:

Note: *If the license server will also function as a workstation, perform this step for the server too.*

1. Copy the Library package and software components on all your workstations.

Important! *The easiest way to make sure all resource files can be located by the program is to maintain the same folder structure and include in your distribution package the SharedFiles.ini file you will find in the **Bin** (or **Bin64**) folder. If you prefer not to use this file, instead of maintaining the folder structure do the following:*

- *Copy the **Data\Resource** folder with all subfolders into **Bin** (**Bin64**) folder.*
- *Copy contents of the **Data\ExtendedDictionaries** folder directly into **Bin** (**Bin64**) folder.*
- *Copy other files contained in the **Data** folder into **Bin** (**Bin64**) folder.*

2. Install the necessary software components:
   - the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.
   - .NET Framework if it is needed.
3. Copy the folder with the ABBYY FineReader Engine files from the Library Package to the directory on the workstation you expected the library to be.
4. Register the dynamic libraries you will need:
   - If your application loads the **Engine** object by means of the **IEngineLoader** interface, register **FREngine.dll** using the following command line:

```
regsvr32 /s /n /i:"<path to the Inc folder>" "<path to FREngine.dll>"
```

   - If your application uses Visual Components, register **VisualComponentsX.dll** using the following command line:

```
regsvr32 /s "<path to VisualComponentsX.dll>"
```

   - If your application uses OfficeConverters, register the .NET assembly **DcAooConverter.dll** using the following command line:

```
regasm /nologo /tlb /codebase "<path to DcAooConverter.dll>"
```

Important! *In case you have problems when using COM loaders, working with OfficeConverters or Visual Components, please check the paths to these components in the registry and, if there are errors, repeat the procedure of registration.*

5. Create the folders which will be used by ABBYY FineReader Engine. In the list below, the default values for these folders are denoted:
   - folder %ProgramData%\ABBYY\SDK\12\FineReader Engine (full control permission is required)
   - folder %ProgramData%\ABBYY\SDK\12\Licenses (full control permission is optional)

Note: *To specify the license data folder during installation, see an example in Working with the LicensingSettings.xml File. You may also use the **InitializeEngine** function or method of the **IEngineLoader** interface.*

6. Copy the license token to a default folder on each workstation or to a folder, the path to which is used by your application in the **InitializeEngine** function.

Creating a LicensingSettings.xml file:

Create a LicensingSettings.xml file, setting up the same type of connection protocol you specified for the server and the address of the server:

```
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="[your protocol type]"/>
</LicensingServers>
```

If you set another TCP/IP port for the server, repeat the same setting here.

```
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LicensingServers>
```

Copying the configured LicensingSettings.xml file:

Copy the configured LicensingSettings.xml file into the folder with ABBYY FineReader Engine files next to FREngine.dll on every workstation:

Note: *If the license server will also function as a workstation, the* ***LicensingSettings.xml*** *file for the server must contain both LicensingServers and LocalLicenseServer tags similar to those specified above. The contents of the file will look like this:*

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="[your protocol type]"/>
</LicensingServers>
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="[your protocol type]"/>
</LocalLicenseServer>
</LicensingSettings>
```

Software protection

In this scenario, you are installing your application which uses ABBYY FineReader Engine library on a server and workstations connected to this server through a network manually. You will need:

- the serial number of a Network Runtime License.

To implement this scenario, you need to install the Licensing Service on the license server — a computer that will manage and distribute licenses among workstations in a network. After that, install the library on the workstations, set up the connection to the license server, and copy the license token to each workstation.

Note: *To simplify creating a custom package, unpack the files using the /extract option for the installRnt**.exe or installLS.exe in the command line from the distribution package to a certain folder without installation. Inside it, there will be the following folders:*

1. *for installRnt**.exe:*
   - *FineReader Engine - with the ABBYY FineReader Engine files*
   - *Inc - with the FREngine.tlb (for registering of the FREngine.dll)*
   - *Licensing - with the the Licensing Service files*

Use these folders and the instruction below to create your Library package.

2. *for installLS.exe:*
   - *Licensing - with the the Licensing Service files*
   - *License Server - with the License Server files*

Use these folders and the instruction below to create your License Server package.

Creating the ABBYY FineReader Engine packages on your computer:

1. Library package. It is a folder with the ABBYY FineReader Engine files listed in ABBYY FineReader Engine Distribution Kit section in Developer's Help. The list will include:
   - Files marked as "mandatory" in ABBYY FineReader Engine Distribution Kit. They are system modules and main recognition databases.
   - Recognition databases for handprinted text, if you want to recognize handprinted text.
   - Resource files for interface languages that will be used in your application.
   - Dictionary support files for recognition languages that your application will support. If the recognition languages include languages with the Latin alphabet, make sure that you select the Univers.amd and Univers.amm files.
   - Scanning modules, scanning-specific resources and Twain modules if your application will perform scanning via the ABBYY FineReader Engine interface.
   - If your application uses the OfficeConverters module (opens input digital documents with the same methods that open the images), copy the files for internal office converter.
   - Visual Components modules and corresponding specific resources, if your application uses ABBYY FineReader Engine Visual Components.

     You can also create the list of files automatically, with the help of the **FREngineDistribution.csv** file and the unpacked FineReader Engine folder. See Developer's Help for instructions.
2. License Server package. It consists of two folders with the Licensing Service and the License Manager utility files:
   - **Licensing Service** folder with all files copied from the unpacked Licensing folder.
   - **License Manager** (**License Manager64**) folder with the following files copied from the unpacked License Server folder: AbbyyZlib.dll, Awl.dll, concrt140.dll, FineFormats.dll, FineNet.dll, FineObj.dll, LicenseManager.exe, LicenseManager12.chm, LicensingSchema.xe, msvcp140.dll, ProductLicensingSchema.xe, Protection.dll, ProtectionRes0.dll, ProtectionResShared.dll, ProtectionUI.dll, vccorlib140.dll, vcruntime140.dll.

     Use ABBYY FineReader Engine and License Server from the same package. Otherwise, compatibility is not guaranteed.
3. Other software components:
   - KB2999226 update or the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912. If you are not installing the Windows Update package, you can install the redistributable from the **External Components\VC_Redist** folder in the distribution package:
     - For 32-bit Windows, use **vc_redist.x86.exe**.
     - For 64-bit Windows, use both **vc_redist.x86.exe** and **vc_redist.x64.exe**.
   - .NET Framework 4.5 or above if you intend to use the OfficeConverters module.

Installing the Licensing Service on the license server:

Important! *Administrator access rights are necessary for the installation.*

1. Copy the License Server package and the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912 on the license server.
2. Install the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.

3. Copy the folders from the License Server package to the directory on the license server you expected the Licensing Service to be.
   Note: *We recommend you organizing ABBYY FineReader Engine files in a folder structure, easy to find and use. See the example below:*
   - *<YourApplicationFolder>/License Server/Licensing Service.*
   - *<YourApplicationFolder>/License Server/License Manager (License Manager64).*
4. Create a folder for storing the licensing data (%ProgramData%\ABBYY\SDK\12\Licenses folder by default). Every user running your application must have full control permissions on this folder. See an example of setting of license data folder in Working with the LicensingSettings.xml File.
5. Create a **LicensingSettings.xml** file. The XML schema to which it must conform can be found in the **LicensingSettings.xsd** file, which is located in the **Inc** folder of your developer installation or the **CADF\Inc** folder in the distribution package. In this file you need to set up the type of connection protocol your license server will use. The supported types are LocalInterprocessCommunication, NamedPipes, TCP/IP, and Sockets:

```
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="[your protocol type]"/>
</LocalLicenseServer>
```

   By default, the 3022 TCP/IP port is used. If this port is closed on your server, specify another open port:

```
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="TCP/IP" EndPointName="1025"/>
</LocalLicenseServer>
```

6. Copy the configured **LicensingSettings.xml** file into the following folders:
   - **Licensing Service**.
   - **License Manager** (**License Manager64**).
7. Run **LicensingService.exe** with the "/install" parameter:

```
LicensingService.exe /install
```

Note: *You also have the option of setting up a two-server configuration which will allow one of the license servers to act as a backup to another in case it goes offline. For details, see Setting up a redundant license server configuration.*

Activating the license on the license server:

Run **LicenseManager.exe**. Click on the **Activate License...** button and follow the wizard instructions to activate your license. If your server has access to the Internet you only need to enter the serial number and wait a few moments. For details about the License Manager, see License Manager Utility.

You can also activate the license through the command line. For details, see Working with the License Manager from the command line.

Installing the ABBYY FineReader Engine library on the workstations:

Note: *If the license server will also function as a workstation, perform this step for the server too.*

1. Copy the Library package and software components on all your workstations.

Important! *The easiest way to make sure all resource files can be located by the program is to maintain the same folder structure and include in your distribution package the SharedFiles.ini file you will find in the **Bin** (or **Bin64**) folder. If you prefer not to use this file, instead of maintaining the folder structure do the following:*

- *Copy the **Data\Resource** folder with all subfolders into **Bin** (**Bin64**) folder.*
- *Copy contents of the **Data\ExtendedDictionaries** folder directly into **Bin** (**Bin64**) folder.*

- *Copy other files contained in the **Data** folder into **Bin** (**Bin64**) folder.*

2. Install the necessary software components:
   - the Microsoft Visual C++ 2015-2019 Redistributable 14.28.29912.
   - .NET Framework if it is needed.
3. Copy the folder with the ABBYY FineReader Engine files from the Library Package to the directory on the workstation you expected the library to be.
4. Register the dynamic libraries you will need:
   - If your application loads the **Engine** object by means of the **IEngineLoader** interface, register **FREngine.dll** using the following command line:

```
regsvr32 /s /n /i:"<path to the Inc folder>" "<path to FREngine.dll>"
```

   - If your application uses Visual Components, register **VisualComponentsX.dll** using the following command line:

```
regsvr32 /s "<path to VisualComponentsX.dll>"
```

   - If your application uses OfficeConverters, register the .NET assembly **DcAooConverter.dll** using the following command line:

```
regasm /nologo /tlb /codebase "<path to DcAooConverter.dll>"
```

Important! *In case you have problems when using COM loaders, working with OfficeConverters or Visual Components, please check the paths to these components in the registry and, if there are errors, repeat the procedure of registration.*

5. Create the folders which will be used by ABBYY FineReader Engine. In the list below, the default values for these folders are denoted:
   - folder %ProgramData%\ABBYY\SDK\12\FineReader Engine (full control permission is required)
   - folder %ProgramData%\ABBYY\SDK\12\Licenses (full control permission is optional)

Note: *To specify the license data folder during installation, see an example in Working with the LicensingSettings.xml File. You may also use the **InitializeEngine** function or method of the **IEngineLoader** interface.*

Creating a LicensingSettings.xml file:

Create a LicensingSettings.xml file, setting up the same type of connection protocol you specified for the server and the address of the server:

```
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="[your protocol type]"/>
</LicensingServers>
```

If you set another TCP/IP port for the server, repeat the same setting here.

```
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LicensingServers>
```

Copying the configured LicensingSettings.xml file:

Copy the configured LicensingSettings.xml file into the folder with ABBYY FineReader Engine files next to FREngine.dll on every workstation:

Note: *If the license server will also function as a workstation, the* ***LicensingSettings.xml*** *file for the server must contain both LicensingServers and LocalLicenseServer tags similar to those specified above. The contents of the file will look like this:*

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
<LicensingServers>
   <MainNetworkLicenseServer ServerAddress="[your server address]"
ProtocolType="[your protocol type]"/>
</LicensingServers>
<LocalLicenseServer>
    <ConnectionProtocol ProtocolType="[your protocol type]"/>
</LocalLicenseServer>
</LicensingSettings>
```

Uninstalling the library

Uninstalling the developer installation

For developer installation, use the **Uninstall or change a program** Windows menu (**Control Panel > Programs and features**).

Uninstalling the automatic runtime installation

Automatic runtime uninstallation is the same for standalone and network installations. For network installation, perform the required steps on every workstation.

If ABBYY FineReader Engine library was installed using a built-in installer (automatic runtime installation), it must be uninstalled also with the help of that installer. Run the **installRnt32.exe** (for both 32-bit and 64-bit versions of Windows) or **installRnt64.exe** (for 64-bit version of Windows) file from the distribution package with the following options:

- **/quiet** (or **/passive** if you want a progress bar to be displayed)
- **INSTALLDIR="<*destination path*>"**
  The path to the folder where the ABBYY FineReader Engine library was installed.
- **DeinstallRuntime=Yes**
  Indicates that the runtime installation must be removed.

Uninstalling the manual runtime installation

Manual runtime uninstallation is the same for standalone and network installations. For network installation, perform the required steps on every workstation, and those concerned with Licensing Service on the license server.

Follow the instructions:

1. Unregister the dynamic libraries (**FREngine.dll**, **VisualComponentsX.dll**, **DcAooConverter.dll**) if they were registered during manual installation.

```
regsvr32.exe /u "<path to FREngine.dll>"
regsvr32.exe /u "<path to VisualComponentsX.dll>"
regasm /u "<path to DcAooConverter.dll>"
```

Uninstall the Licensing Service if it was installed (i.e., from the workstation for standalone installation, and from the license server for network installation):

```
LicensingService.exe /uninstall
```

Then remove the folder with the Licensing Service files.

2. Remove all files of ABBYY FineReader Engine library. They may be located in **Bin**, **Bin64** (only for 64-bit operating systems), and **Data** folders in your installation folder.
3. Delete the folders used to store ABBYY FineReader Engine data (%ProgramData%\ABBYY\SDK\12\FineReader Engine and %ProgramData%\ABBYY\SDK\12\Licenses by default) or their analogs, if the paths to these folders were changed.

Specifications

Workstation requirements

The system requirements for the workstation on which your application is installed are the same for all scenarios:

Hardware

- PC with x86-compatible processor (1 GHz or higher)
- Memory:
  - for processing one-page documents — minimum 400 MB RAM, recommended 1 GB RAM
  - for processing multi-page documents — minimum 1 GB RAM, recommended 1,5 GB RAM
  - for parallel processing — 450 + (number of cores) × 350 MB RAM
  - for parallel processing of documents in Arabic or CJK languages — 750 + (number of cores) × 850 MB RAM
- Hard disk space:
  - for library installation:
    - 1600 MB for core functionality
    - additional 400 MB for the OfficeConverters module
  - for program operation:
    - 100 MB for running the program
    - additional 15Mb for every page when processing a multi-page document
- 100% TWAIN-compatible scanner, digital camera, or fax modem — for scanning or image import only
- Video card and monitor (minimum resolution 1024×768 — for pattern training, dictionary editing, scanning with a GUI displayed, Visual Components)
- Keyboard, mouse or other input device

Operating system

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows 11
- Windows 10
- Windows Server 2012 R2 with the April 2014 update rollup (KB2919355)
- Windows 8.1 with the April 2014 update rollup (KB2919355)
- Windows Server 2012

ABBYY FineReader Engine has been tested on the following cloud computing platforms:

- Microsoft Azure:
  - Azure App Services
  - Azure Cloud Services
  - Azure Service Fabric

  - Azure Virtual Machines
- Amazon EC2

ABBYY FineReader Engine has been tested in the following virtual environments:

- Microsoft Hyper-V Server 2012
- Microsoft Hyper-V Server 2012 R2
- Microsoft Hyper-V Server 2016
- Oracle VM VirtualBox 6.1.32
- Parallels Desktop for Mac 16.1.3
- VMware ESXi 6.7
- VMware Workstation Player 16.2.3
- VMware Workstation Pro 16.2.2

ABBYY FineReader Engine can also be run in a Docker container on supported platforms.

Other software components

ABBYY FineReader Engine installer uses Windows Installer XML Toolset (WiX), that requires .NET Framework for its operation:

- For Developer installation:
  - .NET Framework 4.6.1 and above is required
- For Runtime installation:
  - automatic installation - .NET Framework 4.6.1 and above is required
  - manual installation with the OfficeConverters module - .NET Framework 4.5 and above is required
  - manual installation without the OfficeConverters module - .NET Framework is not required

The following versions of .NET Framework are supported:

- 4.8, 4.7, 4.6, 4.5, 4.0, 3.5

The following components should be installed:

- Microsoft® Internet Explorer 8.0 or higher
- .NET Framework 4.5 — required for opening digital documents

For correct operation of the font detection mechanism the fonts needed for the languages you use should be installed; for the list of recommended font families see the description of the **FontNamesFiltersEnum** enumeration in the Developer's Help.

For ABBYY FineReader Engine Visual Components only:

- Microsoft Windows Common Controls must have version 6.0 or later
- If you use Microsoft Visual Studio 2010 and .NET Framework 4.0 for development of your application, you may need to install COM Interop assemblies for Visual Components manually. Refer to Using Visual Components in Different Versions of Visual Studio section in the Developer's Help for details.

Working with the .NET Core wrapper

ABBYY FineReader Engine .NET Core wrapper supports working with the following versions of .NET Core:

- 3.1

Working with the Java wrapper

ABBYY FineReader Engine Java wrapper has been tested on the following Java Development Kits:

- for 32-bit operating systems:
  - Oracle Java SE Development Kit 8u331 x86
- for 64-bit operating systems:
  - Open JDK 17.0.2 x64
  - Open JDK 18.0.1 x64
  - Oracle Java SE Development Kit 8u331 x64

- Oracle Java SE Development Kit 11.0.15 (LTS) x64
- Oracle Java SE Development Kit 17.0.3 x64
- Oracle Java SE Development Kit 18.0.1 x64

Permissions

Full control permissions for the following folders are required:

- %TEMP% folder
- folder %ProgramData%\ABBYY\SDK\12\FineReader Engine
- folder %ProgramData%\ABBYY\SDK\12\Licenses (required for license server, optional for workstations)

All users running your application should have read&execute and write permissions to:

- %ProgramData%\ABBYY folder.

The following registry branches should be accessible from the workstation:

- "HKEY_CURRENT_USER\Software\ABBYY\SDK\12\FineReader Engine" — full control
- "HKEY_CURRENT_USER\Software\ABBYY\SDK\12" — full control for installation only
- "HKEY_LOCAL_MACHINE\Software\ABBYY\SDK\12" — full control for installation only

License server requirements

- PC with x86-compatible processor (1 GHz or higher)
- Operating system:
  - Windows Server 2022
  - Windows Server 2019
  - Windows Server 2016
  - Windows 11
  - Windows 10
  - Windows Server 2012 R2 with the April 2014 update rollup (KB2919355)
  - Windows 8.1 with the April 2014 update rollup (KB2919355)
  - Windows Server 2012

  ABBYY SDK 12 Licensing Service has been tested in the following virtual environments:

  - Microsoft Hyper-V Server 2012
  - Microsoft Hyper-V Server 2012 R2
  - Microsoft Hyper-V Server 2016
  - Oracle VM VirtualBox 6.1.32
  - Parallels Desktop for Mac 16.1.3
  - VMware ESXi 6.7
  - VMware Workstation Player 16.2.3
  - VMware Workstation Pro 16.2.2
- 25 MB of free hard disk space
- Full control access to a folder with licensing information (%ProgramData%\ABBYY\SDK\12\Licenses folder by default)
- For activation/deactivation of the License:
  - allowed connections to *.abbyy.com
  - port: 80 (except Online License), 443 (HTTPS)
  - protocol: http
  - GoDaddy Trusted Root Certification Authority (for Online License only)

Command line installation options

This section describes all supported command-line options of the built-in installer.

Option	Default Value	Description
INSTALLDIR="<*destination path*>"		The path to the folder where the ABBYY FineReader Engine library will be installed. ***Important!*** *This parameter is required.*
DeinstallRuntime=Yes | No	No	Specifies if the existing runtime installation of the library must be uninstalled.
MODULES=<*list of library modules*> | None | All	All	The list of library modules that you want to install. See the list of available modules below. The modules must be separated with commas (,). For example, MODULES=ICR,PDF,BasicLang. If you do not want to install any of these modules, set this option to None. If you want to install all the modules, set this option to All. ***Important!*** *If you intend to open the office formats, install the OfficeConverters module together with the PDF module.*
WIBUDR=Yes | No	No	Specifies whether hardware key drivers (CodeMeter) must be installed.
LICENSESRV=Yes | No	Yes	Specifies whether Licensing Service must be installed. If WIBUDR=Yes, the Licensing Service is installed automatically and cannot be excluded from the installation. The Licensing Service is installed to the **%ProgramFiles%\Common Files\ABBYY\SDK\12\Licensing** folder on a 32-bit operating system, and **%ProgramFiles(x86)%\Common**

Option	Default Value	Description
		Files\ABBYY\SDK\12\Licensing folder on a 64-bit operating system. ***Important!*** *Licensing Service is required for working with ABBYY FineReader Engine library, and for correct installation you must do one of the following:* • *If you have a Standalone or an Online license, you should install the Licensing Service on the same computer on which ABBYY FineReader Engine is installed. Set this parameter to Yes.* • *In the case of Network license, you should install the Licensing Service beforehand on a network server — a computer which will manage and distribute licenses among workstations in a network. Set this parameter to No and specify the address of the server in the SERVERNAME parameter during installation on your workstation. If you set this parameter to No and omit the server address, an error will be returned and installation will not be completed.*
SERVERNAME=<*the DNS name or IP address*>		The DNS name or IP address of the computer where the Licensing Service is installed. ***Note:*** *When specifying SERVERNAME, LICENSESRV is automatically set to No.*
LICENSEDATADIR="<*destination path*>"		The path to the folder where auxiliary information about licensing will be stored. By default, it is the %ProgramData%\ABBYY\SDK\12\Licenses. You may change the folder with this option when performing Runtime installation or after, through the

Option	Default Value	Description
		InitializeEngine function.
RegisterCOM=Yes | No | Force	Yes	Specifies whether the COM wrappers should be registered. The registration is performed by calling the DllRegisterServer function from FREngine.dll. If the library is already registered with the different path: • if this option is set to Yes, an error is returned; • if this option is set to Force, the library is re-registered to the new path and the installation is continued. If the MODULES option contains VC or is set to All, the library must be registered. Setting this option to No will cause an error and the installation will not be completed.
/quiet		Silent mode without a progress bar. This is mandatory option because the runtime installation can be performed only in silent mode. Use the **/passive** option if you want the progress bar to be displayed during the installation. No other dialog boxes will be displayed. If these options are not specified, the /passive option is used by default.
/extract		Specifies whether the ABBYY FineReader Engine files should be extracted from the distribution package to a certain folder without installation. No entries will be made in the Windows registry, services, or package cache. However, the procedure requires administrator rights. Use this option to unpack the ABBYY

Option	Default Value	Description
		FineReader Engine files and create your custom package.

Library modules

Each library module determines the license modules which must be available in a Runtime License, and resource files which will be installed. The license modules and resource files corresponding to each library module are listed in the table below:

Library module	The license modules which must be available in a Runtime License	The resource files which will be installed
ICR	Index, ICR, OMR	The files for recognition of checkmarks and handprinted text.
PDF	PDF Opening	The files which are listed in the ABBYY FineReader Engine Distribution Kit: PDF section as mandatory for PDF opening.
VC	Scanning, User Patterns Training	The files for scanning and user patterns training.
Classification	Document Classification	The files for document classification.
BCR	Business Card Recognition	The files for business card recognition.
BasicLang	Natural	The files for basic predefined languages, except the ones defined in special groups.
DataCaptureLang	Natural for Data Capture	The files for predefined languages for special language units.

Library module	The license modules which must be available in a Runtime License	The resource files which will be installed
OfficeConverters	Office Formats Opening, PDF Opening	The files for opening digital documents.
CompareDocuments	Compare Documents	The files for the documents comparison.
Arabic	Arabic	The files for recognition of texts in Arabic language.
Burmese	Burmese	The files for recognition of texts in Burmese language.
Chinese	Chinese	The files for recognition of texts in Chinese language.
Farsi	Farsi	The files for recognition of texts in Farsi language.
Japanese	Japanese	The files for recognition of texts in Japanese language.
Korean	Korean	The files for recognition of texts in Korean language.
FRXIX	FineReader XIX	The files for recognition of texts in Old European languages.
Hebrew	Hebrew	The files for recognition of texts in Hebrew and Yiddish languages.
Thai	Thai	The files for recognition of texts in Thai language.
Vietnamese	Vietnamese	The files for recognition of texts in Vietnamese language.

Example

```
installRnt64.exe /quiet MODULES=PDF,ICR WIBUDR=Yes INSTALLDIR="C:\MyFolder"
```

This command line will install the PDF and ICR library modules to C:\MyFolder in silent mode, without a progress bar. Hardware key drivers will be installed.

Licensing Reference

License Activation

ABBYY FineReader Engine 12 activation depends on the type of license. If you have a Standalone license, you should activate ABBYY FineReader Engine on the same computer where it is installed. In the case of a Network license, you should activate ABBYY FineReader Engine on a network server — a computer which will manage and distribute licenses among workstations in a network. However, ABBYY FineReader Engine may be installed both on the network server and on workstations.

All licenses require the Licensing Service (LicensingService.exe) for correct operation of ABBYY FineReader Engine. The Licensing Service can be installed automatically during the Developer installation and the Runtime installation of the ABBYY FineReader Engine library in automatic mode. If you need to install it manually, see Installing the Licensing Service for details.

Note: *The Licensing Service settings are provided in the LicensingSettings.xml file. The file is required for network installation and for standalone installation if Hardware or Online protection is used. This file is generated automatically during automatic installation. When installing manually, you must specify the correct settings in this file. The XML schema of the Licensing Service settings is described in the* ***LicensingSettings.xsd*** *file. You can find this file in the Inc folder (****Start > Programs > ABBYY FineReader Engine 12 > Installation Folders > Include Files Folder****). The detailed description of the settings is provided in the Working with the LicensingSettings.xml File section.*

For managing licenses ABBYY FineReader Engine provides the License Manager utility. With the help of this utility you can add, remove, activate, deactivate, update licenses and view license properties. The License Manager utility allows you to work with licenses of all protection types:

- **Software protection**
  Uses an activation file that should be obtained from the ABBYY server during a license activation process.
- **Hardware protection**
  Uses a USB dongle (hardware key) that contains the license parameters. In this case, license activation is not required.
- **Online protection**
  Uses a password-protected file that contains the license parameters. In this case, license activation is not required.

If you choose hardware protection

If you choose the hardware protection key, the Wibu CodeMeter drivers must be installed on the computer where the Licensing Service is installed, which is currently possible only during automatic runtime installation. Once the installation is completed, connect the hardware protection key to the USB port of the computer. Make sure that you do it before the first run of the program. No license activation is required. To view license properties, use the License Manager utility.

If you choose online protection

If you choose online protection, you need to ensure that the Licensing Service installed on your computer can connect to ABBYY Online licensing services: allow connections to ***.abbyy.com** on port **443** (HTTPS).

With online protection, license activation is not required, but ABBYY Online licensing service synchronizes with your workstation in set time intervals. Online License appears in the License Manager utility after ABBYY FineReader Engine transfers the connection data to ABBYY Online licensing service. The presence of an Online License depends on the state of the connection to ABBYY Online licensing service. With a

continuous connection to ABBYY Online licensing service, the license is always present in the License Manager utility. If the connection to ABBYY Online licensing service is lost, there is a certain timeout during which it can be re-established. Otherwise, the license is invalidated and will disappear from the License Manager utility until the local Licensing Service connects to ABBYY Online licensing service. The synchronization and timeout periods are the parameters of your license.

The same Online License can be used on multiple computers which number is limited by the license; no specific actions are required to allow concurrent usage.
If your Online License has a limit on the number of concurrent users, it will become available to another user after a time equal to the offline working time from the moment the workstation or license server was shut down. If you need to release an Online License faster, use the **"Release Online Licenses..."** option in the License Manager utility or call the **ReleaseOnlineLicense** method of the **Engine** object. The time it takes to complete the operation depends on your network bandwidth, however, you may specify the OperationTimeout parameter of the **ReleaseOnlineLicense** method for your convenience.

If you choose software protection

A software protection key requires the activation of its serial number by means of the License Manager utility.

How is activation carried out?

Activation takes very little time and is carried out with the help of an **Activation Wizard**. This wizard is built into the License Manager utility. The Activation Wizard has a friendly interface and is used for sending the necessary activation information to ABBYY. The same wizard is used for loading the **ABBYY License File** (*.ABBYY.License file) which you receive from ABBYY during activation.

Activation information is sent as a code (Installation ID) which is generated on the basis of information about the computer on which the program is being installed. No personal information about the user or computer is used for generating this code and this code cannot be used for identifying the user.

Activation methods:

- **Via the Internet**
Activation is carried out automatically and takes only a few seconds. An Internet connection is required for this type of activation.
- **By e-mail**
The user needs to send an e-mail message generated by the program and containing information required for activation to product-activation-robot@abbyy.com. To ensure a quick reply from the mail robot, do not alter the information in the message body or Subject field.
- **By e-mail from another computer**
This method is suitable, if your computer does not have an Internet connection. The program will generate an e-mail message containing information required for activation and offer you to copy the message and send it to ABBYY from another computer.

In the case of activation via the Internet, the whole process is carried out automatically. In the case of activation by e-mail, the user needs to enter the path to the Activation File received from ABBYY in the corresponding field of the Activation Wizard.

Once the activation is complete, the program can be used.

Note: *If you need to activate or deactivate ABBYY FineReader Engine 12 using a proxy server, specify the correct proxy settings in Internet Explorer (**Tools > Internet Options > Connections**).*

Reactivation

ABBYY FineReader Engine 12 can be reinstalled on one and the same computer an unlimited number of times without reactivation. However, if you make major upgrades, format your hard drive, or reinstall the operating system on the computer where the Licensing Service is installed, an additional activation may be required.

Deactivation

ABBYY FineReader Engine 12 license can be deactivated. The deactivated license can be then activated on another computer. The number of allowed deactivations can be restricted by your license.

Deactivation takes very little time and is carried out with the help of a **Deactivation Wizard**. This wizard is built into the License Manager utility. During the deactivation the Activation File (*.ABBYY.License file) which you receive from ABBYY during activation is deleted. Any copy of this file cannot be used for activation again.

The deactivation can be performed only via the Internet. Deactivation is carried out automatically and takes only a few seconds. An Internet connection is required. Once the deactivation is complete, the license can be activated on another computer.

License update

If you have purchased additional modules or an additional amount of pages for ABBYY FineReader Engine 12 and your license does not allow you to use them, you need to update the license. The license update process is similar to the activation process. The update process is carried out with the help of the **Update Wizard** and can be performed via the Internet or by e-mail. Once the update is complete, the newest functionality of the program can be used.

Note: *Online License is updated automatically during the synchronization period after changing the license parameters on the ABBYY FineReader Engine side.*

Working with the LicensingSettings.xml File

The **LicensingSettings.xml** file contains the ABBYY FineReader Engine protection settings. This file is used for correct work of the Licensing Service:

- (required) when you use a Network license with any type of protection or Standalone license with online protection.
- (optional) when you use a Standalone license with hardware protection (see Licensing).

The file is generated automatically during Developer or Runtime installation in automatic mode. Both for network and standalone installation this file is generated in two locations:

- for a 32-bit operating system:
  - **%CommonProgramFiles%\ABBYY\SDK\12\Licensing**
  - **<Installation Folder>\Bin**
- for a 64-bit operating system:
  - **%CommonProgramFiles(x86)%\ABBYY\SDK\12\Licensing**
  - **<Installation Folder>\Bin64** (if available)
  - **<Installation Folder>\Bin** (if available)

When installing manually, you will have to create this file in the **Bin** (**Bin64**) folder and specify correct settings in it.

The XML schema of the Licensing Service settings is described in the **LicensingSettings.xsd** file. You can find this file in the Inc folder (**Start > Programs > ABBYY FineReader Engine 12 > Installation Folders > Include Files Folder**).

Changing the Licensing Service settings

The Licensing Service settings include two main parts:

1. The settings of the Licensing Service installed on the current computer which are specified in the <LocalLicenseServer> tag.
2. The parameters of connection with the network server, where the Licensing Service is installed. These parameters are specified in the <LicensingServers>\<MainNetworkLicenseServer> tag.

If ABBYY FineReader Engine is installed on a local computer, you should specify the settings only in the <LocalLicenseServer>, if necessary. While if ABBYY FineReader Engine works in a network, <LocalLicenseServer> parameters are specified on the server, and <LicensingServers>\<MainNetworkLicenseServer> parameters on workstations. If a computer is used both as a server and as a workstation, both <LocalLicenseServer> and <LicensingServers>\<MainNetworkLicenseServer> parameters should be specified. See the descriptions of tags below for details.

To change port number

The Licensing Service uses the 3022 TCP/IP port by default. If this port is closed on the computer where the Licensing Service is installed, the Licensing Service will not work. In this case, specify an open port in EndPointName attribute of <LocalLicenseServer> or <LicensingServers>\<MainNetworkLicenseServer> tag, for example:

```
<LocalLicenseServer>
  <ConnectionProtocol ProtocolType="TCP/IP" EndPointName="1025"/>
</LocalLicenseServer>
```

or

```
<LicensingServers>
  <MainNetworkLicenseServer ServerAddress="computername"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LicensingServers>
```

To disable hardware protection keys

The Licensing Service enables using both hardware and software protection keys by default. If you do not use hardware protection keys, you can disable them. Set the Enable attribute of the <LocalLicenseServer>\<EnableCodeMeterLicenses> and <LocalLicenseServer>\<EnableIKeyLicenses> tags to "no":

```
<LocalLicenseServer>
  <ConnectionProtocol ProtocolType="<Protocol type>" />
  <EnableCodeMeterLicenses Enable="no" />
  <EnableIKeyLicenses Enable="no" />
</LocalLicenseServer>
```

Using an Online License with a proxy server

An Online License does not require any special settings, but to connect via a proxy server you should specify its parameters in <OnlineLicensing> tag.

The protocol supported is HTTP for which you can use Anonymous, Basic or Digest authentication schemes.

Here is a sample file with the proxy server settings for Online Licensing:

```
  <LocalLicenseServer>
    <OnlineLicensing Timeout="timeout_time"
ProxyServer="http://<server_name>:<port_number>"
ProxyAuth="<login>:<password>" />
    <ConnectionProtocol ProtocolType="<Protocol type>" />
  </LocalLicenseServer>
```

Setting a path to license data

By default, ABBYY FineReader Engine stores the auxiliary information about licensing in %ProgramData%\ABBYY\SDK\12\Licenses. To set another folder, specify the Path attribute in <LicensesFolder> tag. The full control permissions to a license data folder:

- required for the folder located on license server
- optional for the folder located on workstations

Note: *You may also use the* ***InitializeEngine*** *function to redefine the license data folder.*

Sample of specifying the license data folder

For the workstation:

```
<LicensingServers>
 ...
</LicensingServers>
<LicensesFolder Path="path_to_workstation_logs_folder"/>
```

For the license server:

```
<LocalLicenseServer>
 ...
</LocalLicenseServer>
<LicensesFolder Path="path_to_server_logs_folder"/>
```

Setting up a redundant license server configuration

For licenses distributed via the network, you have the option of setting up a two-server configuration which will allow one of the servers to act as a backup to another in case it goes offline. To do this, you will need a single Network Runtime License of a special type with redundancy support enabled.

- On servers:
  - The same license must be activated on both the main and the backup server.

  - Each of the servers must specify the address of the other in the <LocalLicenseServer>\<RedundancyNetworkLicenseServer> tag.
- On workstations:
  - All workstations must specify the addresses of both servers in the <LicensingServers>\<MainNetworkLicenseServer> and <LicensingServers>\<BackupNetworkLicenseServer> tags.

While using the license, the workstations will interact with the main license server, sending the license parameters and usage statistics to it. The main license server will regularly synchronize this data with the backup server. If the main license server fails, the backup server takes over the license management and continues to store information coming from the workstations. When the main server goes online, the license data is synchronized with the backup, and the main server again starts managing the licenses.

Note that the backup server will stop providing licenses, if the main server is not restored before a certain timeout. This timeout is specified in internal license settings and can range from 4 to 5000 hours.

If the backup license server fails while the main server is functional, the system will continue to work as if there is only one license server. When the backup server is restored, it will be synchronized with the main server and will start working as a backup again.

Sample of a redundant configuration

For the workstation:

```
<LicensingServers>
  <MainNetworkLicenseServer ServerAddress="mainserver"
ProtocolType="TCP/IP" EndPointName="1025"/>
  <BackupNetworkLicenseServer ServerAddress="backupserver"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LicensingServers>
```

For the main license server:

```
<LocalLicenseServer>
  <ConnectionProtocol ProtocolType="TCP/IP" EndPointName="1025"/>
  <RedundancyNetworkLicenseServer ServerAddress="backupserver"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LocalLicenseServer>
```

For the backup license server:

```
<LocalLicenseServer>
  <ConnectionProtocol ProtocolType="TCP/IP" EndPointName="1025"/>
  <RedundancyNetworkLicenseServer ServerAddress="mainserver"
ProtocolType="TCP/IP" EndPointName="1025"/>
</LocalLicenseServer>
```

Note: *If you need to use this configuration, please make sure that your license supports it. Contact your sales manager to find out.*

Description of Tags

Tag	Type	Multiplicity	Parent Tag	Description
LicensingSettings	LicensingSettings. Elements: • LocalLicenseServer • LicensingServers • LicensesFolder	1	no	Protection settings.
LocalLicenseServer	LocalLicenseServerSettings. Elements: • ConnectionProtocol • EnableCodeMeterLicenses • OnlineLicensing • RedundancyNetworkLicenseServer	0...1	LicensingSettings	The parameters of the Licensing Service located on the same computer.
ConnectionProtocol	Complex Type. Attributes: • *ProtocolType* — the protocol type: LocalInterprocessCommunication, NamedPipes, TCP/IP, or Sockets. ***Note:*** *This is an additional protocol type for the local Licensing Service. It is not necessary to specify this attribute for a standalone installation, as Standalone licenses are always used with the LocalInterprocessCommunication protocol type.* • *EndPointName* — (optional) port	0...1	LocalLicenseServer	The parameters of the connection protocol.

Tag	Type	Multiplicity	Parent Tag	Description
	number. By default, "3022".			
OnlineLicensing	Complex Type. Attributes: • *Timeout* — (optional) specifies reconnection period Currently the only supported protocol for connection via a proxy server is HTTP, for which you may specify the following attributes: • *ProxyServer* — (optional) the address of the proxy server and the connection settings, as a string "*protocol://server:port*", where the substrings' meanings are: ○ *protocol* — the protocol used to connect to the proxy server. Currently the only supported value is HTTP. ○ *server* — the name or IP-address of the proxy server. ○ *port* — the connection port. • *ProxyAuth* — (optional) authentication to a proxy server as a string "*login:password*", where the	0...1	LocalLicenseServer	Specifies whether online licenses can be used on the computer.

Tag	Type	Multiplicity	Parent Tag	Description
	substrings' meanings are: ○ *login* — the username for authorization. ○ *password* — the password for authorization.			
EnableCodeMeterLicenses	Complex Type. Attributes: • *Enable* — specifies whether CodeMeter protection keys can be used on the computer (set it to "yes" or "no"). By default, "yes".	0...1	LocalLicenseServer	Specifies whether CodeMeter hardware protection keys can be used on the computer.
EnableIKeyLicenses	Complex Type. Attributes: • *Enable* — specifies whether iKey protection keys can be used on the computer (set it to "yes" or "no"). If this tag was not specified, the value of *Enable* attribute is "yes". But it switches to "no" after automatic installation of ABBYY FineReader Engine.	0...1	LocalLicenseServer	Specifies whether iKey hardware protection keys can be used on the computer.
RedundancyNetworkLicenseServer	NetworkServerAddress. Attributes: • *ServerAddress* — the DNS name or IP address of	0...1	LocalLicenseServer	Specifies the settings of connection to another server and synchronization with it for each of the network license servers which work together in a

Tag	Type	Multiplicity	Parent Tag	Description
	another server in the redundant configuration. • *ProtocolType* — the protocol type: LocalInterprocess Communication, NamedPipes, TCP/IP, or Sockets. • *EndPointName* — (optional) port number. By default, "3022".			redundant two-server configuration.
LicensingServers	Complex Type. Elements: • MainNetworkLicenseServer • BackupNetworkLicenseServer	0...1	LicensingSettings	The list of network servers where the Licensing Service is installed.
MainNetworkLicenseServer	NetworkServerAddress. Attributes: • *ServerAddress* — the DNS name or IP address of the computer where the Licensing Service is installed. • *ProtocolType* — the protocol type: LocalInterprocess Communication, NamedPipes, TCP/IP, or Sockets. • *EndPointName* — (optional) port number. By default, "3022".	1	LicensingServers	The parameters of the connection with the main network server where the Licensing Service is installed.
BackupNetworkLicenseServer	NetworkServerAddress. Attributes:	0...1	LicensingServers	The parameters of the connection with the backup network license server.

Tag	Type	Multiplicity	Parent Tag	Description
	• *ServerAddress* — the DNS name or IP address of the computer where the Licensing Service is installed. • *ProtocolType* — the protocol type: LocalInterprocess Communication, NamedPipes, TCP/IP, or Sockets. • *EndPointName* — (optional) port number. By default, "3022".			

LicensesFolder	Complex Type.
	
	Attributes:
	
	• *Path* - the path to a folder with the licensing information. By default, "%ProgramData%\ABBYY\SDK\12\Licenses".
	The full control permissions to a license data folder:
	  ○ required for the folder located on license server
	  ○ optional for the folder located on workstations
		0...1	LicensingSettings	Specifies where a folder with auxiliary information about licensing should be stored.

Samples

The sample below shows a simple **LicensingSettings.xml** file for a standalone installation. Local inter-process communication is used. Hardware protection keys are disabled.

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
```

```
  <LocalLicenseServer>
    <ConnectionProtocol ProtocolType="LocalInterprocessCommunication" />
    <EnableCodeMeterLicenses Enable="no" />
    <EnableIKeyLicenses Enable="no" />
  </LocalLicenseServer>
</LicensingSettings>
```

The samples below show simple **LicensingSettings.xml** files for a network installation: a file for workstations and a file for a server. Licensing Service is located on the computer with the name "computername". The TCP/IP protocol is used for communication between the server and workstations.

For a workstation:

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
  <LicensingServers>
    <MainNetworkLicenseServer ServerAddress="computername"
ProtocolType="TCP/IP" />
  </LicensingServers>
</LicensingSettings>
```

For the server:

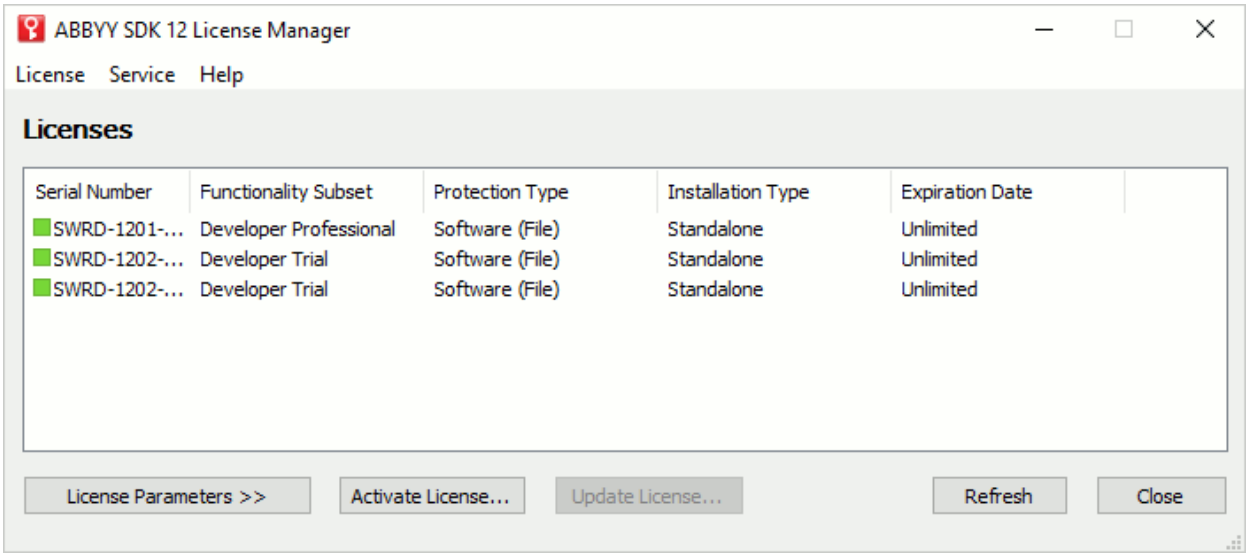
```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
  <LocalLicenseServer>
    <ConnectionProtocol ProtocolType="TCP/IP" />
  </LocalLicenseServer>
</LicensingSettings>
```

License Manager Utility

The License Manager utility (LicenseManager.exe) allows you to manage ABBYY SDK licenses of all types. In the ABBYY SDK 12 License Manager dialog box you can activate, deactivate, or update a license and view the properties of an activated license.

The License Manager utility is installed automatically during a Developer installation or during a Runtime ABBYY FineReader Engine library installation in automatic mode together with the Licensing Service. This utility is accessible through **Start > Programs > ABBYY FineReader Engine 12 > License Manager** or in the **Bin** (**Bin64**) folder. This utility is distributed along with other ABBYY FineReader Engine 12 files allowed for distribution and is used for Runtime Licenses activation. Developers can also use the License Manager utility via the command line to create their own registration utilities.

ABBYY SDK 12 License Manager dialog box



The following license information is available in the **ABBYY SDK 12 License Manager** dialog box:

Column	Description
Serial number	The ABBYY FineReader Engine 12 serial number. ***Note:*** *The icon in front of the serial number shows whether the license is currently in use:* The license is not in use now, all available CPU cores are free. The license can be used for processing. The license is in use, but there are some free CPU cores. Some recognition processes can be run with this license. The license is in use, all available CPU cores are busy. No more recognition processes can be run with this license. You can view detailed license usage statistics in the **Service > License Use Statistic** menu.
Functionality subset	The functionality subset of the license (Developer or Runtime, and any license limitations, e.g., Trial).
Protection type	The protection type: • **Software (File)** — software protection key. • **CodeMeter Key** — hardware protection key. • **Online licensing** — online protection.

Column	Description
Installation type	The installation type: • **Standalone** — the license is used on a local computer. • **Network** — the license is located on a network computer.
Expiration date	The expiration date.

You can find more details about the license in the License Parameters table. To show or hide license parameters, use the **License Parameters/Hide License Parameters** button.

Activating, updating, or deactivating the license

To activate, update, or deactivate the license, press the corresponding button, or select the corresponding item in the menu, and follow the instructions in the dialog box that opens. See details about license activation, deactivation and update in the Activation section.

Buttons

- **License Parameters/Hide License Parameters**
  Shows or hides license parameters.
- **Activate license...**
  Starts the License Activation Wizard.
- **Update license...**
  Starts the License Update Wizard for the selected license.
- **Refresh**
  Updates the license list.
- **Close**
  Closes the License Manager.

Menu items

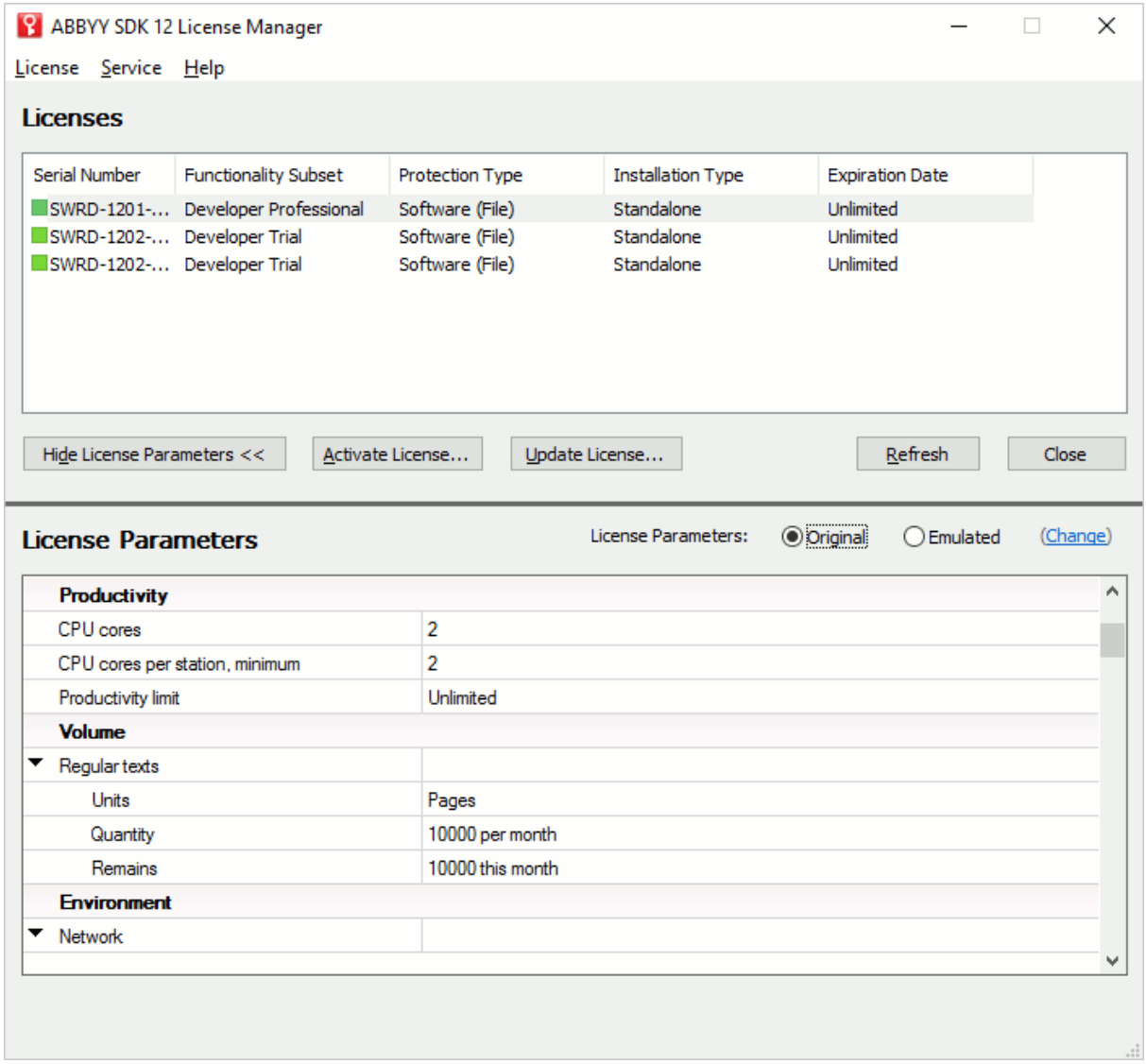
Item		Description
License	**Activate...**	Starts the License Activation Wizard.
	Update...	Starts the License Update Wizard for the selected license.
	Deactivate...	Starts the License Deactivation Wizard for the selected license.
	Copy Serial Number	Copies the selected license.

Item		Description
	Close	Closes the License Manager.
Service	**License Use Statistic...**	Shows the statistics of license usage on the workstations. Available only for the Network licenses with the CPU cores limitation.
	Refresh	Updates the license list.
	Release Online Licenses...	Disables and removes the Online Licenses along with the associated information.
Help	**Help**	Opens the License Manager Help.

License Parameters

The license parameters are displayed in the table below the list of the licenses in the License Manager. To view or hide license parameters, use the **License Parameters/Hide License Parameters** button in the main window of the License Manager.

License Parameters



The License Parameters table provides information about your license. The following information about your ABBYY FineReader Engine 12 license is available:

- Functionality subset of the license (Developer or Runtime, and any license limitations, e.g., Trial).
- Type of protection (software, hardware, or online protection).
- ABBYY FineReader Engine 12 serial number.
- Customer Project ID.
- License expiration date.
- Performance limitation: CPU core limit (the number of CPU cores which can be used for recognition), minimum number of CPU cores which can be used on a station, performance limit (e.g., characters per second).
- Environment limitation: usage in a network, on virtual machines, on Amazon EC2 and Microsoft Azure.
- List of features that are allowed by your license: text types, export formats, additional modules, etc. (see the Modules section).

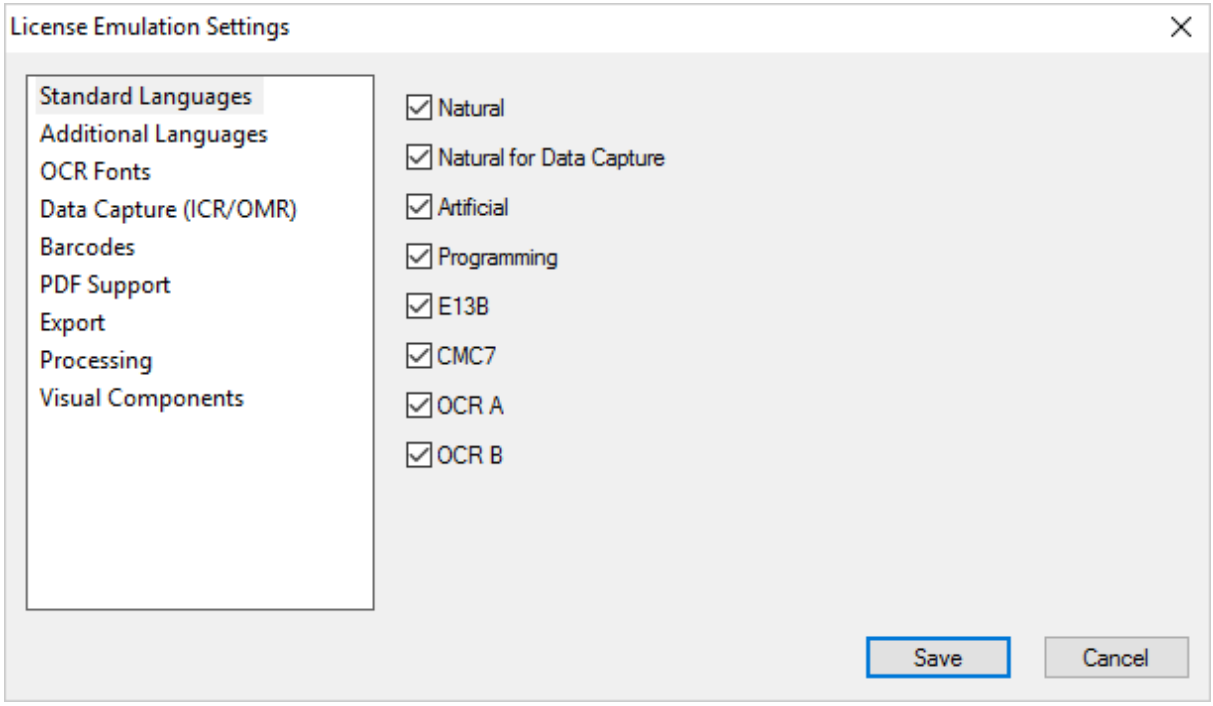
You can also access all these license parameters via the API. Use properties and methods of the **License** object.

Runtime license emulation

You can use your Developer's license in the Runtime emulated mode. This allows you to test your application which uses the ABBYY FineReader Engine 12 library.

To try emulation, select **Emulated** mode for license parameters. To change settings, click **Change** and in the **License Emulation Settings** dialog box that appears, select modules that will be enabled for your Runtime License (you can select only those modules which are supported by your Developer License). If some modules are not used for your Runtime License, disable them.

License Emulation Settings



Working with the License Manager from the Command Line

The current version of the License Manager allows you to activate and deactivate licenses from the command line. The following command line parameters are supported:

Parameter	Action
/SilentActivation	Use this parameter to activate the license that corresponds to the serial number specified after /SN.
/SN:<serial number>	Use this parameter to specify the serial number of the license to be activated or deactivated. Specify the serial number in one of the following

Parameter	Action
	formats: XXXX-XXXX-XXXX-XXXX-XXXX-XXXX or XXXXXXXXXXXXXXXXXXXXXXXX.
/SaveActivationEMailToFile:<file name>	Use this parameter to save to file the text of the message generated by the program and containing the information required for activation of the number specified in the /SN parameter.
/LoadActivationFile:<file name>	Loads the ABBYY License File and activates the serial number corresponding to this file.
/SilentDeactivation	Use this parameter to deactivate the license that corresponds to the serial number specified after /SN.

When launched from the command line, the License Manager may return the following return codes:

Return code	Description
-1	No command line parameters were specified.
0	No errors.
1	Invalid serial number.
2	The serial number has been already activated.
3	Unable to activate the serial number via the Internet. Try to activate it by e-mail.
4	The license with hardware protection cannot be activated by e-mail.
5	The specified file has not been found or is unavailable.
6	Incorrect license file.

Return code	Description
7	Unable to load the application.
8	Unable to save to file the text of the activation request.
9	Unknown command line parameter.
11	Unable to find the USB key to activate the license with hardware protection.
12	Several USB keys have been found. Please, leave only one USB key plugged to the computer.
13	Unable to connect to Licensing Service.

Running ABBYY FineReader Engine 12 inside a Docker container

This section provides the instructions on running ABBYY FineReader Engine 12 inside the Docker container in the scenario of using two containers: the first one with ABBYY FineReader Engine, and the second one with the Licensing Service. Implementing of this scenario increases the fault tolerance and ensures the continuous operation of all containers. If one of the containers fails, you can restart it without interrupting the other.

An Online License, which connects to ***.abbyy.com** license server, is used for this scenario.

Note: *ABBYY Licensing Service can work with only one Online License at the same time.*

You need:

- your Customer Project ID
- an Online License token file
- the password to the license token file

You need to fulfill the following conditions for using an Online License wherever ABBYY Licensing Service is installed:

- Active Internet connection
- Allowed connections to ***.abbyy.com** on port **443** (HTTPS)
- GoDaddy Trusted Root Certification Authority

Important! *GoDaddy root certificate for the Certification Authority should be installed in the local machine version of the Trusted Root Certification Authorities certificate store. See the detailed information about the certificate on the GoDaddy website.*

To run ABBYY FineReader Engine 12 in Docker container:

1. Download and install Docker for Windows on your machine.
2. Enable Hyper-V and Containers Windows features.
3. Select the "Use Windows containers instead of Linux containers" option during the container installation.
4. Create an empty directory and put into it:

- **docker-compose.yml** — a configuration of Docker Compose (See its listing below)
- **TestAppFolder** — a directory with
  - **Online License token file**
  - **test application** compiled to use this token file as well as the password to it and your Customer Project ID
- **Dockerfile_ls** — a Dockerfile for container with the Licensing Service (See its listing below)

  Important! ***Dockerfile_ls*** *is intended for building a container that will be working with only one container built from* ***Dockerfile_worker****. Do not configure multiple replicas of container with ABBYY FineReader Engine to work with a single Licensing Service.*

- **LicensingSettings_ls.xml** - a file with parameters of licensing for the Licensing Service (See its listing below)
- **DeployLS.ps1** — a script for launching the Licensing Service (See its listing below)

  Note: *This script runs the Licensing Service with* ***/standalone*** *key. With this key, the Licensing Service works as a foreground process, which is the best practice for working in Docker containers.*

- **Dockerfile_worker** — a Dockerfile for container with ABBYY FineReader Engine and test application (See its listing below)

- **LicensingSettings_worker.xml** - a file with parameters of licensing for the ABBYY FineReader Engine library (See its listing below)
- **SetUpWorkerLicensing.ps1** - a script for replacing the server name in **LicensingSettings_worker.xml** with the name specified in **docker-compose.yml** (See its listing below)
- **RunTestApp.ps1** — a command-line script to run the test application (See its listing below)

5. Unpack the ABBYY FineReader Engine distributive package to the previously created directory with the command:

```
installRnt64.exe /extract InstallDir="<your_directory>/FRE"
```

6. To build and run the containers, use the following command:

```
docker-compose up
```

The results of running the test application will be displayed in the console.

Important! *If you use Windows Server Core as a container base image, set the fonts after deploying ABBYY FineReader Engine, as described in the* ***Working with Fonts*** *article in Developer's Help.*

docker-compose.yml

```
version: '3'
services:
  ls:
    build:
      context: .
      dockerfile: Dockerfile_ls
    hostname: servername
    networks:
      frenet:
        aliases:
          - servername
    restart: on-failure
  worker:
    build:
      context: .
      dockerfile: Dockerfile_worker
      args:
        - service_address=servername
    depends_on:
      - ls
    networks:
      frenet:
    restart: on-failure
networks:
  frenet:
```

Dockerfile_ls

```
# Copy the Licensing Service
FROM mcr.microsoft.com/windows/servercore:ltsc2019 as fre12_ls
WORKDIR /app
ADD ./FRE/Licensing /app/FRE_Licensing
```

```
ADD ./LicensingSettings_ls.xml /app/FRE_Licensing/LicensingSettings.xml
WORKDIR /app/FRE_Licensing


# Installing the Licensing Service
COPY ./DeployLS.ps1 /app/DeployLS.ps1
EXPOSE 3022
SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop';
$ProgressPreference = 'SilentlyContinue';"]
ENTRYPOINT /app/DeployLS.ps1
```

LicensingSettings_ls.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
  <LocalLicenseServer>
    <ConnectionProtocol ProtocolType="TCP/IP" />
  </LocalLicenseServer>
</LicensingSettings>
```

DeployLS.ps1

```
echo "Preparing logging folder:"
$folder = "C:\ProgramData\ABBYY\SDK\12\Licenses"
echo $folder
if( (Test-Path -Path $folder) -ne $True ) {
    New-Item -Path $folder -ItemType Directory
}


echo "LicensingSettings.xml found:"
cd "C:/app/FRE_Licensing"
type LicensingSettings.xml


echo "Starting license service..."
.\LicensingService.exe /standalone

echo "License service stopped."
return 500
```

Dockerfile_worker

```
FROM mcr.microsoft.com/windows/servercore:ltsc2019 as fre12installation


# Copy extracted library to the folder C:/app/FRE12
ARG library_src="./FRE/FineReader Engine"
ADD $library_src /app/FRE12
```

```
ADD
./LicensingSettings_worker.xml /app/FRE12/Bin64/LicensingSettings.xml


# Replace <service_address> in worker LicensingSetting.xml
ARG service_address
COPY ./SetUpWorkerLicensing.ps1 /app/SetUpWorkerLicensing.ps1


SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop';
$ProgressPreference = 'SilentlyContinue';"]
RUN /app/SetUpWorkerLicensing.ps1 -ServerAddress $service_address


# Copy and run a test application
COPY ./TestAppFolder /app/TestAppFolder
COPY ./RunTestApp.ps1 /app/RunTestApp.ps1


CMD /app/RunTestApp.ps1
```

LicensingSettings_worker.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LicensingSettings
xmlns="http://www.abbyy.com/Protection/LicensingSettings">
  <LicensingServers>
    <MainNetworkLicenseServer ServerAddress="<server_address>"
ProtocolType="TCP/IP" />
  </LicensingServers>
</LicensingSettings>
```

SetUpWorkerLicensing.ps1

```
Param (
[string]$service_address
)


# Script for preparing client licensing log folder and setting
ServerAddress to LicesingSettings.xml


echo "Preparing logging folder:"
echo "C:\ProgramData\ABBYY\SDK\12\Licenses"
New-Item -Path "C:\ProgramData\ABBYY\SDK\12\Licenses" -ItemType
Directory


echo "LicensingSettings.xml found:"
cd "C:/app/FRE12/Bin64"
type LicensingSettings.xml
```

```
echo "Setting service address to: "
echo $service_address


((Get-Content -path LicensingSettings.xml -Raw) -replace
'<server_address>',$Env:service_address) `
| Set-Content -Path LicensingSettings.xml


echo "LicensingSettings.xml was set:"
type LicensingSettings.xml
```

RunTestApp.ps1

```
# Script for running a test application from TestAppFolder folder


# Launching the test application with arguments
(CommandLineInterface.exe is used as the test application)
cd C:\app\TestAppFolder
.\CommandLineInterface.exe  -pi -if Demo.tif -f PDF -of Demo.pdf
# For using the CommandLineInterface code sample in a Docker container,
you need to:
# 1)specify data for Docker container in SamplesConfig.h:
# - GetLicensePath, GetCustomerProjectId, GetLicensePassword
# - L"C:\\app\\FRE12\\Bin64\\FREngine.dll" as GetFreDllPath
# 2)rebuild your project for x64 configuration.
# 3)copy the CommandLineInterface.exe, *.ABBYY.ActivationToken license
file
#  and demo image (Demo.tif) in the folder \TestAppFolder.
```

Running ABBYY FineReader Engine inside Azure Services

This section provides the instructions on deploying and running ABBYY FineReader Engine 12 in Azure Services: Azure App Service and Azure Cloud Service. The following scenarios considering specifics of each service are prepared:

1. Azure App Service scenario using as an example a pair of WebJob projects.
2. Azure Cloud Service scenario using as an example a Worker role project.

For both services, you need almost the same toolset to develop your application. You will also need an App Service Plan with at least 1 GB of space to upload and store the archive with the ABBYY FineReader Engine files.

For detailed information about Azure Services, please visit Azure official website.

Running in Azure App Service

This section provides instructions on how to deploy **ABBYY FineReader Engine 12**-based application to Azure App Service. As an example, a pair of WebJob projects that use data from Azure Storage account is represented. Files processing is performed using a Blob container.
Using this scenario, you will obtain the best recognition results for small one-page documents, such as invoices, receipts, etc.

Important! *A special license in combination with a special license agreement is required. For more details, please contact your sales manager before implementing this scenario.*

Deploying your application to App Service includes several steps:

1. Organizing your local machine and app instance using prerequisites
2. Performing the preparatory steps before deploying your application
3. Deploying and running your application in App Service

Use the code samples represented in instruction below.

Prerequisites

Local machine

Before creating your App Service, use the following specification to organize your local machine:

- Visual Studio 2019 and its modules for developing applications for Azure (check Azure feature in Visual Studio or use Visual Studio Installer to download such modules)
- Azure SDK (download here)
- .NET Framework 4.7.2
- NuGet Package for working with Azure Storage and Blob containers:
  - Azure.Storage.Blobs (download here)
  - Azure.Storage.Queues (download here)
  - Newtonsoft.Json (download here)
  - System.IO.Compression.ZipFile (download here)
- ABBYY FineReader Engine wrapper for .Net Framework 4.7 (in the C:\ProgramData\ABBYY\SDK\12\FineReader Engine\Inc\.NET interops folder after developer installation)
- **IFileWriter** interface overridden for working with the Blob containers (see the sample below)
- Azure Storage Explorer (optional - download here)

Preparatory steps

Preparatory steps are to be done on your local machine. By completing these steps, you will prepare all necessary settings and files to start deploying your application:

1. Create an archive with the ABBYY FineReader Engine Library (for example, LibraryPackage.zip). List of files represents in the FREngineDistribution.csv file.
   Important! *If you have limited storage space (for example, you use an App Service Plan with 1GB space), we recommend using the /extract option to create your custom ABBYY FineReader Engine package with minimal size. The rest of the storage space will be used for processing the files.*
2. Create an Azure Storage account (**frestorage** in this article). All needed instructions you can find on Azure website.
3. Create your App Service as desired (see instructions here).
4. Create two Blob containers inside **frestorage**:
   - **fre-lib** - for the ABBYY FineReader Engine files
   - **processing-container** - for processing results
5. Upload LibraryPackage.zip to the **fre-lib** container in the most convenient way (using .NET, Powershell, Python script or Azure Storage Explorer/Azure Portal applications).
6. Create two queues inside **frestorage**:
   - **processing-queue** - for setting the tasks of files processing
   - **status-queue** - for notifying about task completion
7. Create two Azure WebJob (.NET Framework) projects in Visual Studio 2019 to work with **frestorage**:
   - **FreDeployerJob** - for deploying LibraryPackage.zip to App Service (see listing of its files: Config.cs, Functions.cs, Program.cs below)
   - **FreProcessorJob** - for document processing (see listing of its files: Config.cs, Functions.cs, Program.cs, EngineLoader.cs, IFileWriter.cs, Processor.cs below)

Deploying and running ABBYY FineReader Engine in App Service

To deploy ABBYY FineReader Engine:

1. Publish **FreDeployerJob** to Azure App Service using Visual Studio (set *Triggered* for WebJob Type).
2. Open your App Service in the Azure portal.
3. Open WebJobs of your App Service.
4. Find **FreDeployerJob** in the list of WebJobs.
5. Launch **FreDeployerJob** by the right-clicking+**Run** command on the WebJobs tab.

You may access the **Logs** tab to check the result of deployment. If it succeeds, **LibraryPackage.zip** is uploaded from the **fre-lib** container and deployed inside the %HOME_EXPANDED% folder available for all entities in App Service.

To deploy **FreProcessorJob**, publish **FreProcessorJob** to Azure App Service using Visual Studio (set *Continuous* for WebJob Type). As a result, **FreProcessorJob** will be in the list of WebJobs tabs of your App Service.

To process a file:

1. Upload the file you intend to process to the **processing-container**.
2. Add a JSON message for a new task of processing in format {"blob-item-name" : "file_name"} to the **processing-queue**. If you upload Demo.tif to the **processing-container**, your message should be:

```
{"blob-item-name" : "Demo.tif"}
```

3. Wait for the task to complete. As soon as the new task is set, **FreProcessorJob** starts to process the specified file in memory. The **status-queue** will contain entries about the execution of this task.
4. Find the output file in the **processing-container**.

Important!
*1. **FreProcessorJob** operates as a single-threaded process. If you intend to process your files in parallel, you need to create several **FreProcessorJob** that will be listening to the same queue.*
*2. Every additional **FreProcessorJob** consumes extra memory. Take into account this fact when buying your Service Plan. For example, in Azure Free Service Plan is nice to have only one **FreProcessorJob** that consumes a little memory and thus ensures the stability of file processing.*
*3. Using single **FreProcessorJob** is not suitable for processing large multi-page documents. In this case, consider recognition of your document in Azure Cloud Service or Azure Virtual Machine instead of App Service.*

Code Samples

This section includes code samples used to deploy and implement the ABBYY FineReader Engine API in App Service.

FreDeployerJob:

Config.cs

```
using System.IO;


class Config
{
    // Connecting string to blob container
    public static readonly string ConnectionString =
"your_connection_string";
    // HOME_EXPANDED directory is common for all WebJobs folder
    public static readonly string LibraryFolder =
Path.Combine(System.Environment.GetEnvironmentVariable("HOME_EXPANDED"),
"FRE");
    // Input and output containers name in your storage
    public static readonly string LibraryContainerName = "fre-lib";
```

Functions.cs

```
namespace FreDeployerJob
{
    public class Functions
    {
        // This function won't be triggered automatically - you should do
it manually
        [NoAutomaticTrigger]
        [Timeout("01:00:00")]
        public static void DeployFRE()
        {
            Console.WriteLine("Deploying FRE");
            // Connecting to existing input container <InputContainerName>
via Storage Account connection string
            BlobContainerClient inputContainerClient = new
BlobContainerClient(Config.ConnectionString, Config.LibraryContainerName);
```

```
            // Creating library directory as well as AppData and Temp
folders for ABBYY FineReader Engine initialization
            if (Directory.Exists(Config.LibraryFolder) == true)
            {
                Directory.Delete(Config.LibraryFolder, true);
            }
            Directory.CreateDirectory(Config.LibraryFolder);
            Directory.CreateDirectory(Path.Combine(Config.LibraryFolder,
"Temp"));
            Directory.CreateDirectory(Path.Combine(Config.LibraryFolder,
"AppData"));


            // Iterating via blobs in container. Blob in
<InputContainerName> is equal to some image file
            foreach (BlobItem blobItem in inputContainerClient.GetBlobs())
            {
                Console.WriteLine("\t" + blobItem.Name);
                // Searching for light version of the ABBYY FineReader
Engine library
                if (blobItem.Name == "LibraryPackage.zip")
                {
                    Console.WriteLine("LibraryPackage.zip was found.");
                    // Connecting to blob to access its contents
                    BlobClient blobClient = new
BlobClient(Config.ConnectionString, Config.LibraryContainerName,
blobItem.Name);
                    Console.WriteLine("Downloading to memory...");


                    // Download zip to memory
                    using (MemoryStream memoryStream = new MemoryStream())
                    {
                        blobClient.DownloadTo(memoryStream);
                        Console.WriteLine("LibraryPackage.zip was
downloaded.");
                        Console.WriteLine("Unzipping...");
                        // Unzip without using temp folder (only memory
processing)
                        using (ZipArchive archive = new
ZipArchive(memoryStream))
                        {
                            foreach (ZipArchiveEntry entry in
archive.Entries)
                            {
                                string subDirectory =
Path.GetDirectoryName(Path.Combine(Config.LibraryFolder, entry.FullName));
                                if (Directory.Exists(subDirectory) ==
false)
                                {
                                    Directory.CreateDirectory(subDirectory)
;
                                }
                                if (entry.Name.Length != 0)
                                {
```

```
                                        entry.ExtractToFile(Path.Combine(subDir
ectory, entry.Name));
                                    }
                                }
                            }
                            Console.WriteLine("LibraryPackage.zip was unzipped
to HOME_EXPANDED.");
                        }
                    }
                }
            }
        }
```

Program.cs

```
namespace FreDeployerJob
{
    // To learn more about Microsoft Azure WebJobs SDK, please see
https://go.microsoft.com/fwlink/?LinkID=320976
    class Program
    {
        // Please set the following connection strings in app.config for
these WebJobs to run:
        // AzureWebJobsDashboard and AzureWebJobsStorage
        static void Main()
        {
            // When triggered this WebJob will only call this method
            Functions.DeployFRE();
        }
    }
}
```

FreProcessorJob:

Config.cs

```
using System;
using System.IO;


namespace FreProcessorJob
{
    class Config
    {
        // Connecting string to blob container
        public static readonly string ConnectionString =
"your_connection_string";
        // HOME_EXPANDED directory is common for all WebJobs folders
        // It is the same as in FreDeployerJob project
        public static readonly string LibraryFolder =
Path.Combine(System.Environment.GetEnvironmentVariable("HOME_EXPANDED"),
```

```
"FRE");
        // Processing container name in your storage
        public static readonly string ProcessingContainerName =
"processing-container";
        // Processing queue name
        public static readonly string ProcessingQueueName = "processing-
queue";
        public static readonly string StatusQueueName = "status-queue";
        // Return Customer Project ID for ABBYY FineReader Engine
        public static String GetCustomerProjectId()
        {
            return "";
        }


        // Return name of license token
        // Licensing is located in Bin64 folder of the library package
        public static String GetLicenseTokenName()
        {
            return "your_licence_for_ABBYY FineReader Engine";
        }


        // Return license password
        public static String GetLicensePassword()
        {
            return "license_password";
        }


        // Return engine path
        public static String GetEngineFolder()
        {
            string engineSubfolder = "Bin64";
            string engineDllFolder = Path.Combine(LibraryFolder,
engineSubfolder);
            return engineDllFolder;
        }
    }
}
```

Functions.cs

```
using Microsoft.Azure.WebJobs;
using System;
using System.IO;
using Azure.Storage.Blobs;
using Azure.Storage.Queues;
using Newtonsoft.Json.Linq;


namespace FreProcessorJob
{
```

```
    public class Functions
    {


        // This function will get triggered/executed when a new message is
written
        // on an Azure Queue called processing-queue
        // The message is expected to be a JSON message with 'blob-item-
name' key
        // processing results will be saved to processing container
        // processing status will be sent to status-queue in JSON format
        public static void ProcessQueueMessage([QueueTrigger("processing-
queue")] string message)
        {
            // First of all, connecting to status-queue
            QueueClient queueClient = new
QueueClient(Config.ConnectionString, Config.StatusQueueName);


            try
            {
                // This will be logged to WebJob logs on Azure portal
                Console.WriteLine("Accepted task: " + message);
                JObject task = JObject.Parse(message);
                task["processor_id"] =
Environment.GetEnvironmentVariable("WEBJOBS_NAME");

                // This will be send to status-queue
                task["status"] = "accepted";
                queueClient.SendMessage(task.ToString());

                // Getting blob-item-name - the name of file in Processing
container
                string blobFileName = task["blob-item-name"].ToString();
                BlobClient blobClient = new
BlobClient(Config.ConnectionString, Config.ProcessingContainerName,
blobFileName);


                // Loading blob into memory
                Console.WriteLine("\t Downloading blob to memory: " +
blobFileName);
                MemoryStream memoryStream = new MemoryStream();
                blobClient.DownloadTo(memoryStream);
                Console.WriteLine("\t Downloaded.");


                // Updating status to processing
                Console.WriteLine("\t Processing in FRE: " + blobFileName);
                task["status"] = "processing";
                queueClient.SendMessage(task.ToString());


                // Processing the downloaded blob in ABBYY FineReader
Engine using memory processing methods
```

```
                // The output is the name of processing result saved as
blob in <ProcessingContainerName>
                string resultBlobName = "";
                using (FreProcessor.Processor freProcessor = new
FreProcessor.Processor())
                {
                    resultBlobName =
freProcessor.ProcessBlobFromMemory(memoryStream, blobFileName);
                    Console.WriteLine("\t Result blob name in output
container: " + resultBlobName);
                }


                // Deleting input image
                Console.WriteLine("\t Deleting from input container: " +
blobFileName);
                blobClient.Delete();


                // Updating status to succeeded
                // in Azure portal logs
                Console.WriteLine("Succeeded");

                // in status-queue
                task["status"] = "succeeded";
                task["result-blob-name"] = resultBlobName;
                queueClient.SendMessage(task.ToString());
            }
            catch (Exception error)
            {
                // In case of any errors reporting
                // to Azure portal logs
                Console.WriteLine("Failed: " + error.Message);
                // to status-queue
                JObject task = new JObject();
                task["processor_id"] =
Environment.GetEnvironmentVariable("WEBJOBS_NAME");
                task["status"] = "failed";
                task["error"] = error.Message;
                task["task"] = message;
                queueClient.SendMessage(task.ToString());
            }
        }
    }
}
```

Program.cs

```
using Microsoft.Azure.WebJobs;


namespace FreProcessorJob
{
    // To learn more about Microsoft Azure WebJobs SDK, please see
```

```
https://go.microsoft.com/fwlink/?LinkID=320976
    class Program
    {
        // Please set the following connection strings in app.config for
these WebJobs to run:
        // AzureWebJobsDashboard and AzureWebJobsStorage
        static void Main()
        {
            var config = new JobHostConfiguration();


            if (config.IsDevelopment)
            {
                config.UseDevelopmentSettings();
            }
            // ABBYY FineReader Engine is not thread-safe, so we cannot
process more than one message simultaneously
            config.Queues.BatchSize = 1;


            var host = new JobHost(config);
            // The following code ensures that the WebJob will be running
continuously
            // as one of functions is attached to Azure queue and listens
for new tasks
            host.RunAndBlock();
        }
    }
}
```

EngineLoader.cs

```
using System;
using System.IO;
using System.Runtime.InteropServices;
using FREngine;


namespace FreProcessorJob.FreProcessor
{
    // Class for loading/unloading FREngine.dll and
initializing/deinitializing Engine
    // Loading is performed in constructor, unloading in Dispose()
    // Throws exceptions when loading fails
    public class EngineLoader : IDisposable
    {
        // Load ABBYY FineReader Engine with settings stored in
SamplesConfig.cs
        public EngineLoader()
        {
            string enginePath = Path.Combine(Config.GetEngineFolder(),
"FREngine.dll");
            string customerProjectId = Config.GetCustomerProjectId();
```

```
            string licensePath = Path.Combine(Config.GetEngineFolder(),
Config.GetLicenseTokenName());
            string licensePassword = Config.GetLicensePassword();


            try
            {
                // Load the FREngine.dll library
                dllHandle = LoadLibraryEx(enginePath, IntPtr.Zero,
LOAD_WITH_ALTERED_SEARCH_PATH);


                if (dllHandle == IntPtr.Zero)
                {
                    int error = Marshal.GetLastWin32Error();
                    Console.WriteLine("The last Win32 Error was: " +
error);
                    throw new Exception("Can't load " + enginePath);
                }

                IntPtr initializeEnginePtr = GetProcAddress(dllHandle,
"InitializeEngine");
                if (initializeEnginePtr == IntPtr.Zero)
                {
                    throw new Exception("Can't find InitializeEngine
function");
                }
                IntPtr deinitializeEnginePtr = GetProcAddress(dllHandle,
"DeinitializeEngine");
                if (deinitializeEnginePtr == IntPtr.Zero)
                {
                    throw new Exception("Can't find DeinitializeEngine
function");
                }
                IntPtr dllCanUnloadNowPtr = GetProcAddress(dllHandle,
"DllCanUnloadNow");
                if (dllCanUnloadNowPtr == IntPtr.Zero)
                {
                    throw new Exception("Can't find DllCanUnloadNow
function");
                }


                // Convert pointers to delegates
                initializeEngine = (InitializeEngine)
Marshal.GetDelegateForFunctionPointer(
                    initializeEnginePtr, typeof(InitializeEngine));
                deinitializeEngine = (DeinitializeEngine)
Marshal.GetDelegateForFunctionPointer(
                    deinitializeEnginePtr, typeof(DeinitializeEngine));
                dllCanUnloadNow = (DllCanUnloadNow)
Marshal.GetDelegateForFunctionPointer(
                    dllCanUnloadNowPtr, typeof(DllCanUnloadNow));
```

```
                // Call the InitializeEngine function
                string dataFolder = Path.Combine(Config.LibraryFolder,
"AppData");
                string tempFolder = Path.Combine(Config.LibraryFolder,
"Temp");
                int hresult = initializeEngine(customerProjectId,
licensePath, licensePassword,
                    dataFolder, tempFolder, false, ref engine);


                Marshal.ThrowExceptionForHR(hresult);
            }
            catch (Exception)
            {
                // Free the FREngine.dll library
                engine = null;
                // Deleting all objects before FreeLibrary call
                GC.Collect();
                GC.WaitForPendingFinalizers();
                GC.Collect();
                FreeLibrary(dllHandle);
                dllHandle = IntPtr.Zero;
                initializeEngine = null;
                deinitializeEngine = null;
                dllCanUnloadNow = null;


                throw;
            }
        }


        // Unload ABBYY FineReader Engine
        public void Dispose()
        {
            if (engine == null)
            {
                // Engine was not loaded
                return;
            }
            engine = null;
            int hresult = deinitializeEngine();


            // Deleting all objects before FreeLibrary call
            GC.Collect();
            GC.WaitForPendingFinalizers();
            GC.Collect();
            hresult = dllCanUnloadNow();
            if (hresult == 0)
            {
                FreeLibrary(dllHandle);
            }
            dllHandle = IntPtr.Zero;
            initializeEngine = null;
```

```
            deinitializeEngine = null;
            dllCanUnloadNow = null;


            // throwing exception after cleaning up
            Marshal.ThrowExceptionForHR(hresult);
        }


        // Returns pointer to ABBYY FineReader Engine's main object
        public IEngine Engine
        {
            get
            {
                return engine;
            }
        }


        // Kernel32.dll functions
        [DllImport("kernel32.dll")]
        private static extern IntPtr LoadLibraryEx(string dllToLoad, IntPtr
reserved, uint flags);
        private const uint LOAD_WITH_ALTERED_SEARCH_PATH = 0x00000008;
        [DllImport("kernel32.dll")]
        private static extern IntPtr GetProcAddress(IntPtr hModule, string
procedureName);
        [DllImport("kernel32.dll")]
        private static extern bool FreeLibrary(IntPtr hModule);


        // FREngine.dll functions
        [UnmanagedFunctionPointer(CallingConvention.StdCall, CharSet =
CharSet.Unicode)]
        private delegate int InitializeEngine(string customerProjectId,
string licensePath, string licensePassword,
            string dataFolder, string tempFolder, bool
isSharedCPUCoresMode, ref FREngine.IEngine engine);
        [UnmanagedFunctionPointer(CallingConvention.StdCall)]
        private delegate int DeinitializeEngine();
        [UnmanagedFunctionPointer(CallingConvention.StdCall)]
        private delegate int DllCanUnloadNow();


        // private variables
        private FREngine.IEngine engine = null;
        // Handle to FREngine.dll
        private IntPtr dllHandle = IntPtr.Zero;


        private InitializeEngine initializeEngine = null;
        private DeinitializeEngine deinitializeEngine = null;
        private DllCanUnloadNow dllCanUnloadNow = null;
    }
}
```

IFileWriter.cs

```
using System;
using System.IO;
using Azure.Storage.Blobs;


namespace FreProcessorJob.FreProcessor
{
    public class FileWriter : FREngine.IFileWriter, IDisposable
    {
        public FileWriter(string _resultBlobName, string _fileExtension)
        {
            resultBlobName = _resultBlobName;
            fileExtension = _fileExtension;
        }
        public void Open(string fileName, ref int bufferSize)
        {
            stream = new MemoryStream();
        }
        public void Write(byte[] data)
        {
            stream.Write(data, 0, data.Length);
        }
        public void Close()
        {
            // Creating connection to new blob in
<ProcessingContainerName>. The processing result will be stored there
            BlobClient resultBlobClient = new
BlobClient(Config.ConnectionString,
                Config.ProcessingContainerName,
                resultBlobName + fileExtension);


            // Rewrite existing file
            resultBlobClient.DeleteIfExists();


            // Setting position to 0 to write file from beginning
            stream.Position = 0;
            resultBlobClient.Upload(stream);


            stream.Close();
        }


        public void Dispose()
        {
            // Closing memory stream on disposal to be able to access it
after data was written
            stream.Close();
        }
```

```
        private string resultBlobName;
        private string fileExtension;
        private MemoryStream stream;
    }
}
```

Processor.cs

```
using System;
using System.Runtime.InteropServices;
using System.IO;
using FREngine;


namespace FreProcessorJob.FreProcessor
{
    class Processor : IDisposable
    {
        private EngineLoader engineLoader = null;
        private void displayMessage(string text)
        {
            Console.WriteLine("\t" + text);
        }
        private void setupFREngine()
        {
            displayMessage("Loading predefined profile...");
            // this is optional
            engineLoader.Engine.LoadPredefinedProfile("DocumentConversion_A
ccuracy");
            // this is mandatory on low-performing App Service Plans as we
will be getting errors on parallel processing
            engineLoader.Engine.MultiProcessingParams.MultiProcessingMode =
MultiProcessingModeEnum.MPM_Sequential;
        }
        private void LoadEngine()
        {
            try
            {
                if (engineLoader == null)
                {
                    engineLoader = new EngineLoader();
                }
                setupFREngine();
            }
            catch (Exception error)
            {
                displayMessage("error: " + error.Message);
            }
        }
        private void UnloadEngine()
        {
            try
            {
                if (engineLoader != null)
```

```
                {
                    engineLoader.Dispose();
                    engineLoader = null;
                }
            }
            catch (Exception error)
            {
                displayMessage("error: " + error.Message);
            }
        }


        public Processor()
        {
            LoadEngine();
        }
        public string ProcessBlobFromMemory(MemoryStream inputMemoryStream,
string inputBlobName)
        {
            FRDocument document = engineLoader.Engine.CreateFRDocument();
            string resultBlobName = "";
            try
            {
                document.PageFlushingPolicy =
FREngine.PageFlushingPolicyEnum.PFP_KeepInMemory;



                // Add image file to document
                displayMessage("Loading image...");
                IntPtr handle =
Marshal.AllocHGlobal(inputMemoryStream.GetBuffer().Length);
                Marshal.Copy(inputMemoryStream.GetBuffer(), 0, handle,
inputMemoryStream.GetBuffer().Length);


                document.AddImageFileFromMemory(handle.ToInt64(), null,
null);


                // Recognize the document
                displayMessage("Recognizing...");
                document.Process(null);


                // Save results
                displayMessage("Saving results...");


                FileWriter fileWriter = new FileWriter(inputBlobName,
".pdf");
                resultBlobName = inputBlobName + ".pdf";
                document.ExportToMemory(fileWriter,
FREngine.FileExportFormatEnum.FEF_PDF, null);
            }
            catch (Exception error)
```

```
            {
                displayMessage("error: " + error.Message);
                throw error;
            }
            finally
            {
                // Close the document
                document.Close();
            }
            return resultBlobName;
        }
        public void Dispose()
        {
            UnloadEngine();
        }
    }
}
```

Running in Azure Cloud Service

This section provides instructions on how to deploy **ABBYY FineReader Engine 12**-based application to Azure Cloud Service. As an example, a Worker processing files by a container in Azure Storage account is represented.

To implement this scenario, an Online License that connects to ***.abbyy.com** license server is used.

Note: *ABBYY Licensing Service can work with only one Online License at the same time.*

Deploying your application to Cloud Service includes several steps:

1. Organizing your Online License, local machine and cloud instance using prerequisites
2. Performing the preparatory steps before deployingyour application
3. Deploying your application to Cloud Service

See the code samples illustrating how to use the ABBYY FineReader Engine methods for document processing.

Prerequisites

Online License

To use an Online License, you should receive the following information from sales:

- your Customer Project ID
- an Online License token file
- the password to the license token file

You need to fulfill the following conditions for using an Online License wherever ABBYY Licensing Service is installed:

- Active Internet connection
- Allowed connections to ***.abbyy.com** on port **443** (HTTPS)
- GoDaddy root certificate for the Certification Authority (It must be installed in the local machine version of the Trusted Root Certification Authorities certificate store. See the detailed information about the certificate on the GoDaddy website).

Local machine

Before creating your Cloud Service, use the following specification to organize your local machine:

- Visual Studio 2019 and its modules for developing applications for Azure (check Azure feature in Visual Studio or use Visual Studio Installer to download such modules)
- Cloud Service emulators and storage for easy debugging (see information here and download via Visual Studio Installer)
- Template solution for Azure Cloud Service (extended preview) with single Worker role project (see information here)
- .NET Framework 4.7.2
- Azure SDK (download here)
- NuGet package for working with Azure Storage and Blob containers - Azure.Storage.Blobs (download here)
- ABBYY FineReader Engine wrapper for .Net Framework 4.7 (in the C:\ProgramData\ABBYY\SDK\12\FineReader Engine\Inc\.NET interops folder after developer installation)
- **IFileWriter** interface overridden for working with Blob containers (see the sample below)
- Azure Storage Explorer (optional - download here)

Cloud Service instance

Cloud Service instance is for storing your WorkerRole project in Azure. Use the following specification to organize this instance:

- .NET Framework version of your Service
- PowerShell for deploying ABBYY FineReader Engine
- NuGet 2.8.5.201 or above for uploading Azure SDK
- Azure SDK for working with an Azure Storage account using PowerShell

Preparatory steps

Preparatory steps are to be done on your local machine. By completing these steps, you will prepare all necessary settings and files to start deploying your application:

1. Create two archives with the ABBYY FineReader Engine Library and Licensing Service (for example, LibraryPackage.zip and LSPackage. zip). You may create the list of files automatically, with the help of the FREngineDistribution.csv file. Use ABBYY FineReader Engine and License Server from the same package; otherwise, compatibility is not guaranteed.
2. Create the Azure Storage account (**frestorage** in this article). All needed instructions you can find on Azure website.
3. Create three Blob containers inside **frestorage**:
   - **fre-lib** - for the ABBYY FineReader Engine files
   - **fre-input** - for incoming files
   - **fre-output** - for processing results
4. Upload **LibraryPackage.zip** and **LSPackage.zip** to the **fre-lib** container in the most convenient way (using .NET, Powershell, Python script, or Azure Storage Explorer/Azure Portal applications).

As an example, a WorkerRole project is used for working in a configured environment. All necessary configuration files (.csdef and Cloud.cscfg) are generated automatically after creating your project.

Deploying ABBYY FineReader Engine to Cloud Service

To deploy ABBYY FineReader Engine to your new WorkerRole project:

1. Specify the parameters of the Cloud.cscfg as desired.
2. Specify the parameters of the .csdef file:
   - (optional) your WorkerRole settings

- (optional) the size of your virtual machine
- (required) the local storage of your role (*LocalStorage* section - in this article, storage named **LocalStorage1**). Set it to 3 GB at least to ensure that the ABBYY FineReader Engine package will be deployed properly.
- (required) your role startup order (use the code samples and settings from code samples section)

3. Implement your WorkerRole project to process files from the **fre-input** container and publish the results of processing to the **fre-output** container:
   - the **OnStart** method to prepare your Cloud Service for working. This method is used to initialize the Engine and set up the TLS protocol.
   - the **RunAsync** method to cyclically process the files taken from the **fre-input** container. This method detects the files in the **fre-input** container, processes them in memory, and places them in the **fre-output** container (see Processor.cs, IFileWriter.cs and Config.cs).
   - the **OnStop** method to complete working of Cloud Service. This method is used to deinitialize the Engine.

Code Samples

This section includes code samples used to configure role startup order and ABBYY FineReader Engine settings:

- **CleanUpOnStart** - deletes the previous version of ABBYY FineReader Engine and simplifies its update procedure (see its listing in the CleanUpOnStart.cmd and CleanUpOnStart.ps1 files below).
- **PreparePoShModules** - downloads the SDK needed for Powershell to work with the Azure Storage (see its listing in the PreparePoShModules.cmd and PreparePoShModules.ps1 files below).
- **PrepareLibrary** - uploads and unpacks the ABBYY FineReader Engine Library from the **fre-lib** container to local storage (see its listing in the PrepareLibrary.cmd and PrepareLibrary.ps1 files below).
- **PrepareLS** - uploads, unpacks and launches the Licensing Service from the **fre-lib** container (see its listing in the PrepareLS.cmd and PrepareLS.ps1 files below).

It is required that the scripts listed above are run exactly in the proposed order, before launching your application. To customize the order of running the scripts, specify the following attributes:

- taskType="*simple*" - tasks are executed synchronously, one at a time.
- executionContext="*elevated*" - running a startup script with administrator rights (required to be able to install any application and running LicensingService.exe)

As a result, a folder that Cloud Service can manage will be created. This folder is used for uploading ABBYY FineReader Engine by the scrips listed above and loading the FREngine.dll for its further implementation in the sample.

CleanUpOnStart.cmd

```
rem   Script to clean up previous ABBYY FineReader Engine and Licensing
Service packages


echo Cleaning up before launching service >> ".\CleanUpOnStart.log" 2>&1
PowerShell -ExecutionPolicy Unrestricted .
\StartupScripts\CleanUpOnStart.ps1 >> ".\CleanUpOnStart.log" 2>&1
echo Cleaning up beafore launching service. >> ".\CleanUpOnStart.log"
2>&1

exit /B %errorlevel%
```

CleanUpOnStart.ps1

```
Write-Host "CleanUpOnStart.ps1 started...";


# path to local storage (see ServiceDefinition.csdef)
$local_storage_name = "LocalStorage1";
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.WindowsAzur
e.ServiceRuntime");
$local_storage_path =
([Microsoft.WindowsAzure.ServiceRuntime.RoleEnvironment]::GetLocalResour
ce($local_storage_name)).RootPath.TrimEnd('\\');


# stopping Licensing Service process if it was running
$processes = Get-Process;
foreach( $process in $processes ) {
    if( $process.Name -eq "LicesingService" ) {
        Stop-Process $process;
    }
}

# cleaning and recreating the fre_packages folder
$destination_path = Join-Path -Path $local_storage_path -ChildPath
'fre_packages';
if( Test-Path -Path $destination_path ) {
    Remove-Item $destination_path -Recurse -Force;
}
New-Item -ItemType Directory -Path $destination_path;


# cleaning and recreating the input folder
$destination_path = Join-Path -Path $local_storage_path -ChildPath
'Input';
if( Test-Path -Path $destination_path ) {
    Remove-Item $destination_path -Recurse -Force;
}
New-Item -ItemType Directory -Path $destination_path;


# cleaning and recreating the results folder
$destination_path = Join-Path -Path $local_storage_path -ChildPath
'Results';
if( Test-Path -Path $destination_path ) {
    Remove-Item $destination_path -Recurse -Force;
}
New-Item -ItemType Directory -Path $destination_path;


Write-Host ("Resource folder was cleaned up.");
```

- PreparePoShModules.cmd

```
rem   Script to prepare modules to be able to connect to Storage account
and download ABBYY FineReader Engine and Licensing Service packages
```

```
echo Preparing PoSh Modules >> ".\PreparePoShModules.log" 2>&1
PowerShell -ExecutionPolicy Unrestricted .
\StartupScripts\PreparePoShModules.ps1 >> ".\PreparePoShModules.log"
2>&1
echo PoSh Modules wer prepared. >> ".\PreparePoShModules.log" 2>&1

exit /B %errorlevel%
```

PreparePoShModules.ps1

```
# installing modules
Write-Host "PreparePoShModules.ps1 started...";


# installing NuGet provider for downloading Azure modules
Write-Host "Installing NuGet package provider...";
Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force;

# installing Azure Storage module for operating the blob
Write-Host "Installing Az.Storage module...";
Install-Module -Name Az.Storage -Scope CurrentUser -Repository PSGallery
-Force -AllowClobber;


Write-Host ("PoSh modules were prepared.");
```

PrepareLibrary.cmd

```
rem   Script to prepare ABBYY FineReader Engine package


echo Preparing Library >> ".\PrepareLibrary.log" 2>&1
PowerShell -ExecutionPolicy Unrestricted .
\StartupScripts\PrepareLibrary.ps1 >> ".\PrepareLibrary.log" 2>&1
echo Library was prepared. >> ".\PrepareLibrary.log" 2>&1


exit /B %errorlevel%
```

PrepareLibrary.ps1

```
Write-Host "PrepareLibrary.ps1 started...";


# container with ABBYY FineReader Engine and Licensing Service packages
$container_name = 'fre-lib';
# storage account connecting string
$connection_string = "<connection_string_to_frestorage>";
```

```
Write-Host "Configuration";
Write-Host ("Container name: " + $container_name);
Write-Host ("Connection string: " + $storage_account);


# connecting to the storage object
Write-Host ("Connecting to storage account...");
$storage_account = New-AzStorageContext -ConnectionString
$connection_string;


# getting blobs (basically files)
Write-Host ("Getting blobs from container...");
$fre_blobs = Get-AzStorageBlob -Container $container_name -Context
$storage_account;


# path to local storage (see ServiceDefinition.csdef)
$local_storage_name = "LocalStorage1";
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.WindowsAzur
e.ServiceRuntime");
$local_storage_path =
([Microsoft.WindowsAzure.ServiceRuntime.RoleEnvironment]::GetLocalResour
ce($local_storage_name)).RootPath.TrimEnd('\\');
$destination_path = Join-Path -Path $local_storage_path -ChildPath
'fre_packages';


# downloading the only blob needed
Write-Host ("Downloading FRE package from container...");
foreach( $blob in $fre_blobs ) {
    Write-Host $blob.Name;
    if( $blob.Name -eq "LibraryPackage.zip" ) {
        Write-Host ("Downloading blob: " + $blob.Name + "to " +
$destination_path);
        Get-AzStorageBlobContent -Container $container_name -Blob
$blob.Name `
            -Destination (Join-Path -Path $destination_path -ChildPath
$blob.Name) -Context $storage_account;
    } else {
        Write-Host ("Downloading blob " + $blob.Name + " was skipped.");
    }
}


# unzipping the downloaded package (old library will be forcibly
rewritten)
Write-Host ("Unzipping library package...");
Expand-Archive -Path (Join-Path -Path $destination_path -ChildPath
"LibraryPackage.zip") `
    -DestinationPath (Join-Path -Path $destination_path -ChildPath
"LibraryPackage") -Force;


Write-Host ("Library package was prepared.");
```

PrepareLS.cmd

```
rem   Script to prepare Licensing Service package and start
LicensingService.exe


echo Preparing LS >> ".\PrepareLS.log" 2>&1
PowerShell -ExecutionPolicy Unrestricted .\StartupScripts\PrepareLS.ps1
>> ".\PrepareLS.log" 2>&1
echo LS was prepared. >> ".\PrepareLS.log" 2>&1


exit /B %errorlevel%
```

PrepareLS.ps1

```
Write-Host "PrepareLS.ps1 started...";


# container with ABBYY FineReader Engine and Licensing Service packages
$container_name = 'fre-lib';
# storage account connecting string
$connection_string = "<connection_string_to_frestorage>";


Write-Host "Configuration";
Write-Host ("Container name: " + $container_name);
Write-Host ("Connection string: " + $storage_account);


# connecting to the storage object
Write-Host ("Connecting to storage account...");
$storage_account = New-AzStorageContext -ConnectionString
$connection_string;


# getting blobs (basically files)
Write-Host ("Getting blobs from container...");
$fre_blobs = Get-AzStorageBlob -Container $container_name -Context
$storage_account;


# path to local storage (see ServiceDefinition.csdef)
$local_storage_name = "LocalStorage1";
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.WindowsAzur
e.ServiceRuntime");
$local_storage_path =
([Microsoft.WindowsAzure.ServiceRuntime.RoleEnvironment]::GetLocalResour
ce($local_storage_name)).RootPath.TrimEnd('\\');


$destination_path = Join-Path -Path $local_storage_path -ChildPath
'fre_packages';
```

```
# downloading the Licensing Service package
Write-Host ("Downloading Licensing Service package from container...");
foreach( $blob in $fre_blobs ) {
    Write-Host $blob.Name;
    if( $blob.Name -eq "LSPackage.zip" ) {
        Write-Host ("Downloading blob: " + $blob.Name + "to " +
$destination_path);
        Get-AzStorageBlobContent -Container $container_name -Blob
$blob.Name `
            -Destination (Join-Path -Path $destination_path -ChildPath
$blob.Name) -Context $storage_account;
    } else {
        Write-Host ("Downloading blob " + $blob.Name + " was skipped.");
    }
}


# unzipping the downloaded package
Write-Host ("Unzipping library package...");
Expand-Archive -Path (Join-Path -Path $destination_path -ChildPath
"LSPackage.zip") `
    -DestinationPath (Join-Path -Path $destination_path -ChildPath
"LSPackage") -Force;


# creating folders for licenses
$program_data_path = [System.Environment]::ExpandEnvironmentVariables("%
programdata%");
$path = Join-Path -Path $program_data_path -ChildPath
"ABBYY\SDK\12\Licenses";
New-Item -ItemType Directory -Path $path -Force;


# starting Licensing Service as standalone
Write-Host ("Starting Licensing Service...");

$licensing_service_app = Join-Path -Path $destination_path -ChildPath
"LSPackage\LicensingService.exe";
Start-Process -FilePath $licesing_service_app -ArgumentList
"/standalone";


Write-Host ("LS package was prepared and LS was started.");
```

- Implementation of the IFileWriter Interface

```
namespace WorkerRole1.Engine
{
    public class FileWriter : FREngine.IFileWriter, IDisposable
    {
        public FileWriter(string _resultBlobName, string _fileExtension)
        {
            resultBlobName = _resultBlobName;
            fileExtension = _fileExtension;
```

```
        }
        public void Open(string fileName, ref int bufferSize)
        {
            stream = new MemoryStream();
        }
        public void Write(byte[] data)
        {
            stream.Write(data, 0, data.Length);
        }
        public void Close()
        {
            // Creating connection to a new blob in
<OutputContainerName>
            // The processing result will be stored there
            BlobClient resultBlobClient = new
BlobClient(Config.ConnectionString, Config.OutputContainerName,
                resultBlobName + fileExtension);


            // Rewrite existing file
            resultBlobClient.DeleteIfExists();


            // Setting position to 0 to write file from beginning
            stream.Position = 0;
            resultBlobClient.Upload(stream);

            stream.Close();
        }


        public void Dispose()
        {
            // Closing memory stream on disposal to be able to access it
after data was written
            stream.Close();
        }


        private string resultBlobName;
        private string fileExtension;
        private MemoryStream stream;
    }
}
```

- Sample of processing the files using ABBYY FineReader Engine

```
namespace WorkerRole1
{
    public class WorkerRole : RoleEntryPoint
    {
        private readonly CancellationTokenSource cancellationTokenSource
= new CancellationTokenSource();
        private readonly ManualResetEvent runCompleteEvent = new
```

```
ManualResetEvent(false);
        Processor processor = new Processor();

        public override void Run()
        {
            Trace.TraceInformation("WorkerRole1 is running");

            try
            {
                this.RunAsync(this.cancellationTokenSource.Token).Wait()
;
            }
            finally
            {
                this.runCompleteEvent.Set();
            }
        }


        public override bool OnStart()
        {
            // Set the maximum number of concurrent connections
            ServicePointManager.DefaultConnectionLimit = 12;

            // Set TLS to 12 to be able to connect to storage account
            System.Net.ServicePointManager.SecurityProtocol =
System.Net.SecurityProtocolType.Tls12;

            // For information on handling configuration changes
            // see the MSDN topic at https://go.microsoft.com/fwlink/?
LinkId=166357

            bool result = base.OnStart();
            Trace.TraceInformation("WorkerRole1 has been started");

            processor.LoadEngine();
            return result;
        }

        public override void OnStop()
        {
            processor.UnloadEngine();
            Trace.TraceInformation("WorkerRole1 is stopping");
            this.cancellationTokenSource.Cancel();
            this.runCompleteEvent.WaitOne();


            base.OnStop();
            Trace.TraceInformation("WorkerRole1 has stopped");
        }

        private async Task RunAsync(CancellationToken cancellationToken)
        {
            // TODO: Replace the following with your own logic
            while (!cancellationToken.IsCancellationRequested)
            {
```

```
                Trace.TraceInformation("Working");

                try
                {
                    // Create the container and return a container
client object
                    BlobContainerClient inputContainerClient = new
BlobContainerClient(Config.ConnectionString, Config.InputContainerName);


                    foreach (BlobItem blobItem in
inputContainerClient.GetBlobs())
                    {
                        Trace.TraceInformation("\t" + blobItem.Name);
                        BlobClient blobClient = new
BlobClient(Config.ConnectionString, Config.InputContainerName,
blobItem.Name);



                        Trace.TraceInformation("\t Downloading blob to
memory: " + blobItem.Name);
                        MemoryStream memoryStream = new MemoryStream();
                        blobClient.DownloadTo(memoryStream);


                        Trace.TraceInformation("\t Processing in FRE: "
+ blobItem.Name);
                        string resultBlobName =
processor.ProcessImage(memoryStream, blobItem.Name);
                        Trace.TraceInformation("\t Result blob name in
output container: " + resultBlobName);


                        Trace.TraceInformation("\t Deleting from input
container: " + blobItem.Name);
                        blobClient.Delete();
                    }
                }
                catch (Exception error)
                {
                    Trace.TraceInformation("error: " + error.Message);
                }

                await Task.Delay(1000);
            }
        }
    }
}
```

Processor.cs

```
namespace WorkerRole1
{
```

```
    class Processor : IDisposable
    {
        EngineLoader engineLoader = null;
        private void displayMessage(string text)
        {
            File.AppendAllText(".\\FRE.log", text + "\n");
            Trace.TraceInformation("\t" + text );
        }


        private void loadProfile()
        {
            engineLoader.Engine.LoadPredefinedProfile("DocumentConversio
n_Accuracy");
        }


        private void setupFREngine()
        {
            displayMessage("Loading predefined profile...");
            loadProfile();
        }
        public void LoadEngine()
        {
            try {
                if (engineLoader == null)
                {
                    // The same EngineLoader, as in the Hello sample
                    engineLoader = new EngineLoader();
                }
                setupFREngine();
            }
            catch (Exception error)
            {
                displayMessage("error: " + error.Message);
            }
        }


        public void UnloadEngine()
        {
            try
            {
                if (engineLoader != null)
                {
                    engineLoader.Dispose();
                    engineLoader = null;
                }
            }
            catch (Exception error)
            {
                displayMessage("error: " + error.Message);
            }
        }

        public string ProcessImage( MemoryStream inputMemoryStream,
```

```
string inputBlobName )
        {
            FRDocument document =
engineLoader.Engine.CreateFRDocument();
            string resultBlobName = "";
            try
            {
               document.PageFlushingPolicy =
FREngine.PageFlushingPolicyEnum.PFP_KeepInMemory;

                // Add image file to document
                displayMessage("Loading image...");
                IntPtr handle =
Marshal.AllocHGlobal(inputMemoryStream.GetBuffer().Length);
                Marshal.Copy(inputMemoryStream.GetBuffer(), 0, handle,
inputMemoryStream.GetBuffer().Length);
                document.AddImageFileFromMemory(handle.ToInt64(), null,
null);

                // Recognize the document
                displayMessage("Recognizing...");
                document.Process(null);

                // Save results
                displayMessage("Saving results...");
                // Save results to pdf using 'balanced' scenario
                // FREngine.PDFExportParams pdfParams =
engineLoader.Engine.CreatePDFExportParams();
                // pdfParams.Scenario =
FREngine.PDFExportScenarioEnum.PES_Balanced;

                WorkerRole1.Engine.FileWriter fileWriter = new
WorkerRole1.Engine.FileWriter(inputBlobName, ".pdf");
                resultBlobName = inputBlobName + ".pdf";
                document.ExportToMemory(fileWriter,
FREngine.FileExportFormatEnum.FEF_PDF, null);

            }
            catch (Exception error)
            {
                displayMessage("error: " + error.Message);
            }
            finally
            {
                // Close the document
                document.Close();
            }

            return resultBlobName;
        }

        public void Dispose()
        {
            UnloadEngine();
        }
```

```
    }
}
```

Config.cs

```
class Config
{
    // Local storage as defined in ServiceDefinition.csdef
    private static string LocalStorageName = "LocalStorage1";
    // Connecting string to blob container
    public static readonly string ConnectionString =
"<connection_string_to_frestorage>";
    // Input and output containers name in your storage
    public static readonly string InputContainerName = "fre-input";
    public static readonly string OutputContainerName = "fre-output";

    // Return Customer Project ID for ABBYY FineReader Engine
    public static String GetCustomerProjectId()
    {
        return "<Your_Customer_Project_ID>";
    }

    // Return name of license token
    public static String GetLicenseTokenName()
    {
        return "<Token_number>.ABBYY.ActivationToken";
    }

    // Return license password
    public static String GetLicensePassword()
    {
        return "<Your_Online_License_token_password>";
    }


    // Return engine path
    public static String GetEngineFolder()
    {
        string engineSubfolder = "fre_packages\\LibraryPackage\\Bin64";
        string engineDllFolder =
Path.Combine(RoleEnvironment.GetLocalResource(LocalStorageName).RootPath
, engineSubfolder);
        return engineDllFolder;
    }
}
```

Technical Support

If you have any questions regarding the use of ABBYY FineReader Engine 12, first of all consult the documentation provided with this product (this Administrator's Guide and the Readme file). Useful information can also be found in the technical support section of our website at www.abbyy.com.

If you cannot find the answer to your question, please contact the ABBYY office serving your region by e-mail. Please provide the following information when contacting technical support:

- your first and last name
- the name of your organization
- your phone number (or fax, or e-mail)
- the serial number of your ABBYY FineReader Engine 12 license
- the protection type of your ABBYY FineReader Engine 12 package (software, hardware, or online)
- the build number (to determine the build number, see or Properties in the FREngine.dll local menu)
- a description of the problem
- a project that demonstrates the problem (with the necessary data files). This may be a slightly modified ABBYY FineReader Engine sample. We recommend that you compress the files using any popular archive format (ZIP, RAR, etc.)
- the name of your development tool
- the type of your computer and processor
- the version of your Windows operating system

You can gather some of the above information automatically:

1. Run the **AInfo** utility (AInfo.exe) from the **<Installation folder>/Bin/Support** folder (or **<Installation folder>/Bin64/Support** folder on 64-bit systems).
2. A dialog box will open displaying some of the above information. Save this information to a ZIP file.

Note: *No personal information or information about the user's computer is collected. You can view all the saved information in the created archive.*

You can also provide any additional information you consider important.

Support contacts

North/Central Americas	Customers from USA, Canada, Japan, Mexico or other Central American countries, please contact ABBYY North American Headquarters at dev_support@abbyyusa.com
Western Europe	Customers from Austria, Benelux, Denmark, France, Germany, Italy, Ireland, Norway, Portugal, Spain, Sweden, Switzerland, the United Kingdom or other Western European countries, please contact ABBYY European Headquarters at TechSupport_eu@abbyy.com
Eastern Europe and the Mediterranean	Customers from Ukraine, Moldova, Turkey, Israel or Eastern European countries, please contact ABBYY Eastern European Headquarters at engine_support@abbyy.ua
All other regions	Customers from the countries not mentioned above, please contact ABBYY International Headquarters at SDK_Support@abbyy.com