

# Kofax Communications Manager

## Installation Guide

Version: 5.4.0

Date: 2020-08-26

The logo for Kofax, consisting of the word "KOFAX" in a bold, blue, sans-serif font.

© 2013–2020 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

# Table of Contents

Preface.....	6
Related documentation.....	6
Training.....	7
Getting help with Kofax products.....	8
<b>Chapter 1: Before installation.....</b>	<b>9</b>
Ports.....	9
Databases.....	10
Databases for KCM Batch & Output Management.....	10
Manage databases.....	11
License.....	11
<b>Chapter 2: Install Kofax Communications Manager.....</b>	<b>13</b>
Deployment parameters.....	13
Example deployment command.....	17
Additional deployment parameters for KCM B&OM.....	17
Install KCM Repository on the Oracle database.....	19
Oracle user.....	19
Predefined setup options.....	19
Install KCM Designer for Windows.....	20
Install KCM on more than one computer.....	20
Install KCM instances on a computer using remote Contract Manager.....	20
Install dedicated Contract Manager on a computer with no instances.....	21
Install KCM without default contracts.....	21
Install KCM without the Contract Manager.....	22
Install KCM B&OM on a computer with no instances.....	22
Install KCM B&OM on multiple computers.....	23
Create a primary installation.....	23
Add a secondary installation.....	24
<b>Chapter 3: Upgrade your installation.....</b>	<b>26</b>
Perform the upgrade.....	26
Parameters for the upgrade.....	27
Perform the KCM B&OM upgrade.....	29
Additional upgrade parameters for KCM B&OM.....	31
Switch to an older installation.....	32
Upgrade a customized setup.....	32

Upgrade contract types and interfaces not in use by any contract.....	33
Upgrade KCM when installed on more than one computer.....	33
<b>Chapter 4: Install Kofax Communications Manager on Docker.....</b>	<b>35</b>
Restrictions.....	36
Create prerequisite images.....	36
Install fonts on prerequisite images.....	36
Install the Contract Manager.....	37
Create the Contract Manager image.....	37
Create the Contract Manager container.....	37
The InstallContractManager.ps1 installer.....	38
The DockerStartContractManager.exe entry point process.....	39
Install Kofax Communications Manager instances.....	39
Create an instance image.....	40
Create an instance container.....	40
The AddInstance.exe installer.....	41
The DockerStartInstance.exe entry point process.....	42
<b>Chapter 5: Create Docker containers with existing configuration.....</b>	<b>44</b>
Export KCM instance configuration.....	44
Export the Contract Manager configuration.....	45
<b>Chapter 6: Post-installation steps.....</b>	<b>47</b>
Test the installation.....	47
Remove the test web application.....	48
Access product documentation.....	48
Offline documentation.....	48
Add extra instances after installation.....	49
Extra instances parameters.....	50
Add extra instances after installation in combination with Oracle.....	52
Remove specific instances.....	52
Parameter to remove specific instances.....	52
Add custom KCM B&OM service hosts.....	52
Add custom service hosts.....	53
List custom service hosts.....	54
Remove custom service hosts.....	55
Required step to run KCM B&OM processes.....	55
Export and import KCM Batch & Output Management content.....	55
Export KCM Batch & Output Management content.....	56
Import KCM Batch & Output Management content.....	57
Move the KCM Repository database.....	57

<b>Chapter 7: Other settings</b> .....	<b>59</b>
Enable authentication and authorization for Contract Manager.....	59
Enable SSL for connections to the Contract Manager.....	59
Change an account for the WebServer services.....	61
Configure request rate limitation.....	61
Install and configure KCM Toolbox for Word.....	61
<b>Chapter 8: Use of Microsoft Word by KCM Core</b> .....	<b>63</b>
Adjust settings in Microsoft Word.....	63
<b>Chapter 9: Uninstallation</b> .....	<b>64</b>
<b>Chapter 10: Troubleshooting</b> .....	<b>65</b>
Package cannot connect to database.....	65
Services cannot be created.....	65
Cannot start KCM Core for development or test license.....	65
Warning messages.....	65
Deployment log files are empty.....	66
CMRestAPI does not start during system startup.....	66
Conversion from PDF to XPS fails.....	66

# Preface

This guide contains instructions on how to install Kofax Communications Manager (also known as KCM). You can use the KCM package to deploy a single instance of the software.

**Note** For information on supported operating systems and other system requirements, see the *Kofax Communications Manager Technical Specifications* document on the [Kofax Communications Manager documentation site](#).

## Related documentation

The documentation set for Kofax Communications Manager is available here:<sup>1</sup>

<https://docshield.kofax.com/Portal/Products/KCM/5.4.0-cli2a1c07m/KCM.htm>

In addition to this guide, the documentation set includes the following items:

***Kofax Communications Manager Release Notes***

Contains late-breaking details and other information that is not available in your other Kofax Communications Manager documentation.

***Kofax Communications Manager Technical Specifications***

Provides information on supported operating system and other system requirement for Kofax Communications Manager.

***Kofax Communications Manager Getting Started Guide***

Describes how to use Contract Manager to manage instances of Kofax Communications Manager.

***Kofax Communications Manager Batch & Output Management Getting Started Guide***

Describes how to start working with Batch & Output Management.

***Help for Kofax Communications Manager Designer***

Contains general information and instructions on using Kofax Communications Manager Designer, which is an authoring tool and content management system for Kofax Communications Manager.

***Kofax Communications Manager Repository Administrator's Guide***

Describes administrative and management tasks in Kofax Communications Manager Repository and Kofax Communications Manager Designer for Windows.

---

<sup>1</sup> You must be connected to the Internet to access the full documentation set online. For access without an Internet connection, see "Offline documentation."

***Kofax Communications Manager Repository User's Guide***

Includes user instructions for Kofax Communications Manager Repository and Kofax Communications Manager Designer for Windows.

***Kofax Communications Manager Template Scripting Language Developer's Guide***

Describes the KCM Template Script used in Master Templates.

***Kofax Communications Manager Core Developer's Guide***

Provides a general overview and integration information for Kofax Communications Manager Core.

***Kofax Communications Manager Core Scripting Language Developer's Guide***

Describes the KCM Core Script.

***Kofax Communications Manager Batch & Output Management Developer's Guide***

Describes the Batch & Output Management scripting language used in KCM Studio related scripts.

***Kofax Communications Manager Repository Developer's Guide***

Describes various features and APIs to integrate with Kofax Communications Manager Repository and Kofax Communications Manager Designer for Windows.

***Kofax Communications Manager API Guide***

Describes Contract Manager, which is the main entry point to Kofax Communications Manager.

***Kofax Communications Manager ComposerUI for HTML5 JavaScript API Web Developer's Guide***

Describes integration of ComposerUI for HTML5 into an application, using its JavaScript API.

***Kofax Communications Manager DID Developer's Guide***

Provides information on the Database Interface Definitions (referred to as DIDs), which is an alternative method retrieve data from a database and send it to Kofax Communications Manager.

***Kofax Communications Manager ComposerUI for ASP.NET and J2EE Customization Guide***

Describes the customization options for KCM ComposerUI for ASP.NET and J2EE.

***Kofax Communications Manager ComposerUI for ASP.NET Developer's Guide***

Describes the structure and configuration of KCM ComposerUI for ASP.NET.

***Kofax Communications Manager ComposerUI for J2EE Developer's Guide***

Describes JSP pages and lists custom tugs defined by KCM ComposerUI for J2EE.

## Training

To ensure your best experience in working with KCM Designer, complete our complimentary and interactive e-learning modules located at [www.kofax.com](http://www.kofax.com).

## Getting help with Kofax products

The [Kofax Knowledge Base](#) repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the Kofax [website](#) and select **Support** on the home page.

**Note** The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.  
Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.  
Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).  
Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).  
Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.  
Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.



## Chapter 1

# Before installation

## Ports

When deployed, Kofax Communications Manager uses the following ports.

Port	Description
80	In use by the KCM Content Management API service
443	In use by Tomcat for the Contract Manager for runtime purposes when using the SSL protocol
2587	In use by KCM Repository Server
3000-3003	In use by KCM Core
8007	In use by Tomcat. Shutdown port for KCM Designer instance
8009	In use by Tomcat for AJP connectivity
8080	In use by Tomcat for KCM ComposerUI for HTML5 and KCM ComposerUI for J2EE
8081	In use by Tomcat for the Contract Manager for runtime purposes when not using the SSL protocol
8180	In use by Tomcat for the LogServer to retrieve logs
8280	In use by Tomcat for KCM Designer
8282	In use by Tomcat for the Contract Manager to accept new remote instance registrations.

### Note

- Inbound TCP/IP Port 8081 or 443, depending on whether or not the SSL protocol is being used, should be opened in the firewall and available for use by Kofax Communications Manager. All other ports mentioned above should not be exposed externally.
- KCM additional instances use incremental ports, apart from those specified above, for the KCM Repository Server and KCM Core. A new KCM Repository Server requires one additional port. A new KCM Core requires two additional ports.
- To prevent a Slow HTTP Denial of Service attack, you should set up your firewall to limit the amount of connections per host for port 8081. For example, you can limit the amount to 50 connections per host.

## Databases

The deployment process uses a database for KCM Repository, which is configured during installation.

**Note** The installer does not create this database automatically. You need to create an empty database manually. For SQL Server, create the database with a case-insensitive collation.

The database can be either on SQL Server or on Oracle. On SQL Server, the SQL account used must have owner rights for the database and use the "dbo" schema. SQL Server must have both SQL Server and Windows Authentication mode enabled. Also, the TCP/IP protocol for SQL Server must be enabled.

Consider the following when adding KCM extra instances:

- KCM additional instances require their own KCM Repository database.
- The content in the KCM Repository databases is not removed when an instance is uninstalled.

## Databases for KCM Batch & Output Management

This section is only relevant if you are installing KCM Batch & Output Management along with the main product.

The Batch & Output Management deployment process requires two empty databases:

- One database for the B&OM Repository
- One database for the runtime information

You should create empty databases manually. They can be either SQL Server or Oracle, but must both be of the same DBMS. Also, you should create the databases with a case-insensitive collation.

**Note** KCM B&OM requires the Oracle 32-bit client. To perform a complete installation of KCM, if you are deploying KCM Repository on Oracle, you need both the 32-bit and 64-bit Oracle clients.

When installing on SQL Server, the user specified with the `Database!User` deployment parameter requires one following privilege, in addition to the regular requirements:

```
ALTER TRACE
```

When installing on Oracle, the following privileges are required on both database users, in addition to the regular requirements:

- ALTER SESSION
- CREATE VIEW
- CREATE PROCEDURE
- CREATE SEQUENCE

Also, the Oracle user for the B&OM Repository also requires these privileges:

```
CREATE TYPE
```

```
CREATE TRIGGER
```

```
EXECUTE ON DBMS_LOCK  
  
EXECUTE ON DBMS_LOB  
  
EXECUTE ON DBMS_OUTPUT  
  
EXECUTE ON CTX_DDL
```

## Manage databases

All databases that you use for KCM contain important production data. Unless following specific instructions from Kofax Technical Support, please do not make database modifications, because they may cause irreversible damage and loss of data. Also, be sure to back up all databases on a regular basis and follow the best practices recommended by your database provider.

## License

Use of the Kofax Communications Manager package requires a valid license. The deployment process searches for a license file. The correct structure of the license file is shown here.

```
<?xml version="1.0" encoding="utf-8"?>  
<ccm:Licence Version="1" xmlns:ccm="http://www.aia-itp.com/licence">  
  <ccm:Product Name="ITP Server">  
    <ccm:SubLicence Type="Base">  
      <ccm:CompanyName></ccm:CompanyName>  
      <ccm:NumberOfDocumentProcessors></ccm:NumberOfDocumentProcessors>  
      <ccm:ExpirationDate></ccm:ExpirationDate>  
      <ccm:Certificate></ccm:Certificate>  
      <ccm:Environment></ccm:Environment>  
      <ccm:LicenceCode></ccm:LicenceCode>  
    </ccm:SubLicence>  
    <ccm:SubLicence Type="OnLine">  
      <ccm:ExpirationDate></ccm:ExpirationDate>  
      <ccm:Certificate></ccm:Certificate>  
      <ccm:NumberOfUsers></ccm:NumberOfUsers>  
      <ccm:LicenceCode></ccm:LicenceCode>  
    </ccm:SubLicence>  
  </ccm:Product>  
  <ccm:Product Name="ITP MDK Repository">  
    <ccm:SubLicence Type="Base">  
      <ccm:CompanyName></ccm:CompanyName>  
      <ccm:ExpirationDate></ccm:ExpirationDate>  
      <ccm:Certificate></ccm:Certificate>  
      <ccm:Environment></ccm:Environment>  
      <ccm:LicenceCode></ccm:LicenceCode>  
      <ccm:NumberOfUsers></ccm:NumberOfUsers>  
    </ccm:SubLicence>  
    <ccm:SubLicence Type="Model Developer">  
      <ccm:CompanyName></ccm:CompanyName>  
      <ccm:ExpirationDate></ccm:ExpirationDate>  
      <ccm:Certificate></ccm:Certificate>  
      <ccm:Environment></ccm:Environment>  
      <ccm:LicenceCode></ccm:LicenceCode>  
      <ccm:NumberOfUsers></ccm:NumberOfUsers>  
    </ccm:SubLicence>
```

```
<ccm:SubLicence Type="Textblock Editor">
  <ccm:CompanyName></ccm:CompanyName>
  <ccm:ExpirationDate></ccm:ExpirationDate>
  <ccm:Certificate></ccm:Certificate>
  <ccm:Environment></ccm:Environment>
  <ccm:LicenceCode></ccm:LicenceCode>
  <ccm:NumberOfUsers></ccm:NumberOfUsers>
</ccm:SubLicence>
</ccm:Product>
</ccm:Licence>
```

For MDK Repository Base license, the value for `NumberOfUsers` must be set to 1.

For development or test purposes, the value for ITP/Server Base license should be "Development and Test." For all other environment elements, the value should be "Development or Test".

**Note** The KCM package might not deploy the full number of Document Processors that are permitted by the license. The default configuration does not install more than four Document Processors. Use the `Instance!NumberOfDPs` parameter during deployment to specify a different number of Document Processors. Then, you can use KCM Core Administrator to add or remove Document Processors.

## Chapter 2

# Install Kofax Communications Manager

The Kofax Communications Manager package is installed and deployed in phases. In the first phase, prerequisites are checked. If the deployment fails in this phase, you can restart it after the cause of the failure is resolved. If it fails again, you can uninstall the software.

1. To start the deployment, copy the package to the server.
2. On the server, extract the package contents to the root directory where a new directory, InstallServer, is automatically created. The unzipped files appear in the new directory, which is referred to as "package root."
3. Set up a PowerShell command to start the installation. This command line should call `Install.ps1` with the list of parameters (see [Deployment parameters](#) ).
4. Click **Run as Administrator** to open a PowerShell window (64-bit) with elevated rights.
5. Navigate to the package root.
6. To run the prepared command, execute the **Set-ExecutionPolicy AllSigned** command.
7. If the following prompt appears, select **Always Run (A)**.

### Example

```
Do you want to run software from this untrusted publisher?  
File ...Install.ps1 is published by CN=Aia Software B.V., O=Aia Software B.V.,  
L=Nijmegen,  
S=Gelderland, C=NL and is not trusted on your system.  
Only run scripts from trusted publishers.  
[V] Never run [D] Do not run [R] Run once [A] Always run [?] Help (default is  
"D") :  
Navigate to the package root.
```

8. Wait for the process to finish.

**Note** When the installation is finished, you can remove the package root directory.

## Deployment parameters

The following is a description of deployment parameters that you need to install Kofax Communications Manager.

**Note** When a parameter contains spaces and/or special characters, such as \$ or &, enclose the parameter in single quotes to prevent errors.

Parameter	Description
Database!Type	Required. Specifies the type of the databases that deployment uses. The database can be either on SQL Server or on Oracle.
Database!ConnectionString	Required, unless the deployment uses the <code>ContractManagerOnly</code> or the <code>OutputManagementOnly</code> predefined setup. Specifies the connection string that KCM Repository uses to connect to its database. Do not specify username or password in this connection string.  <b>Note</b> This must be an ODBC connection string.
Database!User	Optional. The database user that KCM Repository uses to connect to its database. In case of the SQL Server database, the user must have owner rights to the database.
Database!Password	Optional. The database password that KCM Repository uses to connect to its database.
Services!User	The user account used for the installed services. This user must be a local administrator and must have the "Logon as a service" rights. When specifying a local user, place a period and backslash (.) at the beginning of the parameter. <b>Example</b> .myuser
Services!Password	The password for the services user.
License!File	The location of the .xml file with the license information.
Webserver!ServiceUser	The user account used to run the web server services. This user must have the "Logon as a service" rights. When specifying a local user, add a period and backslash (.) at the beginning of the parameter.  This parameter is required for the <code>RunTimeOnDemand-NoContractManager</code> setup, even if nothing is installed on a web server. In this case the user does not need a Tomcat installation. <b>Example</b> .myuserws  <b>Note</b> <code>Webserver!ServiceUser</code> parameter is not mandatory for the <code>OutputManagementOnly</code> setup.
Webserver!ServicePassword	The password belonging to the user provided through <code>Webserver!ServiceUser</code> .  This parameter is required for the <code>RunTimeOnDemand-NoContractManager</code> setup, even if nothing is installed on a web server. In this case, the user does not need a Tomcat installation.  <b>Note</b> <code>Webserver!ServicePassword</code> parameter is not mandatory for the <code>OutputManagementOnly</code> setup.

Parameter	Description
DesignerAdministratorAccount!Password	The password of the designer Administrator account that is created automatically by the installation. The password must be at least 20 characters long, and may only consist of letters (a-z, A-Z), numbers (0-9), and the "-" and "_" characters. This parameter is required only for default setups (no Deploy!PredefinedSetup).
Deploy!RootPath	Optional. The root drive/directory where the software is deployed. If this argument is omitted, the package is deployed to C:\.
Deploy!PredefinedSetup	Optional. The name of a supported predefined setup, where not all components are installed. Supported values are Runtime-Designtime-NoContractManager, RuntimeOnDemand-NoContractManager, ContractManagerOnly, Runtime-Designtime-RemoteContractManager, Runtime-Designtime-NoContract, and OutputManagementOnly. For a complete setup, omit this parameter. For details, see <a href="#">Predefined setup options</a> .
Instance!NumberOfDPs	Optional. Specifies how many Document Processors should be deployed for KCM Core. The default value is 4 or the number of Document Processors allowed by your license.
ContractManager!KeyStoreFile	Optional. The location of the certificate file (.pfx or .p12) for this server. Default connection to KCM is done securely over HTTPS using the SSL protocol. Required if <code>ContractManager!UseSSL</code> is true. This parameter is ignored for setups, where the Contract Manager is not installed.
ContractManager!KeyStorePassword	Optional. The KeyStore password. This is the password needed for the KeyStore File. Required if <code>ContractManager!UseSSL</code> is true. This parameter is ignored for setups, where the Contract Manager is not installed.

Parameter	Description
ContractManager!UseSSL	<p>Optional. Overrides the default behavior on using the SSL protocol. Defaults to true, indicating that any connections to KCM are secure using the SSL protocol and that KCM can be connected over HTTPS.</p> <p>Set the value to false to connect to KCM over HTTP.</p> <p>The <code>ContractManager!UseSSL</code> option configures the connectors for the service Catalina-CM in Tomcat.</p> <p>For security reasons, it disables TLSv1.0, and it only allows ciphers of at least 128 bits.</p> <p>This configuration can be found in the Tomcat <code>server.xml</code> file, which resides in the Tomcat installation folder under <code>instance-CCMRuntime-&lt;KCM version&gt;\conf</code>. For more information, see the Tomcat documentation available on the Internet.</p> <p>If you configured SSL for connections to the Contract Manager manually, that is, you did not deploy KCM with the <code>ContractManager!UseSSL</code> parameter, and you want to remove the HTTP connectors, follow these steps:</p> <ol style="list-style-type: none"> <li>1. Locate the <code>ManageCM.exe.config</code> file that resides in: <code>&lt;deploy root&gt;\KCM\Programs\5.4\Management</code></li> <li>2. Open the file and change the <code>ReloadURLs</code> setting as follows:</li> </ol> <pre>&lt;add key="ReloadURLs" value=" https://localhost:&lt;CM SSL port&gt;/ccm/ Administration/reloaddatabase;https:// localhost:&lt;PartnerAdmin SSL port&gt;/ccm/ Administration/reloaddatabase"/&gt;</pre> <p>This parameter is ignored for setups, where the Contract Manager is not installed.</p>
ContractManager!UseAuthentication	<p>Optional. If set to false, anyone with access to the Contract Manager can execute any SOAP call on its interface.</p> <p>If set to true, when executing SOAP calls on the Contract Manager, the calling application should supply a user name and a password using an HTTP Basic Authentication header.</p> <p>If omitted, default value is true.</p> <p>This parameter is ignored for setups, where the Contract Manager is not installed.</p>



Parameter	Description
ContractManager!SharedResourceLocation	<p>Optional. Specifies the location of a folder where the shared objects for import and export, such as project files and reference projects, are stored.</p> <p>This parameter is ignored for setups, where the Contract Manager is not installed.</p> <p>By default, the shared objects are stored here:  <code>&lt;deploy root&gt;\KCM\Work\&lt;version number&gt;\SharedResources.</code></p> <p><b>Note</b> If you configure the folder in a location outside <code>&lt;deploy root&gt;\KCM\Work\&lt;version number&gt;</code>, the new location is kept during future upgrades.</p>
ExampleWebApp!Install	<p>Optional. If set to true, an example web application is installed to test the installation and to reference KCM ComposerUI integration resources.</p> <p>If omitted, default value is false.</p> <p>This parameter is ignored for setups, where the Contract Manager is not installed.</p>
OutputManagement!Install	<p>Only relevant if you have the general Kofax Communications Manager package that includes KCM B&amp;OM.</p> <p>Optional. Specifies whether to install B&amp;OM. Default is true.</p> <p>If set to false, this parameter is allowed, but ignored for the OutputManagementOnly setup.</p>

For the KCM B&OM deployment parameters, see [Additional deployment parameters for KCM B&OM](#).

## Example deployment command

The following is an example deployment command.

```

.\Install.ps1 Database!Type=SQLServer
'Database!ConnectionString=Driver=SQL Server;Server=KCMServer;Database=KCM_Repository'
Database!User=itpuser Database!Password=pwd123 Services!User=.myuser
Services!Password=pwd456 'License!File=C:\{path}\My License\license.xml'
Webserver!ServiceUser=.myuserws Webserver!ServicePassword=pwd789
ContractManager!KeyStoreFile=C:\{path}\cert.pfx
ContractManager!KeyStorePassword=pass123
DesignerAdministratorAccount!Password=default-administrator-password

```

## Additional deployment parameters for KCM B&OM

The following is a description of deployment parameters that you might need to customize the KCM B&OM installation.

Parameter	Description
OutputManagement!InstallType	<p>Optional. Specifies the type of KCM B&amp;OM installation to deploy:</p> <ul style="list-style-type: none"> <li>• <code>Full</code>. A complete installation.</li> <li>• <code>StudioOnly</code>. Only deploys KCM Studio. Must be combined with <code>OutputManagement!Secondary=True</code>.</li> <li>• <code>ServerOnly</code>. A complete installation, omitting KCM Studio.</li> </ul> <p>Default is <code>Full</code>.</p>
OMRepositoryDB!ConnectionString	<p>Required. Specifies the connection string that Batch &amp; Output Management uses to connect to its repository database. Do not specify username or password in this connection string.</p> <p><b>Note</b> Depending on the <code>Database!Type</code> parameter, this must be either a valid Oracle or SQL Server connection string. This cannot be an ODBC connection string.</p>
OMRepositoryDB!User	Optional. Specifies the user that Batch & Output Management uses to connect to its repository database.
OMRepositoryDB!Password	Optional. Specifies the password that Batch & Output Management uses to connect to its repository database.
OMRuntimeDB!ConnectionString	<p>Required. Specifies the connection string that Batch &amp; Output Management uses to connect to its runtime database. Do not specify username or password in this connection string.</p> <p><b>Note</b> Depending on the <code>Database!Type</code> parameter, this must be a valid Oracle or SQL Server connection string. This cannot be an ODBC connection string.</p>
OMRuntimeDB!User	Optional. Specifies the user that Batch & Output Management uses to connect to its runtime database.
OMRuntimeDB!Password	Optional. Specifies the password that Batch & Output Management uses to connect to its runtime database.
OutputManagement!Secondary	<p>Optional. When set to <code>True</code>, the new B&amp;OM installation uses the databases and storage folder of an existing B&amp;OM installation. For more information, see <a href="#">Install KCM B&amp;OM on multiple computers</a>.</p> <p>Default is <code>False</code>. Can only be set to <code>True</code> if the <code>OutputManagementOnly</code> predefined setup is used.</p>

## Install KCM Repository on the Oracle database

During installation, you can deploy KCM Repository on the Oracle database.

**Note** Oracle Client versions 11.02.00.01 and 11.02.00.02 are not supported for use with KCM Repository.

1. To install KCM Repository on the Oracle database, set the parameter `Database!Type=Oracle` to the deployment command.
2. Provide a value for the `Database!ConnectionString` parameter. For example, `DSN=MyORA`, where MyORA is the name of a data source created using the 64-bit version of the ODBC Data Source Administrator tool.

This parameter should not contain UserID or Password elements.

**Note** Previous versions of KCM required a 32-bit DSN. Now KCM Repository is a 64-bit program, so a 64-bit DSN is required.

### Oracle user

When logging on to Oracle, the user identifier is entered without quotes. When a user is entered without quotes, Oracle interprets the user name as uppercase. As the predefined user identifier contains a dot, it is likely that you create a user as a quoted identifier. Do not use lowercase characters in the quoted user identifier.

#### Example

"KCM\_01\_5.4" is able to log on.

"Kcm\_01\_5.4" fails to log on because Kcm\_01\_5.4 is automatically interpreted as KCM\_01\_5.4, and that user does not exist.

## Predefined setup options

The installation package contains predefined setup options. The complete installation consists of KCM Core, KCM Repository, KCM Designer, KCM ComposerUI for HTML5, Batch & Output Management, and the Contract Manager.

- The Runtime-Designtime-NoContractManager setup performs a complete installation, but omits the Contract Manager.
- The RuntimeOnDemand-NoContractManager setup installs KCM Core and KCM Repository only.
- The ContractManagerOnly setup installs the Contract Manager only.
- The Runtime-Designtime-RemoteContractManager setup performs a complete installation, omits the Contract Manager, and is configured to have Contract Manager on another machine/server.

- The Runtime-Designtime-NoContract setup performs a complete installation, but omits creating default contracts.
- The OutputManagementOnly setup installs KCM Batch & Output Management only.

**Note** Currently, KCM Designer for Windows is still supported to get access to KCM Repository. KCM Designer for Windows is not installed by default so you have to install it manually. For more information, see [Install KCM Designer for Windows](#).

## Install KCM Designer for Windows

KCM Designer for Windows is a client application used to perform certain tasks for KCM Designer, such as create projects and users. You can install the application manually after the KCM installation package is deployed. For each instance, an installer and detailed information are provided.

1. Once the installation package is deployed, navigate to `<deploy root>\KCM\Work\<version>\Instance_01\designer\Client` and locate the KCM Designer for Windows installer, which is the `setup.exe` file.
2. Run the installer.

The installer prompts you to enter the installation name that will determine the directory name where the client will be residing. When the installer prompts you to provide the necessary values for server and host, open the `ReadMe` file located in the same directory and copy the values from the file.

By default, the client is deployed in `C:\Program Files (x86)` for a 64-bit OS.

## Install KCM on more than one computer

You can install KCM on multiple computers, with one computer containing the Contract Manager and several computers containing one or more instances. This makes KCM scalable and easy to upgrade.

**Note** Ensure that the shared resource folder configured on the Contract Manager is locally accessible from each computer containing the instance. It is required for the project import and export functionality in KCM Designer.

## Install KCM instances on a computer using remote Contract Manager

1. Install the KCM package with no Contract Manager.

To do so, use the predefined setup `Runtime-Designtime-RemoteContractManager`, as shown in this example command.

```
.\install Database!ConnectionString="Driver=SQL
Server;Server=KCMServer;Database=KCM_Repository"
Database!User=itpuser
Database!Password=pwd123
Services!User=.\myuser
Services!Password=pwd456
Webserver!ServiceUser=.\myuserws
Webserver!ServicePassword=pwd789
License!File="c:\{path}\my_licenses\license.xml"
```

```
Deploy!PredefinedSetup=Runtime-Designtime-RemoteContractManager
```

2. Now you need to configure KCM instances on the remote Contract Manager. For information on how to perform the action, see [Install dedicated Contract Manager on a computer with no instances](#).

## Install dedicated Contract Manager on a computer with no instances

1. Install the KCM package on a machine with the Contract Manager only.

To do so, use the predefined setup `ContractManagerOnly`, as shown in this example command.

```
.\install Services!User=. \myuser
      Services!Password=pwd456
      Webserver!ServiceUser=. \myuserws
      Webserver!ServicePassword=pwd789
      ContractManager!KeyStoreFile=C:\{path}\cert.pfx
      ContractManager!KeyStorePassword=pass123
      Deploy!PredefinedSetup=ContractManagerOnly
```

2. To configure the Contract Manager, use the `ManageCM` tool that resides in: `<deploy root>\KCM\Programs\<version>\Management`.
3. Register an instance to your remote Contract Manager with the following command, substituting the example parameters for your actual parameters.

```
ManageCM /RegisterInstance ... /Host=host42 /Instance=5
```

4. Now you need to add contracts to the Contract Manager and associate the instance with the contracts. For information on how to do so, see the chapter "Work with the Contract Manager" in the *Kofax Communications Manager Getting Started Guide*.

## Install KCM without default contracts

The KCM installation package creates default contracts for each instance. To create your own contracts, you need to install KCM without default contracts and then create contracts to access KCM instances.

1. Install the KCM package on a machine using the predefined setup `Runtime-Designtime-NoContract`, as shown in this example command.

```
.\install Database!ConnectionString="Driver=SQL
      Server;Server=KCMServer;Database=KCM_Repository"
      Database!User=itpuser
      Database!Password=pwd123
      Services!User=. \myuser
      Services!Password=pwd456
      Webserver!ServiceUser=. \myuserws
      Webserver!ServicePassword=pwd789
      ContractManager!KeyStoreFile=C:\{path}\cert.pfx
      ContractManager!KeyStorePassword=pass123
      License!File="c:\{path}\my_licenses\license.xml"
      Deploy!PredefinedSetup=Runtime-Designtime-NoContract
```

This predefined setup omits default contracts, but creates one KCM instance. To add extra instances, see [Add extra instances after installation](#).

2. As there are no default contracts, you need to create contracts to access KCM instances.

To do so, you need the `ManageCM` tool that resides in: `<deploy root>\KCM\Programs\<version>\Management`

For information on how to create contracts, see the chapter "Work with the Contract Manager" in the *Kofax Communications Manager Getting Started Guide*.

- Now you need to create interfaces and contract types to add them to your contract.  
For information on how to perform the actions, see the chapter "Work with the Contract Manager" in the *Kofax Communications Manager Getting Started Guide*.

## Install KCM without the Contract Manager

The Contract Manager is designed to manage connections to KCM instances, but you can connect to the various KCM products directly. In this case, if more than one instance is installed on a computer, there will be multiple instances of the various KCM components accessible through different ports. This implies that you need to monitor which ports are needed to access a particular KCM component.

Install the KCM package on a computer using the predefined setup `NoContractManager`, as shown in this example command.

```
.\install Database!ConnectionString="Driver=SQL
Server;Server=KCMServer;Database=KCM_Repository"
    Database!User=itpuser
    Database!Password=pwd123
    Services!User=. \myuser
    Services!Password=pwd456
    Webserver!ServiceUser=. \myuserws
    Webserver!ServicePassword=pwd789
    License!File="c:\{path}\my_licenses\license.xml"
    Deploy!PredefinedSetup=Runtime-Designtime-NoContractManager
```

This predefined setup omits the Contract Manager. Access to the various KCM components will be done directly using the port it is listening on.

- For KCM Core, the port is 3003 for the instance 1; 3005 for the instance 2, and so on. KCM Core offers an integration layer consisting of APIs and a COM library, an AS/400 command, and Java classes used to integrate KCM Core in a business application. For more information on KCM Core, see the *Kofax Communications Manager Core Developer's Guide*.
- KCM Designer is listening on the port 8280 for the instance 1; port 8281 for the instance 2, and so on. You can access KCM Designer with the following example URL:

**http://<kcm machine>:<port>/ccm\_designer\_<instance>**

Where <kcm machine> is the name of a computer on which KCM is installed, and <instance> is the number of an instance in two digits.

**Example** `http://host24:8280/ccm_designer_01`

## Install KCM B&OM on a computer with no instances

You can install KCM Batch & Output Management separately on a dedicated computer, and then configure it to use a KCM instance of an existing remote KCM installation for document composition.

- Install the KCM package on a machine with Batch & Output Management only. To do so, use the predefined setup `OutputManagementOnly` as shown in the example command below.

```
.\install OMRepositoryDB!
ConnectionString="Server=KCMServer;Database=KCM_BOM_Repository"
    OMRepositoryDB!User=itpuser
    OMRepositoryDB!Password=pwd123
    OMRuntimeDB!
ConnectionString="Server=KCMServer;Database=KCM_BOM_Runtime"
    OMRuntimeDB!User=itpuser
```

```
OMRuntimeDB!Password=pwd123
Services!User=.\myuser
Services!Password=pwd456
Deploy!PredefinedSetup=OutputManagementOnly
OutputManagement!InstallType=Full
```

If you want to install only the server components of B&OM, change the `OutputManagement!InstallType` parameter to `ServerOnly`. To only install the KCM Studio component of B&OM, change it to `StudioOnly`. It is only allowed in combination with `OutputManagement!Secondary=True`.

2. Configure the KCM instance used for document composition. In KCM Studio, go to the Administration tab, click System Administration, and then click System Object. Afterwards, locate the KCM/Core and KCM/Repository sections, and adjust the respective Host and Port properties.
3. Configure the referenced KCM/Core to accept remote connections. Open the KCM/Core Administrator of the KCM instance that is referenced by B&OM, go to the DP Manager tab, and enable remote access in the Security section.

## Install KCM B&OM on multiple computers

You can distribute a single B&OM installation over multiple machines. Such installation is created by extending a primary B&OM installation with one or more secondary B&OM installations. Each secondary installation uses the databases and storage folder of the primary installation, instead of having its own databases and a storage folder. Additionally, each secondary installation uses KCM Core and KCM Repository of the primary installation.

### Create a primary installation

The primary installation can be a full KCM installation or a dedicated B&OM installation with a predefined setup `OutputManagementOnly`. All secondary installations must be dedicated B&OM installations.

1. Install the KCM package typically, with a setup containing Batch & Output Management. This can either be the default setup, that installs all KCM components, or the `OutputManagementOnly` predefined setup, that installs B&OM only. In this case, use `OutputManagement!Secondary=False`, and `OutputManagement!InstallType` either `Full` or `ServerOnly`.

During the installation, make sure the database host is referred to either by computer name or IP address. It concerns the `OMRepositoryDB!ConnectionString` and `OMRuntimeDB!ConnectionString` parameters.

**Note** You can also alter the connection strings after installation. First, manually edit the repository connection string in the `repository.config` file in the B&OM programs folder. Then, restart the B&OM services and edit the Odin connection string from the **DB Alias Administration** on the **Administration** tab in KCM Studio.

2. Replace the default storage folder with a shared folder that can be accessed from both the primary and all secondary installations using the exact same path.
  - a. In KCM Studio, on the **Administration** tab, click **System Administration** and then click the system object.
  - b. In the **Object Inspector**, locate the **KCM Configuration** section, and change the **Storage folder** settings to the shared folder preferably using an UNC path.
3. This step is only necessary if you want to run the `CCM_Communication` process on any of the secondary computers.
  - a. In KCM Studio, on the **Administration** tab, click **System Administration** and then click the system object.
  - b. In the **Object Inspector**, locate the **KCM/Core** and **KCM/Repository** sections and adjust both the **Host** properties, if necessary.
  - c. Double-click the Repository database. If the **Connection String** property is set to "localhost", change it to the target computer name or its IP address.
  - d. Click **OK**.
  - e. Configure the **KCM/Core** referenced by the primary installation to accept remote connections. In **KCM/Core Administrator**, go to the **DP Manager** tab, and enable remote access to the **Security** section.
  - f. Click **OK**.
  - g. Save the changes.

## Add a secondary installation

1. To deploy an additional installation on a new computer, install the KCM package, using `Deploy!PredefinedSetup=OutputManagementOnly` and `OutputManagement!Secondary=True`. All of the `OutputManagement!InstallType` are supported. Use the `OMRepositoryDB` and `OMRuntimeDB` parameters to refer to the Batch & Output Management databases of the primary installation. These databases are verified, but are not affected by the secondary installation. Example command for SQL Server:

```
.\install OMRepositoryDB!  
ConnectionString="Server=KCMServer;Database=KCM_BOM_Repository"  
    OMRepositoryDB!User=itpuser  
    OMRepositoryDB!Password=pwd123  
    OMRuntimeDB!  
ConnectionString="Server=KCMServer;Database=KCM_BOM_Runtime"  
    OMRuntimeDB!User=itpuser  
    OMRuntimeDB!Password=pwd123  
    Services!User=. \myuser  
    Services!Password=pwd456  
    Deploy!PredefinedSetup=OutputManagementOnly  
    OutputManagement!Secondary=true
```

2. This step is only necessary if `OutputManagement!InstallType` is either `Full`, which is the default value when the option is omitted. or `ServerOnly`. On the primary KCM B&OM installation, navigate to `<deploy root>\KCM\Programs\5.4\Output Management` and copy the `Repository.Config` and `UserRepository.Config` files. Paste the files into the same folder on the KCM B&OM secondary installation.



3. This step is only necessary if `OutputManagement!InstallType` is `StudioOnly`. In the `CCMe.Studio.exe.config` file, go to the `<deploy root>\KCM\Programs\5.4\OutputManagement` folder. Open the `<appSettings>` node and, under it, find all URLs contained in the value attributes of the `<Add>` subnodes. In the URLs, replace all references to "localhost" with the target computer name or IP address of the primary installation.

**Note** The ports referenced in these URLs also need to be accessible from the secondary installation, through the respective firewalls.

4. If you want to run the processes on both the primary and the secondary installations, make sure that all resource locations are specified by means of UNC paths reachable from both installations.

**Note** It does not apply to the "request" and "output" folders of `CCM_Registration` and `CCM_Distribution`. As these folders are not allowed to be UNC paths, they are always interpreted as local to the computer on which the process is running.

## Chapter 3

# Upgrade your installation

You can upgrade KCM versions 5.0 and later to KCM 5.4.0. The upgrade installs KCM 5.4.0 alongside your existing installation, using the same configuration that the current KCM was installed with.

Note the following when upgrading your KCM installation:

- If you have an older KCM installation, first upgrade it to 5.0. Then, you can upgrade it to 5.4.0.
- You need to manually reinstall any custom additional language packs. We recommend that you execute the script `GetLanguagePack.ps1` prior to upgrading the product, update the custom language packs so they include new translations, and then install the language packs using the script `AddLanguagePack.ps1`. For information on the scripts, see the chapter "Manage language packs" in the *Kofax Communications Manager Getting Started Guide*.

## Perform the upgrade

To upgrade a KCM installation, you need the following:

- Create a new database for the KCM instance
- Enough disk space to accommodate KCM Repository content

**Note** The new databases must be owned by the user who owns the existing databases.

In addition to the preceding requirements, when you upgrade an installation that includes KCM Batch & Output Management, the following is required:

- Create a new database for KCM Batch & Output Management
- Create a new database for KCM Batch & Output Management runtime information
- Enough disk space to accommodate KCM Batch & Output Management content

Also, check the lists of the required and optional parameters prior to performing the upgrade (see [Parameters for the upgrade](#) and [Additional upgrade parameters for KCM B&OM](#) ).

To upgrade a standard installation, execute the following command, substituting the example parameters values with your actual values.

```
.\Upgrade.ps1
'Database!ConnectionString=Driver=SQL Server;Server=KCM Server;Database=NewRepository'
Database!User=itpuser Database!Password=pwd123
Services!User=. \myuser Services!Password=pwd456
'License!File=C:\{path}\MyLicense\license.xml '
Webserver!ServiceUser=. \myuserws Webserver!ServicePassword=pwd789
ContractManager!KeyStoreFile=C:\{path}\cert.pfx
ContractManager!KeystorePassword=pass123
OldCMDDatabase!User=cmuser OldCMDDatabase!Password=pwd011
```

Replace the preceding example parameters with actual values that apply to your currently active KCM setup.

When the upgrade is completed, the 5.4.0 installation is active and available for testing. The previously active installation is disabled.

The KCM Repository content is now copied to the new database. Any changes to the Repository content are not reflected in the older version.

**Important** An upgrade only creates a basic Tomcat Server configuration for new KCM 5.4.0 Tomcat instances. If any customization has been applied to the Tomcat Server configuration in the previous KCM version, you should reapply them to the instances installed for KCM 5.4.0.

If an upgrade encounters an error while upgrading the instances, but the upgraded software has already been installed, the previously active version of KCM is re-activated. After solving any reported problems and uninstalling the failed upgrade, you can retry the upgrade.

## Parameters for the upgrade

The following is a description of the parameters that you need to upgrade the installation.

Parameter	Description
Database!User	<p>The SQL Server user used to access all databases. At the deployment stage, the user must have owner rights to the database.</p> <p>Optional. Specifies the new user that KCM Repository uses to connect to its database.</p> <p><b>Note</b> When upgrading multiple instances, specify a user for each instance using the notation 'XX#Database!User=...', where XX is the instance number.</p>
Database!Password	<p>Optional. Specifies the new password that KCM Repository uses to connect to its database.</p> <p><b>Note</b> When upgrading multiple instances, specify a password for each instance using the notation 'XX#Database!Password=...', where XX is the instance number.</p>
Database!ConnectionString	<p>The parameter is required when upgrading at least one instance. Specifies the new connection string that KCM Repository uses to connect to its database. This must be an ODBC connection string and should not contain a username or password.</p> <p><b>Note</b> When upgrading multiple instances, specify a connection string for each instance using the notation 'XX#Database!ConnectionString=...', where XX is the instance number.</p>

Parameter	Description
OldCMDatabase!User	The parameter is required when upgrading from KCM 5.3 or earlier, unless one of the -NoContractManager predefined setups is used. Specifies the user that the previous KCM version uses to connect to its Contract Manager database.
OldCMDatabase!Password	The parameter is required when upgrading from KCM 5.3 or earlier, unless one of the -NoContractManager predefined setups is used. Specifies the password that the previous KCM version uses to connect to its Contract Manager database.
Services!User	The user account used for the installed services. This user must be a local administrator and must have the "Logon as a service" rights. When specifying a local user, place a period and backslash (.) at the beginning of the parameter. <b>Example</b> .myuser
Services!Password	The password for the services user.
License!File	The parameter is required when upgrading from KCM 5.0 and higher.
Webserver!ServiceUser	The user account used to run the web server services. This user must have the "Logon as a service" rights. When specifying a local user, add a period and backslash (.) at the beginning of the parameter. <b>Example</b> .myuserws
Webserver!ServicePassword	The password belonging to the user provided through Webserver!ServiceUser.
Upgrade!Version	Optional. The parameter is used to specify the version to upgrade from if no active installation can be found. If an active installation can be found, this parameter is ignored.
ContractManager!KeyStoreFile	Optional. The location of the certificate file (.pfx or .p12) for this server. Default connection to KCM is done securely over HTTPS using the SSL protocol. Required if ContractManager!UseSSL is true.
ContractManager!KeystorePassword	Optional. The KeyStore password. This is the password needed for the KeyStore File. Required if ContractManager!UseSSL is true.

Parameter	Description
ContractManager!UseSSL	<p>Optional. Overrides the default behavior on using the SSL protocol. Defaults to true, indicating that any connections to KCM are secure using the SSL protocol and that KCM can be connected over HTTPS.</p> <p>Set the value to false to connect to KCM over HTTP.</p> <p>The <code>ContractManager!UseSSL</code> option configures the connectors for the service Catalina-CM in Tomcat.</p> <p>For security reasons, it disables TLSv1.0, and it only allows ciphers of at least 128 bits.</p> <p>This configuration can be found in the Tomcat <code>server.xml</code> file, which resides in the Tomcat installation folder under <code>instance-CCMRuntime-5.4\conf</code>. For more information, see the Tomcat documentation available on the Internet.</p> <p>If you configured SSL for connections to the Contract Manager manually, that is, you did not deploy KCM with the <code>ContractManager!UseSSL</code> parameter, and you want to remove the HTTP connectors, follow these steps:</p> <ol style="list-style-type: none"> <li>1. Locate the <code>ManageCM.exe.config</code> file that resides in: <code>&lt;deploy root&gt;\KCM\Programs\5.4\Management</code></li> <li>2. Open the file and change the <code>ReloadURLs</code> setting as follows:</li> </ol> <pre>&lt;add key="ReloadURLs" value="https://localhost:&lt;CM SSL port&gt;/ccm/Administration/reloaddatabase;https://localhost:&lt;PartnerAdmin SSL port&gt;/ccm/Administration/reloaddatabase"/&gt;</pre>

For the KCM B&OM upgrade parameters, see [Additional upgrade parameters for KCM B&OM](#) .

## Perform the KCM B&OM upgrade

When upgrading a KCM installation that includes KCM Batch & Output Management, it is important that you note the following limitations and consequences:

- **History of the previous systems not transferred**

The contents of the previous B&OM installation, which includes systems and their objects, are transferred to the new installation without their history, and the transfer history details are lost. This information is still present in the previous installation if you reactivate it. Also, changes to the objects in the new version have no effect on the previous version.

- **Paths must be manually updated**

Paths to files or folders on the file system, such as the path to the Compart license file on the OdinConversion component, are not updated automatically. You need to update them manually.

- **Runtime information remains in use**

The runtime information of the previous KCM B&OM installation, including the runtime databases, existing jobs, processes, envelopes and stacks, as well as custom databases, remain in use in the new version. The transferred systems in the new installation continue to use the same runtime databases as before. If strict database separation is needed, you must manually copy the databases and adjust the connection in KCM Studio.

- **New system also installed**

The upgrade also installs a new system into KCM B&OM, which includes standard processes and objects from the new version. It uses the new runtime database that you created before the upgrade.

- **Multiple content databases no longer supported**

Multiple KCM B&OM content databases are no longer supported. When upgrading from a KCM B&OM installation with multiple content databases, you are prompted to select only one with the `OutputManagement!RepositoryAlias` parameter (see [Additional upgrade parameters for KCM B&OM](#) ). You can transfer any systems stored in different content databases after the upgrade using the `ExportOutputManagementContent` and `ImportOutputManagementContent` management scripts (see [Export KCM Batch & Output Management content](#) and [Import KCM Batch & Output Management content](#), respectively). Any system-transfer links between systems residing in different content databases are broken in the new version, and you need to reapply them manually.

- **Upgrading a B&OM secondary requires a manual copy of configuration files**

When upgrading a setup with `OutputManagement!Secondary=true`, perform the following steps:

1. Stop the B&OM services on the secondary machine.
2. Upgrade the primary machine to which the secondary setup refers.
3. Extract the KCM package on the secondary machine. Next to the extracted `InstallServer` folder, create a new folder "Output Management Primary".
4. From the primary machine, copy the "repository.config" and "UserRepository.config" files from the `Programs` folder of Output Management. Place these files in the folder created in the previous step.
5. Upgrade the secondary machine.

- **Custom service hosts recreated under a new name**

Custom service hosts created in the previous version are recreated in the new version under a different name and run the same processes as before. Services that were running processes from multiple different systems are no longer supported. Such services are not automatically recreated, and you need to split them up manually. Customized configuration, such as log levels or Remote Control, is not transferred, and you need to reapply it manually. To view active services, create or remove them, use the `ListOutputManagementServices`, `AddOutputManagementService`, and `RemoveOutputManagementService` scripts (see [Add custom KCM B&OM service hosts](#)).

- **Removing the B&OM services from an inactive installation requires the uninstallation script**

To remove the B&OM services from an inactive installation, use uninstallation script.

The `ListOutputManagementServices`, `AddOutputManagementService`, and `RemoveOutputManagementService` scripts always affect the active installation. For example, the `RemoveOutputManagementService.ps1` script removes the B&OM services from the active installation. For more information, see [Uninstallation](#) and [Switch to an older installation](#).

- **Customized Odin parameters saved to a new configuration file**

When upgrading from KCM 5.0 or 5.1, customized Odin parameters in the previous version are taken over, but saved to a new configuration file specific to the previous version to prevent compatibility issues between the different version parameters. To modify the Odin parameters configuration for systems transferred from the previous version, use the new configuration file **odinparameters\_<previous version>.config**.

- **Customized configuration for CM Studio not automatically reapplied**

Customized configuration for KCM Studio and the two standard services, Core and Output, is not transferred, and you have to reapply it manually. Also, the system used for assessing the user rights in KCM Studio is now configured to use the newly created system allowing users to immediately access the new objects. The configured user roles and role mappings are transferred as part of the general content transfer, but to continue using them, you have to update them manually to include rights for all new objects in KCM B&OM.

- **Compart configuration files are not automatically transferred to the new version**

Customized DocBridge profiles are transferred to the new version, but other Compart configuration files, such as license files, are not moved.

## Additional upgrade parameters for KCM B&OM

The following is a description of the parameters that you might need to upgrade the KCM B&OM installation.

Parameter	Description
OutputManagement!RepositoryAlias	Optional. Specifies the content database to use to transfer systems and objects from the previous version. Specify the content database as the name of the database alias. Required if multiple content databases are present in the previous version.
OMRepositoryDB!ConnectionString	Required. Specifies the connection string that Batch & Output Management uses to connect to its new repository database. Do not specify username or password in this connection string.  <b>Note</b> Depending on the database, this must be either a valid Oracle or SQL Server connection string. This cannot be an ODBC connection string.
OMRepositoryDB!User	Optional. Specifies the user that Batch & Output Management uses to connect to its new repository database.
OMRepositoryDB!Password	Optional. Specifies the password that Batch & Output Management uses to connect to its new repository database.

Parameter	Description
OMRuntimeDB!ConnectionString	<p>Required. Specifies the connection string that Batch &amp; Output Management uses to connect to its new runtime database. Do not specify username or password in this connection string.</p> <p><b>Note</b> Depending on the database, this must be either a valid Oracle or SQL Server connection string. This cannot be an ODBC connection string.</p>
OMRuntimeDB!User	Optional. Specifies the user that Batch & Output Management uses to connect to its new runtime database.
OMRuntimeDB!Password	Optional. Specifies the password that Batch & Output Management uses to connect to its new runtime database.

## Switch to an older installation

You can switch to an older installation if the new installation does not work properly.

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management`
2. Run the command `\Activate.ps1 -Version <version number>`

After running this command, the older installation is active and the new installation is disabled. To switch back to 5.4.0, run the preceding command with 5.4 as a value for `<version number>`.

## Upgrade a customized setup

A customized setup is a setup to which custom configuration has been added after the installation of KCM. To upgrade a customized setup, perform a standard upgrade (see [Upgrade your installation](#)) and customize the new deployment. After performing the upgrade, a standard setup of KCM 5.4.0 is available that might not contain all custom configuration. In this case, you need to manually deploy the missing custom configuration to the newer version.

You need to deploy some custom configuration of the previous version to the newer version:

- Manually created files and folders within the KCM Work directory.

The Work directory for the new deployment resides in: `<deploy root>\kcm\work\5.4\<instance_number>`.

The Work directory for the previous KCM version is located at: `<deploy root>\kcm\work\<previous_version>\<instance_number>`.

**Note** If your customization refers to these folders using paths that contain the KCM version number, verify that you change the version number when you move these folders.



- Changes made through KCM Core Administrator other than constants and Environments settings. For example, changes to the DP Manager, Logging, Monitor, HTTP Monitor, Job Recovery, or Advanced tabs.
- Extra installed KCM components such as KCM ComposerUI for ASP.NET, KCM Core Web Services interface, and KCM Designer for Windows.

## Upgrade contract types and interfaces not in use by any contract

An upgrade skips contract types and interfaces (or other configuration) that are not in use by active contracts. Only the configuration needed to duplicate the current setup is moved to a newer version. To duplicate unused contract types and interfaces, follow these steps:

1. Start the ManageCM tool.  
For information on the tool, see the chapter "Use the ManageCM tool" in the *Kofax Communications Manager Getting Started Guide*.
2. Export the contract types belonging to the KCM version you are upgrading from.  
Execute the following command.

```
ManageCM /ExportDatabase /CMVersion=5.0 /File=Export.xml  
/ContractTypes
```

This command creates an XML file containing all the contract types and their interfaces.

3. Import the created XML file into the new version of the Contract Manager.  
Execute the following command, substituting the example parameters for your actual parameters.

```
ManageCM /ImportDatabase /File=Export.xml
```

You can use this procedure for other types of configuration, if necessary.

## Upgrade KCM when installed on more than one computer

When KCM is installed on more than one computer, it is not necessary to upgrade all computers in one step. You can upgrade all the computers gradually, taking into consideration the order in which KCM components are upgraded.

## Upgrade the Contract Manager before the instances

To upgrade the Contract Manager prior to upgrading the instances, perform the following steps.

1. Upgrade the Contract Manager.  
To do so, execute the `.\Upgrade` script on the computer on which the Contract Manager is installed. For more information on the script, see [Perform the upgrade](#). The setup still functions after the upgrade.
2. Upgrade the instances by executing the `.\Upgrade` script on the machine on which the instances are running.
3. Start the ManageCM tool and use the `/Update` flag as shown in this example to complete the upgrade.

```
ManageCM /RegisterInstance /Host=host42 /Instance=1 /Update
```

The upgrade is completed. You can now access the new instances.

## Upgrade the instances before the Contract Manager

The following topic describes how to upgrade the instances prior to upgrading the Contract Manager.

1. Upgrade the instances.

Execute the `.\Upgrade` script on the computer on which the instances are running. For more information on the script, see [Perform the upgrade](#).

2. Upgrade the Contract Manager.

Execute the `.\Upgrade` script on the computer on which the Contract Manager is installed. Your setup automatically switches to newer versions of the instances, also registering the instances.

# Install Kofax Communications Manager on Docker

Kofax Communications Manager supports an installation and deployment of KCM software on Docker images and containers. An installation on Docker must always consist of a single Contract Manager on its own container, and one or more instances each on their own container.

To create a KCM installation on Docker for the Contract Manager, perform the following steps:

1. Create a Docker image that contains the prerequisites for the Contract Manager.
2. Create a Docker image that contains the Contract Manager software. At this step, the Contract Manager software is installed, but not yet configured.
3. From this image, start a Docker container for the Contract Manager. At this step, the configuration of the Contract Manager is applied.

To create a Kofax Communications Manager installation on Docker for the instances, perform the following steps:

1. Create a Docker image that contains the prerequisites for a KCM instance.
2. Create a Docker image that contains the software for a single KCM instance. At this step, the instance software is installed, but not yet configured. It is neither connected to a database, nor registered on the Contract Manager.
3. From this image, start one or more Docker containers for the instances. This step applies the configuration of the instance, which includes connecting it to its database and registering it on the Contract Manager.

**Note** KCM deployment on Docker requires technical knowledge of the Docker virtualization technology. For more information, see <https://www.docker.com>.

## Restrictions

Kofax Communications Manager provides almost the same support for KCM functionality on Docker as for installations on regular servers, with the following restrictions:

- Batch & Output Management cannot be installed on a Docker container. Instead, it is possible to install Batch & Output Management on a regular server and connect it to a KCM instance on a Docker container.

**Note** The following interfaces can only be used if Batch & Output Management is installed on the Docker host itself, as they require the Contract Manager to communicate to Batch & Output Management through a shared folder:

- DocumentPack.Distribute.OutputManagement
  - KTA.ComposeAndDistribute.OM.DocumentPack
  - KTA.Distribute.OM.DocumenPack
  - KTA.Distribute.OM.DocumenPackFromSession
- 
- Kofax Communications Manager does not support installation of Microsoft Word on Docker containers. The KCM instances on Docker do not have Word Processor support, and certain KCM/KCM Core scripting features are not available. For example, printing Word documents, running macros, or converting to PDF using Word. For Word to PDF conversion, Rendition will be used automatically.
  - When installed on Docker containers, Kofax Communications Manager only supports Microsoft Word DOCX documents, which is the default XML-based file format. When installed on Docker containers, Kofax Communications Manager does not support legacy DOC documents, which is the Microsoft Word 97-2003 Document file format. This restriction applies to the usage of DOC files, including but not limited to templates, import documents, file conversion, printing, and Batch & Output Management communications.
  - KCM ComposerUI for J2EE can be installed on a KCM instance of a Docker container manually. KCM ComposerUI for ASP.NET is not supported in combination with a Docker KCM installation.

## Create prerequisite images

Prior to installation of Kofax Communications Manager, create two Docker files, one that installs the software prerequisites for the Contract Manager, and another that installs the software prerequisites for the instances. Use these Docker files to build the prerequisite images. These Docker files must be based on a Windows *servercore* image, as the *nanoserver* image is not enough for KCM. Here is an example of a supported Windows image:

```
mcr.microsoft.com/windows/servercore
```

### Install fonts on prerequisite images

When creating a prerequisite image from the Docker file, you may need to additionally install Windows fonts. Some Windows Server Docker images, such as Windows Server 2019, have a very limited set of fonts installed by default. If system fonts are required during PDF processing and these fonts are not

installed in a Docker image, an error occurs and the process fails. For example, during the conversion from DOCX document to PDF.

## Install the Contract Manager

The installation of the Contract Manager on Docker consists of two steps:

### 1. The creation of an image.

Install the software by the Docker specific installer in an inert and not yet configured state:

```
InstallServer\Software\DockerTools\InstallContractManager.ps1
```

### 2. The creation of a container from the image.

At the second step, the configuration is applied to the software, and the software is brought into a running state. These actions are managed by the Docker specific entry point process:

```
InstallServer\Software\DockerTools\DockerStartContractManager.exe
```

These actions are included in Docker files and Docker commands.

## Create the Contract Manager image

To create the Contract Manager image, follow these steps:

1. Unpack the KCM installation package on the Docker host.
2. Copy the template Docker file `InstallServer\Dockerfiles\example-contract-manager` to the extracted `InstallServer` folder and rename it.
3. Find the `FROM` instruction in the Docker file and edit the name of the prerequisite image.
4. Find the `RUN` instruction in the Docker file and edit the parameters of the installation command.
5. Find the `ENTRYPOINT` instruction in the Docker file and edit, add or remove the parameters of the entry point process. This is a partial list of parameters, which are fixed on the image. When a container is started, the remaining parameters will be supplied and appended to this list.

**Note** If a parameter references a file, this file should be accessible from the container when it is started. For example, you can add a `COPY` instruction to the Docker file to embed the file on the image.

6. From the `InstallServer` folder, run the following Docker command: `docker build -t <image-name> -f <docker-file> .` If any error occurs during the "Docker build" command, then the Docker build needs to be started again from the beginning.

## Create the Contract Manager container

To create the Contract Manager container, run the following command:

```
docker run -d <network-parameters> <image-name> <entry-point-parameters>
```

Use the *network-parameters* to ensure that the Contract Manager is externally reachable through its public port, and that the Contract Manager and the instances can internally connect to each other, without

using public connections. The *network-parameters* can be configured in different ways within the customer infrastructure. See the following examples of *network-parameters* configuration:

- Add `-p <n>:443`. This parameter forwards connections to port `n` on the Docker host through port 443 of the Contract Manager container. Replace 443 with 8081 when SSL is disabled.
- Add `--network <name>`. This parameter adds the Contract Manager container to the Docker network name.
- Add `-ip <address>`. This parameter assigns `address` as the internal IP address of the container in the Docker network.

The *entry-point-parameters* are the parameters that passed to the entry point process. They must be placed at the end of the command, after the image name. If a parameter references a file, then this file should be accessible from the container when it is started. For example, add a bind mount or volume to the Docker run command and place the file on it.

**Important** *It is an error if a parameter of the entry point process is both given an embedded value on the image and a variable value on creating the container.*

**Note** The entry point parameters do not require syntax and are passed directly to the entry point. Alternatively, parameters can also be replaced with environment variables, using `--env` parameters on the Docker run command. It is an error if a parameter is specified both as a normal argument and as environment variable.

The entry point will be executed as soon as the container starts. It logs its output to the standard Docker log, which is visible with the command `docker logs <container-name>`. If an error occurs in the entry point process, the created container is erroneous and should be removed. The Docker run command should be run again to create the container anew.

Example commands with all options included:

```
docker run -d -p 443:443 --network nat --ip 123.45.67.89 kcm-contractmanager
ContractManager!KeystorePassword=<password>
```

```
docker run -d -p 443:443 --network nat --ip 123.45.67.89 --env
ContractManager!KeystorePassword=<password> kcm-contractmanager
```

## The InstallContractManager.ps1 installer

The `InstallContractManager.ps1` script has the following parameters.

Parameter	Description
<code>Deploy!RootPath</code>	Optional. Specifies the root path where the KCM installation is placed. Defaults to <code>C:\</code> when omitted.
<code>Webserver!ServiceUser</code>	Required. Specifies the name of the user account for the Tomcat Windows services.
<code>Webserver!ServicePassword</code>	Required. Specifies the password of the user account for the Tomcat Windows services.

## The DockerStartContractManager.exe entry point process

The DockerStartContractManager.exe entry point has the following parameters.

Parameter	Description
LDAP!Mode	Required, boolean value. Specifies whether LDAP should be enabled. If it is enabled, the LDAP!PropertiesFile parameter is mandatory.
LDAP!PropertiesFile	Required only when LDAP!Mode is set to true. Specifies the location of the LDAP properties file to use to configure LDAP on the Contract Manager. For details, see the <i>KCM Getting Started Guide</i> . This file needs to be accessible by the container during the initial run only.
ContractManager!UseSSL	Required, boolean value. Specifies whether SSL should be enabled. If this is enabled, the ContractManager!KeyStoreFile and ContractManager!KeyStorePassword parameters are mandatory.
ContractManager!KeyStoreFile	Required only when ContractManager!UseSSL is set to true. Specifies the location of the certificate store file to use for configuring SSL. Contrary to the LDAP properties file, this file needs to be accessible by the container at all times.
ContractManager!KeyStorePassword	Required only when ContractManager!UseSSL is set to true. Specifies the password for the certificate store file specified in ContractManager!KeyStoreFile.
Configuration!ImportFile	Optional. Specifies the location of the Contract Manager configuration export created by the ExportContractManagerConfiguration tool. This configuration is imported into the container.
Configuration!ImportInstances	Optional. Specifies whether the imported Contract Manager configuration includes instances. Default value is False.

## Install Kofax Communications Manager instances

The installation of KCM instances on Docker includes two steps:

### 1. The creation of an image.

At the first step, the software is installed in an inert and not yet configured state by the Docker installer:

```
InstallServer\Software\DockerTools\AddInstance.exe
```

**Note** At this step, the image is not yet connected to a Repository database.

## 2. The creation of one or more containers from the image.

At the second step, the configuration is applied to the software, and the software is brought into a running state. These actions are performed by the Docker entry point process:

```
InstallServer\Software\DockerTools\DockerStartInstance.exe
```

**Note** At this step, the container is connected to the Repository database. Also, the instance container is registered on the Contract Manager.

These actions are included in the Docker files and Docker commands.

## Create an instance image

To create an instance image, perform the following steps:

1. Unpack the KCM installation package on the Docker host.
2. Copy the template Docker file `InstallServer\Dockerfiles\example-instance` to the extracted `InstallServer` folder and rename it.
3. Find the "FROM" instruction in the Docker file and edit the name of the prerequisite image.
4. Find the "RUN" instruction in the Docker file and edit the parameters of the installation command.
5. Find the "ENTRYPOINT" instruction in the Docker file and edit, add, or remove the parameters of the entry point process. This is a partial list of parameters, which are fixed on the image. The remaining parameters will be supplied and appended to this list when a container is started.

**Note** If a parameter references a file, then this file should be accessible from the container when it is started. For example, you can add a "COPY" instruction to the Docker file to embed the file on the image.

6. From the `InstallServer` folder, run the docker command `docker build -t <image-name> -f <docker-file>`. If any error occurs during the Docker build command, then the Docker build needs to be re-started from the beginning.

## Create an instance container

To create an instance container, ensure that the Contract Manager container is running for them. Perform the following steps:

1. Run the command `docker run -d <network-parameters> <image-name> <entry-point-parameters>`.

The *network-parameters* ensure that the Contract Manager and the instances can connect to each other via internal, not public, connections. This can be configured in different ways within the infrastructure of the customer. See the following examples:

- Add `--network <name>`. This parameter adds the instance container to the Docker network *name*.
- Add `-ip <address>`. This parameter assigns *address* as the internal IP address of the container in the Docker network.
- Add `-p <n>:2587` to use the KCM Designer for Windows on client machines. This parameter forwards connections to port *n* on the Docker host through port 2587 of the instance container. When installing the KCM Designer for Windows, fill in the Docker host for the KCM Repository host and port *n* for the KCM Repository port.



- When the *entry-point-parameters* are passed to the entry point process, place these parameters at the end of the command, after the image name. If a parameter references a file, then this file should be accessible from the container when it is started. For example, you can add a bind mount or volume to the Docker run command and place the file on it.

**Important** *It is an error if a parameter of the entry point process is both given an embedded value on the image and a variable value on creating the container.*

**Note** The entry point parameters do not require syntax and will be passed directly to the entry point. Alternatively, parameters can also be replaced with environment variables, using the `--env` parameters on the Docker run command. It is an error if a parameter is specified both as a normal argument and as an environment variable.

- The entry point will be executed as soon as the container starts. It logs its output to standard Docker log, which is visible with the command `Docker logs <container-name>`. If an error occurs in the entry point process, the created container is erroneous and should be removed. The Docker run command should be run again to create the container anew.

**Tip** The entry point process connects to the KCM Repository database and modifies it. If the entry point process fails after this point, it can still be run again. In this case, you need to change the `Database!Mode` parameter to "Check".

**Note** The entry point process registers the instance on the Contract Manager. If the entry point process fails after this point, it can still be run again. The second instance overwrites the registration of the first one.

See the following examples of commands with all options included:

- `docker run -d --network nat --name instance01 --ip 123.45.67.98 instance Database!Mode=New Database!ConnectionString="Driver=SQL Server;Server=123.45.67.88;Database=DB;" Database!User=ccm Database!Password=<pwd> Contract!Partner=CCM Contract!Customer=Docker01 DesignerAdministratorAccount!Password=0fec6d3d54b8df23067a7f7e79db8b0f ContractManager!Host=123.45.67.89 ExampleWebApp!Name=start01`
- `docker run -d --network nat --name instance01 --ip 123.45.67.98 --env Database!Mode=New --env Database!ConnectionString="Driver=SQL Server;Server=123.45.67.88;Database=DB;" --env Database!User=ccm --env Database!Password=<pwd> --env Contract!Partner=CCM --env Contract!Customer=Docker01 --env DesignerAdministratorAccount!Password=0fec6d3d54b8df23067a7f7e79db8b0f --env ContractManager!Host=123.45.67.89 --env ExampleWebApp!Name=start01 instance`

## The AddInstance.exe installer

The `AddInstance.exe` installer has the following parameters.

Parameter	Description
Deploy!RootPath	Optional. Specifies the root path where the KCM installation is placed. Defaults to C:\ when omitted.
Webserver!ServiceUser	Required. Specifies the name of the user account for the Tomcat Windows services.
Webserver!ServicePassword	Required. Specifies the password of the user account for the Tomcat Windows services.
Services!User	Required. Specifies the name of the user account for the KCM Windows services.
Services!Password	Required. Specifies the password of the user account for the KCM Windows services.
Core!NumberOfDPs	Optional. Specifies the number of document processors to create for the instance, between 1 and 99. Defaults to 4 when omitted.

## The DockerStartInstance.exe entry point process

The `DockerStartInstance.exe` entry point has the following parameters.

Parameter	Description
License!File	Specifies the location of the KCM license file. Required, but set automatically in the example. May be adjusted when using Docker volumes or something similar to access the license. This location must be accessible from the container when it is first started.
Database!Mode	Required. Describes how to treat the KCM Repository database. Possible values: <ul style="list-style-type: none"> <li>• New. Sets up the database for the first time use. Expects an empty database, and creates the default Administrator user for KCM Designer, as well as import an example project.</li> <li>• Check. Uses an existing database and verifies that it matches the contract. Sets the database to use the new instance instead.</li> <li>• Overwrite. Works with an existing database used by a different contract. Sets the database to use the new contract and the new instance instead.</li> </ul>
Database!ConnectionString	Required. Specifies the connection string to use for connecting to the KCM Repository database. This must be an ODBC connection string, and should not contain a username or password.
Database!User	Optional. Specifies the username to use for connecting to the KCM Repository database.
Database!Password	Optional. Specifies the password to use for connecting to the KCM Repository database.

Parameter	Description
<code>Services!User</code>	Required. Specifies the user to use for the KCM services. Must include the domain, and the user must have the rights to logon as a service.
<code>Services!Password</code>	Required. Specifies the password for the user defined in <code>Services!User</code> .
<code>Contract!Partner</code>	Required. Specifies the partner of the contract which uses this instance.
<code>Contract!Customer</code>	Required. Specify the customer of the contract which uses this instance.
<code>Contract!Mode</code>	Optional, default is <code>New</code> . Can be either <code>New</code> or <code>Overwrite</code> . Specifies how the instance should be registered to the remote Contract Manager. If <code>New</code> , it only creates a new contract and expects no existing contract to be present. If <code>Overwrite</code> , it expects an existing contract and configures itself to be a new active instance for the contract.
<code>Instance!Host</code>	Optional, defaults is the host name of the container running the instance. This is the name that the Contract Manager uses to access the instance, which should be visible to it from the network. Depending on a network configuration, it is possible to enter an IP address here.
<code>ExampleWebApp!Name</code>	Optional, defaults is none. If given, an example web application with this name is added to the remote Contract Manager upon registration. This name may not yet exist.  <b>Note</b> An example web application is tied to a contract. Usually, this parameter only needs to be specified once per contract.
<code>DesignerAdministratorAccount!Password</code>	Optional. Specifies the password to use for the standard Administrator account in KCM Designer. Only specified if <code>Database!Mode</code> is set to <code>New</code> . If this is not set, a password is generated and written to the standard streams visible in Docker logs. Change the generated password as soon as possible.
<code>ContractManager!Host</code>	Required. Specifies the host of the Contract Manager, which is used to automatically register the instance. This host must be reachable through the network from the container.
<code>Configuration!ImportFile</code>	Optional. Specifies the location of an instance configuration export created by the <code>ExportInstanceConfiguration</code> tool. This configuration is imported into the container.

## Chapter 5

# Create Docker containers with existing configuration

You can use the configuration of an existing KCM installation to initialize a new KCM Docker container. To do so, perform the following steps:

1. Export the configuration of an existing KCM installation into a single file.
2. Refer to this file from the `Configuration!ImportFile` parameter of the Docker run command created with the new Docker container.

This configuration transfer is supported for the Contract Manager and KCM instances starting from KCM version 5.2.0 and later. The support is provided both for regular and Docker KCM installations.

## Export KCM instance configuration

To transfer a KCM instance configuration to a new Docker container, use the `ExportInstanceConfiguration` tool. It can be executed manually on an existing instance container or on a KCM instance installed with the package. This tool resides in:

```
<deploy root>:\KCM\Programs\<version>\Management\ConfigurationTools
```

In a Docker container, this location is part of the PATH environment variable and can be called directly during the Docker execution. The transferred configuration consists of the following content:

- Core configuration, including networking, security, logging, services, DIDs, watermarks, custom or customized scripts, environment variables and constants, environments themselves, and custom-built script libraries. It also includes the contents of the newly added Resources subfolder of the KCM Core work folder. License and Windows user information is excluded.

**Note** When an instance configuration is imported into a Docker container on the initial run, any pre-configured network remote access restrictions are cleared, and no access restriction is configured instead.

- Repository configuration, including networking, security, and logging. The password blacklist is also included.

- Any customized logging configuration.

**Note** For KCM Designer, the syntax of this configuration is incompatible between versions 5.2.0 and 5.3.0, so an export made for a version earlier 5.3.0 does not transfer the KCM Designer logging configuration to 5.3.0 or later.

For ComposerUI for HTML5, the syntax of this configuration is incompatible between versions 5.3.0 and 5.4.0, so an export made of a version earlier 5.4.0 does not transfer the KCM ComposerUI logging information to 5.4.0 or later.

To create an export for an older KCM version that does not include this tool, use an extracted KCM 5.4.0 package. In this case, the tools reside in:

```
InstallServer/Software/DockerTools
```

Use the `Export!Version` parameter to specify the version to export.

The following table lists the parameters to perform the export.

Parameter	Description
<code>Export!File</code>	Required. Specifies the file to save the exported configuration to.
<code>Export!Version</code>	Optional. Specifies the version of KCM to create the export for.
<code>Instance!Number</code>	Optional. Specifies a particular instance to export the configuration for. This parameter is mandatory if more than one instance is available.

## Export the Contract Manager configuration

Use the `ExportContractManagerConfiguration` tool to transfer data and configuration of the Contract Manager to a new Docker container. This tool can be executed manually on an existing Contract Manager container or KCM instance installed with the package. The tool resides in:

```
<deploy root>:\KCM\Programs\<version>\Management\ConfigurationTools
```

In a Docker container, this location is a part of the PATH environment variable, and can be called directly during the Docker execution. The transfer configuration includes the following components:

- All contracts, contract types, interfaces, and authentication settings
- Optionally, all instances
- Any customized logging configuration

**Note** The syntax of this configuration is incompatible between versions 5.3.0 and 5.4.0, so an export made of a version earlier 5.4.0 does not transfer its logging configuration to the 5.4.0 version or later.

To create an export for an older KCM version that does not include this tool, you can use an extracted KCM 5.4.0 package. In this case, the tools reside in:

InstallServer/Software/DockerTools

Use the `Export!Version` parameter to specify the version to export.

The following table lists the parameters needed to run the tool.

Parameter	Description
<code>Export!File</code>	Required. Specifies the file to save the exported configuration to.
<code>Export!Version</code>	Optional. Specifies the KCM version to create the export for. If the version is earlier 5.4.0, the <code>Database!</code> parameters are also required.
<code>Export!Instances</code>	Optional. Default is <code>False</code> . Specifies whether to include instances in the export, which comprises both the instance definition and the association with a contract.
<code>Database!User</code>	Required if the version is earlier 5.4.0 and if the <code>Export!Version</code> parameter is given. Specifies the user name for the Contract Manager database.
<code>Database!Password</code>	Required if the version is earlier 5.4.0 and if the <code>Export!Version</code> parameter is given. Specifies the password for the Contract Manager database.

# Post-installation steps

## Test the installation

When the KCM package is installed, an example web application is available to test the installation and to reference KCM ComposerUI integration resources. The web application that gives you the ability to test the installation is available when the `ExampleWebApp!Install` deployment parameter is set to true. For production and acceptance systems, this web application is not required, and you should not install it. To test the installation, use the following web pages:

- `https://<kcm server>:443/start/home.html`
- `http://<kcm server>:8081/start/home.html` (when `ContractManager!UseSSL` is set to false)

**Important** Default KCM Contract Manager's Tomcat instances (TCP port 443) are configured to use HTTPS. You should not configure any other Tomcat instances to use HTTPS. Also, configure Tomcat instances to suppress extended error messages.

You can change the default port value 443 to the appropriate port value to avoid possible conflicts in case other third-party software on your machine is using the same ports.

If you want to use alternate ports, ensure that you first consult with your administrator.

The default user is "Administrator". The default password is entered through the `DesignerAdministratorAccount!Password` parameter during installation. After installation you can change the password through KCM Designer, see the section "Change your password" in the *KCM Designer online Help*.

To test an interactive composition of a document, under **ComposerUI HTML5**, click **Test**. Afterwards, fill in the forms and a document is opened by the end of the run. The first time you do this test run, you are prompted to install an ActiveX component. This component is used to open Microsoft Word when the composition run is completed.

KCM additional instances install their own test web page based on the number of the instance. For example, instance 5 receives the following web page: `https://<kcm server>:443/start5/home.html`

To remove the test web application, see the next section.

## Remove the test web application

We recommend that you remove the test web application in production environments. To do so, perform the following steps:

1. Navigate to `<deploy root>\KCM\Programs\5.4\Management\ConfigurationTools` and run the `RemoveExampleWebApp.exe` tool.
2. Use the `Application!Name` parameter to specify the name of the test web application.

## Access product documentation

You can access the online KCM documentation using the link (see [Related documentation](#)) or directly from the product. When you click the help button in KCM Designer, KCM Designer for Windows, KCM Core, or KCM Studio, online documentation appears in a new browser window.

To access the online documentation, you must have an active Internet connection. If the security policy for your organization restricts Internet access or the Internet connection is not stable, you can access the documentation in offline mode while using the product.

### Offline documentation

To make the documentation available for use in offline mode, obtain the documentation files from the product package that you download from the [Kofax Fulfillment Site](#).

The product package includes the following documentation files for offline use:

- `KofaxCommunicationsManagerDocumentation_5.4.0_EN.zip`: Contains the entire KCM documentation set in English. This file is required for all users working in offline mode.
- Zip files that contain KCM Designer documentation translated to German, Spanish, French, Italian, Japanese, Dutch and Brazilian Portuguese. For example, the Spanish zip file name is `KofaxCommunicationsManagerDocumentation_5.4.0_ES.zip`.

**Note** The KCM Designer online Help is the only item in the KCM documentation set that is translated to multiple languages at this time.

After you install the KCM product, follow the procedures below to extract the documentation .zip files for each KCM component as required. When you finish extracting the files, the documentation folder in each location will contain one or both of the following folders:

- `print` with the PDF documentation
- `help` with the KCM Designer Help

You can access the documentation in offline mode from these locations. Also, when you click a help button in a KCM component and select the required document, the system will open an offline, local copy of that document.



**Note** The names of the folders where you extract the documentation .zip files are hard-coded and case-sensitive. Do not rename the folders.

### Offline documentation for KCM Designer

1. Extract the contents of the help folders from English documentation .zip file to:

```
<deploy root>\<Tomcat installation folder>\Tomcat <version>\instance-WebDesigner-5.4\webapps\ccm_designer_<instance number>\doc\en
```

2. If you require the Help in another language, extract the contents of the applicable documentation .zip file to the language folder, which resides here:

```
<deploy root>\<Tomcat installation folder>\Tomcat <version>\instance-WebDesigner-5.4\webapps\ccm_designer_<instance number>\doc\<language>
```

Supported languages:

- de: German
- es: Spanish
- fr: French
- it: Italian
- ja: Japanese
- nl: Dutch
- pt\_BR: Brazilian Portuguese

For example, to use the KCM Designer Help in Spanish, extract the contents of the Spanish documentation .zip file to the **es** folder.

### Offline documentation for KCM Studio and KCM Core Administrator

For KCM Studio and KCM Core Administrator, extract the print folder contents of the English documentation .zip file to:

```
<deploy root>\KCM\Documentation\5.4\en
```

### Offline documentation for KCM Designer for Windows

For the KCM Designer for Windows component, which you install separately after the KCM product is installed, extract the contents of the English documentation .zip file to:

```
<deploy root>\KCM Designer Instance <instance number>\Documentation\en
```

For more information on the KCM Designer for Windows installation and default location, see [Install KCM Designer for Windows](#).

## Add extra instances after installation

If additional KCM instances are required after the software has been installed, you can add them using the `AddInstance.ps1` script that resides in: `<deploy root>\KCM\Programs\<version>\Management`. Instances added with this script use the same configuration as used by the original installation.

If the original installation deployed the standard KCM contract (with partner KCM and customer local), the additional instances are also deployed with the standard KCM contract. These instances will have partner KCM and customer local<n>, where <n> is the number of the instance. If the original installation did not deploy the standard KCM contract, the additional instances will not have the standard KCM contract deployed.

**Note** Adding an instance to an active KCM installation restarts the shared services in use by the other instances, making them temporarily unavailable, until they finish restarting.

Check the required parameters prior to adding extra instances (see [Extra instances parameters](#)).

1. To add an instance, navigate to: <deploy root>\KCM\Programs\<version>\Management  
A license file and a database matching the prerequisites for the base KCM installation must be present.

**Note** When adding an instance to an active KCM installation, ensure that no users are connected to the other instances as the process triggers restart of these instances.

2. Instances are represented by a number, for example, Instance\_01. Determine a number to represent the new instance.
3. Execute the script.

The following is an example command.

```
.\AddInstance.ps1
'Database!ConnectionString=Driver=SQL Server;Server=KCM
Server;Database=AdditionalRepository'
Database!User=itpuser Database!Password=pwd123
Services!User=. \myuser Services!Password=pwd123
'License!File=c:\{path}\My License\License.xml'
Webserver!ServiceUser=. \myuserws Webserver!ServicePassword=pwd789
Instance!Number=5
DesignerAdministratorAccount!Password="password"
```

## Extra instances parameters

This table contains a description of the parameters required to add extra instances after the installation.

Parameter	Description
Database!User	Optional. The database user that the added KCM Repository uses to connect to its database. In case of the SQL Server database, this user must have owner rights to the database.
Database!Password	Optional. The database user that the added KCM Repository uses to connect to its database.
Database!ConnectionString	Required. Specifies the connection string that the added KCM Repository uses to connect to its database. Do not specify username or password in this connection string.  <b>Note</b> This must be an ODBC connection string.

Parameter	Description
<code>Services!User</code>	The user account used for the installed services. This user must be a local administrator and must have the "Logon as a service" rights. When specifying a local user, place a period and backslash (.) at the beginning of the parameter. <b>Example</b> <code>.myuser</code>
<code>Services!Password</code>	The password for the services user.
<code>License!File</code>	The location of the .xml file with the license information.
<code>Webserver!ServiceUser</code>	The user account used to run the web server services. This user must have the "Logon as a service" rights. When specifying a local user, add a period and backslash (.) at the beginning of the parameter. <b>Example</b> <code>.myuserws</code>
<code>Webserver!ServicePassword</code>	The password belonging to the user provided through <code>Webserver!ServiceUser</code> .
<code>DesignerAdministratorAccount!Password</code>	The password of the designer Administrator account that is created automatically for the new instance. The password must be at least 20 characters long, and may only consist of letters (a-z, A-Z), numbers (0-9), and the "-" and "_" characters.  This parameter is required only for installations with a default setup (no <code>Deploy!PredefinedSetup</code> ) and when <code>Contract!CreateContract</code> is not false.  The Administrator account is also created when running in LDAP mode. However, the created account can only be used when LDAP mode is switched off.
<code>Instance!Number</code>	Specifies the number that the instance uses. For example, <code>Instance!Number=5</code> . The maximum value for the parameter is 99. This applies to the <code>AddInstance</code> and the <code>RemoveInstance</code> script specifications.
<code>Instance!NumberOfDPs</code>	Optional. Specifies how many Document Processors should be deployed for KCM Core. The default value is 4 or the number of Document Processors allowed by your license.
<code>Contract!CreateContract</code>	Optional. The parameter is used to indicate whether an additional instance gets the default contract KCM Local<n>. The value can be either true or false.
<code>ExampleWebApp!Install</code>	Optional. If set to true, an example web application is installed to test the installation and to reference KCM ComposerUI integration resources. If omitted, default value is false.

## Add extra instances after installation in combination with Oracle

1. To add extra instances after you installed KCM with the `Database!Type` set to Oracle, create additional users matching the instance numbers you want to create.
2. Execute the command `AddInstance`.  
For more information on adding extra instances, see [Add extra instances after installation](#). For a description of the command `AddInstance`, see the *Kofax Communications Manager Getting Started Guide*.

## Remove specific instances

If a specific KCM instance is no longer needed, you can remove it using the `RemoveInstance.ps1` script that resides in: `<deploy root>\KCM\Programs\<version>\Management`. Removing an instance clears files, services, and settings specific to this instance. The content database is unaffected.

**Note** Removing an instance from an active KCM installation restarts the shared services also in use by other instances, temporarily making them unavailable.

Check the list of the required parameters prior to removing specific instances (see [Parameter to remove specific instances](#)).

1. To remove an instance, navigate to: `<deploy root>\KCM\Programs\<version>\Management`

**Note** When removing an instance from an active KCM installation, ensure that no users are connected to the other instances as the process triggers restart of these instances.

2. Execute the script.

The following command removes the instance with the number 2. It also removes the contract registered for this instance from the Contract Manager.

```
.\RemoveInstance.ps1 Instance!Number=2
```

### Parameter to remove specific instances

This table contains a description of the parameter required to remove specific instances of KCM.

Parameter	Description
<code>Instance!Number</code>	Specifies the number of the instance to remove.

## Add custom KCM B&OM service hosts

To run KCM B&OM processes without using KCM Studio, you can add custom KCM B&OM service hosts.

Log files for KCM Studio and all KCM Batch & Output Management services, including custom service hosts, are written to the Output Management work folder, which resides in: `<deploy root>\KCM\Work\<version>\Output Management\Logs`.

For more information on KCM B&OM and the processes, see the *Kofax Communications Manager B&OM Getting Started Guide*.

## Add custom service hosts

You can add custom KCM B&OM service hosts using the script `AddOutputManagementService.ps1`. New services added using this script execute selected processes in a given system and can be independently managed using Service management tools. They depend on the standard KCM B&OM Core service to run.

Note the following when adding custom service hosts:

- A new service is automatically started and does not require a restart of B&OM or other components of KCM.
- A new executable is created in the `KCM Output Management programs` directory for the new service with the following name: **CCM\_OM\_SVC\_<Identifier>.exe**. A `.config` file is created next to it, which you can customize, if required.
- By default, service events for the new service are logged to the **CCM\_OM\_<Identifier>\_<Version>.wcf.svclog** file, and all other messages are logged to the **CCM\_OM\_<Identifier>\_<Version>.log.txt** file. If starting the service fails, the log files contain information about the failure.

The following table lists the parameters that you need to add custom service hosts.

Parameter	Description
<code>Service!Identifier</code>	Identifier for the new service. The value is used for identifying the service within the KCM B&OM installation. Also, the value is part of the name, default display name, and default description of the Windows service.
<code>Service!DisplayName</code>	Optional. Overrides the default generated display name of the new Windows service. Default is <code>CCM Output Management - &lt;Identifier&gt; - &lt;Version&gt;</code> .
<code>Service!Description</code>	Optional. Overrides the description of the new Windows service generated by default. Default is the same value as the <code>Service!DisplayName</code> parameter has.
<code>Service!User</code>	The user account to be used for the new service. This must be the same user as specified in the original KCM deployment, and must consequently have the "Logon as a service" rights. When specifying a local user, place a period and backslash ( <code>.\</code> ) at the beginning of the parameter value.
<code>Service!Password</code>	The password for the service user.
<code>System!Name</code>	The name of the KCM B&OM system for the new service to use. You can view the names in the System Administration section in KCM Studio.

Parameter	Description
Process!Names	<p>Optional. A colon-separated list of names for KCM B&amp;OM processes for the new service to execute. You can view the names in the Object Viewer pane in KCM Studio.</p> <p><b>Note</b> The separator is the regular colon character (:), not the semicolon character (;). If this parameter is not given, AliasTable!Names must be specified.</p>
AliasTable!Names	<p>Optional. A colon-separated list of names for KCM B&amp;OM alias tables, which should contain one or more KCM B&amp;OM processes.</p> <p>The service executes each process contained in these alias tables. The names of the keys defined in the alias tables are not taken into account. You can view the names of the alias tables in the Object Viewer pane in KCM Studio.</p> <p><b>Note</b> The separator is the regular colon character (:), not the semicolon character (;). If this parameter is not given, Process!Names must be specified.</p>

1. Navigate to: <deploy root>\KCM\Programs\<version>\Management
2. Services are managed using an identifier. Select an identifier to represent the new service.

**Example** Registration

3. Execute the script.

Use the following script substituting the example parameter values with your actual values.

```
.\AddOutputManagementService.ps1 Service!Identifier=Registration
Service!DisplayName=MyService "Service!Description=Runs my
processes" Service!User=.myuser Service!Password=pwd123
System!Name=MySystem Process!Names=MyProcess:MySecondProcess
AliasTable!Names=MyTable:MySecondTable
```

## List custom service hosts

You can get an up-to-date list of custom KCM B&OM service hosts in the current KCM installation using the script ListOutputManagementServices.ps1.

This list contains the identifiers, names, display names, and current status of each service. Also, it includes information on the custom services that failed to be added, so you can remove and re-add them.

This script has no parameters.

1. Navigate to: <deploy root>\KCM\Programs\<version>\Management
2. Execute the script.

## Remove custom service hosts

You can remove custom KCM B&OM service hosts using the script `RemoveOutputManagementService.ps1`.

### Note

- The removed service is automatically stopped, and the removal does not require a restart of B&OM or other parts of KCM.
- Some custom configuration applied to the service executable `.config`, such as log levels, are lost after the service is removed. Log files themselves are kept.

The script has one parameter.

Parameter	Description
<code>Service!Identifier</code>	Identifier of the service to remove. A list of services can be retrieved using the script <code>ListOutputManagementServices.ps1</code> (see the previous section).

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management`
2. Services are managed using an identifier. Specify the identifier of the service to remove.

**Example** `Registration`

3. Execute the script.

Use the following script substituting the example parameter values with your actual values.

```
.\RemoveOutputManagementService.ps1
Service!Identifier=Registration
```

## Required step to run KCM B&OM processes

To run KCM B&OM processes, you need to apply the Microsoft 2954953 fix, which is available from the Microsoft support web page: <https://support.microsoft.com>.

On the website, search for the Knowledge Base article "2954953" to find "Some APIs do not work when they are called in services in Windows," and follow the instructions.

## Export and import KCM Batch & Output Management content

This section provides instructions on using scripts to export and import KCM Batch & Output Management content.

## Export KCM Batch & Output Management content

To facilitate one-time transfer between different installations of KCM B&OM and the upgrade to a newer KCM B&OM version, you can use the `ExportOutputManagementContent.ps1` script. The exported content consists of systems and all their objects.

This script is not intended to provide an alternative to the regular Transfer Jobs option accessible from KCM Studio. Optionally, you can filter the export to only include a specific system, or to only export systems from a specific content database.

Note the following when using the script:

- It does not require KCM B&OM to be running at the time of the export.
- It can be used to export content from KCM B&OM 5.0 or 5.1 installed without any other KCM components. In this case, you can execute the script from an extracted package directory, without performing the deployment.

The following table lists the parameters that you need to perform the export.

Parameter	Description
<code>Destination!Path</code>	The path to a folder to store the export. It must be empty if it already exists. If it does not exist, it is automatically created.
<code>System!Name</code>	Optional. The name of a specific system to export. If this parameter is specified, the export will only contain this system. You cannot use it with the <code>Legacy!Export</code> parameter. By default, no filter is applied.
<code>RepositoryAlias!Name</code>	Optional. The name of a specific content database alias to export. If this parameter is specified, the export will only contain systems from this content database. Only useful when exporting from a KCM B&OM 5.0 or 5.1 installation. By default, no filter is applied.
<code>Legacy!Export</code>	Optional. Boolean. Specifies whether to create the export from KCM B&OM 5.0 or 5.1 on the machine. If specified, an installation is located, and the export is created. No active KCM package is required in this case. Default is false.

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management`  
Or, when exporting from a standalone KCM B&OM 5.0 or 5.1 installation, navigate to the extracted KCM package directory.
2. Select a location to save the export. This location must either not yet exist, or be empty.
3. Execute the script.  
Use the following script substituting the example parameters values with your actual values.

```
.\ExportOutputManagementContent.ps1 Destination!Path=C:\Export
System!Name=MySystem
```



## Import KCM Batch & Output Management content

To import the KCM Batch & Output Management content, use the `ImportOutputManagementContent.ps1` script. You can import the following content:

- A full-systems export created with the `ExportOutputManagementContent.ps1` script
- An export to a disk created with the Transfer Job option in KCM Studio

This script does not require KCM B&OM to be running at the time of the import.

The following table lists the parameters that you need to perform the import.

Parameter	Description
<code>Source!Path</code>	The path to a folder containing an export.
<code>System!Name</code>	Optional. Specifies the name of the system to contain a Transfer Job export. Required if the source directory contains a Transfer Job export. By default, the directory is assumed to have a full-systems export retrieved with the <code>ExportOutputManagementContent.ps1</code> script.

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management`
2. Execute the script.  
Use the following script substituting the example parameters values with your actual values.

```
.\ImportOutputManagementContent.ps1 Source!Path=C:\Import
```

## Move the KCM Repository database

In some cases, we recommend that you configure a running KCM instance to use a different database for its KCM Repository. The `ChangeInstanceDatabase.exe` tool is designed to automatically change the configuration, set up the database if necessary, and connect it to a KCM instance.

For a typical KCM installation, the `ChangeInstanceDatabase.exe` tool resides in:

```
<deploy root>:\KCM\Programs\<Version>\Management\ConfigurationTools
```

Run the `ChangeInstanceDatabase.exe` tool to change the configuration of the KCM instance, connect to the new database, and upgrade it if possible. Running the tool also restarts the KCM Repository service automatically. The configuration is backed up before it is changed, and if an error occurs, this backup is restored. In case of a restoration failure, the backup file is still present in the configuration directory of the KCM Repository instance.

Parameters of the tool are displayed in the `Section!Key=Value` format. Long values may be surrounded by quotes.

### Example

```
Section!Key="Value"
```

The following table lists the `ChangeInstanceDatabase.exe` tool parameters to configure a KCM instance.

Parameter	Description
<code>Database!Mode</code>	Required. Specifies the type of the newly indicated KCM Repository database. Possible values: <ul style="list-style-type: none"> <li>• <b>New.</b> Sets up the database for the first usage. Expects an empty database and creates the default Administrator user for KCM Designer as well as imports an example project.</li> <li>• <b>Existing.</b> Expects an existing KCM Repository database and connects it to the current instance.</li> </ul>
<code>Database!ConnectionString</code>	Required. Connection string to use for connecting to the KCM Repository database. This must be an ODBC connection string without a user name or password.
<code>Database!User</code>	Optional. User name to use for connecting to the KCM Repository database.
<code>Database!Password</code>	Optional. Password to use for connecting to the KCM Repository database.
<code>Services!User</code>	Required. Specifies the user currently running the KCM services. Must include the domain.
<code>Services!Password</code>	Required. Password for the user specified in <code>Services!User</code> .
<code>DesignerAdministratorAccount!Password</code>	Optional. Specifies the password to use for the standard Administrator account in KCM Designer. Only specified if <code>Database!Mode</code> is set to <code>new</code> . If this parameter is not set, a password is generated and written to the standard streams visible in an output console. We recommend that you change the generated password as soon as possible.
<code>Instance!Number</code>	Required if more than one instance is present; otherwise, it is optional. Specifies a particular instance to configure the new database for. Must be a number from 1 to 99, matching an existing KCM instance.

## Chapter 7

# Other settings

## Enable authentication and authorization for Contract Manager

You can enable authentication and authorization for the Contract Manager through the following PowerShell script:

```
.\SetCMAuthentication.ps1 ContractManager!UseAuthentication=True
```

The `SetCMAuthentication.ps1` script resides in: `<deploy root>\KCM\Programs\5.4\Management`

Authentication can be disabled again by calling this script with `ContractManager!UseAuthentication=False` as its argument.

After enabling authentication and authorization for the Contract Manager, it only grants access applications to authorized SOAP calls after they are properly authenticated. For information on how to configure authentication and authorization for a particular application, see the section "Manage calling applications and their access to Contract Manager interfaces" in the *Kofax Communications Manager Getting Started Guide*.

**Note** This mechanism uses HTTP basic authentication. This means that you should only use authentication and authorization in combination with an encrypted connection. For more details, see information on the `ContractManager!UseSSL` parameter in [Deployment parameters](#).

## Enable SSL for connections to the Contract Manager

We advise that you enable SSL for connections to the Contract Manager. To do so, configure the connectors for the service Catalina-CM.

You can find the configuration for these connectors in the Tomcat `server.xml` file, which resides in the Tomcat installation folder under `instance-CCMRuntime-5.4\conf`. For more information, see the Tomcat documentation available on the Internet.

You should at least specify the following:

```
<Service name="...">
  <Connector
    port={HTTPS Port for the Contract Manager web service}
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    SSLEnabled="true"
    maxThreads="150"
    scheme="https"
    secure="true"
```

```

    clientAuth="false"
    sslEnabledProtocols="TLSv1.1,TLSv1.2"
    keystoreFile={The path to a key store file}
    keystorePass={The password for the key store file}
    keystoreType="PKCS12"
    connectionTimeout="20000"
    useServerCipherSuitesOrder="true"
    ciphers="TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, ...
            TLS_EMPTY_RENEGOTIATION_INFO_SCSV"
  />
  ...
</Service>

```

The `KeyStoreFile` parameter specifies the location of a certificate file (.pfx or .p12). The `KeyStore` password is the password needed for the `KeyStore` File.

The `maxTreads` and `connectionTimeout` values are given as examples.

For security reasons, do not include `TLSv1.0` in the `sslEnabledProtocols` and avoid weak ciphers. Only list ciphers of at least 128 bits. Examples are shown here.

```

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
TLS_DHE_DSS_WITH_AES_256_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA,
TLS_ECDH_ECDSA_WITH_RC4_128_SHA,
TLS_ECDH_RSA_WITH_RC4_128_SHA,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256,
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,
TLS_EMPTY_RENEGOTIATION_INFO_SCSV

```

After changing this configuration, restart the "Apache Tomcat CCMRuntime instance 5.4" Windows service.

## Change an account for the WebServer services

The package installer configures the Tomcat services to run under a specified user account. After installation, you can modify the used account with the `ChangeWebserverServiceUser` script, as shown here:

```
.\ChangeWebserverServiceUser.ps1 Webserver!ServiceUser=. \myuser  
Webserver!ServicePassword=myPassword
```

The `ChangeWebserverServiceUser.ps1` script resides in: `<deploy root>\KCM\Programs\5.4\Management`

Verify that the specified account has Logon as a Service rights.

## Configure request rate limitation

If it is required to shield the server on which KCM is installed from (D)DOS attacks, you should consider enabling a request rate limiting policy. KCM does not provide a mechanism for such a policy, so this should be handled by setting up a firewall or proxy server that is capable of request rate limiting between the KCM installation and the outside network.

All communication with the KCM installation is either HTTP or HTTPS.

## Install and configure KCM Toolbox for Word

KCM Toolbox for Word is an add-in to Microsoft Word that gives you the ability to easily add Fields to Rich Text Blocks and Quick Templates in KCM Designer. The functionality is supported on Microsoft Word 2013 and later, and only for the DOCX documents.

### Requirements

- KCM Toolbox for Word requires a web server.
  - All users running KCM Toolbox for Word must have an internet connection.
  - The `manifest.xml` file for the Toolbox must be stored in a shared location accessible to all users and listed in the Trusted App Catalogs section in Microsoft Word.
1. To install KCM Toolbox for Word, in your KCM installation, navigate to the `<deploy root>\KCM\Programs\5.4\ITPMDKRepositoryServer\ccmtoolboxforword\webapp` directory, copy its content, and then deploy it on your web server. Make a note of the URL used to access the application.
  2. Copy the file `ccmforword manifest-5.4.xml` from `<deploy root>\KCM\Programs\5.4\ITPMDKRepositoryServer\ccmtoolboxforword\manifest` to the shared location listed in the Trusted App Catalogs section in Microsoft Word.

3. In the file `ccmforward manifest-5.4.xml`, specify values for the `<AppDomain>` node and the `DefaultValue` attribute of the `<SourceLocation>` node:
  - In the `<AppDomain>` node, specify the URL to access the application.
  - In the `DefaultValue` attribute of `<SourceLocation>`, specify the URL to access the application and add `/app/index.html` to it.
4. Configure a shared location for the `ccmforward manifest-5.4.xml` file using the `/Toolboxpath` flag in ManageCM. The location must be configured for each contract where you want to use KCM Toolbox for Word. For more information, see the sections "Create a contract" and "Change properties of a contract" in the *Kofax Communications Manager Getting Started Guide*.

For information on using KCM Toolbox for Word in KCM Designer, see the KCM Designer online Help.

## Chapter 8

# Use of Microsoft Word by KCM Core

KCM Core uses Microsoft Word for a number of operations such as printing and conversion of documents. To enable these operations, Microsoft Word must be installed locally on the servers that run KCM Document Processors.

**Note** You do not need Microsoft Word for conversion to PDF if the conversion is done with the Rendition technology. For more information on this technology, see the section "DocToPDF" in the *Kofax Communications Manager Core Scripting Language Developer's Guide*.

You should disable grammar and spelling tools in Microsoft Word. For more information, see [Adjust settings in Microsoft Word](#).

Also, Microsoft Word requires the presence of a Desktop directory. The location of this directory depends on the versions of Microsoft Word and Microsoft Windows. For 64-bit versions of Microsoft Word, create the following directory: %WINDIR%\System32\config\systemprofile\Desktop.

KCM Core attempts to create this directory if it does not exist yet. If it fails, KCM Core is unable to open Microsoft Word documents.

**Note** Microsoft Word must be started interactively at least once under the user account used for KCM Core Document Processors. This initializes Microsoft Word for that account so that it can be used by the Document Processors. If this is omitted, certain Core Script calls, such as DocToPDF, may fail.

## Adjust settings in Microsoft Word

Open Microsoft Word and check if the following settings are adjusted correctly.

- Navigate to **File > Options > Display > Printing Options** and select **Update fields before printing**. Click **OK**.
- Navigate to **File > Options > Advanced > Save** and clear **Allow background saves**. Click **OK**.
- Navigate to **File > Options > Proofing > AutoCorrect Options** and clear everything on all tabs. Click **OK**.

## Chapter 9

# Uninstallation

To uninstall KCM 5.4.0, navigate to `<deploy root>` and execute the following command. Place the command on a new line.

```
.\KCM\Programs\5.4\Uninstall\Uninstall.ps1
```

This command removes all Kofax Communications Manager 5.4.0 related programs, files, services, and registry entries.

**Note** The content of the KCM Repository and KCM Batch & Output Management databases is not removed when an instance is uninstalled.



## Chapter 10

# Troubleshooting

This chapter gives troubleshooting tips that may be useful if you encounter an issue while using the product.

## Package cannot connect to database

If the package cannot connect to the database using a SQL Server, verify that SQL Server authentication is enabled, and that SQL Server has been restarted.

Also, the SQL Server user has a password policy that should be set to "never expire."

## Services cannot be created

If the package fails when creating a service, it may be unable to find the user. If a local user is used, verify that the value for the parameter `Services!User` starts with a period and backslash (.).

## Cannot start KCM Core for development or test license

The license document and the XML file required for the package differ. The environment for KCM Core should be set to "Development and Test." Otherwise, the KCM Core service cannot be started due to an invalid license.

## Warning messages

During the deployment, the KCM package may wait for a certain service to stop. This is indicated by warning messages similar to the following:

```
WARNING: Waiting for service '{serviceName}' to finish stopping
```

## Deployment log files are empty

The deployment scripts can fail to write any content to the log files on Windows Server 2012 R2 if the Windows update 2919355 is installed. To resolve this issue, the Windows 3014136 Hotfix must be applied, which is available from the Microsoft support web page: <https://support.microsoft.com>

On the web site, search for the Knowledge Base article "3014136" to find "PowerShell transcript file doesn't contain the correct information in Windows Server 2012 R2," and then download the Hotfix.

## CMRestAPI does not start during system startup

The CMRestAPI service can fail to start during the Windows startup sequence. In this situation, the Service Control Manager logs an event 7000 with the error "The service did not respond to the start or control request in a timely fashion."

This error occurs if the system is underprovisioned causing the loading of the service to exceed the default 30,000 milliseconds, which results in the Service Control Manager terminating the service. To resolve this, assign more resources to the system or increase the startup timeout for services to an appropriate value.

You can find the instructions to change the service timeout on the Microsoft support web page: <https://support.microsoft.com>.

On the web site, search for the Knowledge Base article "922918" to find "A service does not start, and events 7000 and 7011 are logged in Windows Server...," and follow the instructions.

## Conversion from PDF to XPS fails

In Batch & Output Management, PDF documents may intermittently fail to convert to XPS. During the XPS conversion, the Compant DocBridge Mill reports missing CMap files.

For the intermediate conversion to XPS, DocBridge may require additional Adobe CMap resources. You can find Adobe CMap files in the Adobe Reader installation directory. To refer to the list of required files, extend the `<resourcelist>` element in the `mffxps.pro` and `mffpdf.pro` profiles.

**Note** Adobe CMap files are not needed for earlier DocBridge versions such as 2017.02 or earlier.