Keyfactor Command 10.2

Upgrade Overview

Table of Contents

1.0 Introduction

The Keyfactor Command solution by Keyfactor allows organizations to issue and manage certificates across enterprise infrastructures. For a comprehensive description of the components that make up Keyfactor Command, please see the *Keyfactor Command Server Installation Guide*[1] and the *Keyfactor Orchestrators Installation and Configuration Guide*[1]. There are also Keyfactor installation guides for third-party CA gateways that interface with Keyfactor Command. For an overview of the key new features in the latest version of Keyfactor Command, please see the *Keyfactor Command Release Notes*.

This document provides guidance to help you prepare for and complete an upgrade. In most cases, a Keyfactor Solution Architect will assist you with the upgrade and walk you through the process. Please contact your Client Success representative for assistance.

Keyfactor Command version 10.0 and later require an encrypted connection to the SQL server. Upgrades will fail if the SQL server is not correctly configured to support this. See System Requirements on page 5.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

2.0 Preparing

This section describes the steps that need to be taken prior to a Keyfactor Command upgrade to complete the prerequisites, create any required supporting components, and gather the necessary information to complete the Keyfactor Command upgrade process.

The following are some key preparation steps that need to be addressed in order to upgrade to version 10.2:

- Keyfactor Command version 10.0 and later by default connects to SQL with an encrypted connection using an SSL certificate configured on your SQL server. Customers should acquire and install an SSL certificate for the SQL server before upgrading to Keyfactor Command version 10.0 or later (see the *Using SSL to Connect to SQL Server* section of the *Keyfactor Command Server Installation Guide*[1]). If you would prefer not to use an encrypted channel for your connection to SQL, see the *Configurable SQL Connection Strings* section of the *Keyfactor Command Server Installation Guide*[1].
- Upgrade to SQL Server 2016 CU2 or higher and adjust the database compatibility level if needed. For more information, see System Requirements on page 5.
- As of Keyfactor Command version 10.0, Windows Server 2016 is no longer supported. The installer will not check your server version nor prevent installation, but the product will not function properly in some instances. Customers should upgrade to Windows Server 2019 or higher before upgrading to Keyfactor Command version 10.0 or later. If you choose to use Server 2016, any PFXs will need to be configured to use SHA1 and 3DES for encryption for use by Keyfactor Command.
- Before upgrading to a major version, Keyfactor recommends first upgrading to the final incremental version of the previous major version—completing both the software installation and configuration with the configuration wizard—for the optimal upgrade experience. For example, if you are currently on version 9.4, you should upgrade first to version 9.10.1 (the final incremental version of 9.x) before upgrading to a version of 10.0 or later. Contact your Customer Success Manager for more information.
- If you have any saved certificate collections containing any of the following deprecated certificate search fields, these collections will need to be removed or updated to remove use of these fields that are no longer in version 10.0 and later:
  - *KeyfactorRequestId*
  - *RequestResolutionDate*
  - *CARequestId*

  These certificate search fields parsers have been removed to allow for native EJBCA support in Keyfactor Command as of version 10.0.
- If you have the CA Policy module version 7.0 installed on the same server as the Keyfactor Command Management Portal, you'll need to upgrade the module to version 7.1 or later before running the Keyfactor Command version 10.0 or later upgrade.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

- If you are upgrading from an older version of Keyfactor Command, the installation directory changed, as of Keyfactor Command v9, to C:\Program Files\Keyfactor. Move any scripts or files that are held in the old directory structure to the new location.

Note: Upgrading from versions prior to Keyfactor Command 6.1.0 will require upgrading in multiple steps and versions prior to 4.5.1 require extra preliminary steps. Contact your Customer Success Manager for more information.

2.1 Licensing

You will receive a new license file for the new version of Keyfactor Command. Before upgrading, locate your existing license file so that, should you need to revert to your existing software version, you will easily be able to do so without requesting a new license file from Keyfactor. (License files have the file extension ".cmslicense".)

As you begin the upgrade, have both your new license file and your existing license file on hand.

If you need assistance with a license, send a request to support@keyfactor.com.

2.2 Users, Service Accounts and Groups

Review the Active Directory service accounts and groups used by your Keyfactor Command implementation. You will need to have these accounts and groups available during the upgrade process, along with the passwords for the service accounts. A full overview of the required service accounts and groups can be found in the *Keyfactor Command Server Installation Guide* [1]. The most common service accounts are:

Keyfactor Command Service Account

In many environments, a single service account is used for most Keyfactor Command functions, including the application pool service account and the service account for the Keyfactor Command Service[2]. In some environments, separate service accounts are used for these functions.

Keyfactor Command LogiAnalytics Service Account

Keyfactor Command uses a the reporting engine LogiAnalytics. This reporting engine uses the same service account the application pool is configured to use.

Keyfactor Command Orchestrator Service Accounts

If you are using orchestrators, you will need the account(s) the orchestrators are configured to run as and the account(s) used to connect to the Keyfactor Command Orchestrators site.

---

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.
[2]If you're running the Certificate Management System rather than Keyfactor Command, this service will be called the CMS Timer Job Service.

Keyfactor Command Policy Module

If you are using the Keyfactor Command policy module with any of the standard policy handlers or any custom policy handlers, you will need to have access to upgrade these on the CA if you will be upgrading these at the same time.

CA Gateways

Important: All CA Gateways must be upgraded to AnyGateway v22.1 to work with Keyfactor Command v10.

If you are upgrading any of the CA Gateways, you will need to have the correct credentials to connect to the cloud-based certificate authority. The format of these varies depending on the CA provider. Some providers use a username and password while others use client certificate authentication. Some support the choice of either.

If you are unable to locate the existing passwords for your service accounts, you will need to reset the passwords so that the accounts will have known values in preparation for the upgrade. These password changes will need to be coordinated with your existing Keyfactor Command installation to avoid a service interruption. On your Keyfactor Command server(s), the password for the Keyfactor Command service account (assuming you are using just one) will need to be changed:

- In IIS for the CMS/Keyfactor Command application pool.
- In the Services MMC for the Keyfactor Command Service[1].
- Via the Keyfactor Command Configuration Wizard for the LogiAnalytics connectivity.
- Via the Keyfactor Command Orchestrator Configuration Wizard for any orchestrators running in the environment.

Password updates for the Keyfactor Command service accounts can be done via the Keyfactor Command Configuration Wizard during the upgrade process and do not need to be done ahead of the upgrade. The password(s) should be changed in Active Directory as close to upgrade time as possible to limit down time in the existing Keyfactor Command implementation.

If possible, identify the user account that was used to do the original installation of Keyfactor Command (the "installer" account) and use this same account to perform the upgrade. If you are upgrading under a different account than this, the permissions required in SQL will be different. See SQL Permissions below.

2.3 SQL Permissions

The user who upgrades Keyfactor Command must have permissions to administer the SQL server and update databases. The user may need to be able to add users (logins), depending on the features used. Full sysadmin permissions in SQL are needed if you're upgrading from a previous version of Keyfactor Command and the user running the install is not the same user who installed the previous version of Keyfactor Command. If the user is the same, only the dbcreator, public and securityadmin roles are needed.

Once Keyfactor Command has been upgraded, these permissions can be removed for the user.

[1]Or CMS Timer Job Service for older versions of the software.

Connecting to SQL over SSL

By default, Keyfactor Command connects to SQL using an encrypted connection. This requires configuration of an SSL certificate on your SQL server.

If your SQL server is not configured correctly for SSL, you'll see an error message similar to the following when you try to make a connection fromKeyfactor Command:

```
Unable to establish a connection to the database server. Please ensure that the server name
is correct and sufficient privileges have been granted to the connection account.:
Encountered an invalid or untrusted certificate and could not connect to the database.
TLS encryption is enabled by default. Please visit 'Planning and Preparing --> SQL Server'
In the Keyfactor Installing Server guide to resolve this.
```

To acquire a new SSL certificate or check for an existing certificate, see the *Using SSL to Connect to SQL* section of the *Keyfactor Command Server Installation Guide*[1].

If you would prefer not to use an encrypted channel for your connection to SQL, see the *Configurable SQL Connection Strings* section of the *Keyfactor Command Server Installation Guide*[1].

2.4 System Requirements

For a full list of the requirements, see the *System Requirements* section of the *Keyfactor Command Server Installation Guide*[2].

Operating System

Keyfactor Command server is supported on Windows Server 2019 or 2022.

Note: As of Keyfactor Command version 10.0, Windows Server 2016 is no longer supported. The installer will not check your server version nor prevent installation, but the product will not function properly in some instances. Customers should upgrade to Windows Server 2019 or higher before upgrading to Keyfactor Command version 10.0 or later.

PKI Architecture

Please visit Confirm the Architecture on page 8 and review the implications of upgrading with regard to the PKI architecture elements.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

[2]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

SQL Server

As of Keyfactor Command version 10.0, connectivity to the SQL server requires TLS encryption. For information about configuring TLS for SQL server, see:

https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-ver15

Microsoft SQL Server 2016 with cumulative update (CU) 2 or higher, SQL 2017, or SQL 2019 is required. You also need to ensure the database compatibility is updated to support 2016 or higher (level 130 or higher). For more information about SQL 2016 CU 2 see:

https://support.microsoft.com/en-us/topic/kb3182270-cumulative-update-2-for-sql-server-2016-8dc9ecae-4af9-cb12-d058-37cafc7f3758

Tip: To check the compatibility level of the database, run the query:

```
SELECT name, compatibility_level FROM sys.databases
```

The value returned for compatibility_level should match the version of SQL server you are using for your Keyfactor Command database(s). If this needs to be updated, take a backup before updating the compatibility level via SQL query. For example, to update to compatibility_level 150 (SQL 2019):

```
ALTER DATABASE [KeyfactorDB] SET COMPATIBILITY_LEVEL = 150
```

Where [KeyfactorDB] is the name of your Keyfactor Command database and the compatibility_level value matches the version of SQL server you are using.

For more information, see:

https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-ver15

.NET Framework

Microsoft .NET 4.7.2 or greater must be installed on the Keyfactor Command server(s) prior to installation of the latest Keyfactor Command software.

For Windows Server 2019 and Windows Server 2022, .NET is a standard Windows feature added through the Windows Server Manager tool. It can be updated to .NET 4.7.2 or greater with a downloadable update package or through Windows update.

Check .NET Framework Version

To verify the version of .NET installed, either:

1. Open the Registry Editor:

   ```
   regedit
   ```

2. Navigate to:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full
```

3. Validate that the **Release** attribute value indicates a version of .NET Framework that is 4.7.2 or higher is installed, as shown in Table 1: .NET Framework Release Values.

Or:

1. Open a command prompt or PowerShell window and type the following command:

```
reg query "HKLM\Software\Microsoft\NET Framework Setup\NDP\v4\Full"
```

2. Validate that the **Release** attribute value indicates a version of .NET Framework that is 4.7.2 or higher is installed, as shown in Table 1: .NET Framework Release Values.

Table 1: .NET Framework Release Values

.NET Framework	Release Value (Decimal)	Release Value (Hexadecimal)
.NET Framework 4.6.2	394802 or 394806	60632 or 60636
.NET Framework 4.7	460805	70805
.NET Framework 4.7.1	461308 or 461310	709FC or 709FE
.NET Framework 4.7.2	461808 or 461814	70BF0 or 70BF6
.NET Framework 4.8	528040, 528049, 528372, or 528449	8oEA8, 80EB1, 80FF4, 81041

PowerShell Requirement

More recent versions of Keyfactor Command make use of the Active Directory tools for PowerShell to do group membership queries in Active Directory in some functions (e.g. when using a group to create a mapping between a Linux logon for SSH and one or more SSH keys). If this feature is not already installed on your Keyfactor Command server, you will need to install it before upgrading the Keyfactor Command software. The *Active Directory module for Windows PowerShell* is installed as a feature as part of the *Remote Server Administrator Tools*. You may install this through the Roles and Features wizard or using the following PowerShell command:

```
Install-WindowsFeature RSAT-AD-PowerShell
```

2.5 Download the Software

Your Keyfactor contact should provide you with a link to download the updated software versions. Be sure to download all the files you will need ahead of the actual upgrade date. This includes the main Keyfactor Command server software as well as the software for the Keyfactor policy module, any orchestrators (e.g. Keyfactor Universal Orchestrator, Keyfactor Java Agent) or gateways (e.g. AnyGateway), that you will be upgrading at the same time or new software you will be deploying.

2.6 Configuration File

Keyfactor Command can use a file to pass the configuration information into the configuration wizard, which saves a significant amount of typing when you do your initial installation. You may have been provided one of these already configured for your initial implementation, or you may have created one after typing in all the configuration information during the initial implementation. If you can locate this file, it can save some time in the upgrade process. You won't want to import the existing file again, as the file structure may change between versions and importing the file again will overwrite any changes you might have made to the configuration since your initial install, but you can refer to the file for previous configuration information.

The configuration files generally have a .cmscfg extension. When creating the file, you have the option to encrypt and password protect the file. If the file has been password protected, sensitive information in the file, such as any service account passwords, will be encrypted, but the remainder of the file will be human readable. You will need to know the password used to protect the file in order to use the file in its complete state.

2.7 Confirm the Architecture

Before you start your upgrade, make sure you have a clear picture of your Keyfactor Command architecture and all the parts that make up the environment, and carefully consider the following.

Roles

Identify all the servers that play a role in the Keyfactor Command environment, including whether you have duplicates of any server roles to support high availability, and make note of what role or roles will need upgrading on each one. Think about whether you want to make any changes to the architecture at this time, such as adding high availability, or consolidating roles.

Certificate Authorities

Keyfactor Command includes a constraint (introduced in version 9.0) that prevents any two certificate authorities from having the same logical name and host name combination. Think about the logical name and host name of the CAs that will be implemented with Keyfactor Command and check for duplicates.

Important: During upgrade, if duplicates are found, then among the duplicates, if there is only one that has any information tied to it, such as certificates, API applications, etc., then all of the others will be removed by the upgrade script. If more than one of the duplicates has any information associated with it, then the upgrade script will stop with an error. In that instance, you will need to manually fix the data before upgrading can proceed.

Templates

Keyfactor Command 10.0 and later upgrades will fail if the database has duplicate templates, defined as:

- Duplicate CommonName and Forest, or
- Duplicate OID and Forest

This should be a rare case. If it does occur, contact Keyfactor support. Support will be able to identify the duplicate templates, save the desired templates, and remove the duplicates.

2.8 Backup

Immediately before starting the upgrade, make a backup of these items:

- Your Keyfactor Command SQL database
- Your SQL server Service Master Key (SMK) and/or Database Master Key (DMK), if needed (see Important note)
  If you plan to migrate your Keyfactor Command implementation to a different SQL server during the upgrade, you need a thorough understanding of how Keyfactor Command uses the SMK and DMK. Review this data in the *SQL Encryption Key Backup* section of the *Keyfactor Command Reference Guide*[1] and make appropriate plans before beginning your upgrade. If you plan to stay on the same SQL instance for the upgrade, you don't necessarily need to backup the SMK or DMK immediately before starting the upgrade. These can just be backed up as part of your normal disaster recovery planning process. Failing to back up the SMK and/or DMK will result in data loss and require manual re-entry of any secret data into Keyfactor Command in the event that the Keyfactor Command database needs to be restored from a backup to a SQL instance other than the original installed instance of SQL server.

  Note: For more information about how Keyfactor Command uses the SMK and DMK and how to back these up, see the *SQL Encryption Key Backup* section of the *Keyfactor Command Reference Guide*[1]. For more details on the mechanics of SQL Server Encryption and related disaster recovery procedures, see the SQL Server documentation.

- If you're using Keyfactor Command encryption, backup your encryption certificate, with private key (see below)
  More recent versions of Keyfactor Command allow you to encrypt select sensitive data stored in the Keyfactor Command database using a separate encryption methodology. This Keyfactor Command encryption utilizes a Keyfactor-defined certificate on top of the SQL server encryption noted above. This additional layer of encryption (Keyfactor Command encryption) protects the data in cases where the SQL Server master keys cannot be adequately protected. More information is provided in the *SQL Server* section of the *Keyfactor Command Server Installation Guide*[1].
- Backup the NLog configuration file for each application to be upgraded. The location of this file varies depending on the application in question. For older versions of the Keyfactor Command server, it can be found in one of these locations:
  C:\Program Files\Common Files\Certified Security Solutions\Certificate Management System\NLog.config
  C:\Program Files\Common Files\Keyfactor\Keyfactor Platform\NLog.config

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

More recent versions of Keyfactor Command separate the NLog configuration into multiple files, with these locations, by default:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\NLog_Configuration.config
C:\Program Files\Keyfactor\Keyfactor Platform\KeyfactorAPI\NLog_KeyfactorAPI.config
C:\Program Files\Keyfactor\Keyfactor Platform\Service\NLog_TimerService.config
C:\Program Files\Keyfactor\Keyfactor Platform\WebAgentServices\NLog_Orchestrators.config
C:\Program Files\Keyfactor\Keyfactor Platform\WebAPI\NLog_ClassicAPI.config
C:\Program Files\Keyfactor\Keyfactor Platform\WebConsole\NLog_Portal.config

- Make a backup of the Logi configuration file, which is found here by default:

  C:\Program Files\Keyfactor\Keyfactor Platform\Logi_Definitions_Settings.lgx

- If you have any custom extension handlers (e.g. auto-registration, alert events), make a backup of these.
- If you've have any other text-based configuration files that have been modified (this is most common for users who have enabled a third-party PAM provider such as CyberArk), make a backup of these.
- If you're using a custom logo for your Management Portal, make a backup of this image. This file can be found here, by default:

  C:\Program Files\Keyfactor\Keyfactor Platform\WebConsole\Images\Banner.png

- Review the authentication settings you have configured in IIS for each of the Keyfactor Command applications under the Default Web Site (or other web site if you've installed elsewhere) and make notes as to how they are configured so that you can confirm that the configuration is the same following upgrade.
- If you're using a virtualization solution for your Keyfactor Command application server(s), backup each virtual server as an image.
- If you are using a version of Keyfactor Command older than 6 and have existing SSL scans, export them to a file using the following script. Replace the bold italicized parts with the information relevant to your environment. This step is not necessary if you're upgrading from release version 6 or later.

```
$connectionString = "Data Source=SQLServerName;Integrated
Security=SSPI;Initial Catalog=KeyfactorDB"

$connection = new-object
System.Data.SqlClient.SqlConnection($connectionString)

$connection.Open()
# Password can be read from an encrypted file which can be secured as follows:
# Create a password file while logged in as the service account that will run this script:
# $credential = Get-Credential
# $credential.Password | ConvertFrom-SecureString | Set-Content
C:\Keyfactor\PowerShell\encrypted_password1.txt

# use the code below for the credentials
#$password = Get-Content C:\Keyfactor\PowerShell\encrypted_password1.txt | ConvertTo-SecureString
#comment out the below line if using secure credentials
$password = "Password" | ConvertTo-SecureString -AsPlainText -Force
```

```
#Update with the credentials for your environment
$username = "domain\administrator"
$credential = New-Object System.Management.Automation.PSCredential($username, $password)
$passphrase = $username + ":" + $password
$fileName = "DiscoveryGroupsExported.txt"

#The name of the agent in your environment
$AgentName = "kyfagent1.domain.com"

#Update with the URLs for your environment.
$kyfAgentUrl = "http://kyfagent1.domain.com/CMSAPI/SSL/1/Agents"
$kyfGroupUrl = "http://kyfagent1.domain.com/CMSAPI/SSL/1/AddEndpointGroup"
$kyfEndpointUrl = "http://kyfagent1.domain.com/CMSAPI/SSL/1/AddEndpoint"

$Bytes = [System.Text.Encoding]::Unicode.GetBytes($passphrase)
$EncodedText = [Convert]::ToBase64String($Bytes)
$headers = @{"Authorization" = "Basic $EncodedText";
       "Content-Type"  = "application/json;" }

$responseAgent = Invoke-RestMethod -Method Get -Uri $kyfAgentUrl -Header
$headers -Credential $credential
if ($responseAgent)
{
  write-host $responseAgent.Name.ToLower()
  if ($responseAgent.Name.ToLower() -eq $AgentName.ToLower())
  {
    $AgentGUID = $responseAgent.Guid
  }
  write-host $AgentGUID
  $kyfEndpointGroups = "http://ky-
fagent1.domain.com/CMSAPI/SSL/1/EndpointGroups?agentId=$AgentGUID"
  write-host $kyfEndpointGroups

  $responseEndpointGroups = Invoke-RestMethod -Method Get -Uri $kyfEndpointGroups -Header
$headers -Credential $credential
  if ($responseEndpointGroups)
  {
    foreach($res in $responseEndpointGroups)
    {
      write-host $res.Name
      $GroupName = $res.Name
      #write-host $res.guid
      $GroupGuid = $res.Guid
      $sql = "SELECT VALUE, TypeID FROM cms_agents.SslEndpointGroupItems WHERE GroupID =
'$GroupGuid'"
```

```
      #write-host $sql $command = new-object System.Data.SqlClient.SqlCommand($sql,$connection)

      $reader = $command.ExecuteReader()
      while ($reader.Read())
      {
        $value = ""
        $type = ""
        $value = $reader["Value"]
        $typeId = $reader["TypeId"]
        #Add-Content $filename "$GroupName,$GroupGuid,$value,$typeId"
      }
      $reader.Close()
    }
  }
}
else
{
  write-host "Agent not found."
}
$connection.Close()
```

The resulting text file will contain the network definitions you currently have and can be opened in Excel. When Keyfactor Command has been upgraded, you can copy and paste from the file into the newly defined *Networks* that replace the previous *Discovery* and *Monitoring* groups.

3.0 Upgrading

Most Keyfactor Command upgrades are brief with a minimum of changes to existing user accounts, groups, CA templates, firewall settings, etc. The prerequisites have not materially changed from previous versions and the current version can generally be installed using the same hardware and existing instances of the supporting software. The upgrade process is often completed within three to four hours.

Note: For complete details of the upgrade process, review the *Keyfactor Command Server Installation Guide*[1].

Important: Before upgrading, please be sure you have reviewed and addressed the important preparation steps (see Preparing on page 2).

Important: The MicrosoftECCurveUpgradeModule may fail due to a pre-version 10.0 issue in which Certificate Request Contents were truncated to 4k characters when saved to the database. If your upgrade fails when the MicrosoftECCurveUpgradeModule is run contact Keyfactor Support to obtain assistance with the scripts that will have to be run to fix this issue.

Important: During the upgrade process Keyfactor Command prevents duplicate template records from being inserted into the database. Duplicate templates could be found, for example, if there are templates in different forests with the same name. If you receive an error message during upgrade, and the log shows a list of the duplicate templates, contact Keyfactor Support. We will be able to support you through the process of resolving the issue and completing the upgrade.

The overall task flow consists of the following steps:

Upgrade of the Server Software

In most cases the Keyfactor Command server software can be installed over the existing software installation without uninstalling the previous version. Install the software as per the *Keyfactor Command Server Installation Guide*[1], retaining the same installation location. In the configuration wizard, populate the fields while referring to your configuration file open in a text editor (see Configuration File on page 8). Use the existing IIS application pool.

Update Windows Orchestrators

If you're upgrading from a version of Keyfactor Command prior to 8.0, you will need to update any Windows Orchestrators (a.k.a. Windows Agents) that are used for SSL scanning to support the current scanning architecture. Install and configure the Keyfactor Universal Orchestrator software as per the *Keyfactor Orchestrators Installation*

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

and Configuration Guide[1].

The Keyfactor Universal Orchestrator replaces the Keyfactor Windows Orchestrator and runs on both Windows or Linux servers. As of this release, the following functions that were part of the Keyfactor Windows Orchestrator are only supported in the Keyfactor Universal Orchestrator with custom extensions:

- Interact with F5 devices for certificate management (available now on the Keyfactor GitHub site)
- Interact with NetScaler devices for certificate management (coming soon to the Keyfactor GitHub site)
- Interact with Amazon Web Services (AWS) resources for certificate management (coming soon to the Keyfactor GitHub site)

The final release of the Keyfactor Windows Orchestrator was version 8.7. This version of the Keyfactor Windows Orchestrator is fully compatible with Keyfactor Command version 10.2. Keyfactor will continue to support the Keyfactor Windows Orchestrator. However, all new integrations and extensions will be delivered via the new Keyfactor Universal Orchestrator. Keyfactor recommends that customers use the Keyfactor Universal Orchestrator moving forward as new extensions become available. Customers with one or more of these types of certificate stores may wish to retain one or more legacy Keyfactor Windows Orchestrators to manage these types of stores until such time as new extensions become available for the Keyfactor Universal Orchestrator. Currently, to manage NetScaler and AWS certificate stores, an 8.x version of the Keyfactor Windows Orchestrator must be used. If you're upgrading from a version of Keyfactor Command prior to 8.0, contact your Keyfactor representative to obtain the installation media for the 8.7 Keyfactor Windows Orchestrator.

Important: The Keyfactor Universal Orchestrator is only compatible with Keyfactor Command version 9.0 or later. The current version of the Keyfactor Universal Orchestrator is 10.1 and requires .NET 6.

Note: The orchestrator endpoint location changed for Keyfactor Command release 6 and may need to be modified in your orchestrator endpoint configuration—from CMSAgents to KeyfactorAgents.

Keyfactor CA Policy Module

The most recent versions of the Keyfactor CA Policy Module software need to be upgraded using the below method and PowerShell script and can't be installed over an existing implementation of the Keyfactor CA Policy Module as an upgrade method.

To upgrade a Keyfactor CA Policy Module:

1. Make a note of all your existing policy module configuration, including which policy handlers are enabled and what configurations are set within each handler. During the upgrade process, you will uninstall the policy module, which will remove your configuration. The upgrade script should successfully restore the configuration as part of the upgrade process, but you will want to have a complete record of the configuration as a backup.

2. On the Keyfactor CA Policy Module server, open a PowerShell window using the "Run as administrator" option.

3. In the PowerShell window, change to the directory in which you placed the upgrade script included with the latest version of the Keyfactor CA Policy Module and execute it in *archive* mode. For example:

```
.\Keyfactor-CA-Modules-Upgrade-Script.ps1 -Mode archive -InformationAction Continue -
ErrorAction Stop
```

Note: This step is creating a backup of your policy module configuration before you uninstall the old policy module. It will create an output file, *Keyfactor-CA-Policy.dat*, in the current directory.

Tip: Additional options are available in the upgrade script and can be viewed using the *-full* switch with *Get-Help*. For example:

```
Get-Help .\Keyfactor-CA-Modules-Upgrade-Script.ps1 -full
```

4. Unload the existing policy module in the CA MMC, and close the MMC.
5. Uninstall the existing policy module.
6. Install the latest version of the Keyfactor CA Policy Module as per the *Keyfactor Command Server Installation Guide*[1], but do not configure it. Be sure to install all the same policy handlers that were installed previously.
7. Execute the upgrade script included with the latest version of the Keyfactor CA Policy Module again, but this time in *restore* mode. For example:

   ```
   .\Keyfactor-CA-Modules-Upgrade-Script.ps1 -Mode restore -InformationAction Continue -
   ErrorAction Stop
   ```

   Note: This step takes the backup of your policy module configuration from the first run of the upgrade script and restores the information to the correct locations so that you will not need to re-configure the policy module. Be sure that the output file from the first run of the upgrade script, *Keyfactor-CA-Policy.dat*, is in the current directory.
8. Open the CA MMC and load the Keyfactor CA Policy Module as per the *Keyfactor Command Server Installation Guide*[1].
9. Open the Properties for the policy module and, if you've received a new license, install the new license on the License tab. On the Custom Handlers tab, review all the configuration to confirm that it has been correctly restored by the upgrade script.

Tip: New versions of the policy module are not necessarily released at the same time as new versions of Keyfactor Command and so the policy module may not need upgrading at the same time as Keyfactor Command.

EJBCA CA Gateway

If you're using an EJBCA gateway and wish to make use of the new feature in Keyfactor Command for native support of EJBCA CAs, you will need to follow the EJBCA gateway upgrade process to unlink the EJBCA certificates in your Keyfactor Command database from your EJBCA gateway CA to enable them to be relinked to a native CA configured in Keyfactor Command. For more information, contact Keyfactor support.

Other CA Gateways

In most cases, the Keyfactor gateway software can be installed over the existing software installation without uninstalling the previous version. Review the configuration for your gateway, and then install and configure the software as per the Keyfactor gateway guide for the particular gateway, retaining the same installation location. The gateway configuration wizard has significantly changed in recent releases for many of the gateways, which may require modification to your configuration.

Tip: New versions of CA gateways are not necessarily released at the same time as new versions of Keyfactor Command and so gateways may not need upgrading at the same time as Keyfactor Command.

API

Please see the *Release Notes*[1] if you are using any custom scripts that leverage one of the APIs.

Replacing or Re-Updating Customized Files

Files such as the nlog.config file or customized files for third-party PAM integration (e.g. web.config customizations for CyberArk) may have slight changes in the latest version as compared to the previous version, so you should not just copy your old, customized versions of those files over the current stock versions of these files. You will need to compare the files and make your customizations in the current versions of the files.

Post-Install Configuration and Testing

See Post-Upgrade Steps on page 17

The bulk of the time upgrading will be spent verifying that all functions and configurations have correctly carried over and the upgraded instance is performing correctly.

4.0 Post-Upgrade Steps

The recommended best practices for after you finish running the Keyfactor Command configuration wizard(s) are:

- The server should be rebooted to assure that the services have a clean start. If this is not possible:
  - Restart Keyfactor Command Service
  - Restart IIS
- Advise users to clear the cache on their web browser and reload the Keyfactor Command Management Portal.

There is no particular order in which the tasks on the following pages must be accomplished.

Tip: If, following the upgrade, you open a page in the Keyfactor Command Management Portal and find it unexpectedly blank or otherwise displaying incorrectly, try refreshing the page with a CTRL-F5. If this doesn't resolve the problem, try clearing the browser cache and then reloading the page. It may be helpful to advise all end users to do this following an upgrade.

4.1 Testing

Once everything is up and running again, confirm that the following features are operating correctly:

- Does the Keyfactor Command Management Portal load correctly?
- Run a report in the Keyfactor Command Management Portal to confirm that the connectivity to LogiAnalytics is operating correctly.
- Issue a certificate in the Keyfactor Command Management Portal to confirm connectivity to CAs and that Kerberos authentication is operating correctly (assuming the environment is configured for Kerberos authentication).
- Check the Keyfactor Command log files to confirm that no errors are appearing and that logging is occurring correctly.

4.2 Post-Install Configuration

If you are upgrading from any release of Keyfactor Command version 6 or greater, you may want to make some additional configuration changes post-installation:

- Upgrade any Keyfactor CA gateways in your environment that are based on the AnyGateway. The AnyGateway must be upgraded to at least 22.1 to be compatible with Keyfactor Command 10.0 and later.
- Consider whether you wish to implement Keyfactor Command workflows and whether a Keyfactor Command-level workflow could replace CA-level manager approval for any templates that are configured to require CA-level manager approval.

> **Note:** To prevent REST requests from being made to inappropriate locations by malicious users, if you plan to implement REST type workflows, configure a system environment variable of KEYFACTOR_BLOCKED_OUTBOUND_IPS on your Keyfactor Command server pointing to the IP address or range of addresses in CIDR format that you wish to block. Both IPv4 and IPv6 addresses are supported. More than one address or range may be specified in a comma-delimited list. For example:
>
> ```
> 192.168.12.0/24,192.168.14.22/24
> ```
>
> When a REST request is made where the URL is either configured to a blocked IP address or resolves via DNS to a blocked IP address, the REST request will fail.

- Review the new enrollment default and policy settings for enrollment. Enrollment defaults and polices can be defined at two levels:
  - System-wide settings apply to all enrollments done through Keyfactor Command unless they are over-ridden by template-specific settings.
  - Template-level settings allow you to modify any established enrollment defaults or policies on a per-template basis.

  There are several settings available for configuration as part of the template policies:
  - Allow Wildcards
  - Allow Public Key Reuse
  - Enforce RFC 2818 Compliance
  - Supported Key Types

  Enrollment defaults allow you to pre-polulate the subject fields in PFX Enrollment and CSR Generation. Users are allowed to override these at enrollment.
- If you're using certificate metadata or regular expressions, optionally define these for each template. Certificate metadata fields and regular expressions can be defined at two levels:
  - System-wide settings apply to all enrollments done through Keyfactor Command unless they are over-ridden by template-specific settings.
  - Template-level settings allow you to modify any established certificate metadata or regular expressions on a per-template basis (for instance, for a metadata field, whether the field is required, what default value it should provide, or whether to hide the field during enrollment, regardless of system-wide setting).
- Review any alert PowerShell event handlers you may have configured to ensure they are in the path (or subdirectory thereof) as defined in the *Extension Handler Path* application setting value. Changes as of version 9.0 will cause PowerShell event handlers to fail if not located in the defined directory. See the *Adding PowerShell Handlers to Alerts* section in the *Keyfactor Command Reference Guide*[1] for more information.

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

- The enrollment configuration will have been carried over in the upgrade, however you may want to confirm the configuration of Certificate Authority and Template enrollment (PFX, CSR, and CSR generation) and make any changes.
- Review any template that is configured to require manager approval at the CA level and confirm that a Keyfactor Command private key retention policy in place.
- Update any monitoring or other processes that reference the log files to point to the new log file location.
- Review the new reports in the Keyfactor Command Report Manager and add them to the menu or favorite them, if desired.

If you are upgrading from a release prior to Keyfactor Command version 6.1, please contact support (support@key-factor.com) for upgrade assistance.

5.0 Troubleshooting

Typically, an upgrade completes with few hiccups and the new version of Keyfactor Command comes up without incident. If this doesn't happen, start by checking the log file(s) for any errors. By default, these are located in C:\Keyfactor\logs. It is sometimes helpful to enable debug or trace level logging. This is done by editing the nlog.config file for each application. See the *Editing NLog* section in the *Keyfactor Command Reference Guide*[1] for more information.

Error During Upgrade

If you encounter an error during upgrade, this can be the result of a number of different things. Often, it's related to connectivity to SQL or issues on the SQL server. Check the *Command_Configuration_Log.txt* for messages related to upgrading and upgrade failures. This will point you in the right direction to begin troubleshooting.

The following error message indicates that the referenced upgrade script failed because it took longer to run than the allowed limit for SQL tasks:

```
2022-12-07 10:19:07.5078 Keyfactor.Sql.Management.Upgrade.UpgradePlan [Error] - Failed to
run upgrade module CSS.CMS.Install.Upgrade.Scripts.EJBCA_Resolved_Request_Contents_
Removal.sql: Execution Timeout Expired. The timeout period elapsed prior to completion of
the operation or the server is not responding.
```

The Keyfactor Command upgrade process includes multiple scripts, each doing different tasks, and each script is run in batches to limit the time and load of any one SQL request, but it's still possible to encounter a batch that exceeds the limit with vary large or complex databases. To resolve this particular issue, you can increase the timeout limit and restart the upgrade. You do not need to restore and start over.

To increase the timeout limit:

1. On the Keyfactor Command server, open a text editor (e.g. Notepad) using the "Run as administrator" option.
2. In the text editor, browse to open the *SharedSqlConnectionStrings.config* file in the Configuration directory under the installed directory. By default, this is:

   C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\SharedSqlConnectionStrings.config
3. Locate the connection string line that contains *command timeout=360*. This will look something like:

   <add name="SqlDirect" connectionString="data source=dev;initial catalog=test;integrated security=True;persist security info=True;command timeout=360;" />
4. The timeout value is set in seconds, so 360 seconds is 6 minutes. Set it to a new, longer value to allow the upgrade to complete. A value of 3600 seconds (30 minutes) should be more than enough. Don't set it to a value that's too high, as you do want the upgrade to time out if there's some fundamental problem communicating with SQL.

---

[1]Embedded links to external documents point to the document on the Keyfactor Client Portal. Access to the portal requires a login. See your administrator or your Keyfactor Client Success representative to obtain a login to the portal.

5. Save the file, close the configuration wizard, open the configuration wizard again (you should find it on the menu), and begin the upgrade again.

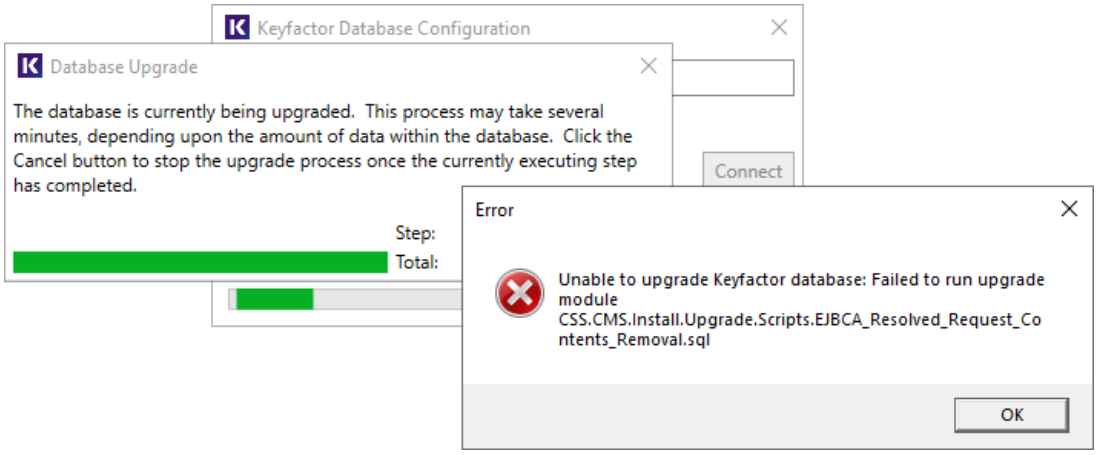

Figure 1: Error During Upgrade

Management Portal Doesn't Load After Upgrade

If the Keyfactor Command Management Portal appears to partially load or does not appear to include expected updates after the upgrade, try clearing the browser cache, closing the browser, and opening a fresh browser session. Try using CTRL-F5 to request the page again without cached content. In some upgrade cases, with Internet Explorer, the Certificate Search page only partially loads. With some browsers, opening the Developer Tools with the F12 key and clearing the cache will resolve the problem.

Certificate Enrollment Fails

If the certificate enrollment fails, this is often an indication that there is a Kerberos authentication problem. Confirm that the service principal name (SPN) is set correctly for the application pool service account and that Kerberos constrained delegation is configured correctly from the Keyfactor Command server(s) to the CA(s). See the *Configure Kerberos Authentication* section of the *Keyfactor Command Server Installation Guide*[1] for more information.

Event Handlers Don't Run

If your alert PowerShell event handlers or renewal event handlers do not run correctly, be sure that you have updated them to the correct new location. Scripted alert handlers will fail to run if not in the path (or a subdirectory of it) specified by the "Extension Handler Path" application setting. By default, this is "C:\Program Files\Keyfactor\Keyfactor Platform\ExtensionLibrary\". See the *Adding PowerShell Handlers to Alerts* section in the *Keyfactor Command Reference Guide*[1] for more information.

500 Error on the Dashboard or in Reports

If you receive a 500 error loading the dashboard and running reports but the remainder of the Management Portal seems to be operating correctly, check to be sure that the IP address(es) configured in the Configuration Wizard on the Dashboard and Reports tab have been entered correctly.

Underlying Connection Closed

If you receive an error when opening the Management Portal that "the underlying connection was closed" please be sure you have all the latest Windows updates installed.

Please refer to the Keyfactor Command *Release Notes* for known issues.

If you need further assistance, please contact support. During normal business hours, support can be reached at support@keyfactor.com or (877)-715-5448.

6.0 Copyright Notice