Kofax Communications Manager

Installation Guide

Version: 5.5.0

Date: 2021-08-16

KOFAX

© 2013–2021 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.

Document revision history

Date	Revision	Description
June 7, 2021	1.0	Initial version
August 16, 2021	1.1	Fixed errors and typos

Table of Contents

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:[1]

https://docshield.kofax.com/Portal/Products/KCM/5.5.0-3py4cyc64a/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.

[1] 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.

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

Overview

Kofax Communications Manager, also known as KCM, is a single integrated solution to automatically produce large numbers of standard documents and compose professional individual correspondence. KCM is a solution that empowers the organization to engage and communicate with customers by generating relevant and personal communication delivered through different channels.

KCM architecture and components

A base KCM installation consists of a Contract Manager and one or more KCM instances.

KCM instance

A KCM instance handles document processing for a particular partner and customer (tenant). In an on-premise deployment, KCM instances are represented by numeric IDs (numbers) from 1 to 99, for example, `Instance_21`. A single KCM instance comprises the following key components:

- KCM Core – a component responsible for creation and processing of PDF files. KCM Core carries out document composition by executing Document Templates and producing Document Packs. In addition, KCM Core allows for the creation of custom Core Services that implement Contract Manager interfaces.
- KCM Repository – a component that stores and manages the Document Template definitions. Each KCM Repository has a Repository server and Repository database that contains data for one particular tenant (that is, *partner-customer* combination explained in the next paragraph). Every KCM instance requires its own Repository database, as instances do not share these databases.
- KCM Designer – a web-based user interface that enables users to create and test Document Templates. It supports document lifecycle management, revision management, and changesets control.
- KCM ComposerUI for HTML5 – a component that supports integration of interactive document composition in a web application.

Each KCM instance registers itself on the Contract Manager, as explained below.

Contract Manager, contracts, and contract types

Contract Manager acts as the main entry point that manages connections to KCM instances and hands out work to these instances. Client applications send requests to the Contract Manager, which will then relay them to the appropriate KCM instance. This is done according to a number of *contracts* that are defined at the Contract Manager. Each contract identifies a partner and a customer operating Kofax Communications Manager.

Generally, a *partner* is a Kofax partner providing content management services by means of KCM, and a *customer* is consuming these services.

For example, a default contract deployed during the KCM installation sets *partner* to `CCM`, meaning that services will be provided by the solution itself. *Customer* is set to `Local`, meaning that services will be consumed by a local organization. You can add extra partners if needed, for example, a reseller.

Note You cannot remove the default partner.

Available operations are defined by *contract types* and their different *versions*. Each version of a contract type provides a consistent set of KCM service calls required to handle specific operations. For example:

- The `CCMInteractive` contract type handles the definition, management, and composition of documents and Document Packs.
- The `CCMAdministration` contract type deals with common administrative tasks, such as adding a new user, assigning a role to the user for a particular project, and starting a KCM Designer session.

If the predefined contract types are not sufficient, customers can add their own contract types and interfaces. These custom interfaces will need a Core Service for their implementation. See the *Kofax Communications Manager Core* guide and the *Kofax Communications Manager Core Scripting Language* guide for details on Core Services and Core Scripting.

So, the Contract Manager is a gateway to the KCM functionality and API for business applications, providing the interface to Kofax Communications Manager for all communications: runtime, design time, and system management. Its key functions are exposing services (SOAP) and being a proxy for HTTP(S) traffic.

All interface definitions and Contract Manager configuration settings are stored in the Contract Manager internal database.

Contract Manager communicates with KCM Core directly over TCP/IP.

Registration process

When a KCM instance tries to register itself for a particular contract, the Contract Manager checks if this KCM instance implements each interface of each contract type in the contract. If it does, the registration is a success, and the Contract Manager starts relaying the web service requests for that contract to the corresponding Core Service of the KCM instance.

KCM Batch & Output Management

KCM Batch & Output Management is an optional component that facilitates advanced multi-channel handling of produced output. It relies on an installed KCM instance to provide Document Pack composition.

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 number of Document Processors that will be deployed depends on the license type: for a Capacity-based license, the number of Document Processors will be set to maximum permitted by the license. For a Volume-based license, it will default to four Document Processors. To specify a different number of Document Processors, you can configure the `Instance!NumberOfDPs` parameter during deployment. Alternatively, you can use KCM Core Administrator to add or remove Document Processors after installation.

Tomcat web servers

KCM uses Tomcat as web servers/web application servers. Ensure that the Tomcat installation uses the default name for its Windows service, depending on the Tomcat version, for example, `tomcat8`. If you change the service name, both KCM installation and additional instances deployment fail.

For more information, see the Tomcat documentation available on the Internet.

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 `SetWebserverServiceUser.exe` tool that resides in: `<deploy root>\KCM\Programs\5.5\Management`

Run the tool, providing the new user account and password, as follows:

```
SetWebserverServiceUser Webserver!ServiceUser=.\myuser Webserver!
ServicePassword=myPassword
```

Important Make sure that the new account has the *Logon as a service* rights.

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

KCM uses databases to store production data.

- If you plan to deploy KCM only (without Batch & Output Management), you will need a KCM Repository database.
- If you also plan to deploy Batch & Output Management, you will need a B&OM Repository database and B&OM Runtime database.

The following versions are supported.

Database	Version	Details
SQL Server	Azure SQL	Not supported for B&OM databases.
	SQL Server 2019 - all editions	
	SQL Server 2017 - all editions	The end of Mainstream Support period is October 11, 2022. Then Extended Support will be required.
	SQL Server 2016 - all editions	The end of Mainstream Support period is July 13, 2021. Then Extended Support will be required.
	SQL Server 2014 - all editions	Extended Support is required.
Oracle	Oracle 19c	Not supported for Batch & Output Management databases.
	Oracle 12c	

Note Starting with KCM 5.5.0, Oracle 11g is not supported.

The deployment does not require a dedicated database server. You can use the database server that is already operating in your environment.

1. Create all required databases manually.
2. Before starting the installation, make sure these databases are empty.

3. Back up all databases on a regular basis and follow the best practices recommended by your database provider.
4. Unless following specific instructions from Kofax Technical Support, do not make database modifications, because they may cause irreversible damage and loss of data.

KCM Repository database

If you plan to deploy multiple KCM instances, consider that they cannot use the same database for KCM Repository. In this scenario, make sure you have a dedicated database prepared for each KCM instance you plan to use.

Whenever you use an Oracle or an SQL Server database for KCM Repository, you must supply an ODBC Connection String. A 64-bit ODBC driver is required.

SQL Server

For SQL Server database, consider the following:

- Database must be created with case-insensitive collation.
- TCP/IP protocol for SQL Server must be enabled.
- SQL Server authentication mode must be set to **Mixed mode** (both SQL Server and Windows Authentication).
- By default, SQL Server authentication is used. SQL Server account for database access must have the `database owner` role assigned and use the `dbo` schema.
- Windows authentication is also supported. To use it, you need to configure specific parameters. See the Install Kofax Communications Manager section for details.

> **Note** 64-bit DBMS drivers are not supported for Database Interface Definition (DID) connections. These drivers must be configured as the 32-bit DBMS drivers.

Oracle

For Oracle database, consider the following:

- Sufficient table space must be available for KCM Repository database, minimum is 60 MB.
- A 64-bit DSN configuration is required.
- Oracle Client versions 11.02.00.01 and 11.02.00.02 are not supported for use with KCM Repository.
- Oracle user you plan to use for database access requires the `Create Session` and `Create Table` privileges.

> **Note** When creating a user in Oracle, do not use lowercase characters in a quoted user identifier. When logging on to Oracle, KCM extends the connection with a user identifier without quotes. This causes Oracle to interpret the user name as uppercase.

Example:

- If you create "`KCM_USER`" in Oracle, KCM will be able to log in with this user.
- If you create “`Kcm_User`” in Oracle, KCM will not be able to log in, because `Kcm_User` is automatically interpreted as `KCM_USER`, and that user does not exist.

Databases for Batch & Output Management

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

Batch & Output Management requires two empty databases:

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

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

The connection strings that you supply during Batch & Output Management installation must be a non-ODBC connection strings.

Note 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.

SQL Server

SQL Server database settings for Batch & Output Management are similar as those for the KCM Repository database. See the KCM Repository database section for details.

In addition, the user accounts used to access the B&OM Repository and B&OM Runtime databases must have the `database owner` role and `ALTER_TRACE` privilege for these databases.

Oracle

For Oracle database, consider the following:

- Batch & Output Management requires the Oracle 32-bit client. Thus, if you plan to deploy both KCM and KCM Batch & Output Management on the same server and use Oracle for their databases, you need both the 32-bit and 64-bit Oracle clients present on that server.
- Batch & Output Management requires the use of the Oracle 11g driver.
- Batch & Output Management requires a non-ODBC connection string.

For both B&OM Repository and B&OM Runtime database users, the following privileges are required:

- `CREATE SESSION`
- `CREATE TABLE`
- `ALTER SESSION`
- `CREATE VIEW`
- `CREATE PROCEDURE`
- `CREATE SEQUENCE`

In addition, for Oracle user for the B&OM Repository database, the following privileges are required:

- `CREATE TYPE`
- `CREATE TRIGGER`
- `EXECUTE ON DBMS_LOCK`

- `EXECUTE ON DBMS_LOB`
- `EXECUTE ON DBMS_OUTPUT`
- `EXECUTE ON CTX_DDL`

Chapter 2

Install Kofax Communications Manager

This chapter describes how to install Kofax Communications Manager on a Windows Server. If you plan the deployment on Docker, see *Chapter 3: Install Kofax Communications Manager on Docker* of this guide.

Kofax Communications Manager base installation consists of a Contract Manager and one or more KCM instances. As described in the Overview section, each KCM instance handles the document processing for a particular partner and customer (tenant), and Contract Manager acts as the main entry point that hands out work to these instances.

Also, KCM features Batch & Output Management (B&OM) – an optional component that provides advanced multi-channel handling of produced output. It relies on an installed KCM instance that provides Document Pack composition.

These two components can be installed separately with their own installation tools:

- `Install.exe` installs a Contract Manager and a single KCM instance on a single server, with a default configuration, and in a running state.

Note This basic installation is similar to pre-5.5.0 installations, but without Batch & Output Management.

- `InstallOutputManagement.exe` that installs Batch & Output Management component on a single server. Starting with version 5.5.0, Batch & Output Management will be installed separately from KCM, to a dedicated installation folder. This enables you to add Batch & Output Management to a KCM 5.5.0 deployment that does not have it, or remove it from the KCM 5.5.0 deployment that has it installed.

Note KCM no longer supports a limited number of predefined setups. Instead, the required setup can be achieved by installing the required components separately. For example, `Install.exe` provides a basic initial KCM installation on a single server. If needed, you can extend this deployment with additional KCM instances or Contract Manager. For advanced setup with an additional Contract Manager or KCM instance installation, see the Advanced installation section of this guide.

KCM Designer for Windows

In addition to the main components, the solution incorporates KCM Designer for Windows, which is a client application used to perform certain administrative tasks for KCM Designer, for example, managing projects and users, or defining a Data Backbone. You can install KCM Designer for Windows manually after installing the KCM main components, using the KCM Designer for Windows setup. See the KCM Designer for Windows installation section of this guide for details.

Note Currently, KCM Designer for Windows is still supported to get access to KCM Repository for content development. However, it is recommended to use the web-based KCM Designer instead. It provides a number of new features that are not supported in KCM Designer for Windows.

The setup package contains several files and two folders:

1. `Dockerfiles` folder for Docker installation.
2. `InstallServer` folder for installation on a Windows Server. This folder includes a `Software` subfolder and several installation executables described below.

Basic KCM installation

This section describes the basic KCM setup tool `Install.exe`.

Overview

`Install.exe` is a command-line tool designed for a streamlined installation and initial configuration of KCM. It enables you to automatically deploy a Contract Manager and a KCM instance to a single server.

The tool works as follows:

1. It checks the requirements needed for the installation.
2. It installs a Contract Manager and one KCM instance ready for use.

   Note If required, you can append additional KCM instances later, as described in the Advanced installation section of this guide.

3. The KCM instance is registered on the Contract Manager with the default contract, with *Partner* set to *CCM* and *Customer* set to *Local*.
4. Both components are started automatically.

Before you start

This section lists the prerequisites and preparatory steps you have to take before you run `Install.exe`.

`Install.exe` can only be started on a server that has no active KCM installation. If you need to upgrade your existing installation, refer to the Upgrade your installation section of this guide.

The account used to run `Install.exe` must be included in the local *Administrators* group on the target server.

Prepare the following user accounts:

- User account for KCM Windows services. This can be a local account or a domain account. It requires:
  1. Membership in the local *Administrators* group.
  2. *Logon as a service* right.
- User account for KCM web applications. This can be a local account or a domain account. It requires:
  1. *Logon as a service* right.
  2. *Read & execute* permission on the Tomcat installation folder.

3. If you plan to use SSL connections to KCM server, then also assign *Read* permission on the SSL keystore file. See the `ContractManager!UseSSL` description in the "Parameters" section below for details.

The installation will automatically create the KCM Designer *Administrator* account. You can change its password as described in the "Change your password" topic in *Help for KCM Designer*.

Prepare a database for KCM Repository:

1. Make sure that the empty database exists on the database server.
2. For SQL Server database, consider the following:
   - If the database is not installed locally, Microsoft SQL Server Command Line Utilities have to be installed on the KCM server. 64-bit DBMS drivers are not supported for DID connections. These drivers must be configured as the 32-bit DBMS drivers.
   - Default authentication method is SQL Server authentication.
   - Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters. See the "Parameters" section below for details.
   - If you plan to access SQL Server database using SQL Server authentication, make sure the account you plan to use for database access has the *database owner* role assigned and uses the *dbo* schema.
3. For Oracle database, consider the following:
   - Oracle user you plan to use for database access requires the `Create Session` and `Create Table` privileges.
4. All communication with the KCM installation is either HTTP or HTTPS. You need to decide whether connections to this KCM server should use SSL.

   > **Note** 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 supports request rate limiting between the KCM installation and the outside network.

5. Decide whether users accessing KCM Designer should be authenticated using LDAP.
6. Decide whether integrating applications will need additional authentication when connecting to Contract Manager.

Parameters

> **Note** To view the list of parameters for `Install.exe`, launch it without any parameters.

Parameter	Required / Optional	Description
`Deploy!RootPath`	Required	The root drive or folder in which the KCM installation will be created.

Parameter	Required / Optional	Description
`Java!Home`	Optional	The folder in which Java is installed in your system, either as JRE or JDK. This Java installation will be used for the KCM web applications. If omitted, KCM will use the `JAVA_HOME` environment variable.
`Services!User`	Required	The user account for KCM Windows services. • This user must be a local *Administrator*. • This user requires *Logon as a service* right. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example** `.\AdminUser`
`Services!Password`	Required	The password for the KCM services user account.
`Webserver!ServiceUser`	Required	The user account to run the KCM web applications. This account requires the following: • *Logon as a service* right. • *Read & execute* permission on the Tomcat installation folder. • If you plan to use SSL connections, then also assign *Read* permission on the SSL keystore file. See the `ContractManager!UseSSL` parameter description below for details. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example** `.\AdminUser`
`Webserver!ServicePassword`	Required	The password for the account that web applications will use.

Parameter	Required / Optional	Description
`ContractManager!UseAuthentication`	Required	Boolean flag that specifies whether to enable authentication for executing SOAP calls on the Contract Manager. • If set to `True`, then the calling application needs to supply a user name and a password using an HTTP Basic Authentication header. **Note** Authentication requires secure connections to be enabled. • If set to `False`, anyone with access to the Contract Manager can execute any SOAP call on its interface without additional authentication.
`ContractManager!UseSSL`	Required	Boolean flag that specifies whether the integrating applications need to use a secure protocol for connections to the Contract Manager. • If set to `True`, connections are secure, using the HTTPS protocol (HTTP over SSL). • If set to `False`, connections are not secure, using the HTTP protocol.
`ContractManager!KeyStoreFile`	Required for HTTPS connections to Contract Manager on this KCM server	The location of the certificate file (`.pfx` or `.p12`).
`ContractManager!KeyStorePassword`	Required for HTTPS connections to Contract Manager on this KCM server	The password of the certificate file (`.pfx` or `.p12`).
`ContractManager!UseLDAP`	Required	Boolean flag that specifies whether users can log in to KCM Designer using LDAP authentication. • If set to `True`, users can log in to KCM Designer with their LDAP user accounts. **Note** LDAP authentication requires secure connections to be enabled. • If set to `False`, users can log in to KCM Designer with internal user accounts that are managed by the KCM Repository server.

Parameter	Required / Optional	Description
`ContractManager!LDAPPropertiesFile`	Required if you want to enable LDAP authentication	The configuration file for LDAP authentication. See the "LDAP properties file" section of the Kofax Communications Manager Getting Started Guide for details.
`Instance!DatabaseType`	Required	Type of the databases that the deployment uses. Enter `SQLServer` or `Oracle`.
`Instance!DatabaseConnectionString`	Required	The ODBC connection string to be used by the KCM Repository server to connect to the database. Make sure that the database exists on the database server and is empty. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below. For SQL Server database, consider the following: • Default authentication method is SQL Server authentication. • Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters.
`Instance!DatabaseUser`	Required for connection to Oracle database or to SQL Server database using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	The database user account that KCM Repository server will use to connect to the database. Required to connect to Oracle database or to SQL Server database using SQL Server authentication. Make sure the account has the required roles and privileges, as described in the Databases section of this guide.

Parameter	Required / Optional	Description
`Instance!DatabasePassword`	Required for Oracle database user and for SQL Server connection using SQL Server authentication.	The database password that KCM Repository server will use to connect to its database.
`Instance!LicenseFile`	Required	The location of the .xml file with the license information.
`Instance!NumberOfDPs`	Optional	The number of Document Processors to create for the instance, between 1 and 99. When using a capacity-based license, this number must not exceed the maximum set in the license. For details, see the License section in this guide. If omitted, default value will be used: • For a capacity-based license — set to maximum allowed. • For a volume-based license — set to 4.
`Instance!DesignerAdministratorAccountPassword`	Required	The password for the KCM Designer *Administrator* account (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. If the specified password does not meet these requirements, a default password will be generated by the installation instead. **Note** This default password will only be available in the output console. To change it, refer to the "Change your password" topic in *Help for KCM Designer*.
`Instance!ExampleProject`	Optional	Boolean flag that specifies if the "InstallationTest" example project should be imported in the KCM Repository database. If omitted, defaults to `True`.
`Registration!ExampleWebAppName`	Optional	The name of the example web application that will be created for the KCM instance. If omitted, the example web application will not be created.

Notes on usage

- To simplify input, you may want to create a script with the required parameter values and then use it to run `Install.exe`.
- The order of parameters is irrelevant.
- If the installation fails, then the unfinished deployment must be removed manually with the `Uninstall.exe` tool described in the Uninstallation section of this guide.

Advanced installation

As explained in the Overview section, basic KCM installation contains the following server-side components:

- Contract Manager, which is the main entry point for Kofax Communications Manager. It accepts applications requests, and hands them out to an appropriate instance for further processing.
- One or more KCM instances, which handle requests for a particular *partner* and *customer*.

As indicated in the Basic KCM installation section, you can use `Install.exe` to implement a basic deployment with just a Contract Manager and single KCM instance on one server.

However, you may need to install KCM on multiple computers, with one computer containing the Contract Manager and several computers containing one or more KCM instances. You may also need to add a KCM instance to any server, either on the server that has the Contract Manager installed, or on a completely separate one. This approach makes KCM scalable and easy to upgrade.

Install required components

To install the required component separately, use the following tools:

- `AddContractManager.exe` — this tool installs the Contract Manager to a single server where it has not yet been installed.
- `AddInstance.exe` — this tool installs a KCM instance to a single server. This may either be the same server that the Contract Manager resides on, or a separate server. The KCM instance will be added to any existing ones on that server, if any.

Start services

In contrast to the basic installation tool, `AddContractManager.exe` and `AddInstance.exe` do not install the related components in a running state. Instead, the appropriate tools must be used to start these components:

- `StartContractManager.exe` — this tool starts the Contract Manager component.
- `StartInstance.exe` — this tool starts the KCM instance component. As part of starting the instance, it will register this instance at its Contract Manager for a particular partner and customer.

Note Manual registration of KCM instances at the Contract Manager is still possible. However, the preferred way is to have `StartInstance.exe` register the instance automatically, or use the instance registration file that `StartInstance.exe` generates. Any manual registration may get overridden by `StartIntance.exe` and `StopInstance.exe` (for details, see the Start KCM instance section of this guide).

- In addition, `StartInstance.exe` can optionally install an example web application for the KCM instance at the Contract Manager.

Note It is still possible to use `AddExampleWebApp.exe` and `RemoveExampleWebApp.exe` to add and remove an example web application at the Contract Manager for a particular instance. However, the use of these tools is generally superseded by the web application that is controlled by `StartInstance` and `StopInstance`, which may undo such installations. `AddExampleWebApp.exe` and `RemoveExampleWebApp.exe` should only be used in cases when `StartIntance` and `StopInstance` are unable to add or remove the example web application.

Configure settings

Depending on the parameters that were passed to `AddContractManager.exe` and `AddInstance.exe`, their launch results in a fully configured installation, or a partially configured one.

Note Using these setups, you can postpone configuring the certain settings while installing the software.

The following tools enable adjusting and completing the configuration after installation. A reconfiguration requires a restart of the component for it to come into effect.

- `SetContractManagerAuthentication.exe` — adjusts Contract Manager authentication settings.
- `SetContractManagerConnection.exe` — adjusts the SSL connection configuration of the Contract Manager.
- `SetContractManagerLDAPMode.exe` — adjusts the LDAP configuration of the Contract Manager.
- `SetInstanceRegistration.exe` — adjusts the Contract Manager that an instance is associated with and specifies the partner and customer that the instance will handle.
- `SetInstanceDatabase.exe` — adjusts the KCM Repository database that an instance uses. This can be an existing or a new database.
- `SetInstanceLicense.exe` — adjusts the license.
- `ManageCM` — enables you to define contracts, contract types and interfaces.
- `ImportInstanceConfiguration.exe` — enables you to import a previously exported configuration. Importing such a configuration may overwrite settings that are also controlled by one of the tools listed above.

These tools are described in the *Kofax Communications Manager Getting Started Guide*.

KCM instance setup tools

This section describes KCM instance setup tools:

- AddInstance.exe
- StartInstance.exe

- StopInstance.exe

Add KCM instance

This section describes how you can use the `AddInstance` tool to add a KCM instance to your installation.

Overview

`AddInstance.exe` is a command-line tool intended to deploy a KCM instance. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management`.

`AddInstance.exe` works as follows:

1. It checks the requirements needed for the KCM instance installation.

   Note If the check fails, the installation quits.

2. It installs the KCM instance and configures it based on the command line parameters.

You can use the `AddInstance` tool to implement the following scenarios:

- Create a new KCM installation that only consists of a KCM instance.
- Add a KCM instance to an existing KCM installation.

The new KCM instance can be installed on the same server with the Contract Manager, or on the remote server.

Note Since the `AddInstance` tool has no default way of handling registration, the *Partner* and *Customer* must always be supplied on the command line.

Before you start

This section lists the prerequisites and preparatory steps you have to take before you run `AddInstance.exe`.

1. The account used to run `AddInstance.exe` must be included in the local *Administrator*s group on the target server.
2. Multiple KCM instances cannot use the same database for KCM Repository, so make sure you have the databases prepared for all KCM instances you plan to deploy.

Parameters

To view the list of parameters for `AddInstance.exe`, launch it without any parameters.

Parameter	Required / Optional	Description
`Instance!Number`	Required for on-premise deployment.**	Numeric ID from 1 to 99 for the new KCM instance. Must differ from the existing instances. **Note** For Docker deployment, it is always set to 1.
`Deploy!Docker`	Optional	Boolean flag that specifies whether a special configuration for Docker will be used. If omitted, defaults to `False`.
`Deploy!RootPath`	Required*	The root drive or folder in which the KCM installation will be created.
`Java!Home`	Optional*	The folder in which Java is installed in your system, either as JRE or JDK. This Java installation will be used for the KCM web applications. If omitted, KCM will use the `JAVA_HOME` environment variable.
`Services!User`	Required	The user account to run KCM services. • This account must be a local *Administrator*. • This account must have the *Logon as a service* right. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning. **Example**: `.\AdminUser`
`Services!Password`	Required	The password for the KCM service account.
`Webserver!ServiceUser`	Required*	The user account to run the KCM web applications. This account requires the following: • *Logon as a service* right. • *Read & execute* permission on the Tomcat installation folder. • If you plan to use SSL connections, then also assign *Read* permission on the SSL keystore file. See `ContractManager!UseSSL` description for details. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example:** `.\AdminUser`
`Webserver!ServicePassword`	Required*	The password for the account that web applications will use.

Parameter	Required / Optional	Description
`Instance!DatabaseType`	Optional	The type of the database to be used by the KCM Repository of the instance. Specify `SQLServer` or `Oracle`.
`Instance!DatabaseMode`	Optional	Allowed values: `New`, `Check`, `Overwrite`. • If set to `New`, the database for the KCM Repository is expected to be empty. • If set to `Check`, the database for the KCM Repository is expected to have content, with the same *Partner* and *Custome*r. • If set to `Overwrite`, the database for the KCM Repository is expected to have content, with different *Partner* and/or *Customer*. If omitted, defaults to `New`.
`Instance!DatabaseConnectionString`	Optional	The ODBC connection string to be used by the KCM Repository server to connect to the database. Make sure that the database exists on the database server and is empty. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below. For SQL Server database, consider the following: • Default authentication method is SQL Server authentication. • Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters.
`Instance!DatabaseUser`	Optional: • Required for connection to Oracle database or to SQL Server database using SQL Server authentication. • Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	The database user account that KCM Repository server will use to connect to the database. Required to connect to Oracle database or to SQL Server database using SQL Server authentication. Make sure the account has the required roles and privileges, as described in the Databases section of this guide.

Parameter	Required / Optional	Description
`Instance!DatabasePassword`	Optional. Required for Oracle database user and for SQL Server connection using SQL Server authentication.	The database password that KCM Repository server will use to connect to its database.
`Instance!LicenseFile`	Optional	The location of the .xml file with the license information.
`Instance!NumberOfDPs`	Optional	The number of Document Processors to create for the instance, between 1 and 99. When using a capacity-based license, this number must not exceed the maximum set in the license. For details, see the License section of this guide. If omitted, default value will be used: • For a capacity-based license — set to maximum allowed. • For a volume-based license — set to 4.
`Instance!ExampleProject`	Optional	Boolean flag that specifies if the "InstallationTest" example project should be imported in the KCM Repository database. **Note** The example project can only be imported if the database is empty (does not have existing content). If omitted, defaults to `True`.
`Instance!ImportConfigurationFile`	Optional	Path to exported configuration file that will be loaded automatically when a KCM instance is installed. See *Chapter 6: Export and import configuration* in the *Kofax Communications Manager Getting Started Guide* for more information. If omitted, default initial configuration will be applied.
`Instance!ImportInfrastructure`	Required if `Instance!ImportConfigurationFile` value was specified.	Boolean flag that specifies whether the exported configuration should be loaded with or without infrastructure. See *Chapter 6: Export and import configuration* in the *Kofax Communications Manager Getting Started Guide* for more information.
`Registration!Partner`	Optional	*Partner* that will be specified when the new KCM instance is registered on the Contract Manager. This information is stored in the KCM Repository database. *Partner* information will be used for verification when the database is connected to another KCM instance.

Parameter	Required / Optional	Description
`Registration!Customer`	Optional	*Customer* that will be specified when the new KCM instance is registered on the Contract Manager. This information is stored in the KCM Repository database. *Customer* information will be used for verification when the database is connected to another KCM instance.
`Registration!ContractManagerLocation`	Optional	Specifies the Contract Manager location when registering the new KCM instance: • If set to `Local`, the instance will be registered on the Contract Manager on the local server. • If set to `Remote`, the instance will be registered on the Contract Manager on a remote server. • If set to `None`, the instance will not be registered on the Contract Manager.
`Registration!ContractManagerHost`	Required for remote Contract Manager Optional for local Contract Manager	The host of the Contract Manager on which the instance will be registered. Specify server name or IP address. Required in the following scenarios: • When the Contract Manager is remote. • If you need to explicitly set or override the host when the Contract Manager is local. • When a configuration with infrastructure is imported. Optional for local Contract Manager. If omitted, defaults to `localhost`.
`Registration!InstanceHost`	Required for remote Contract Manager Optional for local Contract Manager	The KCM instance host with which it will register itself on the Contract Manager. Specify server name or IP address. Required in the following scenarios: • When the Contract Manager is remote. • If you need to explicitly set or override the KCM instance host when the Contract Manager is local. • When a configuration with infrastructure is imported. Optional for local Contract Manager. If omitted, defaults to the local server name.

Parameter	Required / Optional	Description
`Registration!` `ExampleWebAppName`	Optional	The name of the example web application that will be created for the instance. When the other registration parameters are present and this parameter is not supplied, then the example web application will not be created.

* When adding a KCM instance to an existing installation, this parameter is not available.

** When installing an instance on Docker, this parameter is not available.

Notes on usage

- The order of parameters is irrelevant.
- The `AddInstance` tool does not start the new KCM instance automatically. You will need to start it using `StartInstance` tool described in the next section.
- The `AddInstance` tool does not perform the registration of the new KCM instance on the Contract Manager and KCM Repository database initialization. These tasks are performed later when the instance is started by the `StartInstance` tool.
- If you plan to configure a new KCM instance by importing a configuration with infrastructure, this will also include KCM instance registration. In this scenario, the `Registration!Partner`, `Registration!Customer` and `Registration!ContractManagerLocation` parameters of the `AddInstance` tool are not available. However, if you need to override host information, you can use the `Registration!ContractManagerHost` and `Registration!InstanceHost` parameters.
- If for some reason, the database, license and registration configuration parameters are omitted, the KCM instance will not be fully configured and, therefore, cannot be started. To finish the configuration, you need to use the `SetInstanceDatabase`, `SetInstanceLicense` and `SetInstanceRegistration` tools and then start the KCM instance.
- If the installation fails, then the unfinished deployment must be removed manually using the `RemoveInstance.exe` tool described in the Uninstallation section of this guide.

Start KCM instance

This section describes how you can use the `StartInstance` tool to start the KCM instance in your installation.

Overview

`StartInstance.exe` is a command-line tool intended to start Windows services of the KCM instance. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Instance`.

`StartInstance.exe` starts all Windows services of the KCM instance, including shared web services (if not already started), and sets their startup type to *Automatic*.

Note In KCM, *shared web services* (used by several KCM instances) are Windows services implemented on Tomcat web server.

`StartInstance.exe` can be used in the scenarios where you deploy a KCM instance by running `AddInstance.exe`. Also, it can be used with the `SetInstanceDatabase`, `SetInstanceLicense`, `SetInstanceRegistration` tools, which require the KCM instance to be restarted.

The tool works as follows:

1. It starts all Windows services of the KCM instance and sets their startup type to *Automatic*.
2. It connects the KCM instance to its KCM Repository database when starting for the first time (or if the database was changed).
3. A new database is initialized. An existing database is upgraded, and the *Partner/Customer* information is verified.
4. The instance is registered on the Contract Manager (unless `Registration!ContractManagerLocation=None`).
5. If a contract does not exist, it will be created and associated with the KCM instance.

Before you start

Before running the tool, make sure the required settings were configured by the `AddInstance` or by the `SetInstanceDatabase`, `SetInstanceLicense`, `SetInstanceRegistration` tools.

The following components and settings should be checked:

- Database for KCM Repository
- KCM license
- Registration settings

If they were not configured, KCM instance will fail to start.

Parameters

To view the list of parameters for `StartInstance.exe`, launch it without any parameters.

Parameter	Required / Optional	Description
`Instance!Number`	Required	Numeric ID from 1 to 99 for KCM instance to be started. **Note** For Docker deployment, it is always set to 1.

Parameter	Required / Optional	Description
`Instance!DesignerAdministratorAccountPassword`	Optional	The password for the KCM Designer Administrator account. For this password to be accepted by KCM Repository server, it must meet the following requirements: **1.** Have at least the minimal safe password length. Default minimum is 20 characters. **2.** Only contain letters (a-z, A-Z), numbers (0-9), and the "-" (En dash) and "_" (underscore) characters. If omitted, or if the specified password does not meet the requirements, a default password will be generated. **Note** This default password will only be available in the output console. To change it, refer to the "Change your password" topic in *Help for KCM Designer*.
`Registration!NewContract`	Required if the registration was changed and `Registration!ContractManagerLocation` is not `None`. Optional otherwise, with default `False`.	Boolean flag that specifies whether to create a new contract: • If set to `True`, the contract for the instance will be created by the registration process. **Note** If the registration process detects that the contract already exists, registration will fail. • If set to `False`, the contract for the instance is assumed to exist and be available. **Note** The registration will fail if the contract does not exist, or if the contract exists and is already associated with another active KCM instance.

Parameter	Required / Optional	Description
`Registration!Force`	Optional	This parameter is only relevant when `Registration!NewContract=False`. Boolean flag that specifies whether to overwrite an association between the existing contract and KCM instance: • If set to `True`, registration will not fail after detecting an existing contract associated with another active KCM instance – instead, this association will be overwritten. • If set to `False`, registration will fail.

Notes on usage

If registration on the Contract Manager fails, `StartInstance` will not fail as a whole. Instead, it will start the KCM instance without registering it at the Contract Manager.

Registration can be completed manually as follows:

1. Locate the instance registration file at: `<deploy root>\KCM\Work\<version>\Instance_<num>\instance.xml`.
2. Copy the file to the Contract Manager server.
3. Manually run the following command:

```
ManageCM /ProcessRegistrationFile /register /file=<registration file>
```

Stop KCM instance

This section describes how you can use the `StopInstance` tool to stop the KCM instance in your installation.

Overview

`StopInstance.exe` is a command-line tool intended to stop Windows services of the KCM instance (including shared web services if the latest instance is being stopped). This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Instance`.

The tool works as follows:

1. It stops all Windows services of the KCM instance and sets their startup type to *Disabled*.
2. It removes the registration of the KCM instance from the Contract Manager (unless `Registration!ContractManagerLocation=None`).

Before you start

Make sure the account used to run `StopInstance.exe` has the local *Administrator* permissions on the target server.

Parameters

Parameter	Required / Optional	Description
`Instance!Number`	Required	Numeric ID of the KCM instance to stop. **Note** For Docker deployment, it is always set to 1.

Notes on usage

If the tool fails to remove the registration on the Contract Manager due to some reason, it will, however, stop the KCM instance. After that, you can remove the registration manually as follows:

1. Locate the instance registration file at: `<deploy root>\KCM\Work\<version>\Instance_<num>\instance.xml`.
2. Copy the file to the Contract Manager server.
3. Manually run the following command:

```
ManageCM /ProcessRegistrationFile /unregister /file=<registration file>
```

Contract Manager setup tools

This section describes the details of:

- AddContractManager.exe
- StartContractManager.exe
- StopContractManager.exe

Add Contract Manager

This section describes how you can use the `AddContractManager` tool to add the Contract Manager to your installation.

Overview

`AddContractManager.exe` is a command-line tool intended to deploy the Contract Manager.

The Contract Manager can be installed on the same server with the KCM instance, or on the remote server.

You can use the `AddContractManager` tool to implement the following scenarios:

- Create a new KCM installation that only consists of the Contract Manager. In this scenario, `AddContractManager` should be run from the `InstallServer` folder.
- Add a local Contract Manager to an existing KCM installation that does not yet have one. In this scenario, `AddContractManager` should be run from the `<deploy root>\KCM\Programs\<version>\Management` folder.

`AddContractManager.exe` works as follows:

1. It checks the requirements needed for the Contract Manager installation.

   > **Note** If the check fails, the installation quits.

2. It installs the Contract Manager and configures it based on the command line parameters.

> **Note** Since the tool does not start the Contract Manager automatically, you need to start it manually using the `StartContractManager.exe` as described in the Start Contract Manager section of this guide.

Before you start

This section lists the prerequisites and preparatory steps you have to take before you run `AddContractManager.exe`.

1. The account used to run `AddContractManager.exe` must have the local *Administrator* permissions on the target server.
2. Prepare the user account for KCM web applications. This can be a local account or a domain account. It requires:
   - *Logon as a service* right.
   - *Read & execute* permission on the Tomcat installation folder.
   - If you plan to use SSL connections to KCM server, then also assign *Read* permission on the SSL keystore file. See also `ContractManager!UseSSL` parameter description in the table below.
3. Decide whether to enable authentication for executing SOAP calls on the Contract Manager.
4. Decide whether users can be allowed to log in to KCM Designer using LDAP authentication.
5. Decide whether the integrating applications need to use a secure protocol to connect to the Contract Manager. If yes, prepare a certificate file (.pfx or .p12).
6. If it is necessary to import and apply an exported configuration, prepare the file you need. See *Chapter 6: Export and import configuration* in the *Kofax Communications Manager Getting Started Guide* for more information.

Parameters

To view the list of parameters for `AddContractManager.exe`, launch it without any parameters.

Parameter	Required / Optional	Description
`Deploy!RootPath`	Required*	The root drive or folder in which the KCM installation will be created.
`Deploy!Docker`	Optional	Boolean flag that specifies whether special configuration for Docker will be used. If omitted, defaults to `False`.

Parameter	Required / Optional	Description
`Java!Home`	Optional	The folder in which Java is installed in your system, either as JRE or JDK. This Java installation will be used for the KCM web applications. If omitted, KCM will use the `JAVA_HOME` environment variable.
`Webserver!ServiceUser`	Required	The user account to run the KCM web applications. This account requires the following: • *Logon as a service* right. • *Read & execute* permission on the Tomcat installation folder. • If you plan to use SSL connections, then also assign *Read* permission on the SSL keystore file. See `ContractManager!UseSSL` description for details. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example:** `.\AdminUser`
`Webserver!ServicePassword`	Required	The password for the account that web applications will use.
`ContractManager!UseAuthentication`	Optional	Boolean flag that specifies whether to enable authentication for executing SOAP calls on the Contract Manager. • If set to `True`, then the calling application needs to supply a user name and a password using an HTTP Basic Authentication header. **Note** Authentication requires secure connections to be enabled. • If set to `False`, anyone with access to the Contract Manager can execute any SOAP call on its interface without additional authentication. • If omitted, the additional authentication feature is not yet configured. You will need to run `SetContractManagerAuthentication.exe` to complete the configuration.

Parameter	Required / Optional	Description
`ContractManager!UseSSL`	Optional	Boolean flag that specifies whether the integrating applications need to use a secure protocol for connections to the Contract Manager. • If set to `True`, connections are secure, using the HTTPS protocol (HTTP over SSL). • If set to `False`, connections are not secure, using the HTTP protocol. • If omitted, the feature is not yet configured. You will need to run `SetContractManagerConnection.exe` to complete the configuration.
`ContractManager!KeyStoreFile`	Only required if `ContractManager!UseSSL` was set to `True`	The location of the certificate file (`.pfx` or `.p12`).
`ContractManager!KeyStorePassword`	Only required if `ContractManager!UseSSL` was set to `True`	The password of the certificate file (`.pfx` or `.p12`).
`ContractManager!UseLDAP`	Optional	Boolean flag that specifies whether users can log in to KCM Designer using LDAP authentication. • If set to `True`, users can log in to KCM Designer with their LDAP user accounts. **Note** LDAP authentication requires secure connections to be enabled. • If set to `False`, users can log in to KCM Designer with internal user accounts that are managed by the KCM Repository server. • If omitted, the feature is not yet configured. You will need to run `SetContractManagerLDAPMode.exe` to complete the configuration.
`ContractManager!LDAPPropertiesFile`	Required if `ContractManager!UseLDAP` was set to `True`	The configuration file for LDAP authentication. See the *LDAP properties file* section in the *Kofax Communications Manager Getting Started Guide* for details.
`ContractManager!ImportConfigurationFile`	Optional	Path to exported configuration file that will be loaded automatically when the Contract Manager is installed. If omitted, the default initial configuration will be applied. See the *Export Contract Manager configuration* section in the *Kofax Communications Manager Getting Started Guide* for details.

Parameter	Required / Optional	Description
`ContractManager!ImportInfrastructure`	Optional	Boolean flag that specifies whether the exported configuration should be loaded with or without infrastructure. See the *Export Contract Manager configuration* section in the *Kofax Communications Manager Getting Started Guide* for details.
`ContractManager!SharedResourceLocation`	Optional	Specifies the location of a folder where the shared objects for import and export, such as project files and reference projects, are stored. If omitted, defaults to `<deploy root>\KCM\Work\<version>\SharedResources`

* When adding the Contract Manager to the existing installation, this parameter is not available.

Notes on usage

- The order of parameters is irrelevant.
- The `AddContractManager` tool does not start the new Contract Manager automatically. You will need to start it using the `StartContractManager` tool described in the Start Contract Manager section of this guide.
- If the installation fails, then the unfinished deployment must be removed manually using `RemoveContractManager.exe` tool described in the Uninstall Contract Manager section of this guide.

Start Contract Manager

This section describes how you can use the `StartContractManager` tool to start the Contract Manager in your installation.

Overview

`StartContractManager.exe` is a command-line tool intended to start Windows services of the Contract Manager. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Contract Manager`.

`StartContractManager.exe` starts all Windows services of the Contract Manager and sets their startup type to *Automatic*.

It is used in the scenarios where you deploy the Contract Manager by running the `AddContractManager.exe`, as this tool does not start the Contract Manager automatically.

Note Also, if configuration changes are made by the `SetContractManagerAuthentication`, `SetContractManagerConnection`, or `SetContractManagerLDAPMode` tools, this requires the Contract Manager to be restarted. See the related sections of the Kofax Communications Manager Getting Started Guide for details.

Before you start

This section lists the prerequisites and preparatory steps you have to take before you run `StartContractManager.exe`.

1. The account used to run `StartContractManager.exe` must have the local *Administrator* permissions on the target server.
2. Required settings must be specified by the `AddContractManager` tool or by the `SetContractManagerAuthentication`, `SetContractManagerConnection`, `SetContractManagerLDAPMode` tools. These are the following:
    - Application authentication. See the `ContractManager!UseAuthentication` parameter description in the Add Contract Manager section of this guide.
    - User authentication with LDAP. See the `ContractManager!UseLDAP` parameter description in the Add Contract Manager section of this guide .
    - Secure connections to KCM server. See the `ContractManager!UseSSL` description in the Add Contract Manager section of this guide.

    If these Contract Manager configuration settings are not specified, then `StartContractManager` will fail.

Parameters

This command-line tool does not have any parameters.

Stop Contract Manager

This section describes how you can use the `StopContractManager` tool to stop the Contract Manager in your installation.

Overview

`StopContractManager.exe` is a command-line tool intended to stop Windows services of the Contract Manager. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Contract Manager`.

`StopContractManager.exe` stops all Windows services of the Contract Manager and sets their startup type to *Disabled*.

Before you start

Make sure the account used to run `StopContractManager.exe` has the local *Administrator* permissions on the target server.

Parameters

This command-line tool does not have any parameters.

KCM Designer for Windows installation

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 the detailed information are provided.

1. Once the KCM instance is deployed, navigate to `<deploy root>\KCM\Work\<version>\Instance_<num>\designer\Client` and locate the KCM Designer for Windows installer named `KCM Designer for Windows setup.exe`.
2. Run the installer.
3. When prompted, enter the installation name, that is, the directory name where KCM Designer for Windows will reside.

   > **Note** By default, on a 64-bit operating system the client is deployed in `C:\Program Files (x86)`.

4. Also, provide the necessary settings for server and host. To do so, open the ReadMe file (located in the same directory) and copy the values from this file.

KCM Toolbox for Word installation

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 active 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\<version>\ITPMDKRepositoryServer\ccmtoolboxforword\webapp` folder, 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-5.5.0.0.0.xml` from `<deploy root>\KCM\Programs\<version>\ITPMDKRepositoryServer\ccmtoolboxforword\manifest` to the shared location listed in the Trusted App Catalogs section in Microsoft Word.
3. In the file `ccmforword manifest-5.5.0.0.0.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 `ccmforword manifest-5.5.0.0.0.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 *Help for KCM Designer*.

B&OM installation

KCM Batch & Output Management (B&OM) is an optional component that provides advanced multi-channel handling of produced output. It relies on an installed KCM instance to provide Document Pack composition.

This section describes how to deploy Batch & Output Management using the `InstallOutputManagement` tool.

Overview

`InstallOutputManagement.exe` is a command-line tool intended to deploy KCM Batch & Output Management (B&OM). You can use the `InstallOutputManagement` tool to implement the following scenarios:

- Deploy Batch & Output Management as a standalone component on a server where no KCM installation is present. To implement this scenario, run `InstallOutputManagement.exe` that resides on the top-level of the KCM 5.5.0 package.
- Deploy Batch & Output Management side-by-side with a KCM installation on the same server. In this case, the two installations have their own folders, registry entries, and installation and management tools. To implement this scenario, run `InstallOutputManagement.exe` that resides in `<deploy root>\KCM\Programs\<version>\Management\Output Management`.

The tool works as follows:

1. It checks the requirements needed for the installation.

   Note If the check fails, the installation quits with no changes to your environment.

2. It installs B&OM and configures it based on the command line parameters.
3. It starts B&OM.

Since B&OM is an extension of KCM, it requires a KCM installation of the same version to be present in the organization. After installing B&OM, you should configure its connection to that KCM installation as described in the "Post-installation steps" section below.

Before you start

This section lists the prerequisites to check and the steps you need to take before the installation.

1. Prepare the databases. Batch & Output Management requires two databases:
   - One database for the B&OM Repository

- One database for the runtime information

These can be either SQL Server or Oracle databases, but must both be of the same DBMS.

> **Note** To work with Oracle databases, Oracle 32-bit client is required.

You need to create these databases manually, with a case-insensitive collation. Before starting the installation, make sure the databases are empty.

2. Plan for the user accounts that will be used for database access. For SQL Server databases, consider the following:
   - The user accounts for database access are specified with the `OutputManagement!RepositoryDatabaseUser` and `OutputManagement!RuntimeDatabaseUser` deployment parameters.
   - These users must have the `database owner` role and `ALTER_TRACE` privilege for both databases.

   For Oracle databases, consider the following:
   - The user accounts for database access are specified with the `OutputManagement!RepositoryDatabaseUser` and `OutputManagement!RuntimeDatabaseUser` deployment parameters.
   - They must have the following privileges:
     - `CREATE SESSION`
     - `CREATE TABLE`
     - `ALTER SESSION`
     - `CREATE VIEW`
     - `CREATE PROCEDURE`
     - `CREATE SEQUENCE`
   - In addition, the user account for the B&OM Repository database (specified with the `OutputManagement!RepositoryDatabaseUser` parameter) must have the following privileges:
     - `CREATE TYPE`
     - `CREATE TRIGGER`
     - `EXECUTE ON DBMS_LOCK`
     - `EXECUTE ON DBMS_LOB`
     - `EXECUTE ON DBMS_OUTPUT`
     - `EXECUTE ON CTX_DDL`
3. Plan for the account that will be used to run B&OM services.
   - This account must be included in the local *Administrators* group on the target server.
   - This account requires *Logon as a service* right.
4. The account you will use to run `InstallOutputManagement.exe` must be included in the local *Administrators* group on the target server.
5. If you plan to deploy multiple B&OM servers, you need to install your first B&OM server as *primary* and configure its databases and storage folder. Then you can deploy *secondary* B&OM servers that will share the databases and storage folder with the existing (primary) B&OM installation.

Note The storage folder is used by B&OM to store intermediate files for requests that are being processed.

Parameters

Parameter	Required / Optional	Description
`Deploy!RootPath`	Required	The root drive or folder in which the B&OM installation will be created. In the all-in-one scenario, KCM can be installed in the `<root path>\KCM` folder, and B&OM in the `<root path>\KCM_OM` folder.
`Services!User`	Required	The user account for B&OM Windows services. • This user must be a local *Administrator*. • This user requires *Logon as a service* right. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example**: `.\AdminUser`
`Services!Password`	Required	The password for the B&OM services user account.
`OutputManagement!Secondary`	Optional	Boolean flag that defines whether this B&OM installation is *secondary*, sharing the databases and storage folder with an existing (primary) B&OM installation. • If set to `True`, this B&OM installation is configured as *secondary*. This type of installation requires manual post-installation steps described later in this guide. • If set to `False`, this B&OM installation is configured as *primary*, having its own databases and storage folder. If omitted, defaults to `False`. If you are installing your first B&OM server, make sure it is installed as primary, with this parameter set to `False` or omitted.
`OutputManagement!DatabaseType`	Required	The type of the database to be used by B&OM. Specify `SQLServer` or `Oracle`.

Parameter	Required / Optional	Description
`OutputManagement!RepositoryDatabaseConnectionString`	Required	Specifies the connection string that Batch & Output Management uses to connect to its repository database. This cannot be an ODBC connection string. Depending on the `OutputManagement!DatabaseType` parameter, this must be either a valid Oracle or SQL Server connection string. • Primary B&OM installation must be connected to the new empty database. • Secondary B&OM installation must be connected to the same database as the primary installation. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below. For SQL Server database, consider the following: • Default authentication method is SQL Server authentication. • Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters.
`OutputManagement!RepositoryDatabaseUser`	Required for connection to Oracle database or to SQL Server database using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password - for example, for connection to SQL Server database using Windows authentication.	The database user account that B&OM server will use to connect to the B&OM Repository database. Make sure the account has the required roles and privileges, as described in the "Databases" section of this guide.

Parameter	Required / Optional	Description
`OutputManagement!RepositoryDatabasePassword`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	Specifies the password that Batch & Output Management uses to connect to its Repository database. **Note** The password cannot begin with a single quote or contain double quotes.
`OutputManagement!RuntimeDatabaseConnectionString`	Required	Specifies the connection string that B&OM uses to connect to its Runtime database. This cannot be an ODBC connection string. Depending on the `OutputManagement!DatabaseType` parameter, this must be either a valid Oracle or SQL Server connection string. • Primary B&OM installation must be connected to the new empty database. • Secondary B&OM installation must be connected to the same database as the primary installation. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below.

Parameter	Required / Optional	Description
`OutputManagement!RuntimeDatabaseUser`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	The database user account that B&OM server will use to connect to B&OM Runtime database
`OutputManagement!RuntimeDatabasePassword`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	Specifies the password that Batch & Output Management uses to connect to its Runtime database.

To view the list of parameters for the `InstallOutputManagement.exe`, launch it without any parameters.

Example

You may want to specify the `InstallOutputManagement` parameters and run the tool using a .cmd file. The following is an example command:

```
InstallOutputManagement ^
  Deploy!RootPath=D:\ ^
  Services!User=enterprise\appadmin ^
  Services!Password=entlocal1 ^
```

```
 OutputManagement!DatabaseType=SQLServer ^
 OutputManagement!RepositoryDatabaseConnectionString=”Server=sqlsrv02\SQL2017;
Database=OM_5.5_Repo;” ^
 OutputManagement!RepositoryDatabaseUser=kcm ^
 OutputManagement!RepositoryDatabasePassword=entkcm_1 ^
 OutputManagement!RuntimeDatabaseConnectionString=”Server=sqlsrv02\SQL2017;
Database=OM_5.5_RT;” ^
 OutputManagement!RuntimeDatabaseUser=kcm ^
 OutputManagement!RuntimeDatabasePassword=entkcm_1
```

Notes on usage

- The order of parameters is irrelevant.
- B&OM can be installed on the same machine as the KCM server.
- If the installation fails, then the unfinished deployment must be removed manually using the `UninstallOutputManagement.exe` tool described in the Uninstall Batch & Output Management section of this guide.
- Even the installation fails, the `KCM_OM` folder will be created anyway. This happens because in some cases, the `Uninstall` folder is added before the preflight check. The `KCM_OM` folder prevents further installation attempts, so it must be manually deleted before trying to install B&OM again.

Post-installation steps

After installing B&OM, configure its connection to the KCM instance that performs document composition:

1. In KCM Studio, go to the **Administration** tab, click **System Administration**.
2. Click **System Object**.
3. Locate the **KCM/Core** and **KCM/Repository** sections and adjust the respective **Host** and **Port** properties.

Note If you have KCM and B&OM installed on one server, **Host** is set to `localhost`.

Then configure the referenced KCM/Core to accept remote connections:

1. Open the KCM/Core Administrator of the KCM instance that is referenced by B&OM.
2. Go to the **DP Manager** tab.
3. Enable remote access in the **Security** section.

Prepare to run 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.

Distributed B&OM installation

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. A distributed B&OM setup shares a single configuration amongst its installations, as well as the storage folder. In order to create such a setup, you need to set up a primary installation that enables you to share the configuration and storage folder, and then set up the secondary installations.

Create a primary B&OM installation

Each distributed B&OM setup shares a single configuration amongst its installations, as well as the storage folder. In order to create such a setup, you first need to setup a primary installation to share configuration and storage folder.

To create a primary installation, do the following:

1. Run `InstallOutputManagement.exe` with the parameter `OutputManagement!Secondary=False` to create a primary installation. During the installation, make sure the database host is referred to either by computer name or IP address. This concerns the `OutputManagement!RepositoryDatabaseConnectionString` and `OutputManagement!RuntimeDatabaseConnectionString` parameters.
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. Configure the KCM/Core and KCM/Repository host that you want several B&OM processes to use on both primary and secondary computers. You can choose to have them all share a single host that has KCM/Core installed, or have then all refer to the KCM/Core installation that each B&OM installation has installed locally.
   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 **Host** properties, if necessary. Specify `localhost` if you want each B&OM installation to refer to the KCM/Core installation on its own server.
   c. If all B&OM installation share a single KCM/Core installation, configure it to accept remote connections. In KCM/Core Administrator, go to the **DP Manager** tab, and enable remote access to the **Security** section. Make sure the changes are saved and applied.

Add secondary B&OM installations

Each secondary B&OM installation uses the databases and the storage folder of the B&OM primary installation, instead of having its own databases and storage folder.

To add one or more secondary installations, do the following:

1. Run `InstallOutputManagement.exe` with the parameter `OutputManagement!Secondary=True` to create a secondary installation.
   a. Use the `OutputManagement!RepositoryDatabaseConnectionString` parameter to refer to B&OM Repository database of the primary installation.
   b. Use the `OutputManagement!RuntimeDatabaseConnectionString` parameter to refer to B&OM Runtime database of the primary installation.

      Note These databases will be verified for use by the secondary installation.

2. On the primary B&OM installation, navigate to `<deploy root>\KCM_OM\Programs\<version>\Output Management` and copy the `Repository.Config` and `UserRepository.Config` files. Paste the files into the same folder on the B&OM secondary installation.
3. 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 This 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.

Custom 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_OM\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 `AddOutputManagementService.exe` tool. It resides in: `<deploy root>\KCM_OM\Programs\<version>\Management`.

The tool works as follows:

- A new executable for the new service is created in the `<deploy root>\KCM_OM\Programs\<version>\Output Management` folder, with the following name: `CCM_OM_SVC_<Identifier>.exe`.
- A configuration file is created next to the service. If required, you can customize it.
- A new service is automatically started and does not require a restart of B&OM or any components of KCM installation.

- 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.

New services added using this tool will execute selected processes in a given system. They depend on the standard KCM B&OM Core service to run. These services can be independently managed using the Service management tools.

Parameters

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

Parameter	Required / Optional	Description
`Service!Identifier`	Required	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.
`Service!DisplayName`	Optional	Overrides the default generated display name of the new Windows service. Default is `CCM Output Management - <Identifier> - <Version>`.
`Service!Description`	Optional	Overrides the description of the new Windows service generated by default. Default is the same value as the `Service!DisplayName` parameter has.
`Service!User`	Required	The user account to be used for the new service. This must be the same user as specified in the original KCM B&OM deployment, and must consequently have the *Logon as a service* rights. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter value.
`Service!Password`	Required	The password for the service user.
`System!Name`	Required	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.
`Process!Names`	Optional	A colon-separated list of names for KCM B&OM processes for the new service to execute. You can view the names in the **Object Viewer** pane in KCM Studio. **Note** The separator is the regular colon character (':'), not the semicolon character (';'). If this parameter is omitted, then `AliasTable!Names` parameter must be specified. See its description below for details.

Parameter	Required / Optional	Description
`AliasTable!Names`	Optional	A colon-separated list of names for KCM B&OM alias tables, which should contain one or more KCM B&OM processes. 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. **Note** The separator is the regular colon character (':'), not the semicolon character (';'). If this parameter is omitted, then `Process!Names` parameters must be specified. See its description above for details.

Example

1. Plan for the identifier that will represent the new service. See the `Service!Identifier` parameter description in the table above for more information on identifiers.
2. Navigate to: `<deploy root>\KCM_OM\Programs\<version>\Management`.
3. Run the `AddOutputManagementService.exe` tool. Use the command line as shown below, substituting the example parameter values with your actual values.

```
.\AddOutputManagementService.exe 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 tool `ListOutputManagementServices.exe`.

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_OM\Programs\<version>\Management`
2. Run the tool.

Remove custom service hosts

You can remove custom KCM B&OM service hosts using the `RemoveOutputManagementService.exe` tool. It resides in: `<deploy root>\KCM_OM\Programs\<version>\Management`.

Parameters

The tool has one parameter.

Parameter	Required / Optional	Description
`Service!Identifier`	Required	Identifier of the service to remove. A list of services can be retrieved using the `ListOutputManagementServices.exe` tool (see the previous section).

Notes on usage

- The removed service is automatically stopped. Removal does not require a restart of B&OM or other parts of KCM.
- After the service is removed, some custom configuration settings applied to its `.config` file are lost (for example, log level settings). Service log files are preserved.

Example

1. Navigate to: `<deploy root>\KCM_OM\Programs\<version>\Management`.
2. Run the `RemoveOutputManagementService.exe` tool. Use the command line as shown below, substituting the example parameter values with your actual values.

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

Export B&OM content

To facilitate one-time transfer between different installations of KCM B&OM and upgrade to a newer KCM B&OM version, you can perform B&OM export. The exported content consists of B&OM components and all their objects. Optionally, you can configure the export to only include a specific component, or to only export data from the specific content database.

To perform export, use the `ExportOutputManagementContent.exe` tool. It resides in: `<deploy root>\KCM_OM\Programs\<version>\Management`.

Note This tool is not an alternative to the regular Transfer Jobs option accessible from KCM Studio.

Parameters

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

Parameter	Required / Optional	Description
`Destination!Path`	Required	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.

Parameter	Required / Optional	Description
`System!Name`	Optional	The name of a specific system to export. If this parameter is specified, the export will only contain this system. **Note** You cannot use it with the `Legacy!Export` parameter. By default, no filter is applied.

Note Also, there are `RepositoryAlias!Name` and `Legacy!Export` parameters that only refer to the legacy system (versions 5.0.0 and 5.1.0). `RepositoryAlias!Name` is the name of a specific content database alias to export, `Legacy!Export` specifies whether to create the export from KCM B&OM 5.0.0 or 5.1.0 on the machine.

Notes on usage

The `ExportOutputManagementContent.exe` tool does not require KCM B&OM to be running at the time of the export.

Example

1. Navigate to: `<deploy root>\KCM_OM\Programs\<version>\Management`.
2. Select a location to save the export. This location must either not yet exist, or be empty.
3. Run the `ExportOutputManagementContent.exe` tool.
   Use the command line as shown below, substituting the example parameters values with your actual values.

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

Import B&OM content

To import the KCM Batch & Output Management content, use the `ImportOutputManagementContent.exe` tool. It resides in: `<deploy root>\KCM_OM\Programs\<version>\Management`.

You can import the following content:

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

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

Parameters

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

Parameter	Required / Optional	Description
`Source!Path`	Required	The path to a folder containing an export.
`System!Name`	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, it is assumed that the directory contains a full-systems configuration export as a result of running the `ExportOutputManagementContent.exe` tool.

Example

1. Navigate to: `<deploy root>\KCM_OM\Programs\<version>\Management`.
2. Run the `ImportOutputManagementContent.exe` tool.

   Use the command line as shown below, substituting the example parameters values with your actual values.

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

Chapter 3

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. At this step, the Contract Manager 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

- Due to Microsoft limitations, Kofax Communications Manager does not support installation of Microsoft Word on Docker containers. Thus, KCM instances on Docker do not have Word Processor support, and certain KCM/KCM Core scripting features are not available. This includes printing Word documents, running macros, or converting to PDF using Word. For the latter, KCM will automatically use Rendition instead.
- 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. Creation of an image that contains an installation of the Contract Manager

2. Creation of a container from the image. In this step the software will be get fully configured and brought into a running state. These actions are managed by a Docker specific entry point process

Both steps will be described in delail below.

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.

The AddContractManage.exe installer on Docker

The template Docker file uses the `AddContractManager.exe` to install the KCM Contract Manager on Docker. This proceeds is similar to a non-Docker installation, except that for a Docker installation the value of the `Deploy!Docker` parameter of `AddContractManager` must be set to `True`.

In Docker mode the installer predefines a number of configurations that are necessary for Docker. This can be used to delay certain configurations to be performed later, upon creation of the container via the `DockerStartContractManager.exe` entry point process.

Consequently, the `AddContractManager` tool will use less parameters in Docker Mode. Only the following parameters are required:

- `Deploy!RootPath`

- `Java!Home`
- `WebServer!ServiceUser`
- `WebServer!ServicePassword`
- `ContractManager!SharedResourceLocation`

The `AddContractManager` parameters are listed in the "Add Contract Manager" section above.

The DockerStartContractManager.exe entry point process

The `DockerStartContractManager.exe` entry point can be found at: `InstallServer\Software\KCMTools\DockerStartContractManager.exe`

Alternatively, after installation on a regular server it is located at: `<deploy root>\KCM\Programs\<version>\Management\Contract Manager`

This entry point has a number of parameters that should be passed upon container creation.

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

- `ContractManager!UseAuthentication` (required for DockerStartContractManager.exe)
- `ContractManager!UseSSL` (required for DockerStartContractManager.exe)
- `ContractManager!KeystoreFile`
- `ContractManager!KeystorePassword`
- `ContractManager!UseLDAP` (required for DockerStartContractManager.exe)
- `ContractManager!LDAPPropertiesFile`
- `ContractManager!ImportConfigurationFile`
- `ContractManager!ImportInfrastructure`

These are all 'delayed' `AddContractManager` parameters. See the "Add Contract Manager" section of this guide for details.

Note AddContractManager.exe
`AddContractManager.exe`, `DockerStartContractManager.exe` will require only some of the setup parameters.

Entry point parameters 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.

The entry point parameters must either be passed directly to the entry point, or alternatively, be specified as an 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.

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.*

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
```

Install Kofax Communications Manager instances

The installation of KCM instances on Docker includes two steps:

1. The creation of an image.
2. The creation of a container from the image. In this step the software will get fully configured, and be brought into a running state. These actions are managed by a Docker specific entry point process.

Both steps will be detailed below.

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.

The AddInstance.exe installer on Docker

The template Docker file uses the `AddInstance.exe` to install the KCM Instance on Docker. This is similar to a non-Docker installation, except that for a Docker installation the value of the `Deploy!Docker` parameter of `AddInstance` must be set to `True`.

In Docker mode, the installer predefines a number of configurations that are necessary for Docker. This can be used to delay certain configurations that will be set up later, upon creation of the container, via the `DockerStartInstance.exe` entry point process.

Consequently, the `AddInstance` installer will accept less parameters in Docker Mode. Only the following parameters are required:

- `Deploy!RootPath`
- `Java!Home`
- `Webserver!ServiceUser`
- `Webserver!ServicePassword`
- `Services!User`
- `Services!Password`
- `Instance!NumberOfDPs`

For a detailed description of these parameters, see the "Add KCM Instance" section above.

The DockerStartInstance.exe entry point process

The DockerStartInstance.exe entry point can be found at: `InstallServer\Software\KCMTools\DockerStartInstance.exe`

Or alternatively, after installation on a regular server: `<deploy root>\KCM\Programs\<version>\Management\Instance`

This entry point has a number of parameters that should be passed upon container creation. This will be discussed below.

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.

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

   - `Services!Password`
   - `Instance!DatabaseType`
   - `Instance!DatabaseMode`
   - `Instance!DatabaseConnectionString`
   - `Instance!DatabaseUser`
   - `Instance!DatabasePassword`
   - `Instance!ExampleProject`
   - `Instance!DesignerAdministratorAccountPassword`
   - `Instance!LicenseFile`
   - `Instance!ImportConfigurationFile`
   - `Instance!ImportInfrastructure`
   - `Registration!Partner`
   - `Registration!Customer`
   - `Registration!ContractManagerHost`
   - `Registration!InstanceHost`
   - `Registration!ExampleWebAppName`
   - `Registration!NewContract`
   - `Registration!Force`

   These are mostly 'delayed' `AddContractInstance` parameters and the `StartInstance` parameters. See the "Add KCM Instance" and "Start KCM Instance" sections for parameter descriptions.

   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 must either be passed directly to the entry point, or alternatively, be specified as an 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.

3. 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 `Instance!DatabaseMode` 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, provided that you change `Registration!NewContract` to `False`, since the contract already exists at that point. Furthermore, if `Registration!Force` is set to `True`, the second instance overwrites the registration of the first one.

See the following examples of commands with all options included.

Example 1

```
docker run -d

--network nat

--name instance01

--ip 123.45.67.98

instance

Instance!DatabaseType=sqlserver

Instance!DatabaseMode=New

Instance!DatabaseConnectionString="Driver=SQL
Server;Server=123.45.67.88;Database=DB;"

Instance!DatabaseUser=ccm

Instance!DatabasePassword=<pwd>

Services!Password=<pwd>
```

```
instance!DesignerAdministratorAccountPassword=0fec6d3d54b8df23067a7f7e79db8b0f

Registration!Partner=CCM

Registration!Customer=Docker01

Registration!ContractManagerHost=123.34.67.89

Registration!InstanceHost=instance01

Registration!ExampleWebAppName=start01

Registration!NewContract=true
```

Example 2

```
docker run -d

--network nat

--name instance01

--ip 123.45.67.98

--env Instance!DatabaseType=sqlserver

–-env Instance!DatabaseMode=New

--env instance!DatabaseConnectionString="Driver=SQL
Server;Server=123.45.67.88;Database=DB;"

--env Instance!DatabaseUser=ccm

--env Instance!DatabasePassword=<pwd>

--env Instance!
DesignerAdministratorAccountPassword=0fec6d3d54b8df23067a7f7e79db8b0f

--env Registration!Partner=CCM

--env Registration!Customer=Docker01

--env Registration!ContractManagerHost=123.45.67.89

--env Registration!InstanceHost=instance01

–env Registration!ExampleWebAppName=start01

–env Registration!NewContract=true instance
```

Chapter 4

Post-installation steps

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_5.5.0_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_5.5.0_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>\KCM\Programs\<version>\Instance_<num>\webapps\ccm_designer-<instance num>\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>\KCM\Programs\<version>\Instance_<num>\webapps\ccm_designer-<instance num>\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 Core Administrator

For KCM Core Administrator, extract the `print` folder content from the English documentation .zip file to:

`<deploy root>\KCM\Documentation\<version>\`**`en`**

Offline documentation for KCM Studio

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

`<deploy root>\KCM_OM\Documentation\<version>\`**`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 folder of Designer for Windows>\Documentation\`**`en`**

For more information on the KCM Designer for Windows installation and default location, see Install KCM Designer for Windows.

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 helps you to test the installation is available when the `Instance!ExampleWebAppName` parameter was specified. 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/<example web application name>/home.html
- http://<kcm server>:8081/ /<example web application name>//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, 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. For details, see "Change your password" in *Help for KCM Designer*.

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.

> **Note** If you use Internet Explorer 11, 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.

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

Remove the example web application

We recommend that you do not deploy the example web application in production environments. Normally, this means that you would not specify the `Instance!ExampleWebAppName` parameter when adding KCM instances to a production environment.

However, if you need to remove the example web application from an instance that had it installed before, follow these steps:

1. Navigate to `<deploy root>\KCM\Programs\<version>\Management\Instance` and run the `SetInstanceRegistration.exe` tool. Fill out all the relevant information for the Registration parameters but omit `Registration!ExampleWebAppName`. See the information for the `SetInstanceRegistration` tool in the *Kofax Communications Manager Getting Started Guide*.

2. For this change to be applied, stop the KCM instance using the `StopInstance` tool, if the instance was still running.
3. Start the instance using the `StartInstance` tool. Set `Registration!NewContract=False`.

If automatic registration is not possible or does not succeed, navigate to `<deploy root>\KCM\Programs\<version>\Management` and run the `RemoveExampleWebApp.exe` tool. Use the `Application!Name` parameter to specify the name of the example web application that needs to be removed.

Note `AddExampleWebApp` and `RemoveExampleWebApp` should only be used when automatic registration is not used at all or when it fails.

When automatic registration is used, `StartInstance` and `StopInstance` will add and remove the example web application according to the information that is registered upon adding the instance, and via `SetInstanceRegistration`.

This takes precedence over `AddExampleWebApp` and `RemoveExampleWebApp` whenever the instance is started or stopped.

Chapter 5

Initialize new KCM component using existing configuration

You can use the configuration of an existing KCM component to initialize a new KCM component, including KCM Docker containers. For that, perform the following steps:

1. Export the configuration of an existing KCM component into a single file.
2. For non-Docker Contract Manager, either pass this file as the `ContractManager!ImportConfigurationFile` parameter to `AddContractManager.exe`, or use `ImportContractManagerConfiguration.exe` to import the file after installation.
3. For non-Docker KCM instance, either pass this file as the `Instance!ImportConfigurationFile` parameter to `AddInstance.exe`, or use `ImportInstanceConfiguration.exe` to import the file after installation.
4. For Docker Contract Manager containers, have the `ContractManager!ImportConfigurationFile` parameter of the `DockerStartContractManager.exe` entry point refer to this file.
5. For Docker Instance containers, have the `Instance!ImportConfigurationFile` parameter of the `DockerStartInstance.exe` entry point refer to this file.

The configuration export 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. See the *Kofax Communications Manager Getting Started Guide* for details on configuration import and export.

Note A file can only be imported in a particular KCM version if has been created by the export tools that are included in that KCM version. That is, to import a configuration into KCM 5.5.0, you need to have exported that configuration with the KCM 5.5.0 export tools.

Chapter 6

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 7

Upgrade your installation

You can upgrade KCM versions 5.2.0 and later to KCM 5.5.0. The upgrade installs KCM 5.5.0 alongside your existing installation, using the same configuration that the current KCM was installed with.

Note the following when upgrading your KCM installation:

- Before upgrading, back up the databases that the current installation uses.
- If you have an older KCM installation, first upgrade it to 5.2.0. Then you can upgrade it to 5.5.0.
- Starting with KCM 5.5.0, KCM and KCM Batch & Output Management are separately upgraded, each with their own upgrade executable.
- You need to manually reinstall any custom additional language packs. We recommend that you execute the script `GetLanguagePack.ps1` of your KCM pre-5.5 installation prior to upgrading the product, update the custom language packs so they include new translations, and then install the language packs using the tool named `AddLanguagePack.exe`. For information on managing language packs, see the chapter "Manage language packs" in the Kofax Communications Manager Getting Started Guide.
- Upgrading KCM Batch & Output Management may require multiple manual steps. For more information, see the "B&OM upgrade" section below.

Upgrade distributed deployment

When KCM is installed on more than one computer, these should be upgraded in the following order:

1. First, upgrade the server that has the Contract Manager installed.
2. Then, upgrade the servers that have KCM instances installed that connect to this Contract Manager.
3. Finally, upgrade Batch & Output Management, starting with the primary installation, followed by the secondary installations.

KCM upgrade

To prepare for the upgrade of a KCM installation on a particular server, you need to do the following:

1. Create a new database for each KCM instance
2. Allocate enough disk space to accommodate KCM Repository content

Then you can start the upgrade by running the `Upgrade.exe` tool described in the next section.

The automated upgrade process will perform the following operations:

1. Export the configurations of the Contract Manager and KCM instances.

2. Deactivate and Stop the current installation.

3. If the current installation has a Contract Manager, install (that is, add) a new Contact Manager

4. For each KCM instance in the current installation, install (that is, add) a new KCM instance.

5. Import the configurations of the old Contract Manager and the old KCM instances into the new Contract Manager and new instances.

6. Transfer the core security configuration from the old KCM instances to the new ones.

7. Register the new KCM instances at the new Contract Manager according to the old registration that is included in the exported Contract Manager configuration.

8. Activate and Start the new Contract Manager and the new KCM instances.

Note Upgrading the server with the Contract Manager will result in an exported Contract Manager configuration that includes information about the KCM instances that were registered at that server. When upgrading pre-5.5 installations this configuration must be used to upgrade servers with remote instances, so that they know which contracts to register at that Contact Manager.

When the upgrade is completed, the 5.5.0 installation will be active and available for testing. The previously active installation will become disabled.

The upgrade process will copy the KCM Repository content to the new database. Any subsequent changes to the new KCM Repository content will not be reflected in the older version.

If the upgrade process encounters an error, and the old installation was active before the upgrade, the upgrade will rollback as follows:

- Uninstall the new KCM software
- Activate the old KCM installation
- Start the old KCM installation

This enables you to address the reported issue with minimal disruption to the existing installation. You can then try updating again once the issue has been addressed.

If the old installation was not active when an error gets reported by the upgrade, no automatic rollback is attempted. This makes it possible to investigate the partially installed new software installation, fix the issue, and either rollback, or perform the remaining upgrade steps manually.

Upgrade tool

The following is a description of the `Upgrade.exe` tool that you need to run for upgrading your KCM installation.

Overview

`Upgrade.exe` is a command-line tool designed to upgrade an existing KCM installation to version 5.5.0. Upgrade is supported for KCM 5.4.0, 5.3.1, 5.3.0, and 5.2.0.

Note This tool does not upgrade B&OM. Use `UpgradeOutputManagement.exe` for that purpose, as described later in this guide.

The tool works as follows:

1. It upgrades all KCM components on a local server to KCM 5.5.0.
2. Configuration of the existing KCM instances will be preserved, except for the following:
   - For each KCM instance, a new database will be used for the KCM Repository. However, the content of the old database will be transferred automatically to the new one.
   - For each KCM instance, the license file will be configured anew.
3. The Contract Manager configuration will be set as follows:
   - Authentication, SSL, and LDAP flags of the Contract Manager will be configured anew, along with their configuration files.
   - The upgrade will preserve the contracts with which the instances were associated on the Contract Manager. However, this registration information will be moved from the Contract Manager to the KCM instances.
   - KCM instances that were not registered on any Contract Manager will now require a *Partner* and a *Customer*.
4. After the upgrade, KCM 5.5.0 installation will be fully set up and started automatically.

Note Only KCM instances that were not associated with an active contract on the Contract Manager will not be started.

Before you start

Make sure the account used to run `Upgrade.exe` is included in the local *Administrators* group on the target server.

Parameters

Parameter	Required / Optional	Description
`Upgrade!From`	Required if the old KCM installation is not active.	The KCM version to upgrade. Enter version number in the format `Major.Minor` Example: `5.4`
`Deploy!RootPath`	Optional	The root drive or folder in which the upgraded KCM installation will be created. If omitted, defaults to the root path of the old KCM installation.
`Java!Home`	Optional	The folder in which Java is installed in your system, either as JRE or JDK. This Java installation will be used for the KCM web applications. If omitted, KCM will use the `JAVA_HOME` environment variable.

Parameter	Required / Optional	Description
`Services!User`	Required	The user account used for the installed services. - This user must be a member of the local *Administrators* group. - This user must have the *Logon as a service* rights. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example** `.\myuser`
`Services!Password`	Required	The password for the services user.
`Webserver!ServiceUser`	Required	The user account to run the KCM web applications. This account requires the following: - *Logon as a service* right. - *Read & execute* permission on the Tomcat installation folder. - If you plan to use SSL connections, then also assign *Read* permission on the SSL keystore file. See `ContractManager!UseSSL` description for details. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example** `.\myuserws`
`Webserver!ServicePassword`	Required	The password belonging to the user provided through `Webserver!ServiceUser`.
`ContractManager!UseAuthentication`	Required*	Boolean flag that specifies whether to enable authentication for executing SOAP calls on the Contract Manager. - If set to `True`, then the calling application needs to supply a user name and a password using an HTTP Basic Authentication header. **Note** Authentication requires secure connections to be enabled. - If set to `False`, anyone with access to the Contract Manager can execute any SOAP call on its interface without additional authentication.

Parameter	Required / Optional	Description
ContractManager!UseSSL	Required*	Boolean flag that specifies whether the integrating applications need to use a secure protocol for connections to the Contract Manager. • If set to True, connections are secure, using the HTTPS protocol (HTTP over SSL). • If set to False, connections are not secure, using the HTTP protocol.
ContractManager! KeyStoreFile	Required* if ContractManager! UseSSL was set to True	The location of the certificate file (.pfx or .p12).
ContractManager! KeyStorePassword	Required* if ContractManager! UseSSL was set to True	The password of the certificate file (.pfx or .p12).
ContractManager!UseLDAP	Required*	Boolean flag that specifies whether users can login to KCM Designer using LDAP authentication. • If set to True, users can login to KCM Designer with their LDAP user accounts. **Note** LDAP authentication requires secure connections to be enabled. • If set to False, users can log in to KCM Designer with internal user accounts that are managed by the KCM Repository server.
ContractManager! LDAPPropertiesFile	Required if ContractManager! UseLDAP was set to True	The configuration file for LDAP authentication. See the “LDAP properties file” section of the Kofax Communications Manager Getting Started Guide for details.
ContractManager! SharedResourceLocation	Optional	Specifies the location of a folder where the shared objects for import and export, such as project files and reference projects, are stored. If omitted, defaults to <deploy root>\KCM\Work \<version>\SharedResources
OldContractManager! DatabaseUser	Required* only when upgrading from KCM 5.2.0, 5.3.1, or 5.3.0.	The name of the database user that the old KCM installation used for accessing the Contract Manager database.
OldContractManager! DatabasePassword	Required* only when upgrading from KCM 5.2.0, 5.3.1, or 5.3.0.	The password of the database user that the old KCM installation used for accessing the Contract Manager database.
Instance!DatabaseType, Instance!DatabaseMode, Instance!DatabaseUser, Instance!DatabasePassword, Instance!LicenseFile	Optional*	Shared default values for the Instance#n! parameters of the same name. **Note** Instance!DatabaseMode defaults to Overwrite when omitted.

Parameter	Required / Optional	Description
`Instance#n!DatabaseType`	Optional*	The type of the database to be used by the KCM Repository of the instance `#n`. Specify `SQLServer` or `Oracle`. If omitted, defaults to the value set by `Instance!DatabaseType`.
`Instance#n!DatabaseMode`	Optional*	Allowed values are `Check` and `Overwrite`. • If set to Check, the *partner* and *customer* stored in the old repository database of the KCM instance with number **'n'** are expected to be the same as the *partner* and *customer* of the upgraded instance. If not, the upgrade will fail. • If set to `Overwrite`, the *partner* and *customer* that are stored in the old repository database of the KCM instance with number **'n'** are expected to be different from the *partner* and customer of the upgraded instance (and will be overwritten in the new repository database). If not, the upgrade will fail. **Note** *Partner* and *customer* are stored in the repository database for the on-premise deployment starting with KCM 5.5.0, and for Docker deployment - starting with KCM 5.4.0. If omitted, defaults to the value set by `Instance!DatabaseMode`.
`Instance#n!DatabaseConnectionString`	Required	The ODBC connection string to be used by the KCM Repository server of the instance #n to connect to the database. Make sure that the database exists on the database server and is empty. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below. For SQL Server database, consider the following: • Default authentication method is SQL Server authentication. • Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters.

Parameter	Required / Optional	Description
`Instance#n!DatabaseUser`	Optional*	The database user account that KCM Repository Server of the instance #n will use to connect to the database. Required to connect to Oracle database or to SQL Server database using SQL Server authentication. In the latter case, this user must have the *database owner* role assigned and use *dbo* schema. If omitted, defaults to the value set by `Instance!DatabaseUser`
`Instance#n!DatabasePassword`	Optional*	The password that KCM Repository Server of the instance #n needs to connect to the database. If omitted, defaults to the value set by `Instance!DatabasePassword`.
`Instance#n!LicenseFile`	Optional*	The location of the .xml file with the license information. If omitted, defaults to the value set by `Instance!LicenseFile`.
`Instance#n!NumberOfDPs`	Optional*	The number of Document Processors to create for the instance, between 1 and 99. When using a capacity-based license, this number must not exceed the maximum set in the license. For details, see the License section in this guide. If omitted, default value will be used: • For a capacity-based license — set to maximum allowed. • For a volume-based license — set to 4.
`OldInstance!DatabaseUser, OldInstance!DatabasePassword`	Optional*	Shared default values for the `OldInstance#n!` parameters of the same name.
`OldInstance#n!DatabaseUser`	Optional*	The database user that the **old** KCM Repository of instance '**n**' uses to connect to its database. If omitted, defaults to the value set by `OldInstance!DatabaseUser`.
`OldInstance#n!DatabasePassword`	Optional*	The password that the **old** KCM Repository of instance '**n**' uses to connect to its database. If omitted, defaults to the value set by `OldInstance!DatabasePassword`.

Parameter	Required / Optional	Description
`Registration!` `ContractManagerLocation`	Required*	Specifies the Contract Manager location when registering the new KCM instances: • If set to `Local`, the new KCM instances will be registered on the Contract Manager on the local server. • If set to `Remote`, the old KCM instances will be registered on the Contract Manager on a remote server. • I f set to `None`, the old KCM instances will not be registered on a Contract Manager.
`Registration!` `ContractManagerHost`	Optional*	The host of the Contract Manager on which the old KCM instances were registered. This is only needed when the Contract Manager is remote, but can also be used to explicitly set or override the host when the Contract Manager is local. If the old installation has a local Contract Manager, defaults to *localhost*.
`Registration!ContractManagerConfigurationFile`	Optional*	The configuration export file of the Contract Manager, used for transferring registration information to the instances. This is only needed when the Contract Manager is remote.
`Registration#n!Partner`	Optional*	The *partner* with which the KCM instance with number '**n**' will be associated. This is only needed when the instance is not registered on a Contract Manager.
`Registration#n!Customer`	Optional*	The *customer* with which the KCM instance with number '**n**' will be associated. This is only needed when the instance is not registered on a Contract Manager.
`Registration#n!ExampleWebAppName`	Optional	The name of the example web application that will be created for the instance number '**n**'. This information will not to be taken over from the old installation. If omitted, the example web application will not be created.

Parameters marked with (*) are available in the specific cases only:

- The `ContractManager!` parameters only exist if the old installation has a Contract Manager.
- The `OldContractManager!` parameters only exist if the old installation has a Contract Manager, and its version is 5.2.0, 5.3.1 or 5.3.0.
- The `Instance!` parameters only exist if the old installation has KCM instances.
- The `Instance#n!` parameters only exist if the old installation has a KCM instance with number 'n'.
- The `OldInstance!` parameters only exist if the old installation has KCM instances, and its version is 5.4.0.

- The `OldInstance#n!` parameters only exist if the old installation has a KCM instance with number 'n', and its version is 5.4.0.
- The `Registration!` parameters only exist if the old installation has KCM instances, and its version is 5.2.0, 5.3.1, 5.3.0, or 5.4.0.
- The `Registration#n!` parameters only exist if the old installation has a KCM instance with number 'n', and its version is 5.2.0, 5.3.0, or 5.4.0.

Notes on usage

1. When transferring registration from the Contract Manager to the KCM instances, the configuration export file of the Contract Manager will be used. If an instance cannot be located in the export file, then the upgrade will continue, but the instance will not have its registration configured, and will also not be started.
2. If `Upgrade.exe` fails, an automatic rollback to the old installation will be attempted.

Upgrade from KCM 5.4.0 or earlier

An upgrade from KCM 5.4.0 or earlier skips contract types and interfaces (and other configuration) that are not in use by an active contract.

If you need to have them in the upgraded installation, you will need to define these manually again.

B&OM upgrade

Upgrading B&OM is not a fully automated process and requires a number of manual steps described below.

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

- **Upgrades need to be done during non-production hours**

  Runtime information is not maintained during the upgrade. Upgrades must be performed during non-production hours. The following steps need to be taken:
  - Ensure that no new requests are submitted to B&OM.
  - Existing jobs must be allowed to finish. Jobs in error must be cancelled or resubmitted until there are no remaining active jobs, so no remaining work is lost.
  - Perform the upgrade, as described below.
  - Resume submitting requests.

- **The primary installation must be upgraded before the secondary installations**

  Upgrading a secondary B&OM installation requires a manual copy of several configuration files of the upgraded primary installation. Assuming that the primary installation has already been upgraded, take the following steps on the secondary installation:

  1. Extract the KCM package on the secondary B&OM server. Go to the extracted `InstallServer` folder, and under the `\InstallServer\Software` subfolder create a new subfolder named `Output Management Primary`.
  2. On the upgraded primary B&OM server, navigate to `<deploy root>\KCM\Programs\5.5\Output Management` and copy the `Repository.Config` and `UserRepository.Config` files. Paste these files in the folder created in the previous step.
  3. Upgrade the secondary B&OM server.

- **Custom processes need to be transferred manually**

  The upgrade procedure will create two systems in the upgraded KCM B&OM installation:

  1. One system will contain the existing client processes, but without their history. This system refers to the old B&OM Runtime database.

     Both the existing client processes in this system, as well as the existing B&OM Runtime database become incompatible with the upgraded installation and cannot be used by the upgraded installation.
  2. Another system will contain the standard processes and objects of the new KCM version. It uses the new B&OM Runtime database that you created before the upgrade.

     This new B&OM Runtime database is compatible with the upgraded installation, but no runtime information of the old system will be transferred to it.

  Users must transfer their custom processes from system #1 to system #2 manually and start using system #2 after the upgrade.

- **Paths must be manually updated**

  System #1 will contain the original paths. System #2 will contain default paths. Paths to files or folders on the file system, such as the path to the Compart license file on the `OdinConversion` component will need to be set manually, similar to a new installation.

- **Compart configuration files must be manually transferred to the new version**

  Compart configuration files, such as license files, must be moved to the new installation manually.

- **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` tools.

- **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.

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

  To remove the B&OM services from an inactive installation, use the uninstallation tool. The `ListOutputManagementServices`, `AddOutputManagementService`, and `RemoveOutputManagementService` tools always affect the active installation. For example, the `RemoveOutputManagementService.exe` tool removes the B&OM services from the active installation. For more information, see Switch to old installation and Uninstallation.

Upgrade tool for B&OM

The following is a description of the `UpgradeOutputManagement.exe` tool that you can use to upgrade the KCM B&OM installation.

Overview

`UpgradeOutputManagement.exe` is a command-line tool intended to upgrade the Batch & Output Management (B&OM) component of a KCM 5.2.0, 5.3.1, 5.3.0, or 5.4.0 installation into a Batch & Output Management 5.5.0 installation.

Note Only B&OM will be upgraded; KCM components will not be affected.

The `UpgradeOutputManagement` tool can be found:

- in `<deploy root>\KCM\Programs\<version>\Management\Output Management` — if B&OM resides together with KCM, and KCM has been upgraded.
- in the `InstallServer` folder extracted from the installation package — if B&OM resides separately, or KCM was not upgraded.

`UpgradeOutputManagement.exe` works as follows:

1. It upgrades B&OM and configures it based on the command line parameters.
2. It starts B&OM.

The resulting B&OM installation is separate from any KCM installation. It has its own installation folder, registry entries, and installation and management tools.

Before you start

This section lists the prerequisites to check and the steps you need to take before running the tool.

1. Prepare the databases. Batch & Output Management requires two databases:
   - One database for the B&OM Repository.
   - One database for the runtime information.

   These can be either SQL Server or Oracle databases, but must both be of the same DBMS.

   Note To work with Oracle databases, Oracle 32-bit client is required.

   You need to create these databases manually, with a case-insensitive collation. Before starting the upgrade, make sure the databases are empty.

2. Plan for the user accounts that will be used for database access. For SQL Server databases, consider the following:
   - The user accounts for database access are specified with the `OutputManagement!RepositoryDatabaseUser` and `OutputManagement!RuntimeDatabaseUser` upgrade parameters.
   - These users must have the `database owner` role and `ALTER_TRACE` privilege for both databases.

   For Oracle databases, consider the following:
   - The user accounts for database access are specified with the `OutputManagement!RepositoryDatabaseUser` and `OutputManagement!RuntimeDatabaseUser` upgrade parameters.
   - They must have the following privileges:
     - `CREATE SESSION`
     - `CREATE TABLE`
     - `ALTER SESSION`
     - `CREATE VIEW`
     - `CREATE PROCEDURE`
     - `CREATE SEQUENCE`
   - In addition, the user account for B&OM repository database specified with the `OutputManagement!RepositoryDatabaseUser` parameter must have the following privileges:
     - `CREATE TYPE`
     - `CREATE TRIGGER`
     - `EXECUTE ON DBMS_LOCK`
     - `EXECUTE ON DBMS_LOB`
     - `EXECUTE ON DBMS_OUTPUT`
     - `EXECUTE ON CTX_DDL`
3. Plan for the account that will be used to run B&OM services.
   - This account must be included in the local *Administrators* group on the target server.
   - This account requires *Logon as a service* right.
4. The account you will use to run `UpgradeOutputManagement.exe` must be included in the local *Administrators* group on the target server.
5. If you have multiple B&OM servers, you need to upgrade your primary B&OM server first. Then you can upgrade the secondary B&OM servers that will share the databases and storage folder with the upgraded primary B&OM server.

Parameters

To view the list of parameters for `UpgradeOutputManagement.exe`, launch it without any parameters.

Parameter	Required / Optional	Description
`Upgrade!From`	Required	The KCM version to upgrade. Enter version number in the format `Major.Minor` Example: `Upgrade!From=5.4`
`Services!User`	Required	The user account for B&OM Windows services. • This user must be a local *Administrator*. • This user requires *Logon as a service* right. You can specify a domain user account or a local user account. When specifying a local user, place a period and backslash (.\) at the beginning of the parameter. **Example:** `.\AdminUser`
`Services!Password`	Required	The password for the B&OM services user account.
`OutputManagement!DatabaseType`	Required	The type of the database used by B&OM - `SQLServer` or `Oracle`.
`OutputManagement!RepositoryDatabaseConnectionString`	Required	Specifies the connection string that Batch & Output Management uses to connect to its repository database. This cannot be an ODBC connection string. Depending on the `OutputManagement!DatabaseType` parameter, this must be either a valid Oracle or SQL Server connection string. • Primary B&OM installation must be connected to the new empty database. • Secondary B&OM installation must be connected to the same database as the primary installation. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below. For SQL Server database, consider the following: • Default authentication method is SQL Server authentication. • Windows authentication is also supported. To enable it, add '`Integrated Security=True`' to the connection string and leave out the database user and password parameters.

Parameter	Required / Optional	Description
`OutputManagement!` `RepositoryDatabaseUser`	Required for connection to Oracle database or to SQL Server database using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	The database user account that B&OM server will use to connect to B&OM Repository database. Make sure the account has the required roles and privileges, as described in the Databases section of this guide.
`OutputManagement!` `RepositoryDatabasePassword`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	Specifies the password that Batch & Output Management uses to connect to its Repository database. **Note** The password cannot begin with a single quote or contain double quotes.

Parameter	Required / Optional	Description
`OutputManagement!RuntimeDatabaseConnectionString`	Required	Specifies the connection string that B&OM uses to connect to its Runtime database. This cannot be an ODBC connection string. Depending on the `OutputManagement!DatabaseType` parameter, this must be either a valid Oracle or SQL Server connection string. • Primary B&OM installation must be connected to the new empty database. • Secondary B&OM installation must be connected to the same database as the primary installation. **Note** Do not specify user name or password in this connection string. The database user account is configured using the specific parameters described below.
`OutputManagement!RuntimeDatabaseUser`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	The database user account that B&OM server will use to connect to B&OM Runtime database
`OutputManagement!RuntimeDatabasePassword`	Required for Oracle database user and for SQL Server connection using SQL Server authentication. Can be omitted whenever you have a connection string that does not require a user and password, for example, for connection to SQL Server database using Windows authentication.	Specifies the password that Batch & Output Management uses to connect to its Runtime database.

Example

You may want to specify `UpgradeOutputManagement` parameters and run the tool using a cmd file. The following is an example command:

```
UpgradeOutputManagement ^
Upgrade!From=5.4 ^
Services!User=enterprise\appadmin ^
Services!Password=entlocal1 ^
OutputManagement!DatabaseType=SQLServer ^
OutputManagement!RepositoryDatabaseConnectionString=”Server=sqlsrv02\SQL2017;
 Database=OM_5.5_Repo;” ^
OutputManagement!RepositoryDatabaseUser=kcm ^
OutputManagement!RepositoryDatabasePassword=entkcm_1 ^
OutputManagement!RuntimeDatabaseConnectionString=”Server=sqlsrv02\SQL2017;
 Database=OM_5.5_RT;” ^
OutputManagement!RuntimeDatabaseUser=kcm ^
OutputManagement!RuntimeDatabasePassword=entkcm_1
```

Notes on usage

- When upgrading a KCM installation that contains both KCM (that is, Contract Manager and KCM instances) and B&OM components, first upgrade the KCM components using `Upgrade.exe`, and then upgrade B&OM.

Note After upgrading KCM, the B&OM component of the old installation will not function any more.

- The deploy root path is taken over from the old installation and cannot be changed.
- Both database connection strings are not the ODBC connection strings.

Switch to old installation

Since KCM 5.5.0 is installed side-by-side with the previous version, you can easily switch to your old installation if the new installation does not work properly.

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management\Upgrade`
2. Run `Deactivate.exe` to deactivate KCM 5.5.0.

After running this command, KCM 5.5.0 is no longer active, and an older installation can be activated.

Activate the older installation of by using a relevant tool:

- `Activate.54.exe` for activating a KCM 5.4.0 installation
- `Activate.53.exe` for activating a KCM 5.3.0 or 5.3.1 installation
- `Activate.52.exe` for activating a KCM 5.2.0 installation

To switch back to KCM 5.5.0, you first must deactivate the older installation.

Deactivate the older installation by using a relevant tool:

- `Deactivate.54.exe` for deactivating the KCM 5.4.0 installation
- `Deactivate.53.exe` for deactivating the KCM 5.3.0 or 5.3.1 installation

- `Deactivate.52.exe` for deactivating the KCM 5.2.0 installation

After deactivating the older installation, you can activate KCM 5.5.0 again:

1. Navigate to: `<deploy root>\KCM\Programs\<version>\Management\Upgrade`
2. Run `Activate.exe` to activate KCM 5.5.0.

To start KCM services after activation, use the appropriate tools, as described in the next section.

Activate KCM installation

This section describes how to use the `Activate.exe` tool to activate KCM installation.

Overview

`Activate.exe` is a command-line tool that marks the KCM 5.5.0 installation as active. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Upgrade`.

`Activate.exe` is necessary to run if you are not doing a fully automatic deployment with `Install.exe`.

Run `Activate.exe` before starting KCM with `StartInstance.exe` and `StartContractManager.exe`, as only the active KCM installation can be started.

Note Activation does affect Batch & Output Management.

You can have multiple KCM installations on the same server, but only one of them can be active. So if there already is an active KCM installation on the server, then `Activate.exe` will fail.

Parameters

This tool does not have any command-line parameters.

Deactivate KCM installation

This section describes how to deactivate KCM installation using the `Deactivate.exe` tool.

Overview

`Deactivate.exe` is a command-line tool that removes the active flag from the KCM 5.5.0 installation. This tool resides in `<deploy root>\KCM\Programs\<version>\Management\Upgrade`.

Deactivation will prevent the KCM 5.5.0 installation from being started until it is activated again. For example, if you have multiple KCM installations on the same server, you may want to deactivate one that is active in order to allow activation and start of another installation.

The KCM 5.5.0 installation can only be deactivated if its components (Contract Manager and all KCM instances) have been stopped first. For that, use the `StopContractManager.exe` and `StopInstance.exe` tools.

Note Deactivation does affect Batch & Output Management.

Parameters

This tool does not have any command-line parameters.

Chapter 8

Uninstallation

To uninstall KCM 5.5.0, execute the `Uninstall.exe` tool described below. This will remove the Contract Manager and all KCM instances on a single server.

If you need to uninstall just the Contract Manager or a KCM instance, use the `RemoveContractManager.exe` and `RemoveInstance.exe` tools described below.

Uninstall basic KCM deployment

This section describes how you can use `Uninstall.exe` to remove the components of KCM 5.5.0 basic installation from the server. As a result, the Contract Manager and all KCM instances from a single server will be removed at once. If you need to remove these components separately, use `RemoveInstance.exe` or `RemoveContractManager.exe` instead. These tools are described in the next sections.

Note `Uninstall.exe` does not remove Batch & Output Management. To remove it, see the related section later in this guide.

Overview

`Uninstall.exe` is a command-line tool intended to remove all KCM 5.5.0 components from the server. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Uninstall`.

Wherever a Contract Manager and one or several KCM instances are installed, `Uninstall.exe` will remove them both.

This tool can be used in all types of deployment. For example, you can use it to remove a basic (all-in-one) installation, or launch it on different servers in case of distributed deployment.

Note `Uninstall.exe` will not remove Batch & Output Management or KCM components of other versions. For example, if you have upgraded your KCM 5.4.0 deployment, you may have both KCM 5.5.0 and KCM 5.4.0 installed though not active. In that case, only KCM 5.5.0 will be uninstalled.

The tool works as follows:

1. It stops and removes all Windows services of the KCM instance and the Contract Manager.
2. It also removes KCM web applications and stops and removes Tomcat services.
3. It removes all installation files and registry entries of the KCM instance and the Contract Manager.
4. If there are multiple KCM instances on the server, Uninstall.exe will stop and remove them all.

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

Before you start

Make sure the account used to run `Uninstall.exe` is included in the local *Administrators* group on the target server.

Parameters

This tool does not have any command-line parameters.

Uninstall KCM instance

This section describes how you can use RemoveInstance.exe to uninstall KCM instance from the server.

Overview

`RemoveInstance.exe` is a command-line tool intended to remove a stopped KCM instance from the server. This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Uninstall`.

The tool works as follows:

1. It removes all Windows services, installation files and registry entries of the KCM instance.

   > **Note** If there are other KCM components or other KCM versions on that server, they will be preserved.

2. If there are no other KCM instances on the server, then the tool also removes the shared Tomcat services.

> **Note** Removing an instance clears files, services, and settings specific to this instance. The content database is unaffected.

Before you start

Before running the tool, check the following prerequisites:

1. All KCM instances on the server have been stopped, otherwise `RemoveInstance` will not proceed.
2. Shared Tomcat instances have been stopped – this is required for the KCM instance web applications to be removed.
3. The account used to run `RemoveInstance.exe` is included in the local *Administrators* group on the target server.

Parameters

Parameter	Required / Optional	Description
`Instance!Number`	Required	Numeric ID (1-99) of the KCM instance to remove. Must be an existing KCM instance.

Uninstall Contract Manager

This section describes how you can use `RemoveContractManager.exe` to uninstall the Contract Manager.

Overview

`RemoveContractManager.exe` is a command-line tool intended to remove a stopped Contract Manager from the server.

This tool resides in: `<deploy root>\KCM\Programs\<version>\Management\Uninstall`.

The tool works as follows:

1. It removes all Windows services, installation files and registry entries of the Contract Manager. The other KCM components on this server (if any) will not be affected.
2. If there are no other components than the Contract Manager, then the entire KCM installation will be removed.

Before you start

Before running the tool, make sure that the following prerequisites are met:

- The Contract Manager has been stopped, otherwise `RemoveContractManager` will not proceed.
- The account used to run `RemoveContractManager.exe` is included in the local *Administrators* group on the target server.

Parameters

This tool does not have any command-line parameters.

Uninstall Batch & Output Management

This section describes how you can use `UninstallOutputManagement.exe` to uninstall Batch & Output Management.

Overview

`UninstallOutputManagement.exe` is a command-line tool intended to remove the Batch & Output Management installation from the server. This tool resides in: `<deploy root>\KCM_OM\Uninstall\<version>`.

The tool works as follows:

1. It stops and removes all Windows services of the B&OM installation.
2. It removes all installation files and registry entries of the B&OM installation.

Note `UninstallOutputManagement.exe` creates the log files in its own folder `<deploy root>\KCM_OM\Uninstall\5.5`, so this folder is not removed during uninstallation. You will need to remove it manually when the uninstallation completes.

Before you start

Make sure the account used to run `UninstallOutputManagement.exe` is included in the local *Administrators* group on the target server.

Parameters

This tool does not have any command-line parameters.

Notes on usage

1. Batch & Output Management does not need to be stopped manually before uninstallation. The `UninstallOutputManagement.exe` will stop the services automatically.
2. The B&OM Repository database and B&OM Runtime database are not removed when B&OM is uninstalled

Chapter 9

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.

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 Compart 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.

Timeouts during components installation or start

The KCM installation and management tools impose timeouts on certain operations. You can increase these timeouts by means of configuration, as follows:

1. Locate the `KCMTools.ini` file in the `<deploy root>\KCM\Work\<version>\Management\Config` folder.

   Note If no KCM components have been installed yet on the server, then this file is located on the top level of the extracted KCM package.

2. Add or modify the `[Global] IncreaseTimeouts` setting, as follows:

   ```
   [Global]
   IncreaseTimeouts = <value>
   ```

   Here `<value>` is a number in milliseconds with which all timeouts will be increased.

Note The timeout increase only applies to the installation tools on the top level of an extracted KCM package, and to the management tools in the `<deploy root>\KCM\Programs \<version>\Management` folder, with the exception of `ManageCM`. The increase will be applied to all timeouts of these tools.