

Keyfactor Command 10.1

Server Installation Guide

Table of Contents

1.0 Introduction	1
2.0 Installing Servers	2
2.1 Logical Architecture	3
2.2 Physical Architecture	6
2.3 Solution Design	7
2.4 Keyfactor Command Server	8
2.4.1 System Requirements	8
2.4.2 Planning & Preparing	10
2.4.2.1 Certificate Authorities	10
2.4.2.2 SQL Server	11
2.4.2.3 Keyfactor Command Server(s)	18
2.4.2.4 Create Active Directory Service Accounts for Keyfactor Command	19
2.4.2.5 Create Active Directory Groups to Control Access to Keyfactor Command Features	22
2.4.2.6 Configure Certificate Chain Trusts for CAs	24
2.4.2.7 Hostname Identification and Resolution	24
2.4.2.8 Firewall Considerations	26
2.4.2.9 Acquire a Public Key Certificate for the Keyfactor Command Server	28
2.4.2.10 Grant Permissions in SQL	30
2.4.2.11 Install IIS and .NET on the Keyfactor Command Server	30
2.4.2.12 Configure SSL for the Default Web Site on the Keyfactor Command Server	36
2.4.2.13 Configure the Keyfactor Command Server to Require SSL	36
2.4.2.14 Prepare for External Log Shipping over TLS (Optional)	37
2.4.3 Installing	42
2.4.3.1 Install the Main Keyfactor Command Components on the Keyfactor Command Server(s)	42
2.4.3.2 Install the Keyfactor Command Server from the Command Line	67
2.4.4 Initial Configuration	73
2.4.4.1 Configure Kerberos Authentication	73
2.4.4.2 Configure Logging	81
2.4.4.3 Configure CA Certificate Synchronization	94
2.4.4.4 Create or Identify Certificate Templates for Enrollment	102
2.4.4.5 Configure Renewal Handler Permission	103
2.4.4.6 Create a Certificate Template for Mac Auto-Enrollment	105
2.5 Keyfactor CA Policy Module	106
2.5.1 System Requirements	106
2.5.2 Preparing for the Keyfactor CA Policy Module	107
2.5.3 Installing the Keyfactor CA Policy Module Handlers	108
2.5.3.1 Install the Keyfactor RFC 2818 Policy Handler	111
2.5.3.2 Install the Keyfactor SAN Attribute Policy Handler	116
2.5.3.3 Install the Keyfactor vSCEP™ Policy Handler	122
2.5.3.4 Install the Keyfactor Whitelist Policy Handler	128
2.5.4 Configure Logging for the Keyfactor CA Policy Module	135
2.5.5 Add Non-Keyfactor SCEP Servers to the Ignore List	136
2.6 Appendices	137
2.6.1 Appendix - Troubleshooting Logi Log Files	137
2.6.2 Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup	138
2.6.3 Appendix - Configuration Wizard Errors in the Logs	140
3.0 Glossary	142
4.0 Copyright Notice	150

List of Figures

Figure 1: Keyfactor Command Logical Architecture Diagram	3
Figure 2: Keyfactor Command Physical Architecture Diagram	6
Figure 3: Simple Keyfactor Command Solution Design	8
Figure 4: SQL Server Configuration Manager View Active SSL Certificate	14
Figure 5: Registry View Active SSL Certificate	15
Figure 6: View SQL Server Services	15
Figure 7: SQL Server SSL Certificate Details	16
Figure 8: Grant Private Key Permissions for SQL Server	17
Figure 9: Default SQL Connection Strings	18
Figure 10: SQL Connection Strings with Encrypt Channel Disabled	18
Figure 11: Local Security Policy	20
Figure 12: Install CA Chain Certificates on the Keyfactor Command Server	24
Figure 13: Certificate Template with Key Encipherment Key Usage	29
Figure 14: Use Get-WindowsFeature to Determine if All Required Roles and Features are Installed	32
Figure 15: Web Server Role	33
Figure 16: .NET 4.7 Feature	33
Figure 17: Role Services Page One	34
Figure 18: Role Services Page Two	35
Figure 19: Active Directory Module for Windows PowerShell	35
Figure 20: Install: Begin Setup Wizard	42
Figure 21: Install: Select Components	43
Figure 22: Windows Authentication	44
Figure 23: SQL Authentication	44
Figure 24: Configure: Backup Database Master Key	46
Figure 25: Configure: Upload License	46
Figure 26: Configure: Open Data File	47
Figure 27: Configure: Application Pools	48
Figure 28: Configure: Encryption Warning	49
Figure 29: Configure: Database	49
Figure 30: Configure: Service	55
Figure 31: Configure: Email	56
Figure 32: Configure: Keyfactor Portal	58
Figure 33: Configure: Dashboard and Reports	60
Figure 34: Configure: vSCEP Service	61
Figure 35: Configure: Orchestrators with Standard Authentication	62
Figure 36: Configure: Orchestrators with Client Certificate Authentication	63
Figure 37: Configure: APIs	64
Figure 38: Configure: Audit	65
Figure 39: Configure: Configuration Warnings	66
Figure 40: Configure: Save Configuration as a File	66
Figure 41: Configure: Configuration Operations	67
Figure 42: Configure: Configuration Complete	67
Figure 43: Configure Local Intranet Zone in Internet Properties	75
Figure 44: Configure Kerberos Constrained Delegation on the Keyfactor Command Machine Account	78
Figure 45: Add HOST and rpcss Service Types for Kerberos Constrained Delegation	79
Figure 46: Configure Kerberos Constrained Delegation on the Keyfactor Command Service Account	80
Figure 47: C:\Keyfactor\logs logs	82
Figure 48: Nlog_Configuration.config	84
Figure 49: Nlog_KeyfactorAPI.config	86
Figure 50: Nlog_TimerService.config	88
Figure 51: Nlog_Orchestrators.config	90
Figure 52: Nlog_Portal.config	92
Figure 53: Nlog_ClassicAPI.config	94

Figure 54: Certificate Profile for EJBCA Client Certificate	95
Figure 55: Certificate Download for EJBCA Client Certificate	95
Figure 56: Microsoft CA Permissions	97
Figure 57: EJBCA Access Permissions	100
Figure 58: Add Client Certificate as Member of EJBCA Access Rule	101
Figure 59: Keyfactor Command Service	101
Figure 60: Include Expired and Revoked Certificates in Certificate Search	102
Figure 61: Configure Expiration Renewal Handler	104
Figure 62: Configure Expiration Renewal Handler: Add New Identity	104
Figure 63: Configure Expiration Renewal Handler: Assign Role to Identity	105
Figure 64: Keyfactor CA Policy Module Policy Module Handler Ordering	109
Figure 65: Default Policy Module	110
Figure 66: Install RFC 2818 Policy Handler: Begin Setup Wizard	111
Figure 67: Install RFC 2818 Policy Handler: Select Components	112
Figure 68: Enable the Keyfactor CA Policy Module	113
Figure 69: Upload the Keyfactor CA Policy Module License	114
Figure 70: Enable the RFC 2818 Policy Handler	115
Figure 71: Add Templates for Management with the RFC 2818 Policy Handler	116
Figure 72: Install SAN Attribute Policy Handler: Begin Setup Wizard	117
Figure 73: Install SAN Attribute Policy Handler: Select Components	118
Figure 74: Enable the Keyfactor CA Policy Module	119
Figure 75: Upload the Keyfactor CA Policy Module License	120
Figure 76: Enable the SAN Attribute Policy Handler	121
Figure 77: Add Templates for Management with the SAN Attribute Policy Handler	122
Figure 78: Install vSCEP Policy Handler: Begin Setup Wizard	123
Figure 79: Install vSCEP Policy Handler: Select Components	124
Figure 80: Enable the Keyfactor CA Policy Module	125
Figure 81: Upload the Keyfactor CA Policy Module License	126
Figure 82: Enable the vSCEP™ Policy Handler	127
Figure 83: Configure Settings for the vSCEP™ Policy Handler	128
Figure 84: Install Whitelist Policy Handler: Begin Setup Wizard	129
Figure 85: Install Whitelist Policy Handler: Select Components	130
Figure 86: Enable the Keyfactor CA Policy Module	131
Figure 87: Upload the Keyfactor CA Policy Module License	132
Figure 88: Enable the Whitelist Policy Handler	133
Figure 89: Add Templates for Management with the Whitelist Policy Handler	134
Figure 90: Add Machines for Management with the Whitelist Policy Handler	135
Figure 91: Keyfactor CA Policy Module NLog.config File	136
Figure 92: Logi web.config	138
Figure 93: Logi Configuration Settings—Keyfactor Command Portal Tab	139
Figure 94: Logi Configuration Settings—Keyfactor Command Dashboards and Reports Tab	140

List of Tables

Table 1: System Requirements	9
Table 2: Typical Service Accounts	22
Table 3: Protocols Keyfactor Command Uses for Communication	26
Table 4: .NET Framework Release Values	31
Table 5: Keyfactor Command Services	51
Table 6: Features Required for Each Server Role	69
Table 7: Input Parameters XML File Fields	71
Table 8: ConfigurationWizardConsole.exe Options	73
Table 9: Microsoft CA Permission Matrix	97

1.0 Introduction

The *Keyfactor Command Documentation Suite* includes:

- *Keyfactor Command Reference Guide*
- *Keyfactor Web APIs Reference Guide*
- *Keyfactor Command Server Installation Guide*
- *Keyfactor Orchestrators Installation and Configuration Guide*
- *Keyfactor Command Release Notes*

In addition, Keyfactor offers documentation for products that are not part of the *Keyfactor Command Documentation Suite*, including the *Keyfactor Command Upgrade Overview* and installation guides for third-party CA gateways that interface with Keyfactor, which are available upon request.

2.0 Installing Servers

The Keyfactor Command solution by Keyfactor allows you to issue and manage certificates across enterprise infrastructures to allow you to achieve end-to-end visibility, control, and automation across all your machine identities so you can turn the impossible into the possible. It includes a web-based Management Portal running on a SQL backend providing the command and control center for managing certificates in the enterprise.

Keyfactor Command provides:

- **Visibility**
Identify risks and prevent outages more effectively with a complete and continuous inventory of all your cryptographic assets.
- **Control**
Have ultimate flexibility to make all certificates trusted, compliant, and up-to-date—and keep them that way.
- **Automation**
Replace manual, error-prone tasks with automated key and certificate discovery, management, and renewal.
- **Orchestration**
Move from DevOps to DevSecOps by orchestrating and expanding cryptography to secure software delivery pipelines.

In addition to the Management Portal, Keyfactor also offers:

- Several agents and orchestrators for managing certificates in certificates stores via the Management Portal (see *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide*).
- Several certificate authority gateways to support management of and enrollment for certificates from remote and cloud-based certificate providers via the Management Portal.
- APIs that integrate with the product to provide for customization (see *Web APIs Reference* in the *Keyfactor Web APIs Reference Guide*).
- A certificate authority policy module with several policy handlers to provide policy control at the Microsoft CA level (see [Keyfactor CA Policy Module on page 106](#)).
- An SSH Key Manager that extends beyond certificate management and traditional PKI to give security and network teams a simple, centralized solution to discover and manage SSH keys across their server and cloud infrastructure (see *SSH* in the *Keyfactor Command Reference Guide*).

Uniquely designed for PKI administrators to operate an enterprise PKI, it's never been easier to issue, revoke, renew, or replace a digital certificate. With exceptionally robust reporting and management capabilities for all the certificates in an IT environment, the PKI administrator has a truly scalable and entirely secure system for operating an enterprise PKI.

2.1 Logical Architecture

Keyfactor Command is an n-tier application, consisting of a web/presentation layer, application tier, and database tier. In addition, Keyfactor Command optionally includes a number of enrollment and management components to help facilitate secure and/or automated certificate issuance and delivery to various server, client, and mobile platforms. The following sections provide views of the Keyfactor Command architecture from a logical and physical standpoint.

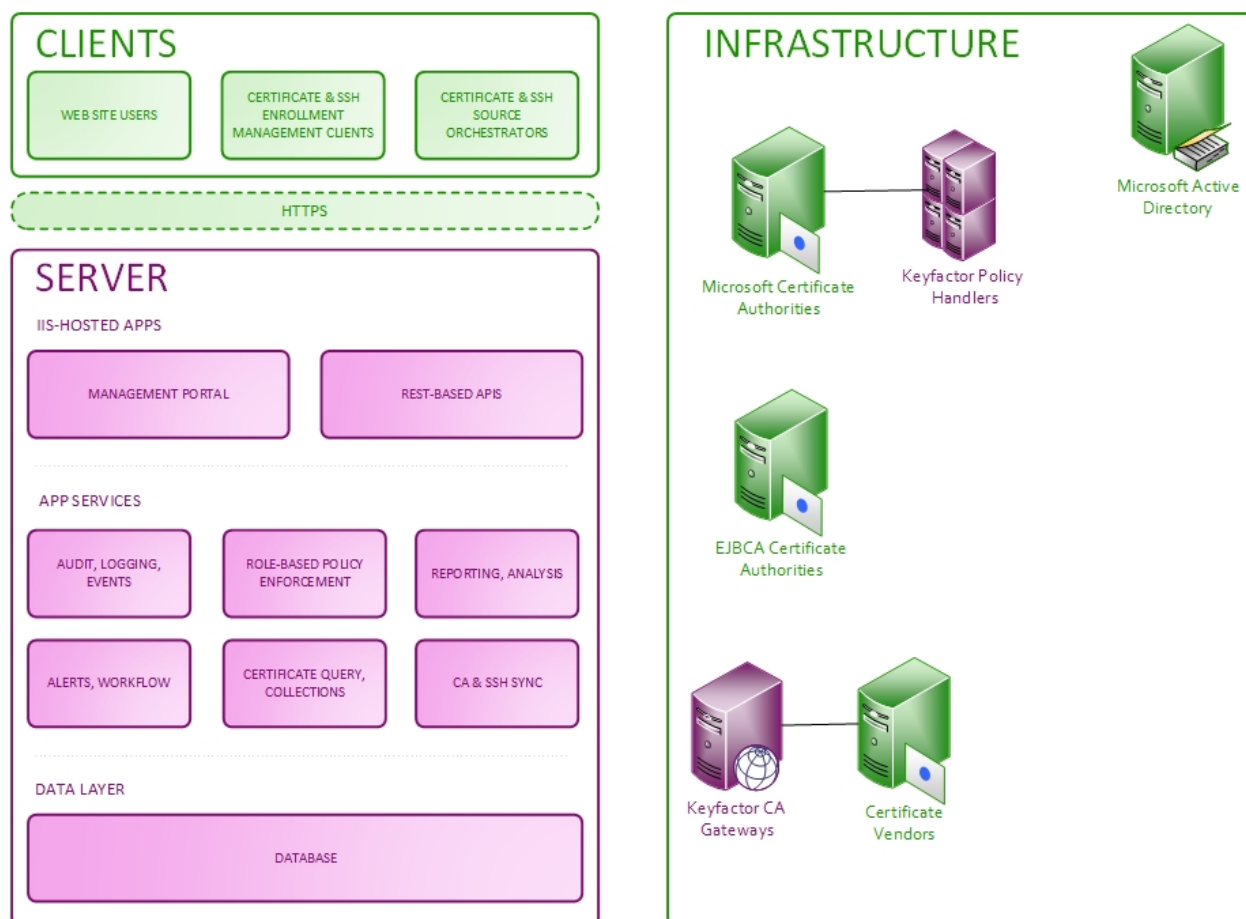


Figure 1: Keyfactor Command Logical Architecture Diagram

The Keyfactor Command solution includes the following logical components:

- Client / Orchestrator Tier:
 - Certificate Enrollment and Management Tools—While many certificate management functions can be performed in a completely agentless fashion, Keyfactor Command provides a number of enrollment and management tools to enable enhanced functionality where needed.
 - Certificate Source Orchestrators (aka Agents) and Gateways—Keyfactor Command gathers information about an enterprise's certificates and SSH keys from a number of different sources, including Microsoft and EJBCA CA databases, SSL scans, SSH key scans, API-based import, Java keystores, PEM certificate

stores, F5 devices, NetScaler devices, Amazon Web Services (AWS) locations, select certificate vendor certificates via gateways, and manual import through the Keyfactor Command Management Portal.

- **Web Tier:**
 - **Management Portal**—Keyfactor Command includes a web-based Management Portal that provides a PKI operations dashboard for administrators. It also enables certificate officers to easily search for and locate certificates and then perform management functions on them such as revocation or recovery. In addition, Keyfactor Command allows every certificate to be tagged with additional customer-defined metadata about the certificate, such as points of contact, certificate/app owners, etc. From within the Management Portal, administrators can inventory and manage secure shell (SSH) keys across the enterprise, while users can issue new SSH keys.
 - **Enrollment Web Pages**—Keyfactor Command includes issuance capabilities to a wide array of platforms, including Mac auto-enrollment, PKCS#12-based certificate issuance, and web-based CSR submission for administrator enrollment. PKCS#12 (PFX) and CSR enrollment are supported against Microsoft CAs in the local forest, remote CAs—with or without trust relationships—and non-domain-joined Microsoft and EJBCA CAs.
 - **Web APIs**—Keyfactor Command is implemented with a robust and continually growing set of APIs that allow integration of Keyfactor Command functionality with the set of Keyfactor Command clients and orchestrators, as well as third-party or customer-created software or scripts.
- **App Tier:**
 - **Event History and Audit Logging**—Keyfactor Command maintains a record of operations that are performed on a certificate and the individual who performed the operation. This includes information such as initial synchronization date, additions to and removals from certificate stores, certificate recovery, and certificate revocation.
 - **Role-Based Policy Enforcement**—Keyfactor Command offers a rich, role-based permissions model that allows you to create your own roles as needed within the Keyfactor Command Management Portal. Users can be assigned to roles based upon Active Directory group memberships or individually, and then each role can be assigned granular Keyfactor Command permissions such as report creation, certificate revocation or renewal, or metadata update.
 - **Dashboard and Report Engine**—Keyfactor Command contains a dynamic dashboard along with several built-in reports generated using the Logi Analytics Platform.
 - **Certificate Query & Collections**—Keyfactor Command allows certificate administrators to query the certificate database using various search criteria. In addition, the bulk of Keyfactor Command's reporting and automated notification functionality can be driven through certificate collections, which are a user-definable mechanism that allows organizations to report on groups of certificates based on selection criteria.
 - **Validation Service**—The Keyfactor Command Validation Service implements a Keyfactor-patented vSCEP™ technology that allows secure control of on-device key generation during certificate enrollment for iOS and Mac devices without the use of client-side orchestrators.
 - **Workflow Builder**—The workflow builder in Keyfactor Command allows you to easily automate event-driven tasks when a certificate is requested or revoked. The workflows can be configured with multiple steps between the start and end of the operation that offer a simple way to configure notifications, approvals, and end-to-end automation. This provides for operational agility in an intuitive and easy-to-configure manner. The workflow builder is highly customizable with options to execute PowerShell

scripts, invoke REST requests, send email messages, and require one or more approvals built in, and facilities to build custom steps to allow many more functions to be built as needed.

- Alert Notice Generator—Keyfactor Command allows you to configure customized email notifications for impending certificate expiration, revocation expiration, pending certificate requests, issued certificate requests and denied certificate requests. These notifications can be sent at configurable intervals, and may contain ASCII or HTML content, along with relevant information about the certificate or request in question (e.g. subject DN, issuer, thumbprint, template, custom metadata, etc.)
- Certificate Request Alerting—Keyfactor Command provides interfaces through which administrators can request certificates that require CA-level manager approval, interfaces where the approvers can either issue or deny the certificate request, and interfaces where the requesters can then download the certificates. This, along with the notice generator, provides an end-to-end flow for certificate requests that require CA-level manager approval.
- Alert Handlers—In addition to the notice generator that provides email alerts for SSH key and certificate expiration and enrollment workflow, Keyfactor Command also provides optional handlers that can be used in the certificate request and expiration alerts to output the information to the event log rather than sending it via email, run a PowerShell script, or automatically renew expiring certificates that are found in certificate store locations.
- Keyfactor Command Service—The Keyfactor Command Service (a.k.a. the timer service) is designed to continually keep the Keyfactor Command SQL database synchronized with the contents of every configured Microsoft and EJBCA CA database in the organization as well as external certificates located on servers it can scan. The service can perform full or partial scans of different CAs at user-defined intervals. This enables a rapidly-accessible, easily queried mirror of CA database information that can then be put to use via Keyfactor Command. Synchronization of CA information is supported for Microsoft CAs joined to the local forest, remote domain-joined Microsoft CAs—with or without trust relationships—and non-domain-joined Microsoft and EJBCA CAs. The Keyfactor Command Service is also responsible for executing a variety of periodic tasks, including scheduled reports, alerts and cleanup jobs.
- Data Tier:
 - SQL Database—Keyfactor leverages a Microsoft SQL Server database to store the information that Keyfactor Command uses.
- Microsoft Certification Authority Components:
 - RFC 2818 Policy Handler—The RFC 2818 Policy Handler integrates with the Microsoft CA to allow you to automate the addition of a DNS SAN matching the CN of the requested certificate for selected templates.
 - SAN Attribute Policy Handler—The SAN Attribute Policy Handler allows the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag.
 - vSCEP™ Policy Handler—The vSCEP™ Policy Handler integrates with the Microsoft CA and facilitates the enforcement of the Keyfactor-patented vSCEP™ technology mentioned above. This component is used to validate certificate request information for certificate requests submitted to the vSCEP API.
 - Whitelist Policy Handler—The Whitelist Policy Handler integrates with the Microsoft CA to allow you to restrict certificate enrollment on that CA for a configured certificate template or templates to only

designated client machines. This allows you, for example, to force certificate enrollment for web server certificates to be accepted only via the Keyfactor Command Management Portal and denied when coming from the Microsoft certificates MMC or IIS on the target servers for web server certificates.

- Enterprise Infrastructure:
 - Certification Authorities—Keyfactor Command has been built from the ground up to make it easier to operate organizational PKIs. This allows you to benefit from Keyfactor Command’s extended features around Microsoft CA capabilities such as certificate templates, enrollment and recovery agents, and private key recovery. Keyfactor Command's integration with EJBCA provides support for capabilities such as certificate profiles, end entity profiles, enrollment, and revocation.
 - Microsoft Active Directory—Keyfactor Command is integrated with Microsoft Active Directory, using AD for authentication, supporting group memberships for Keyfactor Command Role assignments, using AD for Microsoft CA and certificate template enumeration, and for the inclusion of AD account attributes in the content of issued certificates.

2.2 Physical Architecture

Figure 2: Keyfactor Command Physical Architecture Diagram shows the physical architecture of the Keyfactor Command solution.

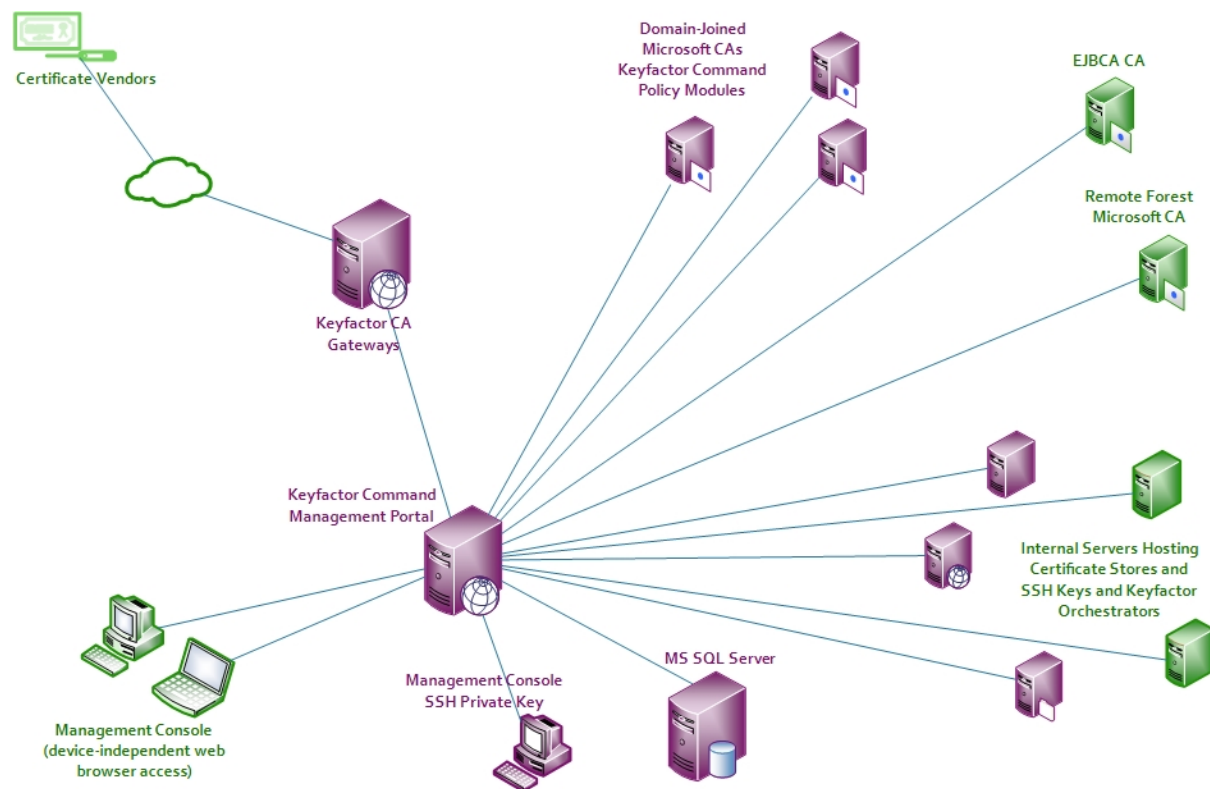


Figure 2: Keyfactor Command Physical Architecture Diagram

For simplicity, the servers in Figure 2: Keyfactor Command Physical Architecture Diagram are shown as single physical instances. In practice, these servers may be virtual machines and may be load balanced or clustered to meet

availability or performance requirements. The diagram includes some optional components—including the Keyfactor vendor gateways and Keyfactor orchestrators—which are not covered in this guide. For more information about these components, see the *Keyfactor Orchestrators Installation and Configuration Guide* and the documentation for each of the gateways.

- Keyfactor Command-Dedicated Servers¹:
 - Keyfactor Command Server—This server hosts the Keyfactor Command Management Portal, the Keyfactor Command vSCEP™ and Services roles, and the Logi Analytics Platform for report generation. These roles run as ASP.NET (4.5 or higher) applications on IIS. Both Windows Server 2019 and 2022 are supported.
- Enterprise-Shared Servers:
 - Microsoft SQL Server—Keyfactor Command supports Microsoft SQL Server 2016 with cumulative update (CU) 2 or higher, 2017, 2019 and 2022 all with TLS encryption enabled for its primary database. While a dedicated SQL deployment is certainly an option, many organizations maintain a well-established SQL server farm to support multiple applications within the organization; if preferred, Keyfactor Command can easily make use of such a service. Keyfactor does not recommend locating the Keyfactor Command roles on the SQL server in a production deployment.
 - Web Reverse Proxy—If Internet-based access is required, the Keyfactor Command services can be published through a variety of reverse proxy products such as Microsoft UAG/TMG, F5, SiteMinder, or NetScaler.
 - Network-based Hardware Security Module (HSM not pictured)—In certain configurations, Keyfactor Command requires the use of Enrollment Agent (EA) and/or Key Recovery Agent (KRA) certificates. To provide additional security over these certificates' private keys, Keyfactor strongly recommends the use of a Hardware Security Module (HSM) such as the Thales NetHSM if these features will be used.

2.3 Solution Design

Keyfactor Command supports a number of different deployment architectures to help provide for different needs from small and simple to highly available. The solution can be as simple as one Keyfactor Command server hosting all the Keyfactor Command roles (other than the policy handlers, which are installed on a Microsoft CA) or the roles can be separated onto different machines to provide increased security or distribute the load. Redundant servers can be added to provide for high availability—either within the same data center or across data centers. Keyfactor expects that the specifics of a high availability deployment plan would be finalized as part of the project rollout.

¹The roles described in this section may be co-located on a single physical or virtual server or may be further separated to multiple machines.

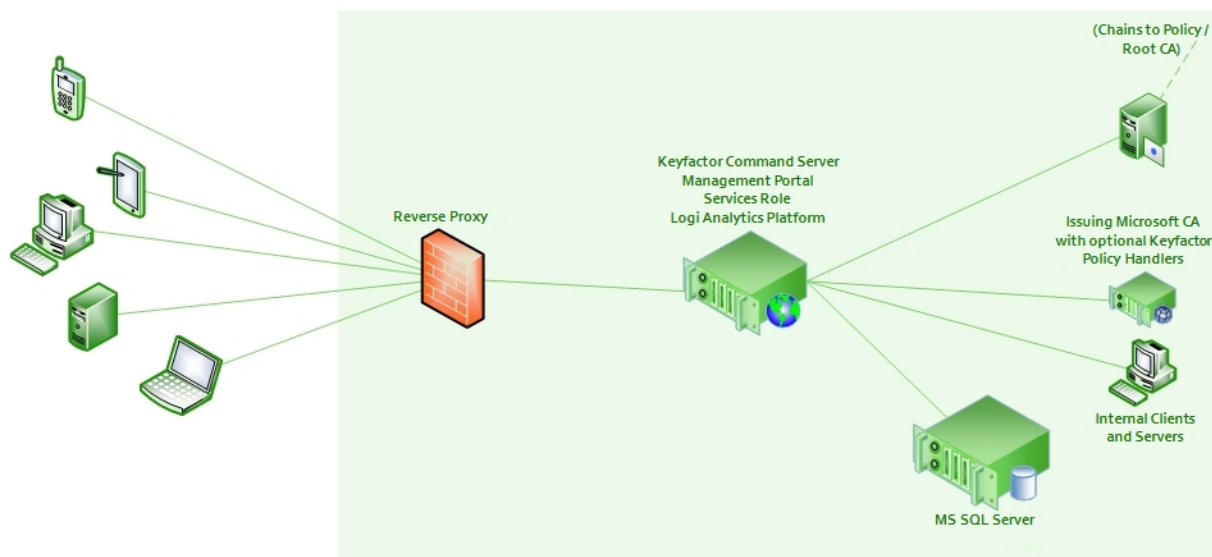


Figure 3: Simple Keyfactor Command Solution Design

2.4 Keyfactor Command Server

The Keyfactor Command solution by Keyfactor allows you to issue and manage certificates across enterprise infrastructures to allow you to achieve end-to-end visibility, control, and automation across all your machine identities so you can turn the impossible into the possible.

2.4.1 System Requirements

[Table 1: System Requirements](#) provides the recommendations for minimum system specifications used by Keyfactor Command components. All servers may be deployed as virtual machines and may be part of a clustering or load-balanced architecture, if desired. If the Keyfactor Command roles are co-located, the specifications may need to be scaled accordingly. All Microsoft-supported methods for making SQL Server highly available are supported. For most high availability requirements, Keyfactor recommends using always on availability groups (see [SQL Server on page 11](#)).

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>

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. 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.

Table 1: System Requirements

Component	Minimum Requirements
Keyfactor Command Server (Management Portal, vSCEP™ and Services roles)	Windows Server 2019 or 2022 Internet Information Services (IIS) with Basic Authentication, Windows Authentication, ASP.NET 4.7 or greater, and the Active Directory Module for Windows PowerShell (see Install IIS and .NET on the Keyfactor Command Server on page 30) .NET Framework 4.7.2 or greater 4 GB RAM, 2 GHz CPU, 40 GB disk
Microsoft SQL Database	Microsoft SQL Server 2016 with cumulative update (CU) 2 or higher, 2017, 2019, 2022 all with TLS encryption enabled and compatibility level 130 or higher. 8 GB RAM, 2+ GHz CPU (>= 2 cores), 500 GB disk
Browser to Access the Management Portal	Chrome 65.0.3325+, Firefox 59.0+, or Microsoft Edge 42.17134+
Keyfactor Command Server Upgrade	Keyfactor Command version 6.1.0 or later is required to upgrade to Keyfactor Command version 9.0 or later.
EJBCA CA	EJBCA Enterprise version 7.8.1 or later is supported. The EJBCA REST API must be enabled to interoperate with Keyfactor Command (see System Configuration -> Protocol Configuration in the EJBCA administration portal).



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>

2.4.2 Planning & Preparing

Before you install Keyfactor Command, you need to consider the components that make it up and its dependencies and decide where you want each role to reside, which roles—if any—you want to be highly available, and which features you're going to enable. It's possible to start with a non-redundant implementation and then add redundancy at a later time, but it's best to plan ahead for this if it's the desired goal.

Your license for Keyfactor Command may not include all the roles described in this document, so some sections of this guide may not apply to your implementation.

Once you've made these planning decisions, you then need to follow the steps outlined in this section that need to be taken prior to a Keyfactor Command implementation to install the prerequisites, create the required supporting components, and gather the necessary information to complete the Keyfactor Command installation and configuration process.

2.4.2.1 Certificate Authorities

In most cases, if you are installing Keyfactor Command then you have at least one Microsoft or EJBCA Certificate Authority (CA) in your environment. As you're planning for Keyfactor Command, you'll need to make the following decisions about the CA(s) and certificate templates for your environment:

- Which CAs should be synchronized to the Keyfactor Command database?

Your license may not allow you to synchronize all of your CAs. Certificates belonging to offline root or policy CA "chain" certificates can be monitored without impacting your license.



Note: As of version 9.0, Keyfactor Command has implemented a constraint that prevents any two certificate authorities from having the same logical name and host name combination. Think about the logical name and host name configuration of the CAs that will be implemented with Keyfactor Command and check for duplicates.

- If you have Microsoft CAs, what authorization method will you use to configure the CAs (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 96](#))?
- If your Keyfactor Command license includes certificate enrollment:
 - Which CA(s) will be used to issue certificates based on CSRs through the Keyfactor Command Management Portal?
 - Which CA(s) will be used to issue PFXs through the Keyfactor Command Management Portal?
 - Which certificate template(s) will be configured for CSR enrollment in the Keyfactor Command Management Portal? In most cases, these templates will already exist.
 - Which certificate template(s) will be configured for PFX enrollment in the Keyfactor Command Management Portal? In most cases, these templates will already exist.
- If your Keyfactor Command license includes Mac auto-enrollment, which CA(s) will be used to automatically issue certificates to Macs running the Mac auto-enrollment agent?

As part of the Keyfactor Command installation preparation, you may need to create a certificate template for this purpose.



Tip: This information is not needed to complete the initial installation and configuration, but will be needed to do post-installation configuration in the Keyfactor Command Management Portal.

2.4.2.2 SQL Server

Keyfactor Command uses a Microsoft SQL Server¹ database to store configuration and synchronized certificate information. Standard edition or above of SQL Server is required. In a production implementation, Keyfactor recommends that SQL Server be installed on a separate server from the Keyfactor Command roles.

Although you can implement a SQL server especially for Keyfactor Command, in many environments an existing shared SQL server or cluster is used. Keyfactor Command creates one database with a user-defined name and can successfully co-exist with other databases in the same SQL instance.

SQL should be installed with a case-insensitive collation setting.



Note: Microsoft SQL 2016 CU2, 2017, 2019, and 2022 all with TLS encryption enabled are supported.

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 from Keyfactor 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.
```

Log message will look something like:

```
2022-09-09 11:35:13.0142 CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel [Error] -
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.
2022-09-09 11:35:13.0142 CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel [Error] -
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.
at CSS.CMS.Install.ConfigurationWizard.ViewModels.DatabaseViewModel.a(Object A_0, RunWork-
erCompletedEventArgs A_1)
A connection was successfully established with the server, but then an error occurred during the
```

¹Microsoft SQL Server 2016 with cumulative update (CU) 2 or higher, 2017, 2019, 2022 all with TLS encryption is supported.

```
login process. (provider: SSL Provider, error: 0 - The certificate chain was issued by an authority that is not trusted.)
```

To acquire a new SSL certificate or check for an existing certificate, see [Using SSL to Connect to SQL Server on the next page](#).

If you would prefer not to use an encrypted channel for your connection to SQL, see [Configurable SQL Connection Strings on page 17](#).

Database Encryption

Keyfactor Command uses Microsoft SQL Server column encryption with the ENCRYPTBYKEY and DECRYPTBYKEY cryptographic functions to protect sensitive data. The type of data protected in this way includes:

- Service account credentials
- SMTP credentials
- Certificate store passwords
- Certificate and pending certificate request private keys
- API secrets
- The 64-byte key used to sign audit log records

SQL encryption is built in to the product and cannot be disabled. In addition to SQL encryption, Keyfactor Command offers optional application-level encryption. This option allows you to encrypt select sensitive data stored in the Keyfactor Command database using a separate encryption methodology utilizing a Keyfactor Command-defined certificate on top of the SQL server encryption. This additional layer of encryption protects the data in cases where the SQL Server master keys cannot be adequately protected. For more information, see [Application-Level Encryption on page 28](#).

Database Backup

Backup of the SQL server Database Master Key (DMK) for the Keyfactor Command database is of critical importance in database backup and recovery operations. The backup file of the DMK and the password for it should be stored in a safe, well-documented location. Without the file and password created with this process, some data that is encrypted within the Keyfactor Command database will be unrecoverable in a disaster recovery scenario. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

High Availability

For a highly available solution, Keyfactor recommends using always on availability groups. The availability groups feature of SQL Server sits on top of Windows Server failover clustering and provides the ability to automatically synchronize multiple copies of databases across geographically dispersed SQL Servers. Although the availability groups feature relies on Windows clustering, it does not require shared storage, so it is appropriate for a geo-redundant deployment. The availability groups feature is the current recommended solution from Microsoft. Because Keyfactor Command makes use of SQL database encryption, when availability groups are configured, the

Keyfactor Command service master key (SMK) must be synchronized between all participating nodes in the availability group. This can be accomplished by backing up the SMK from one SQL server and restoring it to the other servers in the availability group. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

Using SSL to Connect to SQL Server

By default, Keyfactor Command connects to SQL using an encrypted connection using an SSL certificate configured on your SQL server.

You can check whether your SQL server has been configured with an SSL certificate in one of two ways:

SQL Server Configuration Manager

1. On the SQL server, open the SQL Server Configuration Manager and drill down under SQL Server Network Configuration to find *Protocols for [YOUR INSTANCE NAME]*.
2. Right-click on Protocols for [YOUR INSTANCE NAME] and choose **Properties**.
3. Check the Certificate tab of the Properties dialog to see if a certificate has been configured and is still valid. If your certificate has a friendly name, it will appear here listed in the dropdown by its friendly name.



Important: A certificate will only appear here if it has a CN¹, usually the FQDN of the SQL server. If a certificate has been configured without this, it will not appear to be configured through this UI.

¹The Subject property of the certificate must indicate that the common name (CN) is the same as the host name or fully qualified domain name (FQDN) of the server computer or it must match the DNS suffix if using a wildcard certificate. When using the host name, the DNS suffix must be specified in the certificate. If SQL Server is running on a failover cluster, the common name must match the host name or FQDN of the virtual server and the certificates must be provisioned on all nodes in the failover cluster.

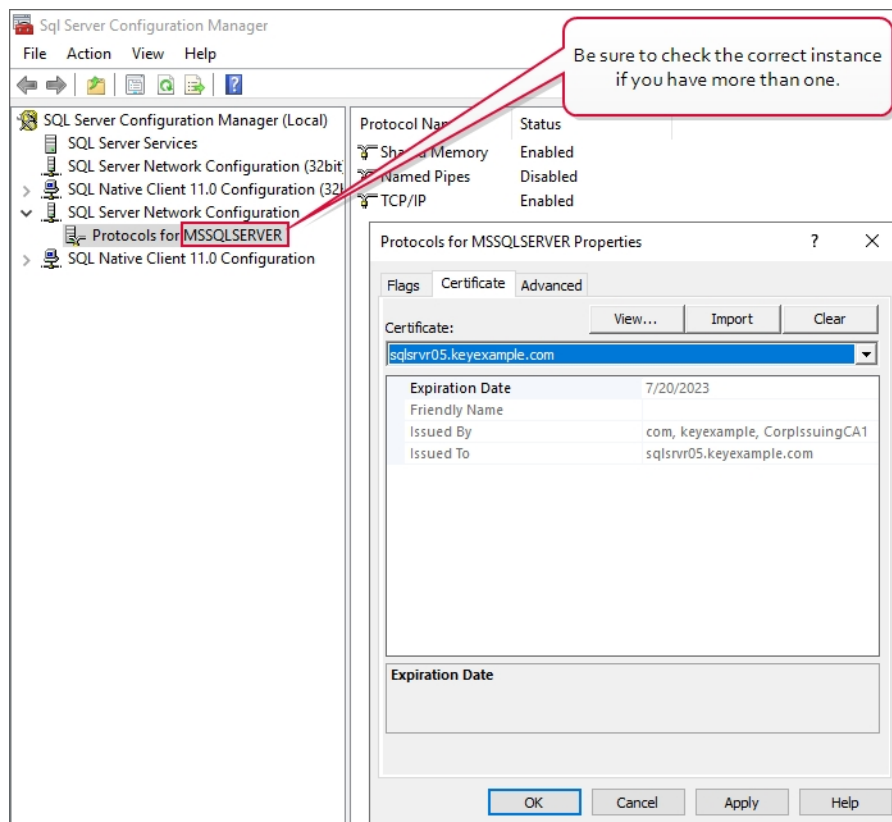


Figure 4: SQL Server Configuration Manager View Active SSL Certificate

Registry

1. On the SQL server, open the registry editor and browse to (where `[MSSQL15.MSSQLSERVER]` is the correct version of SQL server for your server):


```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\
[MSSQL15.MSSQLSERVER]\MSSQLServer\SuperSocketNetLib
```
2. In the SuperSocketNetLib registry key, look for a Certificate value.
3. Validate that the Certificate value has a thumbprint configured. This should match the thumbprint of an active certificate with a Server Authentication EKU in the Local Machine certificate store.

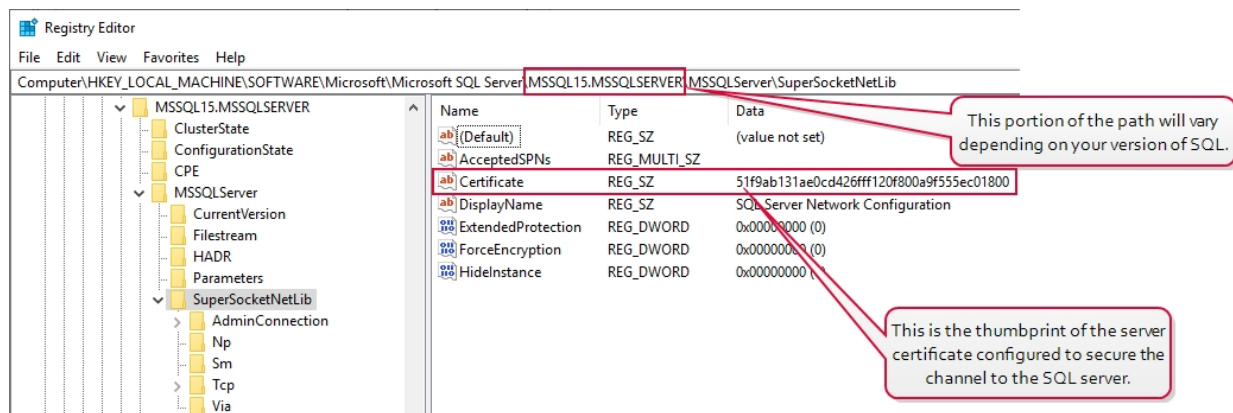


Figure 5: Registry View Active SSL Certificate

To acquire a new certificate for your SQL server:

1. On the SQL server open the Services.msc MMC and scroll down to locate the *SQL Server ([YOUR INSTANCE NAME])* service.
2. Check the *Log On As* column for the name of the service account that the service is running as.

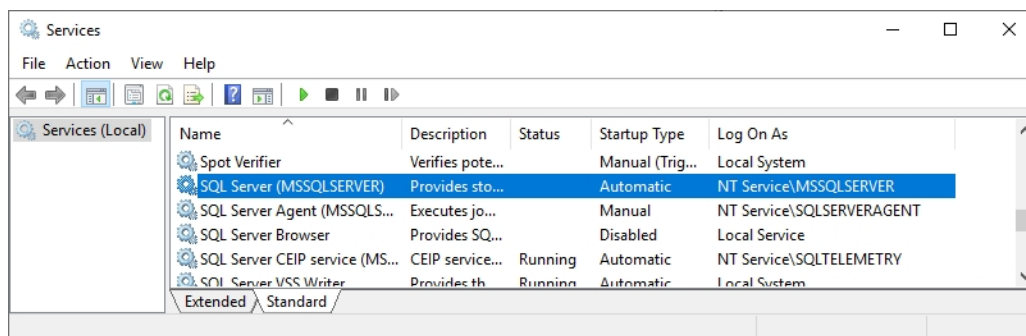


Figure 6: View SQL Server Services

3. Identify a template with a Server Authentication EKU (a typical web server template).
4. On the SQL server, do one of following:
 - Using the GUI:
 - a. Open an empty instance of the Microsoft Management Console (MMC).
 - b. Choose **File->Add/Remove Snap-in....**
 - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
 - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
 - e. Click **OK** to close the Add or Remove Snap-ins dialog.

- Using the command line:
 - a. Open a command prompt using the "Run as administrator" option.
 - b. Within the command prompt type the following to open the certificates MMC:
certlm.msc
- 5. Enroll for the certificate using your preferred method, being sure to give the certificate a CN (it will not appear in the configuration tool without this) and add subject alternative names (SANs) to it for all the IP addresses, server names, and FQDNs that you might use to reference the SQL server when communicating with it, including DNS aliases. Install it, along with its private key, into the Local Machine certificate store on the SQL server. One way to do this is in the certificates MMC:
 - a. Drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate...**
 - b. Follow the certificate enrollment wizard, selecting the template you identified for this purpose, and providing appropriate SANs along with any required information.

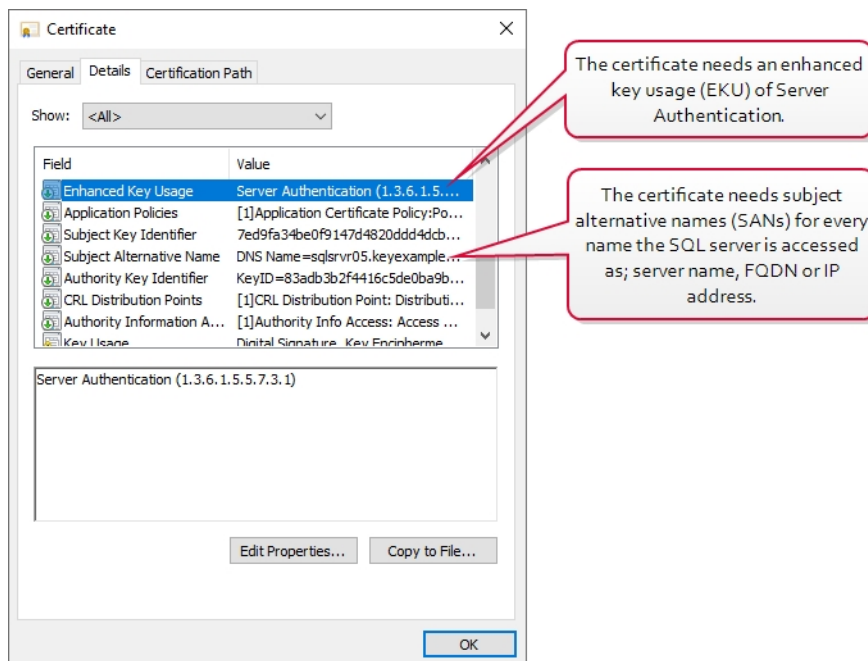


Figure 7: SQL Server SSL Certificate Details

6. Drill down to the Personal folder under **Certificates** for the Local Computer, locate your certificate, right-click, and choose **All Tasks->Manage Private Keys...**
7. In the Permissions for private keys dialog, click **Add**, add the SQL service account, and grant that service account **Read** but not **Full control** permissions. If the SQL server is running as *NT Service\[YOUR INSTANCE NAME]* as shown in [Figure 6: View SQL Server Services](#), be sure to change the location to your local machine and enter the object name as "NT SERVICE\[YOUR INSTANCE NAME]" as shown in [Figure 8: Grant Private Key Permissions for SQL Server](#).

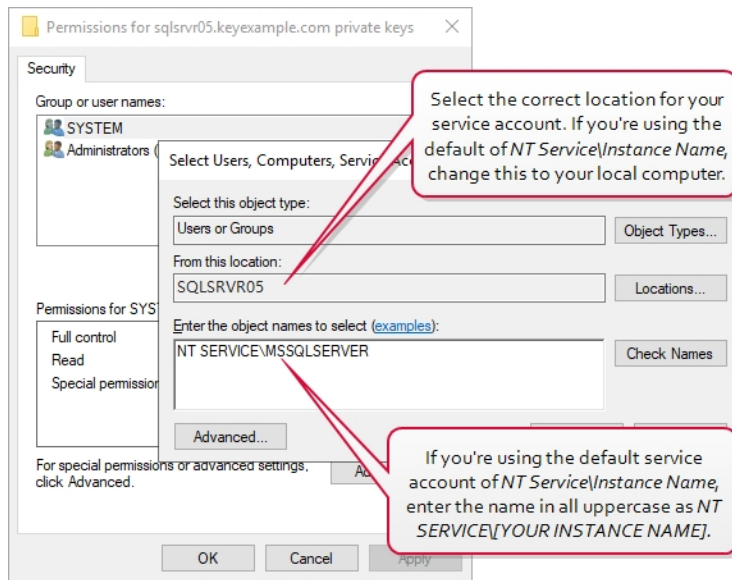


Figure 8: Grant Private Key Permissions for SQL Server

8. Click **OK** to save.
9. Configure the SSL certificate in SQL using either the SQL Server Configuration Manager or registry as shown above for checking whether there is an existing certificate configured (see [SQL Server Configuration Manager on page 13](#)).
10. After you've acquired a new certificate, made the private key permission changes, and associated it in SQL, you'll need to restart the *SQL Server (Instance Name)* service (see [Figure 6: View SQL Server Services](#)) before these changes will take effect.

Configurable SQL Connection Strings

Keyfactor Command supports using a "template" SQL connection string that can be created to fit the needs of the overall deployment. This template will be used as a starting point and will not be overwritten by the configuration wizard. For instance, you can set the timeout setting in one place, and once the configuration wizard is run, this is reflected in all places where a connection string is used. The template can be changed at any time to update the connection strings.

To create a customized connection string, after installing the Keyfactor Command software but before running the configuration wizard, modify both the EFModels and SqlDirect connection strings in the **SharedSqlConnectionStrings.config** file found in the *Configuration* folder under your installation directory. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration

The settings that can be modified are described in the following Microsoft article:

<https://docs.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection.connectionstring?view=dotnet-plat-ext-6.0>



Note: The *Data Source*, *Initial Catalog*, *Integrated Security*, *User ID* and *Password* settings are reserved for the configuration wizard to use to configure and save the authentication for the connection string, but other settings found in the template string are left as-is.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add name="EFModels" connectionString=
    "metadata=res://*/EFModels.csdl|res://*/EFModels.ssdl|res://*/EFModels.msl;provider=Microsoft.Data.SqlClient;provider
    connection string="data source=dev;initial catalog=test;integrated security=True;persist security
    info=True;MultipleActiveResultSets=True;App=EntityFramework";" providerName="System.Data.EntityClient" />
  <add name="SqlDirect" connectionString="data source=dev;initial catalog=test;integrated security=True;persist
    security info=True;timeout=360;" />
</connectionStrings>
```

Note that *Data Source*, *Initial Catalog*, *Integrated Security*, *User ID* (not shown) and *Password* (not shown) are reserved for the configuration wizard to use to configure and save the authentication for the connection string.

Figure 9: Default SQL Connection Strings

If you prefer to connect to your SQL server over a non-encrypted channel (and thus avoid configuring an SSL certificate for your SQL server), you can use the *Encrypt* keyword in the connection strings with a value of *False*.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add name="EFModels" connectionString=
    "metadata=res://*/EFModels.csdl|res://*/EFModels.ssdl|res://*/EFModels.msl;provider=Microsoft.Data.SqlClient;provider
    connection string="data source=dev;initial catalog=test;integrated security=True;persist security
    info=True;encrypt=False;MultipleActiveResultSets=True;App=EntityFramework";" providerName=
    "System.Data.EntityClient" />
  <add name="SqlDirect" connectionString="data source=dev;initial catalog=test;integrated security=True;persist
    security info=True;timeout=360;encrypt=False;" />
</connectionStrings>
```

To connect to make a non-encrypted connection to SQL, add *Encrypt=False* to each of the connection strings.

Figure 10: SQL Connection Strings with Encrypt Channel Disabled

2.4.2.3 Keyfactor Command Server(s)

A Keyfactor Command server implementation is made up of several Keyfactor Command roles:

Keyfactor Command Management Portal

The server with this role provides the web-based administration interface that is used to view and report on certificates issued in the environment and enroll for certificates for use on servers. This role runs under Microsoft IIS. Configuration for the Keyfactor Command implementation as a whole is also done through the Keyfactor Command Management Portal. The Logi Analytics Platform for reporting is hosted on the server with this role.

Keyfactor Command Windows Services

The server with this role hosts back-end services required to support Keyfactor Command. This includes the Keyfactor Command Service, which is used for all periodic tasks throughout Keyfactor Command, including CA

synchronization, monitoring alerts, and report automation.

Keyfactor Command Web API

The server with this role hosts the Web APIs—the newer Keyfactor API and the older Classic API. The newer Keyfactor API is also included in the Management Portal role, since the Management Portal makes extensive use of this API.

Keyfactor Command Orchestrator Service API

The server with this role hosts the back-end service for receiving requests from and sending requests to Keyfactor agents and orchestrators.

Keyfactor Command vSCEP Validation Service

The server with this role hosts the back-end service for validating SCEP requests.

In many environments, the Keyfactor Command Management Portal, Windows Servers, Web API, and Orchestrator Service API roles are collocated on a single server (or pair of servers if redundancy is desired). The vSCEP Validation Service is an optional role that is only installed in environments where SCEP validation is required. Both physical and virtual servers are supported.

Keyfactor does not recommend installing the Keyfactor Command Management Portal, Windows Servers, Web API, Orchestrator Service API, or vSCEP Validation Service role on a CA or on a SQL server in a production environment.

As you plan for Keyfactor Command, you need to decide upon an architecture for the implementation and prepare servers with sufficient resources accordingly. See [System Requirements on page 8](#) for more information about planning for servers with sufficient resources to support the planned roles.

Licensing

The Keyfactor Command product is licensed by component. Your license for Keyfactor Command may not include all the features described in this guide. If you choose to add additional components to your Keyfactor Command license in the future, these features can generally be configured without the need to reinstall Keyfactor Command.

2.4.2.4 Create Active Directory Service Accounts for Keyfactor Command

Several of the Keyfactor Command roles operate under an Active Directory service account. You can either create a single Active Directory service account for all these roles or create separate service accounts for each role. If multiple Keyfactor Command roles will be installed on the same server, some of the below roles will be redundant. The roles that require a service account are:

Keyfactor Command Installer

The user who runs the Keyfactor Command installation must have local administrator permissions on the Keyfactor Command server(s) and must be granted permissions in SQL if Windows authentication for SQL will be used during the installation (see [Grant Permissions in SQL on page 30](#)). You can either grant these permissions to

an existing user or you can create a Keyfactor Command installer account and grant the appropriate permissions to this account.

Additionally, the user installing Keyfactor Command must have the SeBackupPrivilege and SeRestorePrivilege rights on the Keyfactor Command server. Normally, administrators are granted these permissions by default, but you should confirm the permissions prior to starting the install. These permissions can be set through Group Policy or Local Security Policy, and can be found under "Local Policies\User Rights Assignment" as "Back up files and directories" and "Restore files and directories".

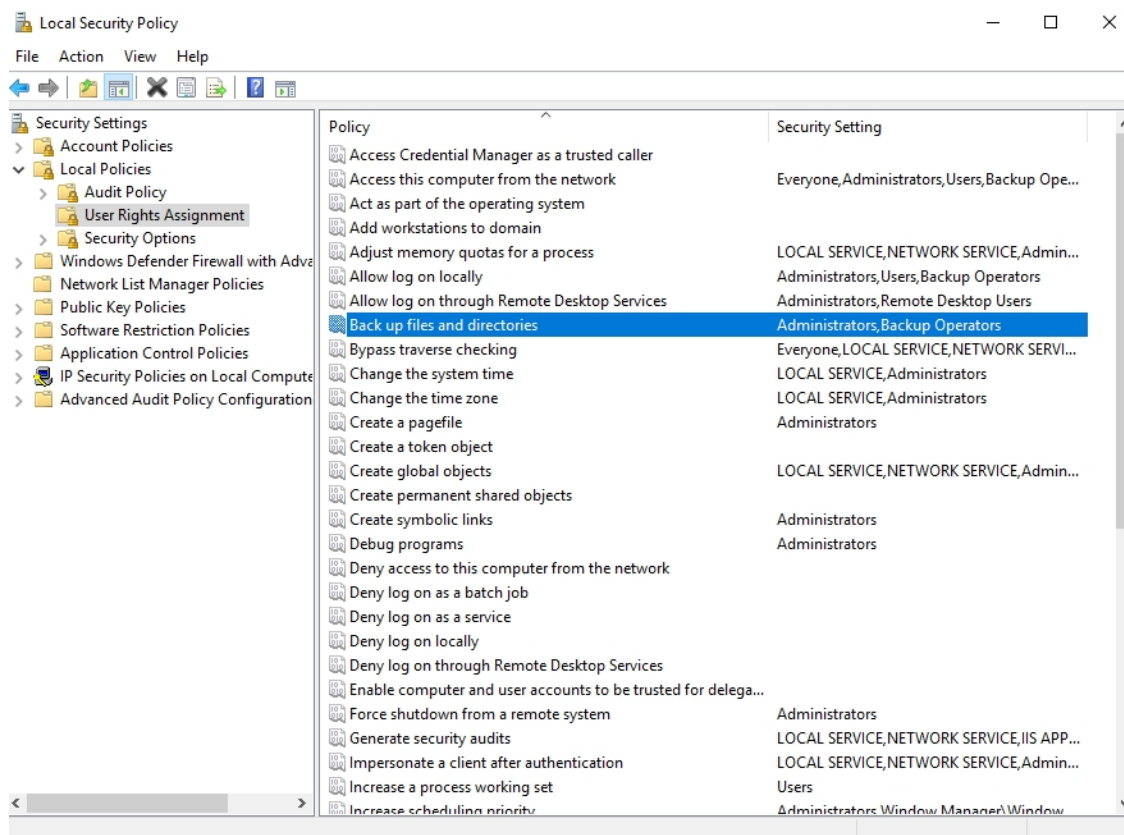


Figure 11: Local Security Policy

For more information on this from Microsoft, see:

<https://docs.microsoft.com/en-us/windows/win32/api/userenv/nf-userenv-loaduserprofilea#remarks>

Keyfactor Command Service

The Keyfactor Command Service (a.k.a. the timer service) runs on the Keyfactor Command services server. It synchronizes certificates to the SQL database and initiates notification and reporting tasks. This service runs in the context of an Active Directory Service account. Users assigned to this role will be granted permission on each of the schemas (dbo, ssl, ssh, cms_agents, etc.) and permission on the encryption certificate in SQL through the keyfactor_db_role which is created during configuration.

Keyfactor Command Management Portal (Application Pool)

The Keyfactor Command Management Portal uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory service account. Users assigned to this role will be granted permission on each of the schemas (dbo, ssl, ssh, cms_agents, etc.) and permission on the encryption certificate in SQL through the keyfactor_db_role which is created during configuration.

Logi Report Access

If Basic authentication will be used to access the Keyfactor Command Management Portal, the Logi Analytics Platform uses a service account to allow Logi to connect to Keyfactor Command via the Keyfactor API to display the dashboard information. This service account is not required if integrated Windows authentication will be used for the Management Portal. The Keyfactor Command analytics application for dashboard and reporting uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory service account. This role is collocated with the Management Portal; a separate service account for this role is not needed as a single application pool will be used for both.

Keyfactor Command Orchestrators API

The Keyfactor Command Orchestrators API IIS application accepts connections from Keyfactor Command orchestrators and uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory service account. If this role will be installed on the server hosting the Keyfactor Command Management Portal role, a separate service account for this role is not needed as a single application pool will be used for both.

Keyfactor Command Keyfactor API

The Keyfactor Command Keyfactor API uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory service account. The Keyfactor API is an integral part of Keyfactor Command and is not an optional installation. The Keyfactor API can be configured to support custom applications. If the Keyfactor Command Keyfactor API role will be installed on the server hosting the Keyfactor Command Management Portal role, a separate service account for this role is not needed as a single application pool will be used for both.

Keyfactor Command Classic API

The Keyfactor Command Classic API (the classic or legacy API) uses an application pool under IIS to operate. This application pool runs in the context of an Active Directory service account. The Keyfactor Command Classic API may have been configured to support custom applications in previous versions of Keyfactor Command. If the Keyfactor Command Classic API role will be installed on the server hosting the Keyfactor Command Management Portal role, a separate service account for this role is not needed as a single application pool will be used for both.

EJBCA End Entity for EJBCA CA Access

Keyfactor Command supports synchronization of certificates and certificate enrollment from EJBCA certificate authorities by configuring a client certificate issued from the EJBCA CA on the CA record in the Management Portal. This client certificate needs to be associated with an end entity in EJBCA that can be assigned sufficient permissions to perform all necessary CA tasks from Keyfactor Command.

Explicit Credentials for Microsoft CA Access

Keyfactor Command supports synchronization of certificates and certificate enrollment from Microsoft certificate authorities in remote forests (forests other than the forest in which Keyfactor Command is installed which are not in a two-way trust with the Keyfactor Command forest) by configuring a service account from the forest in which the CA resides on the CA record in the Management Portal. All communication to retrieve existing certificates, enroll for new certificates, revoke certificates, and recover certificate keys from the remote CA is done in the context of this service account. Explicit credentials for remote CA access is configured in the Keyfactor Command Management Portal after installation is complete rather than in the configuration wizard.

You may need additional service accounts to support the use of Keyfactor Command orchestrators and/or gateways in your environment. Please see:

- *Create Service Accounts for the Universal Orchestrator* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- *Create a Service Account for the Keyfactor Bash Orchestrator* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- *Create Service Accounts for the Java Agent* in the *Keyfactor Orchestrators Installation and Configuration Guide*
- The installation guide for each gateway.

The service account(s) need to be created in Active Directory prior to installation of the Keyfactor Command software, and the person installing the Keyfactor Command software needs to know the service account(s) domain, username and password. The same service account may be used for multiple roles, if desired. For example, you might have one service account for orchestrators, another for gateways, and a third for all server roles.



Note: The Application Pool account must have read permission on any groups being created. This will allow Keyfactor Command to query for group membership on the groups.

Table 2: Typical Service Accounts

Account	Uses
Keyfactor Command Service Account	Keyfactor Command Service, Keyfactor Command Management Portal, Keyfactor Command APIs, Keyfactor Command Logi Report Access
Keyfactor Orchestrator Service Account	Keyfactor Orchestrator access to Keyfactor Command Server and Keyfactor Orchestrator on-machine operations, where applicable

2.4.2.5 Create Active Directory Groups to Control Access to Keyfactor Command Features

Keyfactor Command uses Active Directory groups to control access to the various Keyfactor Command features. The Keyfactor Command Management Portal supports multiple groups with different levels of access to the portal. During the installation, at least one group or user must be entered to grant full administrative access to the portal. After installation, additional groups can be configured through the Keyfactor Command Management Portal to grant more limited access to the portal.



Important: The built-in Active Directory groups Domain Admins and Enterprise Admins cannot be used directly to grant access to the Management Portal due to how these groups function within Windows. You can create a custom Active Directory group, reference that group in the Management Portal, and add the built-in Domain Admins or Enterprise Admins group to that custom group, if desired.

Groups that you may find it useful to identify or add following the initial installation include:

Keyfactor Command Enrollment

Users who are a member of this group or groups may use PFX and/or CSR enrollment through the Keyfactor Command Management Portal. Access control for enrollment is configured in the Keyfactor Command Management Portal after installation is complete.

Keyfactor Command My SSH Key

Users who are a member of this group or groups may acquire SSH keys through the Keyfactor Command My SSH Key portal. Access control for the My SSH Key portal is configured in the Keyfactor Command Management Portal after installation is complete.

Keyfactor Java Agents

Service accounts that are a member of this group are allowed to auto-register as Java agents in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Java and PEM certificate stores. This group is not required if auto-registration with user validation will not be used.

Keyfactor Bash Orchestrators

Service accounts that are a member of this group are allowed to auto-register as Bash orchestrators in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Bash orchestrators. This group is not required if auto-registration with user validation will not be used.

Keyfactor Mac Auto-Enrollment Users

Users who are members of this group are allowed to auto-register for Mac auto-enrollment in Keyfactor Command, if auto-registration is configured, providing for more hands-free management of Mac auto-enrollment. The same group may be used to grant users permissions on the template that will be used for Mac auto-enrollment. This group is not required if auto-registration with user validation will not be used and a different group will be used to grant permission on the template.



Note: The same group may be used for multiple roles. Existing groups may be used. For example, if all employees of your organization are members of the Active Directory Domain Users group and you wish to allow all employees to acquire SSH keys, you may use the Domain Users group for the Keyfactor Command My SSH Key function.



Tip: To grant access in the Management Portal to users from trusted forests, create a domain local group in the Active Directory domain in which Keyfactor Command is installed, put the cross-forest users and groups in this local group and grant access in Keyfactor Command to this domain local group.

2.4.2.6 Configure Certificate Chain Trusts for CAs

The Keyfactor Command server needs to trust the chain certificates for all the CAs you will reference within Keyfactor Command in order for all operations to complete successfully. In many environments, root and intermediate trusts for domain-joined Microsoft CAs are pushed out automatically. If this is not the case in your environment or if you are using EJBCA CAs, you will need to configure these chain trusts on the Keyfactor Command server manually.

The certificate for each root CA must be installed in the Trusted Root Certification Authorities store under Local Computer on the Keyfactor Command server. If your public key infrastructure (PKI) also has issuing CAs, the issuing CA certificates must be installed in the Intermediate Certification Authorities store under Local Computer on the Keyfactor Command server.

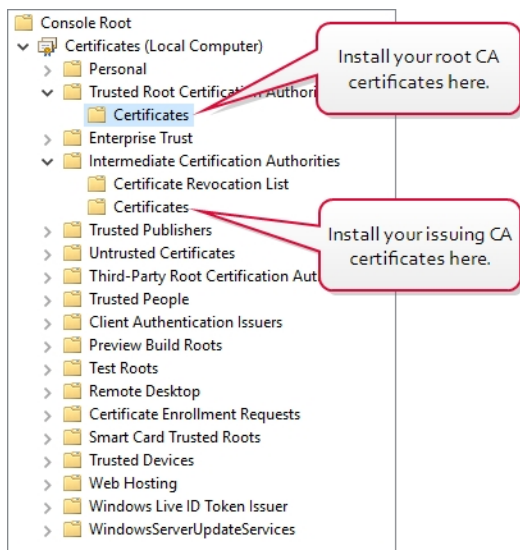


Figure 12: Install CA Chain Certificates on the Keyfactor Command Server

2.4.2.7 Hostname Identification and Resolution

Prior to the installation of Keyfactor Command, you need to determine the DNS alias(es) by which the Keyfactor Command roles will be accessed, if any, and configure them in your hostname resolution solution so that they will be resolvable prior to installation. For example, if you're licensed for SSH key management and wish to publish the My SSH Key portal externally to support SSH key acquisition by users outside the company firewall, you will probably wish to reference the server by a DNS alias rather than its actual hostname. For example, you may wish to use keyfactor.keyexample.com rather than websrvr23.keyexample.local. This is particularly significant if you will be using redundant servers with load balancing. For DNS aliases used internally, you will need to consider whether the servers to be accessed will be authenticated using Kerberos authentication. Out of the box, the Keyfactor

Command Management Portal uses integrated Windows authentication and will default to Kerberos authentication in most environments. Although some features of the Keyfactor Command Management Portal may support NTLM authentication in some environments, the dashboard and enrollment functions do not support NTLM. If you will be using Kerberos authentication, your DNS aliases need to be configured as "A" records rather than CNAME records because Kerberos does not function well with CNAME records under Microsoft IIS.

The roles for which you need hostnames during the Keyfactor Command installation are:

SQL Server

For a small environment you may choose to use the server's actual name. If you plan to use SQL clustering, you will need an alias that represents the cluster. Using an alias for the SQL server allows for database portability in the future.

Email

During the Keyfactor Command installation you configure the email server that will be used to send email notifications.

Keyfactor Command Management Portal

This is the primary management server and may hold all Keyfactor Command roles in a small implementation.

Keyfactor Command Logi Dashboard and Reports

This hostname must match the hostname entered for the Management Portal.

Keyfactor Command vSCEP Service

This hostname is only required if your Keyfactor Command license includes vSCEP™. If all Keyfactor Command roles are combined on one server, this will be the same hostname as used for the Keyfactor Command Management Portal.

Keyfactor Command Orchestrators API

This hostname is only required if your Keyfactor Command license includes orchestrator functionality. If all Keyfactor Command roles are combined on one server, this will be the same hostname as used for the Keyfactor Command Management Portal.

Keyfactor Command Keyfactor API

This hostname must match the hostname entered for the Management Portal unless you are installing a secondary instance of the Keyfactor API.

Keyfactor Command Classic API

This hostname is only required if you choose to enable this option for legacy support. Out of the box, the Keyfactor Command API PowerShell Client use the Classic API role. If all Keyfactor Command roles are combined on one server, this will be the same hostname as used for the Keyfactor Command Management Portal.

Centralized Logging Solution

This hostname is only required if you choose to enable the option to copy Keyfactor Command audit logs entries in real time, as they are generated, to a separate server for collection and analysis by a centralized logging solution (e.g. rsyslog, Logstash).

Prior to beginning the Keyfactor Command installation, ensure that the selected hostnames resolve successfully.

2.4.2.8 Firewall Considerations

Keyfactor Command needs to be able to communicate internally between the various Keyfactor Command components installed on different servers, if applicable, and to the SQL server, certificate authorities, centralized logging server (if applicable), and Active Directory. If there are any firewalls in the environment that control internal traffic, these may need to be updated to allow the appropriate level of communication. [Table 3: Protocols Keyfactor Command Uses for Communication](#) shows each Keyfactor Command component and the protocols they use to communicate. In addition, all Keyfactor Command components require a healthy Active Directory environment with the ability to use Kerberos, LDAP, and DNS.

Table 3: Protocols Keyfactor Command Uses for Communication

Keyfactor Command Component	Protocols and Ports	Target
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	Client browser (e.g. Microsoft Edge)
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	Certificate revocation list (CRL) distribution points
Keyfactor Command Management Portal	HTTP/HTTPS (TCP 80/443)	EJBCA Certificate Authorities
Keyfactor Command Management Portal	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Microsoft Certificate Authorities
Keyfactor Command Management Portal	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Keyfactor vendor gateways to cloud CAs (e.g. Entrust, Symantec)
Keyfactor Command Management Portal	MS SQL (default TCP 1433)	SQL Server
Keyfactor Command Management Portal	Varies depending on the implemented solution (TCP 514 for rsyslog, TCP 5000 for Logstash are some standard defaults)	Centralized logging solution
Keyfactor Command	Active Directory (TCP/UDP 389)	Microsoft Active Directory queries

Keyfactor Command Component	Protocols and Ports	Target
Keyfactor Command SSH Management	Active Directory Web Services (TCP 9389)	Microsoft Active Directory for group membership enumeration
All Orchestrators and Agents	HTTP/HTTPS (TCP 80/443)	Keyfactor Command Orchestrator API endpoint
Keyfactor Windows Orchestrator (IIS Certificate Stores)	PowerShell Remoting (default TCP 5985 and 5986)	IIS Servers to which PFX files will be distributed
Keyfactor Universal Orchestrator (IIS Certificate Stores)	PowerShell Remoting (default TCP 5985 and 5986)	IIS Servers to which PFX files will be distributed
Keyfactor Windows Orchestrator (SSL Endpoint Management)	Any configured for scanning	The SSL endpoint being scanned by the SSL discovery or monitoring job
Keyfactor Universal Orchestrator (SSL Endpoint Management)	Any configured for scanning	The SSL endpoint being scanned by the SSL discovery or monitoring job
Keyfactor Windows Orchestrator (F5 and NetScaler Certificate Store Management)	HTTP/HTTPS (TCP 80/443)	F5 or NetScaler Devices
Keyfactor Windows Orchestrator (Remote Certificate Authority)	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Microsoft Certificate Authorities
Keyfactor Universal Orchestrator (Remote Certificate Authority)	RPC/DCOM (TCP 135 plus random high ports typically in the range 49152 – 65535)	Microsoft Certificate Authorities
Keyfactor Windows Orchestrator (FTP)	FTP (TCP 21)	FTP Servers

Keyfactor Command Component	Protocols and Ports	Target
Keyfactor Universal Orchestrator (FTP)	FTP (TCP 21)	FTP Servers
Keyfactor Bash Orchestrator	SSH (TCP 22 by default)	Remote control targets for SSH management
Keyfactor Gateways to Cloud CAs	HTTP/HTTPS (TCP 80/443)	Cloud providers (e.g. Entrust, Symantec)
Keyfactor Cloud Gateway	Active Directory Web Services (TCP 9389)	Microsoft Active Directory for group membership enumeration

2.4.2.9 Acquire a Public Key Certificate for the Keyfactor Command Server

Keyfactor recommends using HTTPS to secure the channel between clients and the Keyfactor Command server(s). This requires at least one SSL certificate. You will need an SSL certificate or certificates for each of the hostnames you have identified (see [Hostname Identification and Resolution on page 24](#)).

Acquire the certificate(s) using the Fully Qualified Domain Name (FQDN) of the server or alias used for the Keyfactor Command server(s). For example:

```
keyfactor.keyexample.com
```

The certificate(s) may be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software or may be installed at the time of Keyfactor Command installation. See [Configure SSL for the Default Web Site on the Keyfactor Command Server on page 36](#) for more information.

If installed ahead of time, the certificate(s) should be placed in the Personal Certificate store of the Local Computer using the Certificates MMC Snap-In.

Application-Level Encryption

Keyfactor Command uses data encryption for sensitive data—such as private keys for certificates—stored in the Keyfactor Command database (see [SQL Server on page 11](#)). This option encrypts only the data in the database deemed to be of a sensitive nature, not the entire database. By default, the data is encrypted using SQL encryption, but you have the option to add another level of security with application-level encryption. If you choose to enable this option, you will need a certificate for this purpose installed in the Personal Certificate store of the Local Computer on each Keyfactor Command server. The certificate must have a key usage of either key encipherment or data encipherment enabled. Microsoft certificate templates only allow you to configure data encipherment ("Allow encryption of user data") as a suboption to key encipherment ("Allow key exchange only with key encryption"). You do not need to enable both. You may use the certificate acquired in the name of the Keyfactor Command web site (assuming it supports the appropriate key usage) or you may enroll for a separate certificate for this purpose. The same certificate must be used on all Keyfactor Command servers and the certificate must be

available in the certificate store on the machine when you run the Keyfactor Command installation. A hardware security module (HSM) may be used, if desired. To support the use of an HSM, the Windows CSP driver for the HSM must be installed on the Keyfactor Command server. Be aware that transactions accessing the encrypted data—such as enrolling for PFX certificates, downloading PFX certificates, running inventory, and adding certificates to certain types of certificate stores (e.g. F5, NetScaler)—will require accessing the HSM. A slow HSM will slow down these processes.



Note: In an environment where there are multiple copies of Keyfactor Command pointing to the same database, each server running a Keyfactor Command instance will need to have the same encryption certificate AND the corresponding private key.



Note: The thumbprint of the certificate used for application-level encryption is stored in the registry on the Keyfactor Command server(s)—rather than in the Keyfactor Command database—to provide a further level of separation from SQL.



Warning: If the certificate used for application-level encryption or the private key for this certificate are removed from the Keyfactor Command server while data in the database is encrypted with this certificate, access to this data will be lost. Take care to ensure that this certificate and its private key remain in place or that there are backups of both the certificate and private key (with any necessary password) that can be accessed in the event that the certificate needs to be restored.

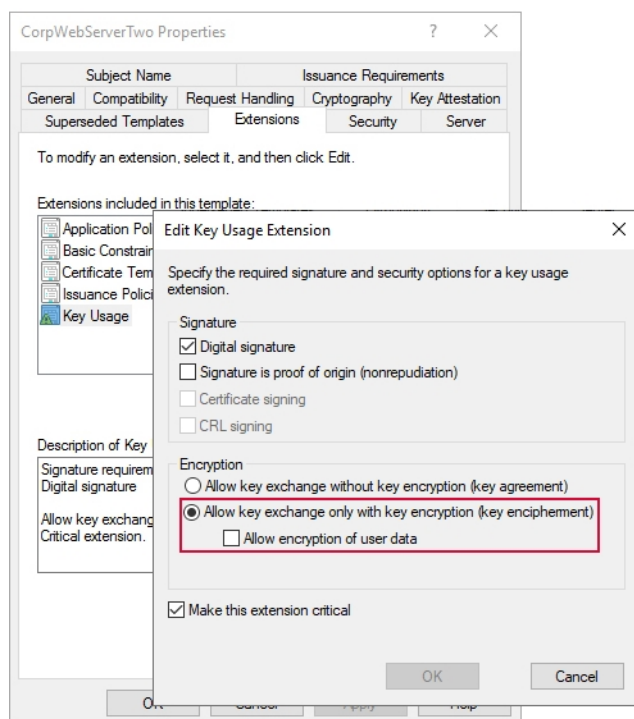


Figure 13: Certificate Template with Key Encipherment Key Usage

2.4.2.10 Grant Permissions in SQL

If you opt to use Windows authentication for the Keyfactor Command connection to SQL during the installation, the user who installs Keyfactor Command must have permissions to administer the SQL server and add databases and users (logins). To grant this, add the user who will install Keyfactor Command to the SQL server login list:

1. On the SQL server open the SQL Server Management Studio, connect to the database, and open **Security**.
2. Right-click on **Logins** and choose **New Login**.
3. Enter the domain name and user name of the administrative user who will be installing Keyfactor Command.
4. On the Login Properties page for this user, open Server Roles and check either the sysadmin role or the dbcreator, public and securityadmin roles. The full sysadmin permissions 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.
5. Accept the remainder of the defaults and click **OK**.

If you opt to use SQL authentication, these permissions need to be granted to the SQL user.

Once Keyfactor Command has been deployed, the Windows user or SQL user used for the install can be removed from the Logins under Security in the SQL Server Management Studio. Ongoing connectivity to the database is maintained using accounts created specifically for the purpose during the installation.



Note: From Keyfactor Command version 9.0, service accounts will not be created with the db_owner role. Instead, a new keyfactor_db_role will be created and granted to the service accounts. This role has permission on each of the schemas (dbo, ssl, ssh, cms_agents, etc.) and permission on the encryption certificate.

2.4.2.11 Install IIS and .NET on the Keyfactor Command Server

Internet Information Services (IIS) and .NET 4.7.2 or greater must be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software.

IIS is a standard Windows role added through the Windows Server Manager tool and .NET is a standard Windows feature added through the Windows Server Manager tool. You may need to update to .NET 4.7.2 or greater with a downloadable update package or through Windows update.



Important: IIS needs to be configured to allow requests using the HTTP verbs DELETE, GET, POST and PUT to reach the Default Web Site (or other web site if you choose to install to an alternate web site). These are enabled by default. To check whether any of these have been disabled, open the IIS Management console, drill down to highlight the Default Web Site, double-click **Request Filtering** in the center pane, and review the information on the **HTTP Verbs** tab.

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 4: .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 4: .NET Framework Release Values](#).

Table 4: .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	80EA8, 80EB1, 80FF4, 81041

Windows Server 2019 and 2022

The following figures show the components of IIS and .NET necessary to support Keyfactor Command on Windows Server 2019 and 2022. Your Keyfactor Command server may have additional roles or features installed that are not shown in these figures.

Keyfactor Command makes 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). The *Active Directory module for Windows PowerShell* is installed as a feature as part of the *Remote Server Administrator Tools*.

Note that it is possible to install IIS and the necessary features using PowerShell rather than the below-referenced GUI-based installation method. The correct PowerShell command for this is:


```
Install-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Errors,
Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-Auth, Web-
Windows-Auth, Web-Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console, RSAT-AD-PowerShell
```



Tip: To check and see if all the required roles and features have been installed, use Get-WindowsFeature with the same list of roles and features like so:

```
Get-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-
Errors, Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-
Auth, Web-Windows-Auth, Web-Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console,
RSAT-AD-PowerShell
```

Output from this command will look something like the following, which shows some required features installed and some missing. Make sure all roles and features in the query output are marked *Installed* before continuing.

```
Get-WindowsFeature Web-Server, Web-Asp-Net45, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Err
ors, Web-Static-Content, Web-Http-Logging, Web-Stat-Compression, Web-Filtering, Web-Basic-Auth, Web-Windows-Auth, Web-
Net-Ext45, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Mgmt-Console, RSAT-AD-PowerShell

Display Name                                         Name                                         Install State
-----
[X] Web Server (IIS)                                Web-Server                                  Installed
    [X] Default Document                            Web-Default-Doc                            Installed
    [X] Directory Browsing                           Web-Dir-Browsing                           Installed
    [X] HTTP Errors                                  Web-Http-Errors                            Installed
    [X] Static Content                               Web-Static-Content                          Installed
    [X] HTTP Logging                                 Web-Http-Logging                           Installed
    [X] Static Content Compression                   Web-Stat-Compression                        Installed
    [X] Request Filtering                            Web-Filtering                              Installed
    [ ] Basic Authentication                         Web-Basic-Auth                             Available
    [ ] Windows Authentication                       Web-Windows-Auth                           Available
    [ ] .NET Extensibility 4.8                       Web-Net-Ext45                              Available
    [ ] ASP.NET 4.8                                  Web-Asp-Net45                              Available
    [ ] ISAPI Extensions                             Web-ISAPI-Ext                              Available
    [ ] ISAPI Filters                               Web-ISAPI-Filter                           Available
[X] IIS Management Console                          Web-Mgmt-Console                           Installed
[ ] Active Directory module for Windows ...         RSAT-AD-PowerShell                         Available
```

Some of the required IIS roles and features are installed on this machine, but several are still missing.

Figure 14: Use Get-WindowsFeature to Determine if All Required Roles and Features are Installed

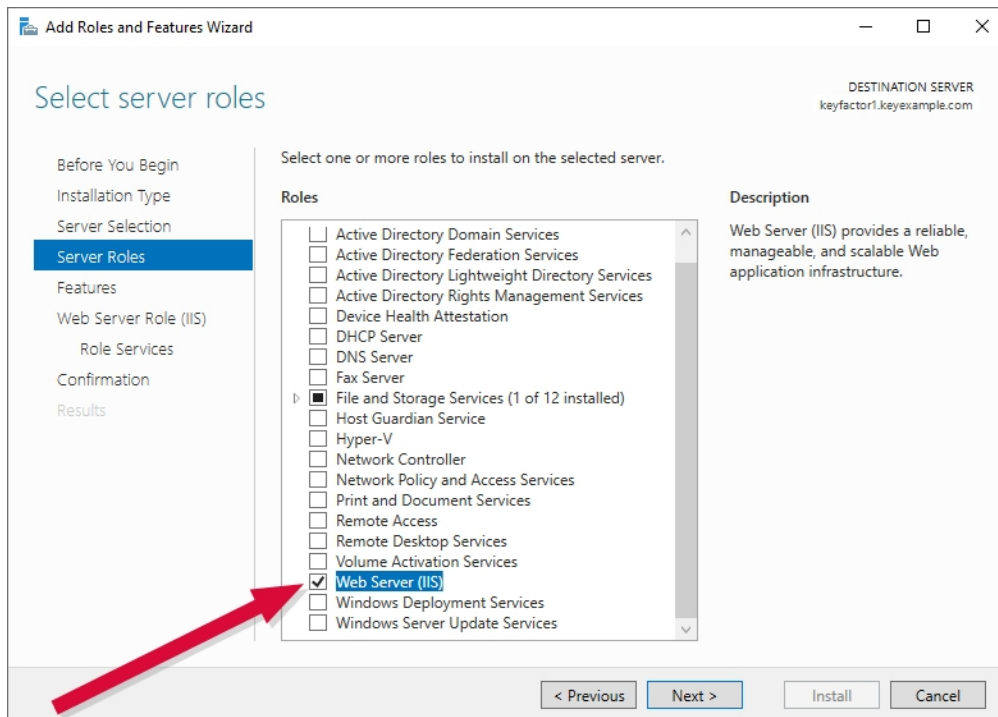


Figure 15: Web Server Role

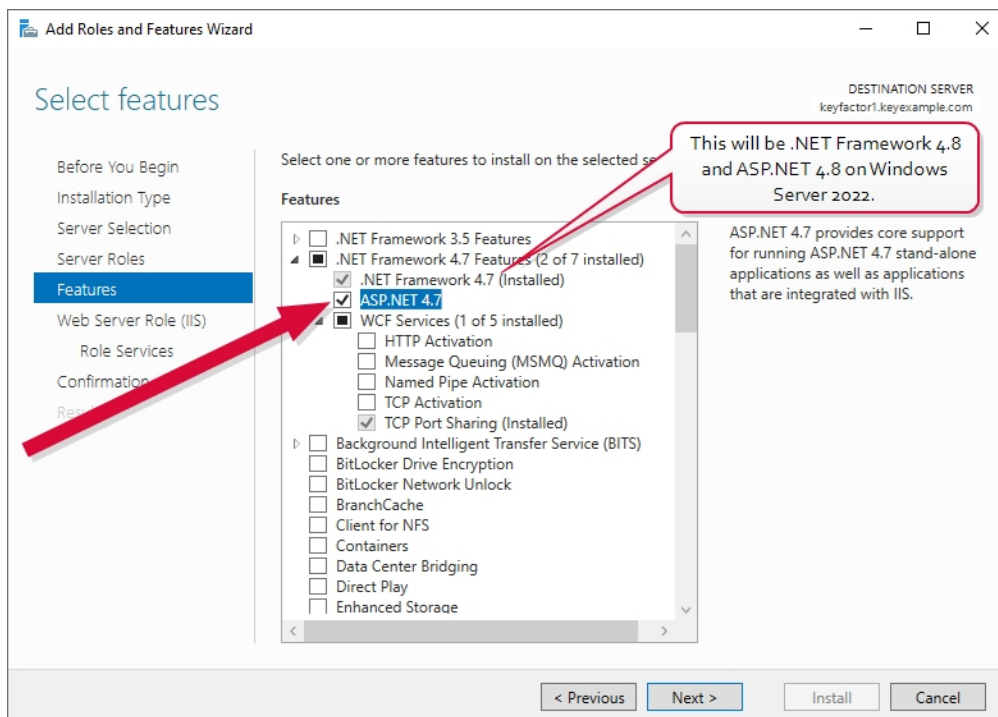


Figure 16: .NET 4.7 Feature

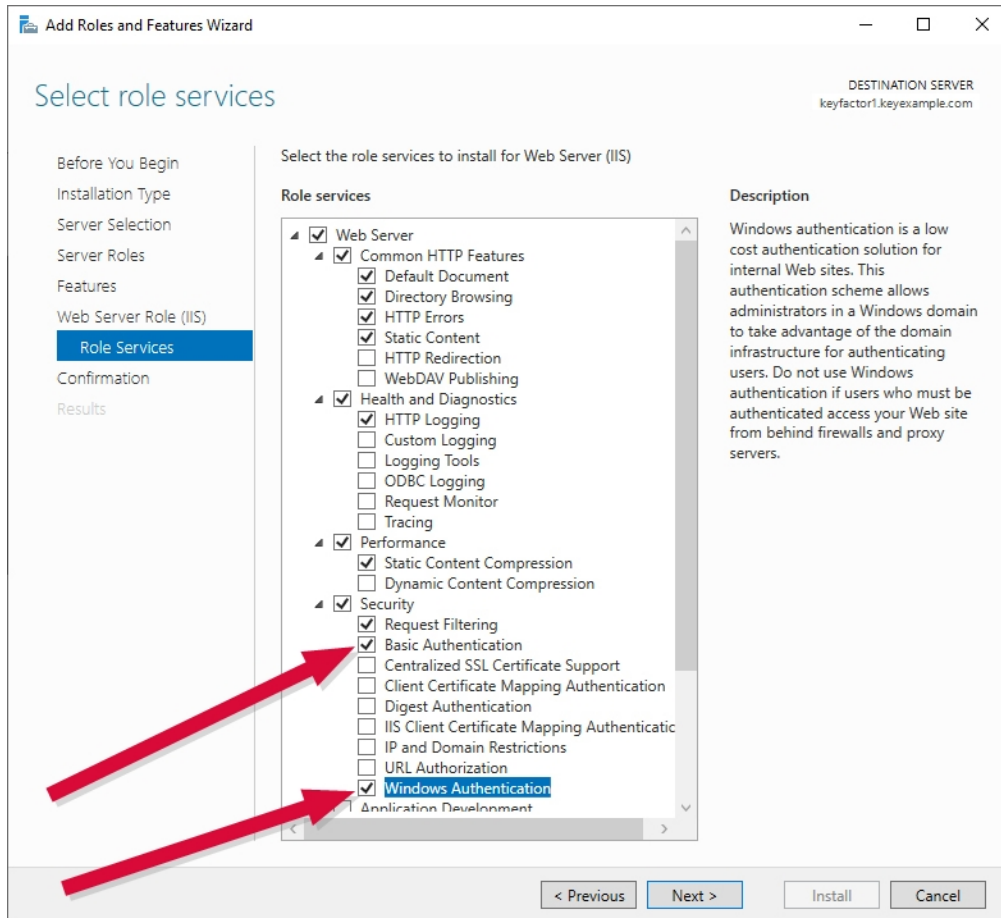


Figure 17: Role Services Page One

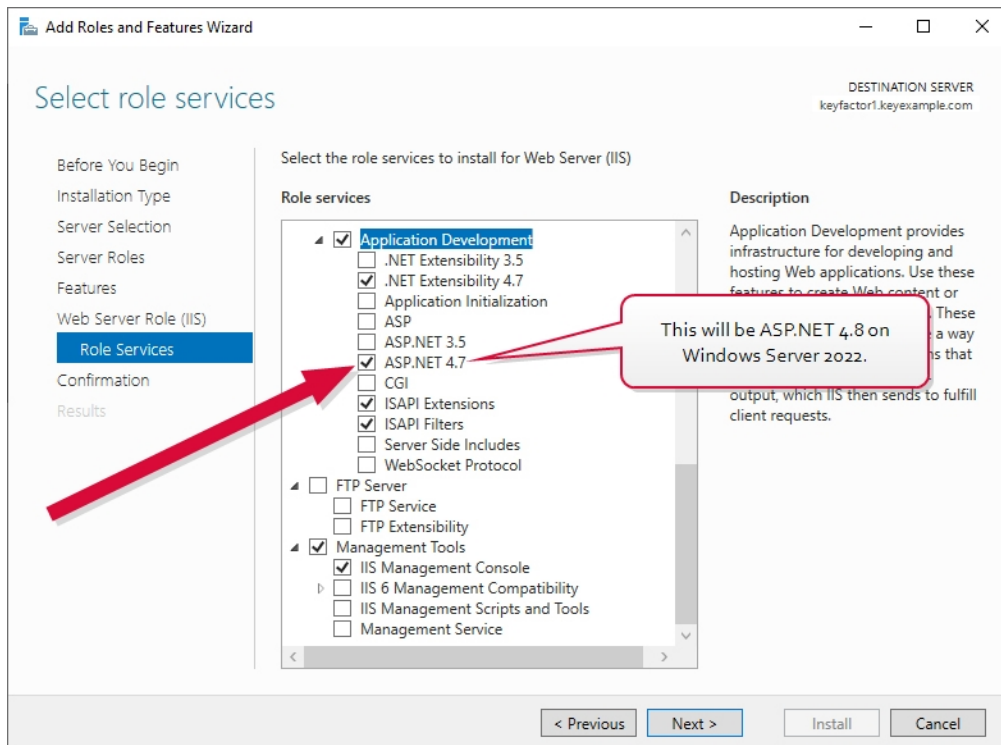


Figure 18: Role Services Page Two

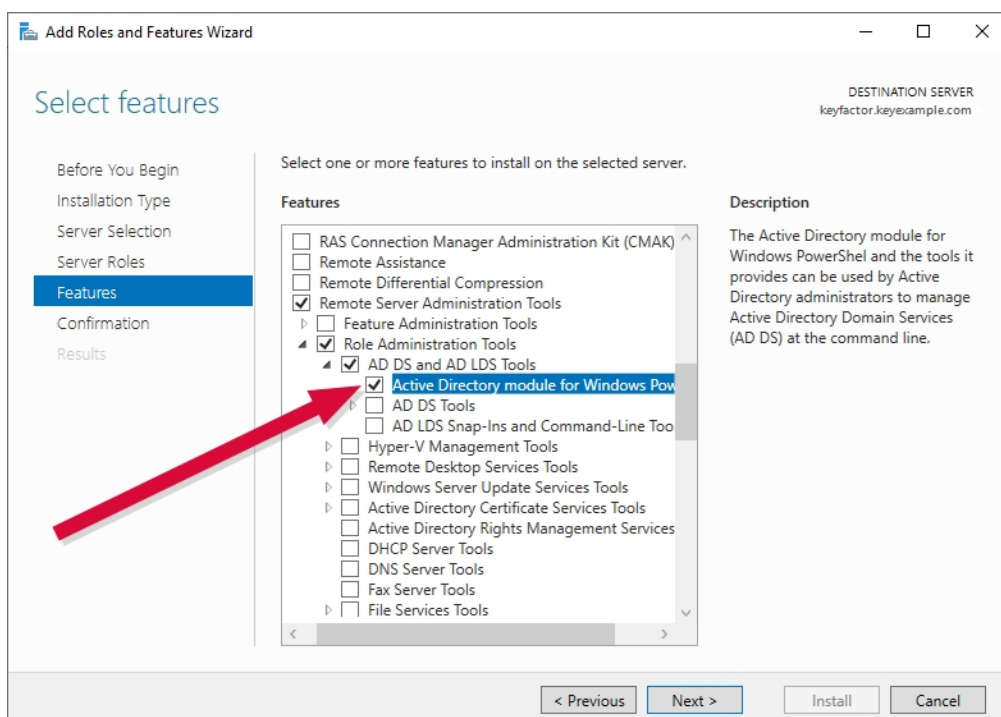


Figure 19: Active Directory Module for Windows PowerShell

2.4.2.12 Configure SSL for the Default Web Site on the Keyfactor Command Server

Once you have acquired an SSL certificate for Keyfactor Command and installed IIS, you can open the IIS Management Console and associate the certificate with the Default Web Site. You can do this either before or after installing Keyfactor Command.

To import your SSL certificate and associate it with the Default Web Site:

1. Open the IIS Manager MMC snap-in.
2. Navigate to the connection for the current host. (The top level in IIS.)
3. On the current host Home page, open (double-click) Server Certificates. If your SSL certificate already appears in this list, you can skip steps 4-7.
4. On the Server Certificates page, select Import... under Actions.
5. In the Certificate file (.pfx) field, choose the browse option and navigate to the .pfx or .p12 file containing your certificate.
6. Enter the password for your certificate, select the Personal Certificate Store, check the Allow this certificate to be exported box if desired, and click **OK**.
7. Your certificate should now appear in the list of Server Certificates. Confirm that the Issued To column shows your certificate name correctly (e.g. keyfactor.domain.com).
8. Navigate to the Default Web Site and on the Default Web Site Home page, select Bindings... under Actions.
9. In the Site Bindings dialog, highlight the https entry if it exists and choose Edit. If an https entry does not exist, click **Add**.
10. In the Edit Site Bindings dialog, select https in the Type dropdown (this will already be selected and grayed out if you selected Edit in the previous step), select the certificate you just imported in the SSL certificate dropdown box, and click **OK**.

Note that these instructions assume that your SSL certificate has been provided in PKCS12 format file. If you are requesting a certificate directly from an on-premise CA through IIS or are generating a CSR through this IIS installation to submit to a CA, the configuration steps will be different.

2.4.2.13 Configure the Keyfactor Command Server to Require SSL

For best security practice, the Keyfactor Command web site should be configured to require SSL for all access. To do this:

1. Open the IIS Manager MMC snap-in.
2. Navigate to the Default Web Site.
3. Under the Default Web Site Home, select **SSL Settings**.
4. On the SSL Settings page, check the **Require SSL** box and, under Actions, click **Apply**.



Important: The Keyfactor Command web application is not configured to support HTTP Strict Transport Security (HSTS) by default. HSTS is a web security policy mechanism that helps to protect websites against protocol downgrade attacks and cookie hijacking. An application enables HSTS by returning an HTTP response header that instructs users' browsers to only interact with the site using secure transport methods. HSTS is supported by all modern browsers. To accommodate this, configure the server to always send the "Strict-Transport-Security" HTTP header on HTTPS connections:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

The max-age parameter is given in seconds; the value shown above is equivalent to one year.

Instructions for implementation of HSTS are beyond the scope of this guide.

2.4.2.14 Prepare for External Log Shipping over TLS (Optional)

Keyfactor Command offers the option to copy audit logs in real time to a separate server for collection and analysis with a centralized logging solution (e.g. rsyslog, Splunk, Elastic Stack). This can be done either over standard UDP/TCP connections, or you can opt to secure the connection to the backend log collection solution using TLS. This requires a backend solution that supports receiving logs over TLS and, typically, a client certificate on the Keyfactor Command server and a server certificate on the backend server.

The following instructions cover using rsyslog on the backend and will differ if you are using an alternative log collection solution.

Acquire the client certificate(s) using the Fully Qualified Domain Name (FQDN) of the server or alias used for the Keyfactor Command server(s) (see [Hostname Identification and Resolution on page 24](#)). For example:

```
keyfactor.keyexample.com
```

The certificate must be installed on the Keyfactor Command server(s) prior to installation of the Keyfactor Command software.

To acquire a client certificate for use in log shipping using a Microsoft CA, first create or identify a template that has an extended key usage (EKU) of *Client Authentication* and make it available for enrollment on a CA to which the Keyfactor Command server has access with enrollment permissions for the Keyfactor Command server. If desired, you can use a template that has both the *Client Authentication* and *Server Authentication* EKUs and use it for certificates on both sides of the communication. Start by copying a computer template if you want to enroll for the certificate using the Microsoft MMC as described below and without needing to set the private key of the certificate as exportable.

To enroll for a client certificate using the MMC:

1. On the Keyfactor Command server, do one of following:
 - Using the GUI:
 - a. Open an empty instance of the Microsoft Management Console (MMC).
 - b. Choose **File->Add/Remove Snap-in....**
 - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
 - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
 - e. Click **OK** to close the Add or Remove Snap-ins dialog.
 - Using the command line:
 - a. Open a command prompt using the "Run as administrator" option.
 - b. Within the command prompt type the following to open the certificates MMC:
certlm.msc
2. Drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate....**
3. Follow the certificate enrollment wizard, selecting the template you created or identified for use for this purpose, and providing any required information, being sure to set the CN to the FQDN of the Keyfactor Command server.



Tip: If you have an existing Keyfactor Command and wish to enroll through Keyfactor Command, you can request the certificate using the PFX enrollment option and either push it out to the Keyfactor Command local machine store using an IIS personal certificate store managed with a Keyfactor Universal Orchestrator installed on Windows or Keyfactor Windows Orchestrator or import it to the certificate store using the PFX generated from Keyfactor Command and one of these orchestrators.



Note: The Keyfactor Command needs to be configured to trust the CA that issued the certificate to the rsyslog server. If you have opted to acquire certificates from a CA for which a root trust is not already configured on the Keyfactor Command server, this will need to be configured.

To acquire a server certificate for use in log shipping using a Microsoft CA, first create or identify a template that has an extended key usage (EKU) of *Server Authentication* and make it available for enrollment on a CA to which the server from which you are requesting the certificate has access with enrollment permissions for the server from which you are requesting the certificate.

To enroll for a server certificate using the MMC:

1. On a Windows server with appropriate enrollment access, do one of following:
 - Using the GUI:
 - a. Open an empty instance of the Microsoft Management Console (MMC).
 - b. Choose **File->Add/Remove Snap-in....**
 - c. In the *Available snap-ins* column, highlight **Certificates** and click **Add**.
 - d. In the Certificates snap-in popup, choose the radio button for Computer account, click **Next**, accept the default of Local computer, and click **Finish**.
 - e. Click **OK** to close the Add or Remove Snap-ins dialog.
 - Using the command line:
 - a. Open a command prompt using the "Run as administrator" option.
 - b. Within the command prompt type the following to open the certificates MMC:
certlm.msc
2. Drill down to Certificates in the Personal folder under **Certificates** for the Local Computer and locate your newly created certificate. Right-click on the certificate and choose **All Tasks->Export...**
3. Follow the certificate export wizard, being sure to answer **Yes, export the private key** and choosing the option to **Include all certificates in the certificate path if possible**. Set a password to secure the exported private key.
4. In the MMC, drill down to the Personal folder under **Certificates** for the Local Computer, right-click, and choose **All Tasks->Request New Certificate...**
5. Securely copy the resulting PFX file to your rsyslog server and place it in a temporary working directory.
6. Use openssl to break the PFX file apart into separate certificate and key files and remove the encryption on the key file (rsyslog does not provide a method for providing the password necessary to use the encrypted file) as follows:
 - a. Execute the following command to pull the key out of the PFX file (provide the input password for the PFX when prompted and the output password for the PEM key file when prompted):
`openssl pkcs12 -in mycertfile.pfx -nocerts -out mykey.pem`
 - b. Execute the following command to pull the certificate out of the PFX file (provide the input password for the PFX when prompted):
`openssl pkcs12 -in mycertfile.pfx -clcerts -nokeys -out mycert.pem`
 - c. Execute the following command to pull the chain certificate(s) out of the PFX file (provide the input password for the PFX when prompted):
`openssl pkcs12 -in mycertfile.pfx -cacerts -nokeys -chain -out cacerts.pem`
 - d. Execute the following command to remove the encryption from the key so that a password will not be required when accessing the key file (provide the PEM key password you set in the first step):
`openssl rsa -in mykey.pem -out mynewkey.key`

7. Identify a secure location on the rsyslog server to store the certificates and key file (e.g. /etc/tls/certs and /etc/tls/private) and copy the certificates and key to these locations, setting appropriately secure permissions on the files. The key needs to be readable by the rsyslog daemon.



Tip: If you have an existing Keyfactor Command and wish to enroll through Keyfactor Command, you can request the certificate using the PFX enrollment option, opt to download it as a ZIP PEM, copy the zip file to the rsyslog server, unzip, and distribute the files as described in the final step, above.

Configuration of rsyslog for TLS support may vary depending on your needs. In addition to the standard rsyslog package for your Linux server, you will need GNU TLS packages to support TLS communication. For example, for Ubuntu the required packages are:

```
rsyslog
rsyslog-gnutls
gnutls-bin
```

The following is an example rsyslog.conf file configured for TLS support:

```
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf
#####
#### MODULES ####
#####

module(load="imuxsock") # provides support for local system logging
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

#####
#### GLOBAL DIRECTIVES ####
#####

#
```

```

# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

# Filter duplicated messages - It can be helpful to set this 'off' during initial Keyfactor Command
testing
$RepeatedMsgReduction off

#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf

#
# Configuration for TLS
#
$DefaultNetstreamDriver gtls

$DefaultNetstreamDriverCAFile /etc/tls/certs/cacerts.pem
$DefaultNetstreamDriverCertFile /etc/tls/certs/mycert.pem
$DefaultNetstreamDriverKeyFile /etc/tls/private/mynewkey.key

$ModLoad imtcp

$InputTCPServerKeepAlive on
$InputTCPServerStreamDriverAuthMode anon
$InputTCPServerStreamDriverMode 1 # run driver in TLS-only mode

```

```
$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverMode 1 # run driver in TLS-only mode

$InputTCPServerRun 10514
```

A filter similar to the following can be used to redirect all Keyfactor Command related traffic to a particular file:

```
:syslogtag, isequal, "Keyfactor" /var/log/keyfactor/audit.log
```

2.4.3 Installing

The following installation instructions cover installing all Keyfactor Command server components on a single server performing all Keyfactor Command roles. You may choose to separate the roles onto different servers. If you do, the installation process will vary from the described process.

2.4.3.1 Install the Main Keyfactor Command Components on the Keyfactor Command Server(s)

The following installation steps show all possible Keyfactor Command features enabled. Your Keyfactor Command license may not cover all Keyfactor Command features. If it does not, unlicensed features will not be shown in the configuration wizard. You may skip those configuration steps.

To begin the Keyfactor Command installation, execute the KeyfactorPlatform.msi file from the Keyfactor Command installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

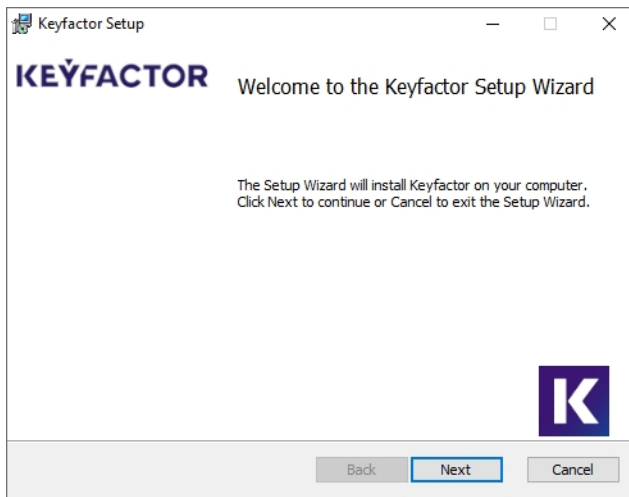


Figure 20: Install: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.

- On the next page, select the components to install. For a server with the default roles collocated, leave the default options and click **Next** to continue. The vSCEP Validation Service component is not selected by default. If desired, you can highlight Keyfactor Command and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor Platform\Certificate Management System



Note: Although [Figure 21: Install: Select Components](#) shows only the default components selected, the remainder of this page covers configuring Keyfactor Command as though all the components have been selected.

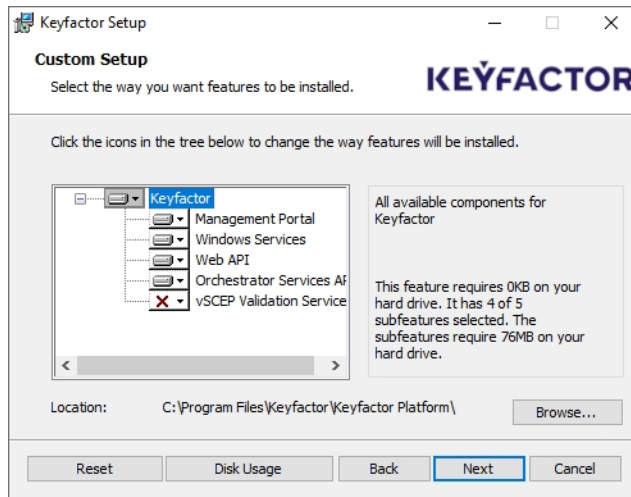


Figure 21: Install: Select Components

- On the next screen, click **Install**.
- On the final installation wizard page, leave the "Launch the Configuration Wizard now" box selected and click **Finish**. The configuration wizard should start automatically. This can take several seconds.
- On the Keyfactor Command Database Configuration page, enter the name, IP address, or fully qualified domain name (FQDN) of your SQL server and select a Credential Type of either **Windows** or **SQL**.



Important: Keyfactor Command uses an encrypted channel to connect to the SQL server by default, which requires configuration of an SSL certificate on the SQL server (see [Using SSL to Connect to SQL Server on page 13](#)). The name or IP address you enter here for your SQL server must be available as a SAN in this certificate unless you have disabled the encrypted connection for Keyfactor Command (see [Configurable SQL Connection Strings on page 17](#)).

- If you select **Windows** as the Credential Type for connecting to SQL, click the **Connect** button.

The dialog box is titled "Keyfactor Database Configuration". It contains the following fields and buttons:

- Server:** A text box containing "sqlsrvr05.keyexample.com".
- Credential Type:** Two radio buttons, "Windows" (selected) and "SQL".
- Database Name:** A text box containing "Keyfactor".
- Buttons:** "Connect", "Browse", "Continue", and "Close".

Figure 22: Windows Authentication

- If you select **SQL** as the Credential Type for connecting to SQL, the window will expand to include fields to enter a SQL username and password. Enter a username and password to authenticate to SQL, and click the **Connect** button.



Note: The password must not contain single or double quotes. An error will be shown if single or double quotes are used in the password. For the permissions required for this user, see [Grant Permissions in SQL on page 30](#).

The dialog box is titled "Keyfactor Database Configuration". It contains the following fields and buttons:

- Server:** A text box containing "sqlsrvr05.keyexample.com".
- Credential Type:** Two radio buttons, "Windows" and "SQL" (selected).
- User:** A text box containing "john_smith".
- Password:** A text box with masked characters (dots).
- Database Name:** A text box containing "Keyfactor".
- Buttons:** "Connect", "Browse", "Continue", and "Close".

Figure 23: SQL Authentication



Note: Keyfactor Command supports configuration of a base SQL connection template that is used for all connections Keyfactor Command makes to SQL. For more information, see [Configurable SQL Connection Strings on page 17](#).



Note: Keyfactor recommends that you accept the default Credential Type of Windows unless you have a strong need to do otherwise. Your SQL server must be configured to support mixed mode authentication in order to use the SQL option.

7. After the **Connect** button is clicked, the database name field will be activated. You can either enter the name of the desired database—for either a new or existing database—or click **Browse** to scroll through a list of existing databases.



Note: On subsequent runs of the configuration wizard, the database name field will be pre-populated with the database name used on the last completed run. Any change to the server connection fields (server name, authentication type, etc.) will require the Connect button to be used again to unlock the database name field and the Continue button.

8. Click the **Continue** button. You will receive a confirmation dialog if any changes will be made to the database at this stage.



Note: If any of the following situations occurs, you will receive a message:

- The selected database does not exist and will be created.
- The selected database is empty and not associated with Keyfactor Command; it will be populated with the Keyfactor Command schema.
- The selected database does not match the current product schema and will be upgraded.
- The selected database is not empty and is not associated with Keyfactor Command.
- The user does not have access to the database.
- An SSL certificate is not correctly configured on the SQL server.

9. On the Keyfactor Command Encryption Warning page, read and understand the warning. Make note of the referenced documents to provide to your SQL team. Take advantage of the option to make a backup of the Database Master Key (DMK) by entering a path to a directory on your SQL server along with a filename for the backup file and a password to encrypt the file and clicking **Backup**. The user running the Keyfactor Command installer must have write permissions to this directory. Click **Continue**.



Important: Keyfactor Command uses Microsoft SQL Server encryption to protect security sensitive data, including service account credentials. Backup of the SQL server Database Master Key (DMK) is of critical importance in database backup and recovery operations. The backup file of the DMK and the password should be stored in a safe, well-documented location. Without the file and password created with this process, some data that is encrypted within the Keyfactor Command database will be unrecoverable in a disaster recovery scenario. For more information, see *SQL Encryption Key Backup* in the *Keyfactor Command Reference Guide*.

If you choose to install Keyfactor Command in the default location, the referenced documents can later be found here:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\DMKBackup.docx
C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\DMKRestore.docx

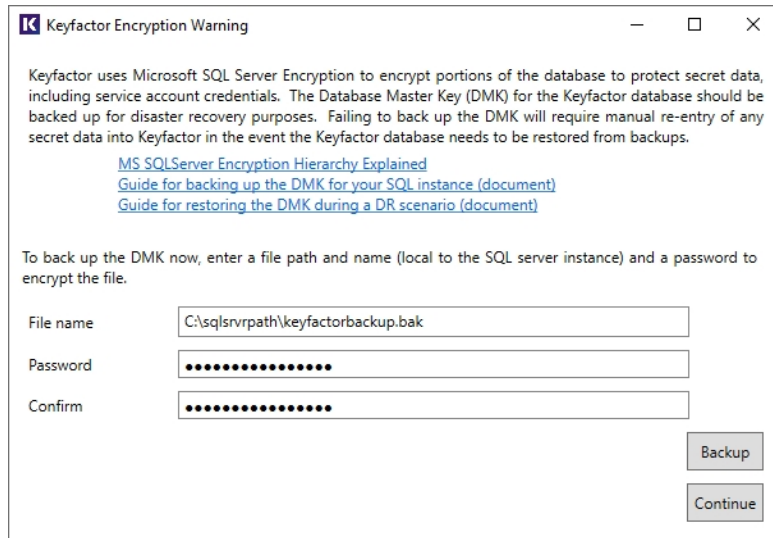


Figure 24: Configure: Backup Database Master Key

10. On the Keyfactor Command License upload page, click **Upload** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE. Once the uploaded license shows as valid, click **Continue**.

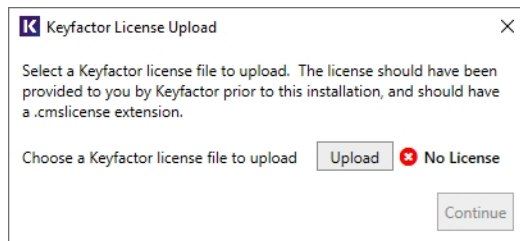


Figure 25: Configure: Upload License

11. In the Keyfactor Command configuration wizard, you can choose to upload a configuration file to populate the fields. You may have a file saved from a previous run of the configuration wizard or you may be provided one by Keyfactor. To upload a file, in the configuration wizard, click **File** at the top of the wizard and choose Open Data File... . Browse to locate the configuration file. Configuration files have an extension of .cmscfg. The file may be protected with a password. If it is, you will need to provide this password to open the file. Continue with the remainder of the steps, reviewing the tabs to assure that the data is complete and correct.



Note: At the bottom of the configuration wizard, if the database server name is longer than will fit in the provided window, it will be truncated and an ellipsis will be added.

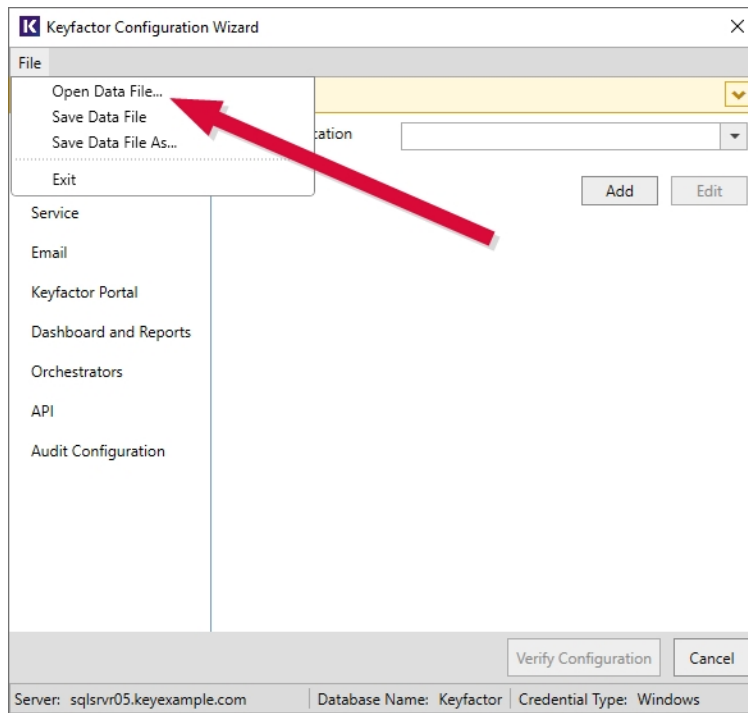
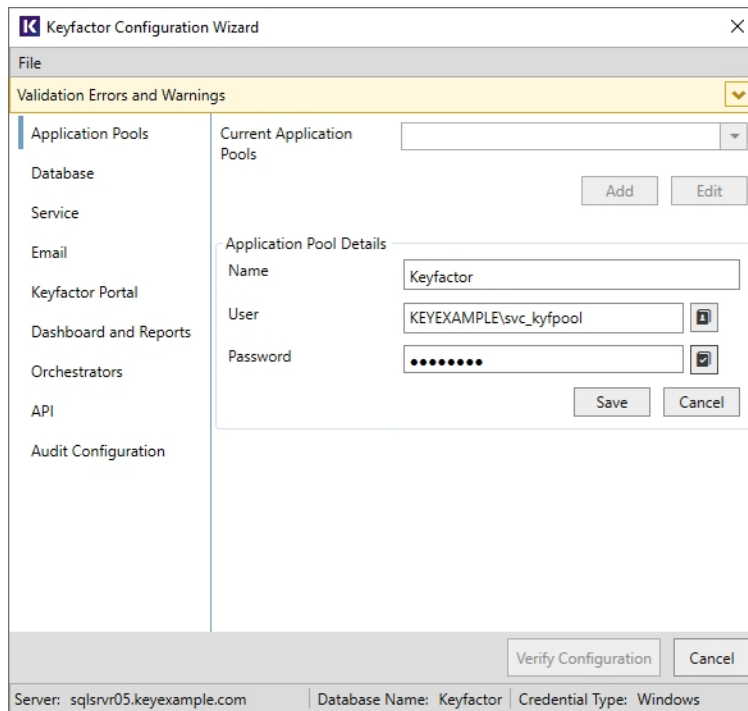


Figure 26: Configure: Open Data File

12. Application Pools Tab

On the Application Pools tab of the configuration wizard, click **Add**, change the default application pool name, if desired, and enter the user name (DOMAIN\username format) and password of the Active Directory service account under which the application pool will run. You may use the people picker button (👤) to browse for the account. Click the verify button (✅) to confirm that the username and password entered are valid. Assuming the verification completes successfully, click **Save**.



The image shows the 'Keyfactor Configuration Wizard' window, specifically the 'Application Pools' tab. The window has a menu bar with 'File' and a 'Validation Errors and Warnings' section. On the left is a sidebar with navigation links: Application Pools, Database, Service, Email, Keyfactor Portal, Dashboard and Reports, Orchestrators, API, and Audit Configuration. The main area is divided into two sections. The top section, 'Current Application Pools', contains a dropdown menu and 'Add' and 'Edit' buttons. The bottom section, 'Application Pool Details', contains fields for 'Name' (set to 'Keyfactor'), 'User' (set to 'KEYEXAMPLE\svc_kyfpool'), and 'Password' (masked with dots). There are 'Save' and 'Cancel' buttons for these details. At the bottom of the window are 'Verify Configuration' and 'Cancel' buttons. A status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 27: Configure: Application Pools

13. Database Tab

On the Database tab in the top section, select an Authentication Mode for ongoing communications to SQL server—**Windows Authentication** or **SQL Server Authentications**. Your SQL server must be configured to support mixed mode authentication in order to use the SQL server authentication option. If you choose SQL server authentication, enter the Username and Password for a SQL administrator for the Keyfactor Command SQL database. If the user does not exist in SQL, it will be created and granted the necessary permissions for management of the Keyfactor Command database (db_owner). If the user already exists in SQL, it will be granted the necessary permissions. If the database you originally connected to is an Azure database, **SQL Server Authentication** is the only option provided.

If desired, check the **Configure Encryption** box. This option allows you to encrypt select sensitive data stored in the Keyfactor Command database using a separate encryption methodology utilizing a Keyfactor Command-defined certificate on top of the SQL server encryption noted above. This additional layer of encryption protects the data in cases where the SQL Server master keys cannot be adequately protected. Read and understand the encryption warning. This warning applies to implementations with more than one Keyfactor Command server.



Note: In an environment where there are multiple copies of Keyfactor Command pointing to the same database, each server running a Keyfactor Command instance will need to have the same encryption certificate AND the corresponding private key.

Select **Application and SQL** for the **Encryption Type** and click the **Select** button to choose a certificate from the Personal Certificate store of the Local Computer with which to encrypt the data. Only valid certificates with the appropriate key usage will appear in the selection dialog. See [Acquire a Public Key Certificate for the Keyfactor Command Server on page 28](#).



Tip: If you need to reset the encryption level to remove application-level encryption, run the configuration wizard again and select the **SQL Only** option. You must ensure that the server you are re-running the configuration wizard on has both the certificate used for application-level encryption and its associated private key. When Keyfactor Command notices that application-level encryption has been disabled, it will process all the secrets in the database and remove the additional encryption. The data will then be re-saved to the secrets table using only SQL-level encryption.

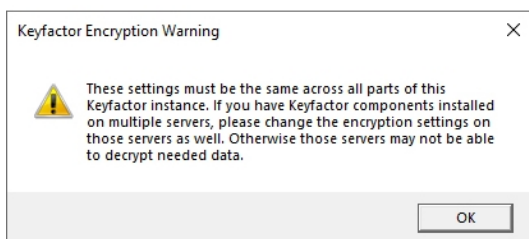


Figure 28: Configure: Encryption Warning

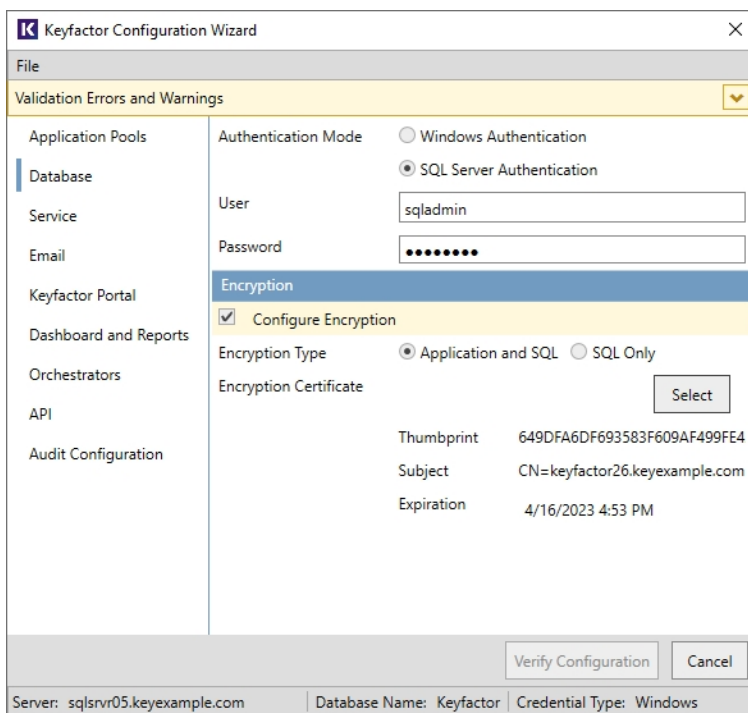


Figure 29: Configure: Database

14. Service Tab



On the Service tab, enter the user name (DOMAIN\username format) and password of the Active Directory service account under which the Keyfactor Command Service will run. This can be the same service account used for the application pool or a different service account. You may use the people picker button () to browse for the account. Click the verify button () to confirm that the username and password entered are valid. If desired, check the **Start service on bootup** box to start the Keyfactor Command Service at system start. If you are installing only one Keyfactor Command server, check the **Everything** box to run all the services on this Keyfactor Command server. If you are installing more than one Keyfactor Command server, uncheck the **Everything** box and check the boxes next to the services that should run on this Keyfactor Command server instance. For example, if you've opted not to use the SSL scanning functionality, you can uncheck the **Endpoint History Purge** box. Services details are shown in [Table 5: Keyfactor Command Services](#). At the bottom of the list of services, modify the default value of 1000 for **Concurrent Workflows**, if desired.

Table 5: Keyfactor Command Services

Type	Service	Description	Notes
Maintenance	Synchronization	Periodically synchronize certificates from certificate authorities.	The schedules for this are user configurable (see <i>Certificate Authorities</i> in the <i>Keyfactor Command Reference Guide</i>).
Maintenance	Bulk Audit Entry Processing	Periodically add audit log entries for large jobs. Most audit log entries are added immediately at the time the activity generating the audit log takes place. However, some large jobs that might generate heavy server load (e.g. bulk revocation) save the audit log entries in a temporary location to reduce server load and then they are added to the audit log by this periodic job.	This job runs every 10 minutes.
Maintenance	Metadata Generation	Periodically generate and assign metadata to certificates when they are imported into Keyfactor Command using a custom metadata extension.	This job runs every 15 minutes.
Maintenance	Private Key Cleanup	Periodically remove any stored private keys in the Keyfactor Command database that have expired and are eligible for deletion.	This job runs daily at 1:00 am. For more information about stored private keys, see <i>Status Tab</i> in the <i>Keyfactor Command Reference Guide</i> .
Maintenance	Purge Audit Log History	Periodically remove any audit log history in the Keyfactor Command database that has expired and is eligible for deletion.	This job runs monthly on the first day of the month at 2:00 am. For more information, see <i>Application Settings: Auditing Tab</i> in the <i>Keyfactor Command Reference Guide</i> . Only audit logs belonging to unprotected categories

Type	Service	Description	Notes
			are eligible for deletion.
Maintenance	Endpoint History Purge	Periodically remove any SSL endpoint history in the Keyfactor Command database that is eligible for deletion, based on the setting in <i>Application Settings: Auditing Tab</i> (SSL > Retain SSL Endpoint History (days)) in the <i>Keyfactor Command Reference Guide</i> .	This job runs daily at 1:00 am.
Maintenance	Report Cleanup	Periodically remove records from temporary files generated while running reports.	This job runs daily at midnight.
Maintenance	Update Stats	Periodically run the Microsoft SQL update statistics function in the Keyfactor Command database.	This job runs monthly on the first day of the month at 1:00 am.
Maintenance	Sync Templates	Periodically synchronize certificate templates from the source (e.g. Active Directory) to pick up new templates.	This job runs every hour.
Maintenance	Schedule SSL Jobs	Periodically identify and schedule SSL discovery and monitoring jobs.	This job runs every 5 minutes.
Maintenance	Workflow Cleanup	Periodically remove any completed workflow instances (both successful and failed) in the Keyfactor Command database that have aged X number of days past the completion date (last modified date), where X is defined by the <i>Workflow Instance Cleanup Days</i> application setting (see <i>Application Settings: Console Tab</i> in the <i>Keyfactor Command Reference Guide</i>). The default value is 14 days.	This job runs daily at midnight.
Alerts	CA Health	Periodically send email alerts when a CA is not responding.	The schedule for this is user configurable (see <i>Certificate Authority Monitoring</i> in the <i>Keyfactor Command Reference Guide</i>).

Type	Service	Description	Notes
Alerts	CA Threshold	Periodically send email alerts when a CA is issuing certificates or experiencing issuance failures outside of the established norms.	The schedule for this is user configurable (see <i>Certificate Authority Operations: Adding or Modifying a CA Record Advanced Tab</i> in the <i>Keyfactor Command Reference Guide</i>).
Alerts	CRL	Periodically send email alerts for certificate revocation lists (CRLs) that are approaching expiration.	The schedule for this is user configurable (see <i>Adding or Modifying a Revocation Monitoring Location</i> in the <i>Keyfactor Command Reference Guide</i>).
Alerts	Pending/Expiration	Periodically send email alerts (typically to certificate approvers) for certificate requests made using a certificate template that requires manager approval. Periodically send email alerts for certificates approaching expiration.	The schedules for these are user configurable. See <i>Pending Request Alert Operations: Configuring a Pending Request Alert Schedule</i> and <i>Expiration Alert Operations: Configuring an Expiration Alert Schedule</i> in the <i>Keyfactor Command Reference Guide</i> .
Alerts	Collection Membership Caching	Periodically update the temporary tables that store information on which certificates are in which certificate collections. These temporary tables (caches) are used to support faster processing of some systems.	This value is user configurable with an application setting (see <i>Application Settings: Console Tab</i> in the <i>Keyfactor Command Reference Guide</i>). The default is 20 minutes.
Alerts	Issued Alerts	Periodically send email alerts (typically to certificate requesters) for certificate requests made using a certificate template that requires manager approval that have been approved.	The schedule for this is user configurable (see <i>Issued Request Alert Operations: Configuring an Issued Request Alert Schedule</i> in the <i>Keyfactor</i>

Type	Service	Description	Notes
			<i>Command Reference Guide</i>). A scheduled alert is only needed to deliver notifications for approvals done outside of Keyfactor Command.
Alerts	SSH Key Rotation Alerts	Periodically send email notifications to SSH key users and/or administrators when a key is nearing the end of the key lifetime.	The schedule for this is user configurable (see <i>Key Rotation Alert Operations: Configuring a Key Rotation Alert Schedule</i> in the <i>Keyfactor Command Reference Guide</i>).
Other	Reporting	Deliver regularly scheduled reports via email or saved to a file system.	The schedules for these are user configurable (see <i>Reports</i> in the <i>Keyfactor Command Reference Guide</i>).
Other	Run Suspended Workflows	Periodically attempt to continue all suspended workflows that may be eligible to continue but have not done so due to locking conflicts. A locking conflict may occur if two users attempt to provide input to a workflow instance (e.g. approve a request) at exactly the same time.	This job runs daily at midnight.
n/a	Concurrent Workflows	Sets the batch size used when suspended workflows are run by the Keyfactor Command service.	The default value is 1000.

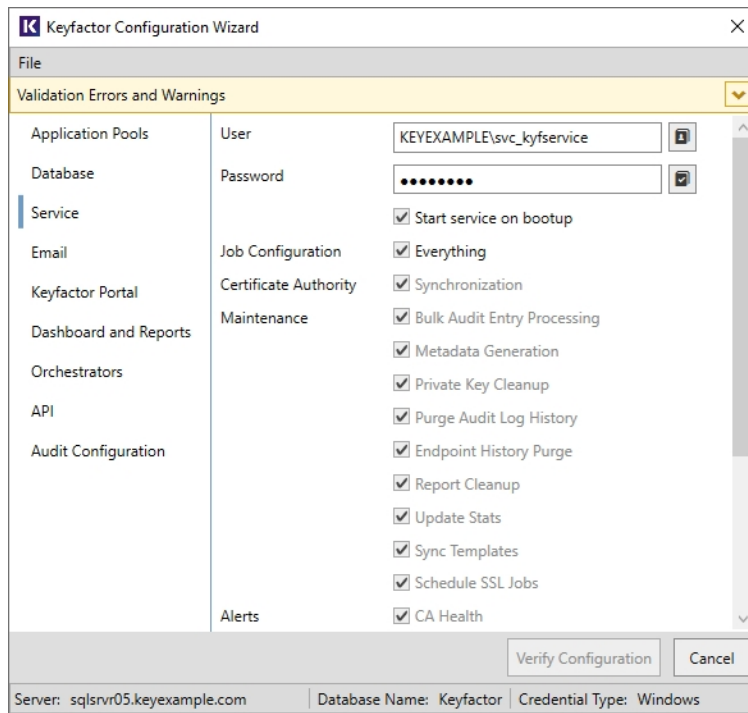
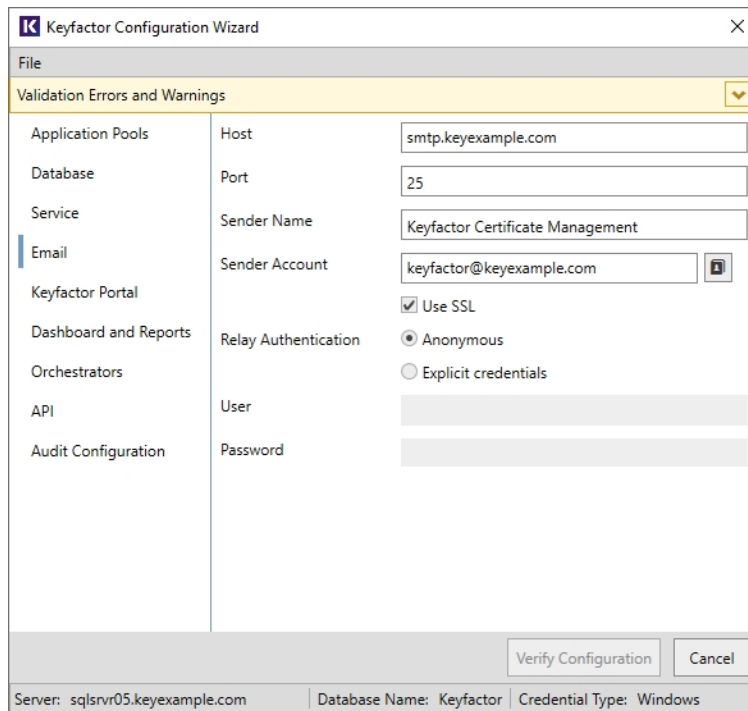


Figure 30: Configure: Service

15. Email Tab

On the Email tab, enter the FQDN of your SMTP server, the SMTP port (default is 25), and the sender name and account. Depending on the email configuration in your environment, the sender account may need to be a valid user on your mail server (using Active Directory credentials) or you may be able to put anything in this field (if your mail server supports anonymous connections). You may use the people picker button (👤) to browse for the sender account if you are using a valid account. Select the **Use SSL** box if this option is supported by your mail server and select the appropriate authentication method for your environment. If your mail server requires that you provide a username and password for a valid user, enter that Active Directory username and password in the fields at the bottom of the page after selecting the **Explicit credentials** radio button. You may use the people picker button (👤) to browse for the account. Click the verify button (👤) to confirm that the username and password entered are valid. The user you select here must match the email address you set in the *Sender Account* field if you select *Explicit credentials*. The information entered on this tab may later be changed in the Keyfactor Command Management Portal.



The image shows the 'Keyfactor Configuration Wizard' window, specifically the 'Email' tab. The window has a title bar with the Keyfactor logo and a close button. Below the title bar is a 'File' menu. A yellow banner at the top indicates 'Validation Errors and Warnings'. The main area is divided into two columns. The left column contains a list of configuration categories: Application Pools, Database, Service, Email (selected), Keyfactor Portal, Dashboard and Reports, Orchestrators, API, and Audit Configuration. The right column contains the configuration fields for the 'Email' tab: Host (smtp.keyexample.com), Port (25), Sender Name (Keyfactor Certificate Management), Sender Account (keyfactor@keyexample.com) with a people picker icon, a checked 'Use SSL' checkbox, 'Relay Authentication' options (Anonymous selected, Explicit credentials unselected), User (empty field), and Password (empty field). At the bottom right are 'Verify Configuration' and 'Cancel' buttons. A status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 31: Configure: Email

16. Keyfactor Portal Tab

On the Keyfactor Portal tab in the top section, enter the FQDN that you will use to access the Keyfactor Command Management Portal in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Administration component or a DNS alias pointing to the server. If you have multiple Keyfactor Command Management Portal servers with load balancing, this will be a DNS name pointing to your load balancer. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created earlier appears in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.

Administration Section

In the Administration section, enter the Active Directory security group or groups that you will use to control administrative access to the Keyfactor Command Management Portal in the **Administrative Users** field. Multiple groups should be separated by commas with no trailing spaces. You may use the people picker button (👤) to browse for groups. Click the verify button (✅) to confirm that any entered groups are valid. Enter only the group(s) to which you want to grant full administrative rights to the Keyfactor Command Management Portal. Following initial configuration, you can create other permission levels and grant those permission levels to other Active Directory users or groups through the Keyfactor Command Management Portal. See *Security Roles and Identities* in the *Keyfactor Command Reference Guide* for more information.



Important: The built-in Active Directory groups Domain Admins and Enterprise Admins cannot be used directly to grant access to the Management Portal due to how these groups function within Windows. You can create a custom Active Directory group, reference that group in the Management Portal, and add the built-in Domain Admins or Enterprise Admins group to that custom group, if desired.



Important: The administrative group must be created as a Global or Universal group. If the administrative group is created as a domain-local group, it will not be recognized by the Management Portal Security Roles and Identities configuration. Configuration of the Management Portal will be incomplete until the group is deleted and recreated as a Global or Universal group.

Enrollment Section

In the Enrollment section of the page, modify the default **Certificate Subject Format** field, if desired. The subject values provided in this field are substituted at processing time for any entered by the user in PFX enrollment or provided with enrollment defaults if the template used is set to supply in request.

The data in the subject format takes precedence over any data entered during PFX enrollment or supplied by enrollment defaults (see *Configuring Template Options: Enrollment Defaults Tab* in the *Keyfactor Command Reference Guide*). For example, if you define the following subject format:

```
CN={CN},E={E},O=Key Example\, Inc.,OU={OU},L=Chicago,ST=IL,C=US
```

The organization for certificates generated through PFX enrollment will always be "Key Example, Inc." regardless of what is shown on the PFX enrollment page during enrollment.

This setting also applies to CSRs generated using the CSR generation method and to CSR and PFX enrollments done using the Classic API.

Data from the default subject *does not* display in the PFX enrollment form. To define defaults that will display in the PFX enrollment form (and can be modified by users), use enrollment defaults (see *Configuring Template Options: Enrollment Defaults Tab* in the *Keyfactor Command Reference Guide*).



Note: Backslashes are required before any commas embedded within values in the subject field (e.g. O=Key Example\, Inc.). Quotation marks should not be used in the strings in the fields except in the case where these are part of the desired subject value, as they are processed as literal values.



Tip: The default subject format *does not* apply to enrollments done using the CSR enrollment method or any requests done with the Keyfactor API.

PFX Enrollment Section

In the PFX Enrollment section of the page, uncheck the **Enabled** box if you do not wish to support PFX enrollment. If you wish to support PFX enrollment, leave the **Enabled** box checked. Select the **Domain** radio button

if you wish PFX files to be protected with the user's Active Directory password or select the **Auto-Generated** radio button if you wish PFX files to be protected with a one-time password. Check the **Alphanumeric Password Characters** box if you wish the one-time password used to protect PFX files to contain numbers and letters. Uncheck the **Alphanumeric Password Characters** box if you wish the one-time password used to protect PFX files to contain numbers, letters and special characters. In the **Password Length** field, enter a number for the number of characters the one-time password should have. The minimum value is 8. If you select the **Domain** radio button, the data entered in the password fields is not relevant.

CSR Enrollment Section

In the CSR Enrollment section of the page, uncheck the **Enabled** box if you do not wish to support CSR enrollment. If you wish to support CSR enrollment, leave the **Enabled** box checked.

The screenshot shows the 'Keyfactor Configuration Wizard' window. The 'Keyfactor Portal' tab is selected in the left-hand navigation pane. The main configuration area is divided into several sections:



- Administration**: Includes 'Host Name' (keyfactor.keyexample.com), 'Web Site' (Default Web Site), 'Virtual Directory' (KeyfactorPortal), and 'Application Pool' (Keyfactor). There is a 'Use SSL' checkbox which is checked.
- Administrative Users**: A text field containing 'KEYEXAMPLE\Keyfactor Administrator' with user selection icons.
- Enrollment**: Includes 'Certificate Subject Format' (CN={CN},E={E},O=Key Example\, Inc,OU=IT,L=).
- PFX Enrollment**: Includes a checked 'Enabled' checkbox, 'PFX Password Type' (radio buttons for Domain and Auto-Generated, with Auto-Generated selected), a checked 'Alphanumeric Password Characters' checkbox, and 'Password Length' (12).
- CSR Enrollment**: Includes a checked 'Enabled' checkbox.

At the bottom, there are 'Verify Configuration' and 'Cancel' buttons. A status bar at the very bottom shows 'Server: sqlsrvr05.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 32: Configure: Keyfactor Portal

17. Dashboard and Reports Tab

On the Dashboard and Reports tab, enter the FQDN of the server hosting the Keyfactor Command Management Portal—where the Logi Analytics Platform is installed—in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Management Portal component or a DNS alias pointing to the server. Check or uncheck the **Use SSL** box as appropriate for your

environment. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created earlier appears in the **Application Pool** dropdown. In the **Keyfactor Site IP Address(es)** field, enter the IPv4 (and IPv6 (if applicable), separated by a comma) IP address(es) of the server hosting the Keyfactor Command Management Portal in a comma-delimited list. If you plan to use integrated Windows authentication (see [Configure Kerberos Authentication on page 73](#)) to access the Management Portal, uncheck the **Use Basic Authentication** box. If you plan to use Basic authentication to access the Management Portal, check the **Use Basic Authentication** box and enter the user name (DOMAIN\username format) and password of the Active Directory service account that the Logi Analytics Platform will use to access Keyfactor Command (using the Keyfactor API). This can be the same service account used for the application pool or a different service account. You may use the people picker button () to browse for the account. Click the verify button () to confirm that the username and password entered are valid.



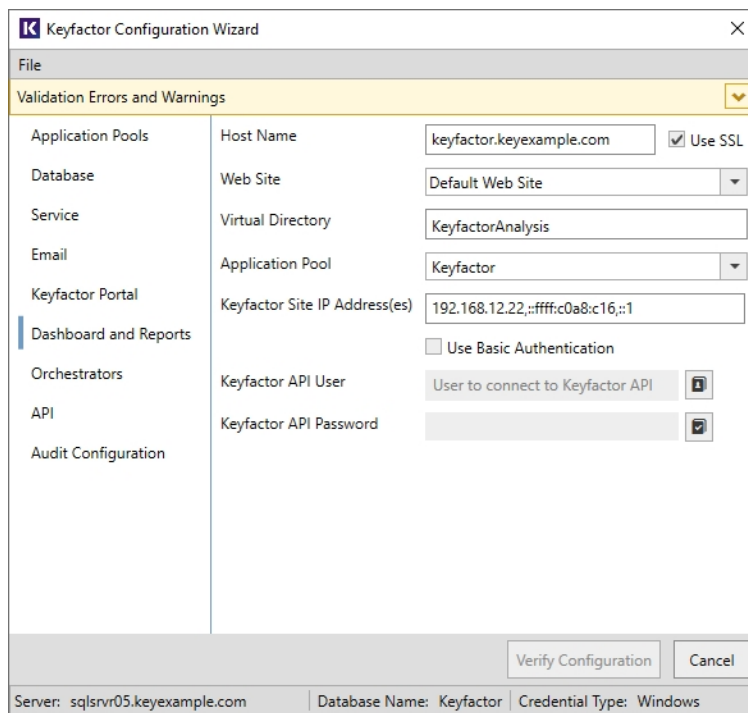
Note: If desired, you can configure the **Host Name** field as *localhost* and then configure the **Keyfactor Site IP Addresses(es)** field as *127.0.0.1,::1* (to cover both the IPv4 and IPv6 loopback addresses). You cannot mix and match actual host names and IP addresses with localhost and loopback addresses—e.g. setting **Host Name** to *keyfactor.keyexample.com* and **Keyfactor Site IP Address(es)** to *127.0.0.1,::1* will not work.



Note: If you are installing the Management Portal in a load balanced configuration, see [Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup on page 138](#).



Note: If you do not enter *::1* (the loopback address for IPv6) in the **Keyfactor Site IP Address(es)** field, the configuration wizard automatically appends this for you. Having extra names and/or addresses by which the Management Portal might be known in this field allows Logi to connect to Keyfactor Command in the most scenarios possible.



The image shows the 'Keyfactor Configuration Wizard' window, specifically the 'Dashboard and Reports' tab. The window has a sidebar on the left with various configuration categories: Application Pools, Database, Service, Email, Keyfactor Portal, Dashboard and Reports (selected), Orchestrators, API, and Audit Configuration. The main area contains several configuration fields:

- Host Name:** A text box containing 'keyfactor.keyexample.com' with a checked 'Use SSL' checkbox.
- Web Site:** A dropdown menu showing 'Default Web Site'.
- Virtual Directory:** A text box containing 'KeyfactorAnalysis'.
- Application Pool:** A dropdown menu showing 'Keyfactor'.
- Keyfactor Site IP Address(es):** A text box containing '192.168.12.22::ffff:c0a8:c16::1'.
- Use Basic Authentication:** An unchecked checkbox.
- Keyfactor API User:** A text box containing 'User to connect to Keyfactor API' with a password icon.
- Keyfactor API Password:** A text box with a password icon.

At the bottom, there are 'Verify Configuration' and 'Cancel' buttons. A status bar at the very bottom shows: 'Server: sqlsrvr05.keyexample.com | Database Name: Keyfactor | Credential Type: Windows'.

Figure 33: Configure: Dashboard and Reports

18. vSCEP Services Tab

On the vSCEP Service tab (this tab won't appear if you installed only the default components), enter the FQDN of the server hosting the Keyfactor Command vSCEP™ service in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (vSCEP Validation Service) components or a DNS alias pointing to the server. Check or uncheck the **Use SSL** box as appropriate for your environment. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created earlier appears in the **Application Pool** dropdown. Select a certificate template you will use with this service in the **SCEP Certificate Template** dropdown. Enter the full path to the SCEP challenge page for the SCEP server in the SCEP Path field. This path should be given in full URL format as follows (where MICROSOFT_NDES_SERVER_FQDN is the FQDN of your Microsoft NDES server or Keyfactor_SCEP_SERVER_FQDN is the FQDN of your Keyfactor SCEP server):

- For Microsoft NDES:
https://[MICROSOFT_NDES_SERVER_FQDN]/certsrv/mscep_admin
- For Keyfactor SCEP:
https://[KEYFACTOR_SCEP_SERVER_FQDN]/scep/challenge

Your Microsoft NDES or Keyfactor SCEP server may have been configured to use HTTP rather than HTTPS. Enter the full path to the SCEP enrollment page for the SCEP server in the **Request Path** field. This path should

be given in full URL format as follows (where MICROSOFT_NDES_SERVER_FQDN is the FQDN of your Microsoft NDES server or Keyfactor_SCEP_SERVER_FQDN is the FQDN of your Keyfactor SCEP server):

- For Microsoft NDES:

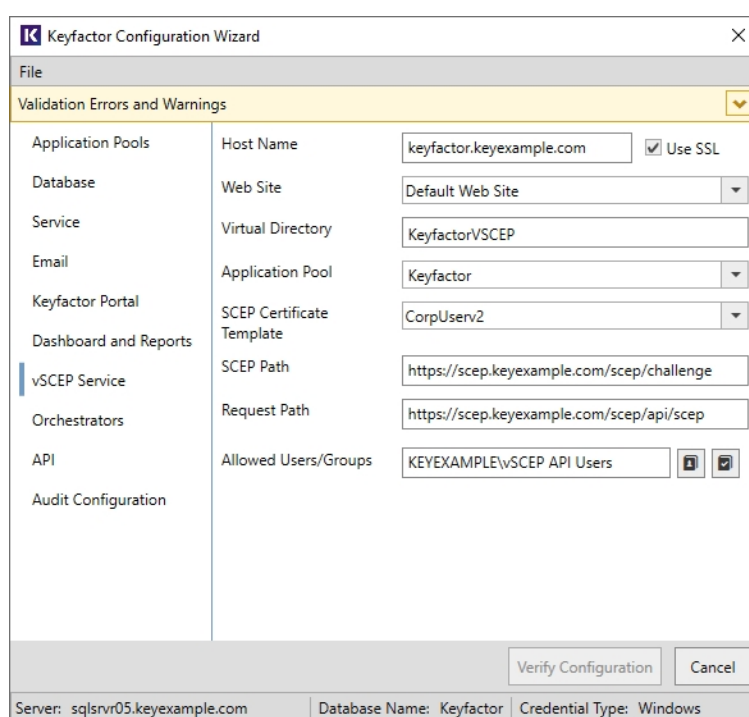
`https://[MICROSOFT_NDES_SERVER_FQDN]/certsrv/mscep/mscep.dll`

- For Keyfactor SCEP:

`https://[KEYFACTOR_SCEP_SERVER_FQDN]/scep/api/scep`

Your Microsoft NDES or Keyfactor SCEP server may have been configured to use HTTP rather than HTTPS.

Enter the Active Directory security group or groups that you will use to control access to the vSCEP API in the **Allowed Users/Groups** field or enter individual users (DOMAIN\username or DOMAIN\group name format). You may use the people picker button (👤) to browse for users or groups. Click the verify button (✅) to confirm that any entered users or groups are valid.



The image shows the 'Keyfactor Configuration Wizard' window, specifically the 'vSCEP Service' configuration tab. The window has a sidebar on the left with various configuration categories, and a main area on the right with configuration fields. At the bottom, there are buttons for 'Verify Configuration' and 'Cancel', and a status bar showing server, database, and credential information.

Category	Field	Value	Options
vSCEP Service	Host Name	keyfactor.keyexample.com	<input checked="" type="checkbox"/> Use SSL
	Web Site	Default Web Site	Dropdown
	Virtual Directory	KeyfactorVSCEP	
	Application Pool	Keyfactor	Dropdown
	SCEP Certificate Template	CorpUser2	Dropdown
	SCEP Path	https://scep.keyexample.com/scep/challenge	
	Request Path	https://scep.keyexample.com/scep/api/scep	
	Allowed Users/Groups	KEYEXAMPLE\vSCEP API Users	People Picker (👤) Verify (✅)
	Buttons: Verify Configuration, Cancel		
	Status Bar: Server: sqlsrvr05.keyexample.com Database Name: Keyfactor Credential Type: Windows		

Figure 34: Configure: vSCEP Service

19. Orchestrators

On the Orchestrators tab, enter the FQDN of the server hosting the Keyfactor Command orchestrators web site in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (Orchestrator Services API) components or a DNS alias pointing to the server. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created

earlier appears in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.

Reenrollment Section (Optional)

In the **Template For Submitted CSRs** field, from the dropdown, select the template to be used for reenrollment requests made from the Certificate Stores page.

In the **CA For Submitted CSRs** field, enter the certificate authority used for reenrollment requests made from the Certificate Stores page. The CA should be entered in the format FQDN\Logical Name.

The screenshot shows the 'Keyfactor Configuration Wizard' window. On the left is a sidebar with navigation links: File, Validation Errors and Warnings, Application Pools, Database, Service, Email, Keyfactor Portal, Dashboard and Reports, Orchestrators (selected), API, and Audit Configuration. The main area is divided into sections. The 'Reenrollment' section is highlighted in blue and contains two dropdown menus: 'Template For Submitted CSRs' set to 'Corp Web Server v2' and 'CA For Submitted CSRs' set to 'corpca01.keyexample.com\CorpIssuingCA1'. Below this is the 'Certificate Authentication' section, also highlighted in blue, which includes an 'Orchestrator Certificate Authentication' checkbox (unchecked), a 'Certificate Authentication HTTP Header' text field, and a 'Certificate Authentication Username' text field containing 'DOMAIN\user'. At the bottom right are 'Verify Configuration' and 'Cancel' buttons. The bottom status bar displays 'Server: sqlsrvr05.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 35: Configure: Orchestrators with Standard Authentication

Certificate Authentication Section (Optional)

In the Certificate Authentication section of the Orchestrators tab, check the **Enabled** box if you wish to support client certificate enrollment from the Keyfactor Universal Orchestrator or Keyfactor Windows Orchestrator. In the **Certificate Authentication HTTP Header** field, enter the HTTP header under which the orchestrator connection proxy should send the client authentication certificate. Keyfactor Command uses the certificate supplied in this header to identify the orchestrator attempting to authenticate. In the **Certificate Authentication Username** and **Certificate Authentication Password** fields, enter the credentials for the Active Directory user configured on the proxy to authenticate the orchestrator(s) to the Keyfactor Command server.

The screenshot shows the 'Keyfactor Configuration Wizard' window. The 'Orchestrators' tab is selected in the left sidebar. The main pane shows the 'Certificate Authentication' section, which is expanded. It includes the following fields and settings:

- Virtual Directory:** KeyfactorAgents
- Application Pool:** Keyfactor
- Reenrollment:**
 - Template For Submitted CSRs:** Corp Web Server v2
 - CA For Submitted CSRs:** corpca01.keyexample.com\CorpIssuingCA1
- Certificate Authentication:**
 - Orchestrator Certificate Authentication:** ☒ Enabled
 - Certificate Authentication HTTP Header:** X-ARR-ClientCert
 - Certificate Authentication Username:** KEYEXAMPLE\svc_kyforch
 - Certificate Authentication Password:** (masked with dots)

At the bottom of the wizard, there are buttons for 'Verify Configuration' and 'Cancel'. The status bar at the very bottom shows: 'Server: sqlsrvr05.keyexample.com | Database Name: Keyfactor | Credential Type: Windows'.

Figure 36: Configure: Orchestrators with Client Certificate Authentication

20. API Tab

On the API tab, enter the FQDN of the server hosting the Keyfactor Command KeyfactorAPI service in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (Keyfactor API) components or a DNS alias pointing to the server. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created earlier appears in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.

Classic API Section

In the Classic API section of the API page, check the **Enabled** box if you wish to make use of the Classic API (a.k.a. the CMS API). The classic API may be needed in your environment if you're upgrading from a previous version of Keyfactor Command and have written API applications using the classic API. Enter the FQDN of the server hosting the Keyfactor Command Classic API service in the **Host Name** field. This can be either the actual host name of the server on which you are installing the Keyfactor Command Services (Classic API) components or a DNS alias pointing to the server. Select the Default Web Site in the **Web Site** dropdown, or other web site as desired. Accept the default for the **Virtual Directory** and confirm that the application pool for Keyfactor Command that you created earlier appears in the **Application Pool** dropdown. Check or uncheck the **Use SSL** box as appropriate for your environment.

Figure 37: Configure: APIs

21. Auditing Configuration Tab

On the Auditing Configuration tab, enter the number of years to retain audit data in the **Audit Entry Retention Period (years)** field. By default, seven years of data is retained. The audit log cleanup job runs once daily and removes any audit log entries older than the time specified in the retention parameter except those in the following protected categories:

- Security
- CertificateCollections
- ApplicationSettings
- SecurityIdentities
- SecurityRoles

Linux SysLog Server Section

In the Linux SysLog Server section of the page, check the **Connect to SysLog** to enable the option to copy audit logs in real time to a separate server for collection and analysis with a centralized logging solution (e.g. rsyslog, Splunk, Elastic Stack). In the **Host Name** field, enter the fully qualified domain name of the server that will be receiving the logs. Set the **Port** to the port on which your log receipt application is listening to receive the logs. The default value is 514 (the default rsyslog port). If desired, turn on **Use TLS SysLogging**. When you click **Save**, Keyfactor Command will verify that a connection can be made to the specified server on the specified port. Additional configuration on both the Keyfactor Command server and log receipt server are

needed to make TLS communications work (see [Prepare for External Log Shipping over TLS \(Optional\) on page 37](#)). If you have not yet completed these configurations, you will receive a validation error on save if the Use TLS SysLogging option is enabled.

The auditing settings can be updated on the auditing tab of the applications settings page following installation (see *Application Settings: Auditing Tab* in the *Keyfactor Command Reference Guide*).

The screenshot shows the 'Keyfactor Configuration Wizard' window with the 'Audit Configuration' tab selected. The left sidebar lists various configuration categories, with 'Audit Configuration' highlighted. The main panel displays the following settings:

- Audit Entry Retention Period (Years):** 7
- Linux SysLog Server:** Selected from a dropdown menu.
- Connect to Linux SysLog:** Checked (checkbox).
- Host Name:** appsvr162.keyexample.com
- Port:** 10514
- Use TLS SysLogging:** Checked (checkbox).

At the bottom right, there are 'Verify Configuration' and 'Cancel' buttons. At the bottom left, a status bar shows: 'Server: sqlsrvr05.keyexample.com | Database Name: Keyfactor | Credential Type: Windows'.

Figure 38: Configure: Audit

22. At this point in the configuration, if you have populated all the required fields, the yellow warning banner at the top of the configuration wizard should have disappeared. If it is still visible, click the dropdown arrow to open the Warnings page and review the warning(s) to see what needs to be corrected. Under some circumstances you will be allowed to continue with the configuration even if the yellow warning banner is still present. You will know this is the case if the **Verify Configuration** button is active. Under these circumstances, you should review the warnings before continuing.

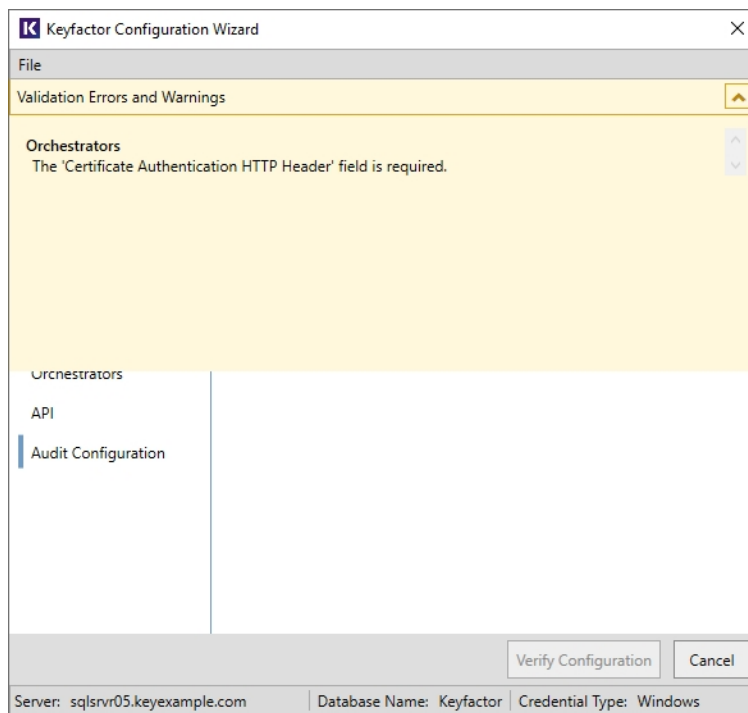


Figure 39: Configure: Configuration Warnings

23. Before completing the configuration wizard, you may choose to save a copy of the configuration as a file for future use. To download the configuration as a file, in the configuration wizard, click **File** at the top of the wizard and choose **Save Data File**. Browse to a location where you want to save the configuration file, enter a file name and click **Save**. You will be prompted to enter a password to encrypt the data in the file. You may choose to protect the file with a password or not. If you use a password at this time, you will need to provide this password to open the file. Keyfactor strongly recommends using a password to protect production files. If you do not wish to use a password to protect a production file, you may edit the file to remove the sensitive information (passwords for the service accounts entered in the configuration wizard). Once you enter a password or uncheck the encryption box, click **OK** to save the file.

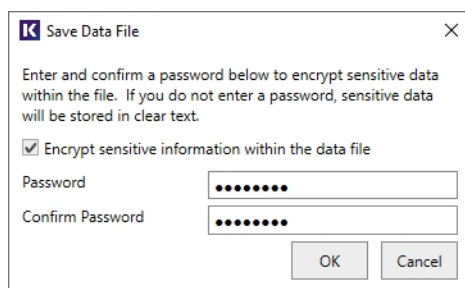


Figure 40: Configure: Save Configuration as a File

24. At the bottom of the Keyfactor Command Configuration Wizard dialog, click **Verify Configuration**.

25. On the Configuration Operations page, review the planned operations and then click **Apply Configuration**. Prior to clicking **Apply Configuration**, you can revisit any of the Configuration Wizard tabs to review or make changes by clicking **Edit Configuration**.

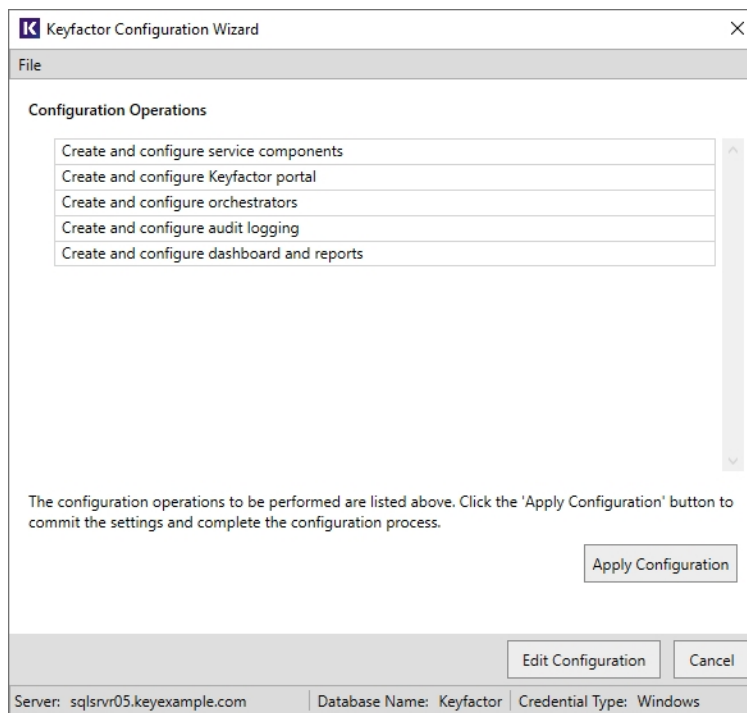


Figure 41: Configure: Configuration Operations

26. When the configuration completes successfully, you will see the below message. If you didn't save a copy of the configuration earlier, you may do so at this time by clicking **Save Settings**. Otherwise, click **Close** to close the dialog.

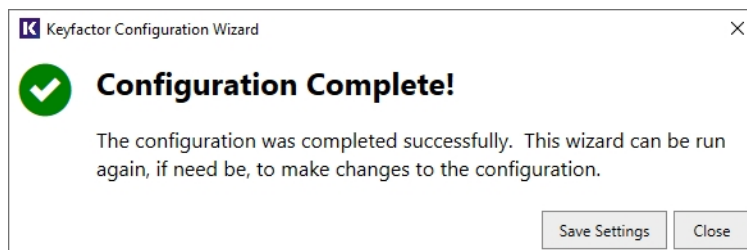


Figure 42: Configure: Configuration Complete

2.4.3.2 Install the Keyfactor Command Server from the Command Line

The Keyfactor Command server can optionally be configured using a pair of configuration files and a command run from the command line. You may be provided one or both of these files by your Keyfactor Customer Success Manager. The configuration files for command-line configuration are:

- Keyfactor Command Configuration File

This file, with an extension of .cmscfg, contains information in XML format to configure the Keyfactor Command database. This file can be generated by installing Keyfactor Command, running the configuration wizard and populating all the fields as desired, and then saving a copy of the configuration either with or without a password to encrypt sensitive information in the file (see [Install the Main Keyfactor Command Components on the Keyfactor Command Server\(s\) on page 42](#)). Keyfactor strongly recommends using a password to protect production files.

- Input Parameters File

This file, with an extension of .xml, contains information in XML format to connect to and configure SQL, open the Keyfactor Command configuration file, locate the Keyfactor Command license, and create application pools, if desired.

To configure and optionally install Keyfactor Command from the command line:

1. Install the Keyfactor Command software using one of these methods:

- Follow the initial instructions for [Install the Main Keyfactor Command Components on the Keyfactor Command Server\(s\) on page 42](#) except on the final installation wizard page, uncheck the *Launch the Configuration Wizard now* box and click **Finish**. The configuration wizard should not open.
- Open an administrative command prompt and execute a command similar to the following:

```
start /wait msiexec /i <full path to install file>\KeyfactorPlatform.msi /Live
<path for msiexec logs> /Quiet
```

This will install the default components of Keyfactor Command in a non-interactive way (/Quiet), output log information to a file (/Live), and wait to return to the command prompt until the installation is complete (start /wait).

If you wish to install a set of features other than the default features, you can add the ADDLOCAL parameter and specify the features you wish to install. For example, the following command will install the *Orchestrator Service API* and *Windows Services* features:

```
start /wait msiexec /i <full path to install file>\KeyfactorPlatform.msi
ADDLOCAL=AgentServicesFeature,ServiceFeature /Live <path for msiexec logs> /Quiet
```

The following features are available:

- AgentServicesFeature
This installs the Orchestrator Service API feature.
- ConfigurationFeature
This installs the configuration wizard and is required for all installations.
- ServiceFeature
This installs the Windows Services feature, which includes the Keyfactor Command Service (a.k.a. the timer service).
- VCRedistFeature
This installs the Microsoft Visual C++ Redistributable and is required for all installations unless it has been separately installed.
- vSCEPValidationFeature

This installs the vSCEP Validation Service feature. It is not included in the default features.

- WebApiFeature

This installs the WebAPI feature, which includes both the Keyfactor API and the Classic API.

- WebConsoleFeature

This installs the Management Portal feature, which includes the Keyfactor Command Management Portal and the Keyfactor API.

The features you decide to install will depend on the role the server will be playing in your Keyfactor Command implementation. [Table 6: Features Required for Each Server Role](#) shows the minimum features that need to be installed for each of the server roles shown in the table columns. If you're installing all the required features on a single server, you need everything except *vSCEPValidationFeature*, which is only required for customers needing to validate SCEP requests, and *WebApiFeature*, which is only required for customers wishing to use the Classic API (see Classic API in the *Keyfactor Web APIs Reference Guide*). If you don't intend to use any orchestrators (see *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide*), you do not need to install the *AgentServicesFeature*.

Table 6: Features Required for Each Server Role

ADDLOCAL Parameter	Single Server	Management Portal	Windows Services	WebAPI	Orchestrator Service API	vSCEP Validation Service
ConfigurationFeature	✓	✓	✓	✓	✓	✓
VCRedistFeature	✓	✓	✓	✓	✓	✓
WebConsoleFeature	✓	✓				
ServiceFeature	✓		✓			
WebApiFeature				✓		
AgentServicesFeature	✓				✓	
vSCEPValidationFeature						✓

2. Acquire a Keyfactor Command configuration file from your Keyfactor Customer Success Manager or create one by installing and configuring Keyfactor Command on a test machine. It's not practical to attempt to generate this file manually, though a file can be edited once generated (other than password-protected fields).

3. Create an input parameters file. See [Table 7: Input Parameters XML File Fields](#). A sample file can be found in the Configuration directory under the directory in which you installed Keyfactor Command. By default, this is:

C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\InputParameters.xml

4. Open an administrative command prompt, change to the Configuration directory under the directory in which you installed Keyfactor Command (by default this is C:\Program Files\Keyfactor\Keyfactor Platform\Configuration), and execute a command similar to the following, referencing your input parameters file and using the appropriate parameters for the ConfigurationWizardConsole tool (see [Table 8: ConfigurationWizardConsole.exe Options](#)):

.\ConfigurationWizardConsole.exe -p C:\Stuff\InputParameters.xml -u



Tip: Check the Keyfactor Command log and the Windows application event log for errors if the installation does not complete successfully (see [Configure Logging on page 81](#)).

Table 7: Input Parameters XML File Fields

Parameter	Description
Protected	A Boolean indicating whether sensitive information in the Keyfactor Command configuration file is protected with a password (true) or not (false).
Password	A string containing the password used to protect the Keyfactor Command configuration file if <i>Protected</i> is set to true .
ConfigurationFile	The full path to the Keyfactor Command configuration file (e.g. C:\Stuff\myconfig.cmscfg).
DatabaseServer	The hostname or IP address of the SQL server where the Keyfactor Command database will be installed, with optional port. For example: <ul style="list-style-type: none"> Local: mysql.keyexample.com Azure SQL myazuresql.database.windows.net,1433
Database	The name of the database in SQL for Keyfactor Command. If a database with this name exists, it will be used (see <i>ForceDatabaseConversion</i>). If it doesn't, it will be created (see <i>CreateDatabaseIfMissing</i>).
CreateDatabaseIfMissing	A Boolean indicating whether the SQL database should be created if it does not exist (true) or not (false). If this is set to false and a database does not exist, an error will be generated and the configuration will not continue.
ForceDatabaseConversion	A Boolean indicating whether a pre-existing SQL database should be converted for use by Keyfactor Command (true) or not (false). If this is set to false and a pre-existing database that has not already been converted for Keyfactor Command use is found, an error will be generated and the configuration will not continue.
ForceDatabaseUpgrade	A Boolean indicating whether a pre-existing SQL database should be upgraded from a previous version of Keyfactor Command (true) or not (false). If this is set to false and a pre-existing database that is running a version of Keyfactor Command that does not match the version being installed is found, an error will be generated and the configuration will not continue.
ContinueOnSqlGrantError	A Boolean indicating whether the configuration should continue if an error is encountered when attempting to set SQL permissions.
SqlUsername	A string containing the SQL username to be used to authenticate to the SQL server if you have opted not to use Windows integrated authentication. For an on-premise SQL server, the server must be configured to support mixed mode authentication in order to use the SQL option. This option is typically used to connect to cloud-based (e.g. Azure) SQL servers. Leave this field blank if you are using Windows integrated authentication. The credentials of the logged on user executing the command will be used to authen-

Parameter	Description										
	authenticate to SQL.										
SqlPassword	A string containing the SQL password to be used to authenticate to the SQL server. Leave this field blank if you are using Windows integrated authentication.										
LicenseFile	The full path to your Keyfactor Command license file (e.g. C:\Stuff\keyexample.cmslicense).										
ApplicationPoolsToCreate	<p>An array of application pools to create (typically only one is needed). Application pool fields include:</p> <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td>Name</td><td>A string containing the name of the application pool to create.</td></tr> <tr> <td>Username</td><td>A string containing the user name (DOMAIN\username format) of the Active Directory service account under which the application pool will run.</td></tr> <tr> <td>Password</td><td>A string containing the password of the Active Directory service account under which the application pool will run.</td></tr> <tr> <td>FailIfExists</td><td>A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).</td></tr> </table> <p>For example:</p> <pre><ApplicationPoolsToCreate> <!--Remove this section if none are to be created--> <WizardApplicationPool> <Name>Keyfactor</Name> <Username>KEYEXAMPLE\svc_keyfactorpool</Username> <Password>MySecurePassword</Password> <FailIfExists>true</FailIfExists> </WizardApplicationPool> </ApplicationPoolsToCreate></pre>	Parameter	Description	Name	A string containing the name of the application pool to create.	Username	A string containing the user name (DOMAIN\username format) of the Active Directory service account under which the application pool will run.	Password	A string containing the password of the Active Directory service account under which the application pool will run.	FailIfExists	A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).
Parameter	Description										
Name	A string containing the name of the application pool to create.										
Username	A string containing the user name (DOMAIN\username format) of the Active Directory service account under which the application pool will run.										
Password	A string containing the password of the Active Directory service account under which the application pool will run.										
FailIfExists	A Boolean indicating whether the configuration will fail if the application pool already exists (true) or not (false).										

Table 8: ConfigurationWizardConsole.exe Options

Switch	Description
-p, --paramfile	The full path to the input parameters XML file. This switch is required .
-u, --unattended	Do not output errors at the console. Errors will be redirected to the Windows event log.
-d, --database	Create the database in SQL but do not configure Keyfactor Command.
-s, --scriptpath	The full path to a non-standard location for the scripts used during a database upgrade. By default, these are found in the following path: <div>C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\DatabaseUpgrade</div> This option is typically only used by Keyfactor Support.
--help	Display the help.
--version	Display the version information.

2.4.4 Initial Configuration

Once the installation and configuration wizards are complete, only a few configuration tasks remain before Keyfactor Command will be up and running at a basic level. This section details the basic post-install configuration steps that need to be completed to get Keyfactor Command up and running. See the *Keyfactor Command Reference Guide* for more advanced configuration guidance. See the separate installation guides for client components such as the Keyfactor Universal Orchestrator, Keyfactor Windows Orchestrator, Keyfactor Java Orchestrator, and Keyfactor CA Gateways.

After you have completed all the steps in this guide, the certificate search and report functions in the Keyfactor Command Management Portal should be functioning. Further configuration, as described in the *Keyfactor Command Reference Guide*, is required to make these features function:

- Using the Keyfactor Command Management Portal *Dashboard*
- Configuring Enrollment through the Keyfactor Command Management Portal (see *Configuring Template Options*)
- *Security Roles and Identities* for the Keyfactor Command Management Portal
- *Revocation Monitoring, Expiration Alerts and Pending Certificate Request Alerts*
- Using the Workflow Builder (see *Workflow*)
- External Certificate Synchronization with *SSL Discovery and Certificate Stores*
- Managing *SSH Keys*

2.4.4.1 Configure Kerberos Authentication

By default, the Keyfactor Command Management Portal uses integrated Windows authentication. Integrated authentication consists of both NTLM and Kerberos authentication types. In some environments, NTLM will work

for integrated authentication and users will be able to open the Keyfactor Command Management Portal without further configuration, though not all aspects of the portal support NTLM, including the dashboard and enrollment. In other environments, NTLM will not work at all for the portal, so only Kerberos will be supported. Further configuration is required to make Kerberos authentication work correctly. Even if NTLM is supported and you don't intend to use the portions of the portal that don't work with NTLM, Kerberos is generally preferred for best security practice.

Common scenarios in which NTLM will not work are multi-domain forests and authentication attempts between domains and servers that support only NTLMv2 using clients attempting NTLM.

Configuring the environment to support Kerberos includes these topics:

- Configure browsers to support Integrated Windows Authentication
- Configure the service principal name (SPN) for the Keyfactor Command server
- Configure Kerberos constrained delegation (optional)



Note: Basic authentication can be used instead of integrated Windows authentication.

Configure Browsers for Integrated Windows Authentication

To support integrated Windows authentication using either NTLM or Kerberos, the browser must be configured correctly to support this integration. This becomes particularly important when only Kerberos is used, as the browser won't allow the user to continue if Kerberos authentication fails, whereas with NTLM authentication, the integration won't work (the user will be prompted to enter a password), but the user will be allowed to continue to the Keyfactor Command portal. Many modern browsers support integrated authentication. The following instructions cover adding the Keyfactor Command server to Windows's trusted sites to support integrated authentication for Microsoft Edge and Google Chrome. Configuring Firefox to support integrated authentication is beyond the scope of this guide.



Important: Internet Explorer is no longer supported for Keyfactor Command. For a list of supported browsers, see [System Requirements on page 8](#).

To configure Windows to support integrated authentication:

1. In Windows either do a search for Internet Options or open Control Panel or Settings and locate Internet Options.
2. In Internet Options, go to the Security tab.
3. On the Security tab, highlight **Local intranet** and click **Sites**.
4. On the Local intranet sites popup, click **Advanced**.
5. On the Local intranet dialog, enter the fully qualified domain name of your Keyfactor Command server and click **Add**.
6. Click **Close** and **OK** until you have closed all the dialogs.

7. Exit your browser (this setting applies to Microsoft Edge and Google Chrome) and open it again to attempt your authentication.

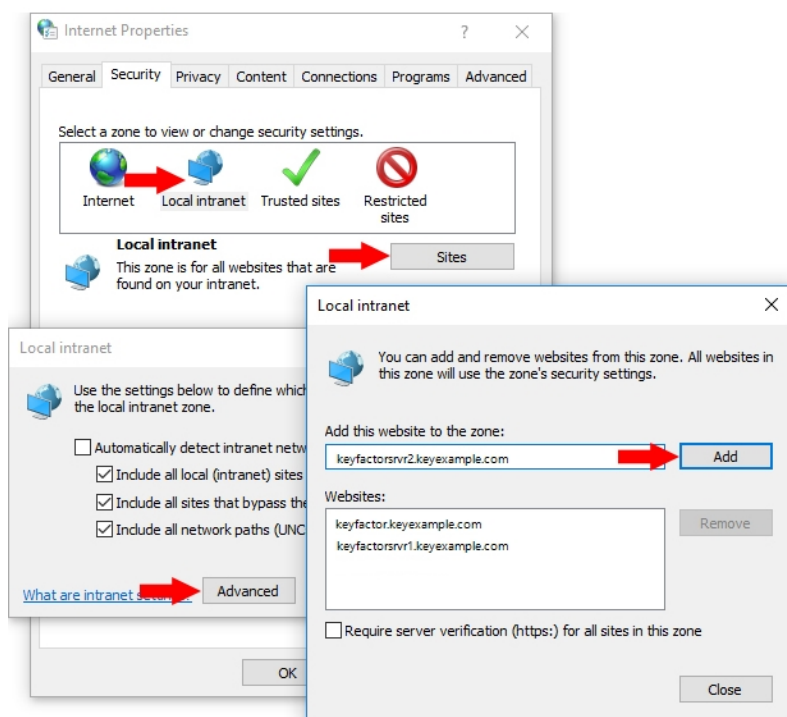


Figure 43: Configure Local Intranet Zone in Internet Properties



Important: It is not sufficient to put the Keyfactor Command server in the **Trusted sites** zone. The server needs to be in the **Local intranet** zone for proper integrated authentication functionality (assuming these zones are still configured as per the default configuration).

Configure the Service Principal Name for the Keyfactor Command Server

On a server that has the `setspn` command available (typically it is available on domain controllers, as it installs as part of the Active Directory Domain Services role), open a command prompt using the "Run as administrator" option and run the following command (where `keyfactor.keyexample.com` is the fully qualified domain name of your Keyfactor Command server or the DNS alias you are using to reference your Keyfactor Command server, if applicable, and `KEYEXAMPLE\svc_keyfactorpool` is the domain name and service account name of the service account under which the Keyfactor Command application pool is running):

```
setspn -s HTTP/keyfactor.keyexample.com KEYEXAMPLE\svc_keyfactorpool
```

Configure Kerberos Constrained Delegation (Optional)

If either of these scenarios is true in your environment, you will need to configure Kerberos delegation to the CAs from the Keyfactor Command server hosting the Keyfactor Command Management Portal:

- You wish to use the option in Keyfactor Command to allow interactions with the CA via the Keyfactor Command Management Portal (e.g. certificate approval or revocation) to be done in the context of the user logged into the Keyfactor Command Management Portal rather than in the context of the Keyfactor Command service account under which the application pool is running or an explicit user configured in the CA configuration within Keyfactor Command.
- You wish to enroll for certificates through the Keyfactor Command Management Portal after authenticating to the portal using Kerberos authentication rather than Basic authentication. If you wish to use the Keyfactor Command Management Portal but don't wish to configure delegation or an explicit user configured in the CA configuration within Keyfactor Command, you will need to set the Keyfactor Command Management Portal to support Basic authentication only.

Configuring Kerberos delegation in Active Directory allows the user's Kerberos credentials to be delegated from the Keyfactor Command server to the CA(s) to allow the Keyfactor Command server to act on behalf of the user.

The types of interactions affected by delegation in the Keyfactor Command Management Portal include:

- Enrollment for certificates
- Approval of pending certificate requests
- Denial of pending certificate requests
- Revocation of certificates
- Certificate key recovery



Note: You have the option to turn off delegation for these functions using the *Delegate* settings on each CA configured in Keyfactor Command (see *Certificate Authority Operations: Adding or Modifying a CA Record Authorization Methods Tab* in the *Keyfactor Command Reference Guide*). Delegation is configured separately for management and enrollment functions.

There are two different approaches to configuring constrained delegation:

- With the traditional version of constrained delegation, you configure the service account under which the Keyfactor Command application pool runs and the machine account of the Keyfactor Command server to be allowed to delegate **to** each of your CAs.
- With the newer resource-based constrained delegation introduced in Windows server 2012, you configure each of your CAs to be allowed to receive delegation **from** the service account under which the Keyfactor Command application pool runs and the machine account of the Keyfactor Command server. This option requires at least one domain controller that's server 2012 or better, though there can be 2008 or 2008 R2 domain controllers in the mix.

With both approaches to constrained delegation, you need to set the service principal name (SPN) for the Keyfactor Command server (see [Configure the Service Principal Name for the Keyfactor Command Server on the previous page](#)).



Note: If you're using a Keyfactor CA gateway and the gateway service is running as an Active Directory service account, delegation to that gateway is configured differently than is described below. Refer to the gateway documentation for more information.

Traditional Delegation



Note: Traditional constrained Kerberos delegation across multiple domains is only supported in newer versions of Windows Server for domain controllers. If yours is a multi-domain environment and you cannot locate your CAs following the below instructions, you may need to configure traditional unconstrained Kerberos delegation or configure traditional constrained delegation using ADSIEdit rather than the below method. To configure traditional unconstrained delegation, you would select "Trust this computer for delegation to any service (Kerberos only)" in each of the step 2s, below, and then skip the remainder of the steps in that set of instructions. For assistance configuring traditional constrained delegation using ADSIEdit, contact Keyfactor support (support@keyfactor.com). If none of the traditional constrained delegation methods work in your multi-domain environment, you may need to pursue resource-based constrained delegation instead, which is more forgiving of multi-domain environments.

To configure Kerberos constrained delegation on the machine account of the Keyfactor Command server:

1. Open Active Directory Users and Computers and browse to locate the **machine** account of the Keyfactor Command server and open its properties.
2. On the Delegation tab for the machine account, choose "Trust the computer for delegation to specified services only" and under that "Use any authentication protocol" and then click **Add**.
3. In the Add Services dialog, click **Users or Computers** and browse to locate the computer account for one of the CAs to which you wish to delegate.

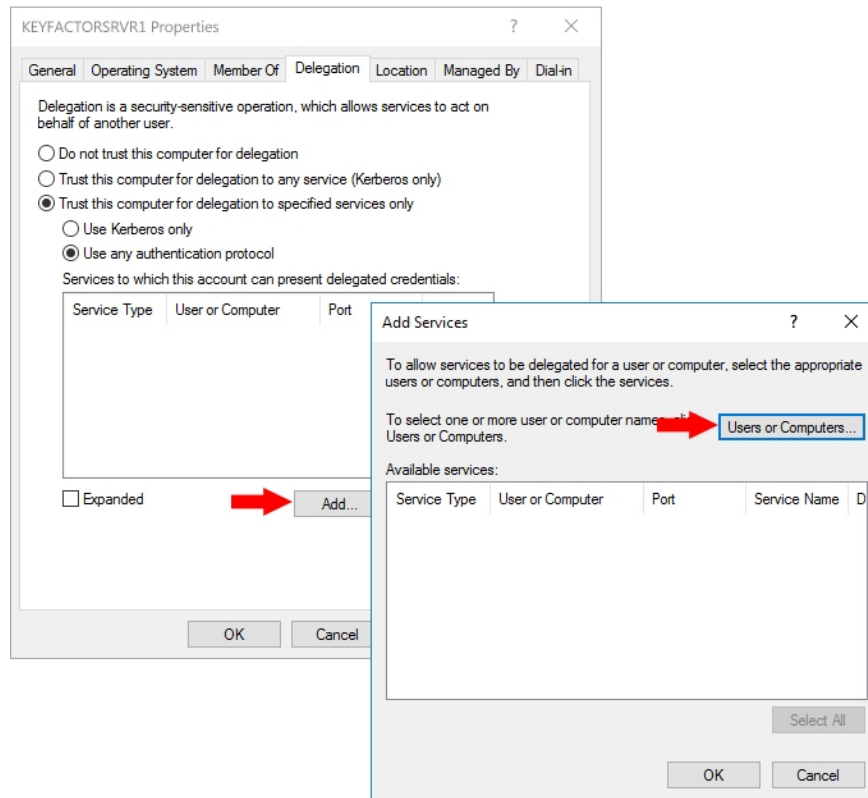


Figure 44: Configure Kerberos Constrained Delegation on the Keyfactor Command Machine Account

4. In the Add Services dialog once the available services have populated, highlight both the **HOST** and the **rpcss** services (hold down the CTRL key when clicking the second service to select both at the same time) and click **OK**.

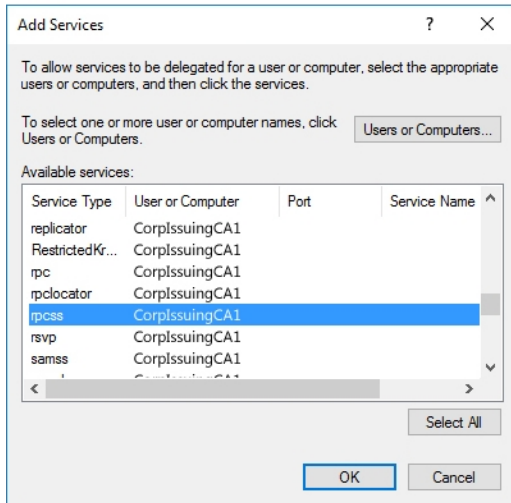


Figure 45: Add HOST and rpcss Service Types for Kerberos Constrained Delegation

5. Back on the Delegation tab of the machine account Properties, you should see the HOST and rpcss services populate the "Services to which this account can present delegated credentials box". The server names do not appear as fully qualified domain names until you close the properties dialog and open it again.
6. Repeat steps 3 and 4 for any other CAs to which you wish to delegate and then click **OK**.

To configure Kerberos constrained delegation on the **service** account under which the Keyfactor Command application pool is running:

1. Open Active Directory Users and Computers and browse to locate the service account under which the Keyfactor Command application pool is running and open its properties.
2. On the Delegation tab for the service account, choose "Trust the computer for delegation to specified services only" and under that "Use Kerberos only" and then click **Add**.



Important: This is a different configuration setting than for the machine account.



Tip: The Delegation tab only appears on the properties sheet after you have configured a custom SPN.

3. In the Add Services dialog, click **Users or Computers** and browse to locate the computer account for one of the CAs to which you wish to delegate.
4. In the Add Services dialog once the available services have populated, highlight both the **HOST** and the **rpcss** services (hold down the CTRL key when clicking the second service to select both at the same time) and click **OK**.

- Back on the Delegation tab of the service account Properties, you should see the HOST and rpcss services populate the "Services to which this account can present delegated credentials box". The server names do not appear as fully qualified domain names until you close the properties dialog and open it again.

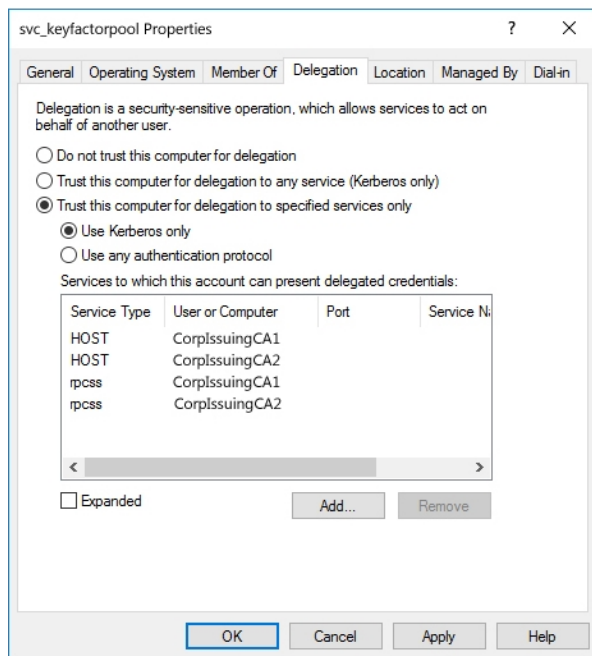


Figure 46: Configure Kerberos Constrained Delegation on the Keyfactor Command Service Account

- Repeat steps 3 and 4 for any other CAs to which you wish to delegate and then click **OK**.

Resource-Based Delegation

To configure Kerberos resource-based constrained delegation:

- Create at least one Active Directory security group that will be used for granting delegation permission.



Tip: The type of group to use and domain to place it in will vary depending on your forest structure and the location of the accounts and resources in the forest. If your Keyfactor Command server(s), CA(s) and application pool service account are all in the same domain, any type of group in that same domain will do. If your Keyfactor Command server(s), CA(s) and/or application pool service account are separated across multiple domains, it becomes more complicated. If you add one or more cross-forest trusts into the mix that you want to do delegation across, that adds another level of complexity. Universal groups cannot be used in a cross-forest scenario, as they are not supported cross-forest. Some possible scenarios include:

- All components (Keyfactor Command server(s), CA(s) and application pool service account) in the same domain: Create a group of any type in the same domain as these components.
- Keyfactor Command server(s) and application pool service account in child domain A and all CAs in the parent domain or child domain B with no cross-forest involvement: Create a universal group in the domain with the CAs (the parent domain or child domain B).



- Keyfactor Command server(s), application pool service account and CA(s) for forest A in the same domain, CAs in a single domain in forest B, forests A and B in a two way trust: Create a group of any type in the forest A domain where the Keyfactor Command server(s) reside and a group of type *domain local* in the forest B domain where the CAs reside; follow the below instructions for both domains and groups.

2. Add the service account under which the Keyfactor Command application pool runs to this new security group.
3. Add the machine account for the Keyfactor Command server to this new security group. Repeat for additional Keyfactor Command servers.
4. On an Active Directory domain controller running Windows Server 2012 or better, open a PowerShell window using the "Run as administrator" option. If you're in a multi-domain or cross-forest environment, use a domain controller in the resource domain where the CAs exist.
5. In the PowerShell window, run the following commands, where *KerberosDelegationGroup* is the name of your group for Kerberos delegation and *IssuingCA* is the machine name (no trailing \$) of the CA you wish to delegate to:

```
$mygroup = Get-ADGroup -Identity KerberosDelegationGroup  
Set-ADComputer IssuingCA -PrincipalsAllowedToDelegateToAccount $mygroup
```
6. Repeat the Set-ADComputer step for any additional CAs.
7. In the PowerShell window, run the following command for each CA to confirm that the group has been associated with the PrincipalsAllowedToDelegateToAccount property on the CA account:

```
Get-ADComputer IssuingCA -Properties PrincipalsAllowedToDelegateToAccount
```

2.4.4.2 Configure Logging

Keyfactor Command provides extensive logging for visibility and troubleshooting. By default, Keyfactor Command places its log files in the C:\Keyfactor\logs directory, generates logs at the "Info" logging level and stores the primary logs for two days before deleting them. If you wish to change these defaults you can open the configuration file for each type of log on each Keyfactor Command server where you wish to adjust logging, and edit the file in a text editor (e.g. Notepad) using the "Run as administrator" option. Each component now has its own NLog configuration file and NLog logging output path.



Note: By default, the filename for each component log is unique. This allows you to isolate and research issues on a component-by-component basis by viewing a specific log file. Alternatively, you may wish to change the default output filename to be the same for all logging components so all activity is reported in a single log file. You will note that the default Audit and Alert filenames for each component (for those components that log audits or alerts) are the same so that all activity is logged in the same file across the platform for this reason.



Tip: If you use the default naming convention, and want to review an event that happened in the management portal, for instance, you would look in the Command_API_Log.txt and/or the Command_Portal_Log.txt.



Important: If you do choose to name the log files the same across the platform, it is recommended that you also set the **maxArchiveFiles** values the same in each Nlog config file. If there is a different value for **maxArchiveFiles** for files with the same filename/location, the smallest value will override all others.

The Nlog.config files are located in the installation directory for the product under a subdirectory for the given type of logging. By default, these locations are:

- **C:\Program Files\Keyfactor\Keyfactor Platform\Configuration\NLog_Configuration.config**
The Configuration file logs activity related to running the Keyfactor Command configuration wizard only.
- **C:\Program Files\Keyfactor\Keyfactor Platform\KeyfactorAPI\NLog_KeyfactorAPI.config**
The Keyfactor API file is the primary file for logging activity related to running the Keyfactor API. The Keyfactor API is used both for some basic underlying functionality of Keyfactor Command and for any API applications written against it by customers, so the log will show activity related to running the Management Portal as well as external API activity.
- **C:\Program Files\Keyfactor\Keyfactor Platform\Service\NLog_TimerService.config**
The Service file logs activity related to scheduled and automated events within Keyfactor Command.
- **C:\Program Files\Keyfactor\Keyfactor Platform\WebAgentServices\NLog_Orchestrators.config**
The Orchestrators, or Orchestrators API, file logs activity related to Keyfactor Orchestrators API.
- **C:\Program Files\Keyfactor\Keyfactor Platform\WebAPI\NLog_ClassicAPI.config**
The Classic API file logs activity involving the Classic API from Keyfactor Command.
- **C:\Program Files\Keyfactor\Keyfactor Platform\WebConsole\NLog_Portal.config**
The Portal file is for logging any activity to do with the Keyfactor Command web portal. Keyfactor is migrating the product to use mostly the Keyfactor API, so this file will have less activity going forward. See [C:\Program Files\Keyfactor\Keyfactor Platform\KeyfactorAPI\NLog_KeyfactorAPI.config](#) above

Once configured, output from the file locations defined will look similar to this:

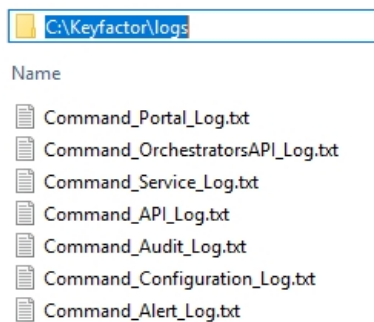


Figure 47: C:\Keyfactor\logs logs

NLog_Configuration.config

The Configuration file logs activity related to running the Keyfactor Command configuration wizard only. The fields you may wish to edit are:

- `fileName="C:\Keyfactor\logs\Command_Configuration_Log.txt"`

The path and file name of the active Keyfactor Command configuration wizard log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- `archiveFileName="C:\Keyfactor\logs\Command_Configuration_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command configuration wizard log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- `fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"`

The path and file name of the active Keyfactor Command log file for auditable configuration wizard events. These logs are generated separately from the configuration log events to allow for separate tracking of auditable events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="C:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command log files for auditable configuration wizard events.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the NLog_Configuration.config file on a server —once for the main logging section and once for the audit logging section. The default number of files to retain is 10 for the main log and 14 for the audit log. The audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs in the configuration file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination

- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files

```
<targets>
  <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command_Configuration_Log.txt" layout="${longdate} ${logger} [{level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Configuration_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
  <target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command_Audit_Log.txt" layout="${longdate} ${logger} [{level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="OutputDebugString" name="String" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Debugger" name="debugger" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Console" name="console" layout="${logger} {message}"/>
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor Command"
    eventId="${event-properties:item=eventID}" category="${event-properties:item=categoryID}" layout="${event-properties:item=message}" />
</targets>
<rules>
  <!-- Don't write events to the log file (log file should contain different, more verbose, logging) -->
  <logger name="CSS, CMS, Install, Configuration Wizard, Console, Wizard" minlevel="Info" writeTo="console" />
  <logger name="*-EVENT*" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
</nlog>
```

Figure 48: Nlog_Configuration.config

NLog_KeyfactorAPI.config

The KeyfactorAPI file is the primary file for logging activity related to running Keyfactor Command API. The fields you may wish to edit are:

- fileName="C:\Keyfactor\logs\Command_API_Log.txt"

The path and file name of the active Keyfactor Command primary log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- archiveFileName="c:\Keyfactor\logs\Command_API_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command primary log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- fileName="C:\Keyfactor\logs\Command_Alert_Log.txt"

The path and file name of the active Keyfactor Command primary log file for alerting events. This entry is only found on servers with the Keyfactor Command Service installed. Info level messages are written to this log whenever alerts (certificate expiration, pending certificate request, issued certificate, denied certificate request, or revocation monitoring) are run either as scheduled tasks or as tests. The log messages include the type of alert (e.g. expiration alert), the recipient of the alert (if an email was scheduled to be sent), and the alert subject line. You can change the level of logging in the log line that references writeTo="alertlogfile". These logs are generated separately from the primary log events to allow for separate tracking and log

shipping of alerting events. By default, the alert log filename/location for all components is the same to allow for a central source for tracking alert events across the platform.

- `archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command primary log files for alert events.

- `fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"`

The path and file name of the active Keyfactor Command primary log file for auditable events. These logs are generated separately from the primary log events to allow for separate tracking of auditable events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command primary log files for auditable events.

- `name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile"`

The level of log detail that should be generated for alert events and written to the alert logs.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the `NLog_KeyfactorAPI.config` file —once for the main logging section, once for the alert logging section, and once for the audit logging section. The default number of files to retain is 10 for the main log and the synchronization log but 14 for the alert log and audit log. The alert and audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs of the KeyfactorAPI file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files

```

<targets>
<target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command_API_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_API_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
<target xsi:type="File" name="alertlogfile" fileName="C:\Keyfactor\logs\Command_Alert_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
<target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command_Audit_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
<target xsi:type="OutputDebugString" name="String" layout="{$(longdate)} ${logger}::${message}"/>
<target xsi:type="Debugger" name="debugger" layout="{$(longdate)} ${logger}::${message}"/>
<target xsi:type="Console" name="console" layout="{$(logger)} ${message}"/>
<target xsi:type="EventLog" name="eventlog" source="Keyfactor Command"
eventId="{$(event-properties:item=eventID)}" category="{$(event-properties:item=categoryID)}" layout="{$(event-properties:item=message)}" />
</targets>
<rules>
<!-- Internal ASP.NET logging, off by default -->
<logger name="CSS.CMS.Web.API.SimpleTracer" minlevel="Off" writeTo="logfile" final="true" />
<logger name="*-EVENT*" minlevel="Info" writeTo="eventlog" final="true" />
<logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
<logger name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile" final="true"/>
<logger name="*" minlevel="Info" writeTo="logfile">
<filters>
<when condition="ends-with('{$(logger)}', 'WebSecurityContext') and level <= LogLevel.Warn" action="Ignore" />
<when condition="ends-with('{$(logger)}', 'AlertsController') and level <= LogLevel.Warn" action="Ignore" />
<when condition="ends-with('{$(logger)}', 'CertStoreController') and level <= LogLevel.Warn" action="Ignore" />
</filters>
</logger>
</rules>
</nlog>

```

Figure 49: Nlog_KeyfactorAPI.config

NLog_TimerService.config

The Timer Service file logs activity related to scheduled and automated events within Keyfactor Command and includes the CA sync logs. The fields you may wish to edit are:

- fileName="C:\Keyfactor\logs\Command_Service_Log.txt"

The path and file name of the active Keyfactor Command timer service log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- archiveFileName="C:\Keyfactor\logs\Command_Service_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command timer service log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- fileName="C:\Keyfactor\logs\Command_Alert_Log.txt"

The path and file name of the active Keyfactor Command timer service log file for alerting events. This entry is only found on servers with the Keyfactor Command Service installed. Info level messages are written to this log whenever alerts (certificate expiration, pending certificate request, issued certificate, denied certificate request, or revocation monitoring) are run either as scheduled tasks or as tests. The log messages include the type of alert (e.g. expiration alert), the recipient of the alert (if an email was scheduled to be sent), and the alert subject line. You can change the level of logging in the log line that references writeTo="alertlogfile". These logs are generated separately from the general timer service events to allow for separate tracking and log shipping of alerting events. By default, the alert log filename/location for all components is the same to allow for a central source for tracking alert events across the platform.

- archiveFileName="C:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command timer service log files for alert events.

- `fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"`

The path and file name of the active Keyfactor Command timer service log file for auditable events. These logs are generated separately from the general timer service events to allow for separate tracking of auditable events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="C:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command timer service log files for auditable events.

- `name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile"`

The level of log detail that should be generated for alert events and written to the alert logs.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the `NLog_Timer-Service.config` file on a server with the Keyfactor Command Service installed—once for the main logging section, once for the alert logging section, and once for the audit logging section. The default number of files to retain is 10 for the main log and the synchronization log but 14 for the alert log and audit log. The alert and audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs of the timer service file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files


```

<targets>
<target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command Service Log.txt" layout="{$(longdate) ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_Service_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
<target xsi:type="File" name="alertlogfile" fileName="C:\Keyfactor\logs\Command_Alert_Log.txt" layout="{$(longdate) ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
<target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command_Audit_Log.txt" layout="{$(longdate) ${logger} [${level}] - ${message}"
archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
<target xsi:type="OutputDebugString" name="String" layout="{$(longdate) ${logger}}:{$(message)"/>
<target xsi:type="Debugger" name="debugger" layout="{$(longdate) ${logger}}:{$(message)"/>
<target xsi:type="Console" name="console" layout="{$(logger) ${message)"/>
<target xsi:type="EventLog" name="eventLog" source="Keyfactor Command"
eventId="{$(event-properties:item=eventID)}" category="{$(event-properties:item=categoryID)}" layout="{$(event-properties:item=message)" />
</targets>
<rules>
<logger name="*-EVENT*" minlevel="Info" writeTo="eventLog" final="true" />
<logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
<logger name="Quartz*" minlevel="Warn" writeTo="logfile" />
<logger name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile" final="true"/>
<logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
</nlog>

```

Figure 50: Nlog_TimerService.config

NLog_Orchestrators.config

The Orchestrators, or OrchestratorsAPI, file logs activity related to orchestrators API. The fields you may wish to edit are:

- fileName="C:\Keyfactor\logs\Command_OrchestratorsAPI_Log.txt"

The path and file name of the active Keyfactor Command orchestrators log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- archiveFileName="C:\Keyfactor\logs\Command_OrchestratorsAPI_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command orchestrators log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- fileName="C:\Keyfactor\logs\Command_Alert_Log.txt"

The path and file name of the active Keyfactor Command orchestrators log file for alerting events. This entry is found on servers with the Keyfactor Command Service installed. Info level messages are written to this log whenever alerts (certificate expiration, pending certificate request, issued certificate, denied certificate request, or revocation monitoring) are run either as scheduled tasks or as tests. The log messages include the type of alert (e.g. expiration alert), the recipient of the alert (if an email was scheduled to be sent), and the alert subject line. You can change the level of logging in the log line that references writeTo="alertlogfile". These logs are generated separately from the general orchestrator events to allow for separate tracking and log shipping of alerting events. By default, the alert log filename/location for all components is the same to allow for a central source for tracking alert events across the platform.

- archiveFileName="C:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command orchestrators log files for alert events.

- fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"

The path and file name of the active Keyfactor Command orchestrators log file for auditable events. These logs are generated separately from the general orchestrator events to allow for separate tracking of auditable

events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="C:\\Keyfactor\\logs\\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command orchestrators log files for auditable events.

- `name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile"`

The level of log detail that should be generated for alert events and written to the alert logs.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the `NLog_Orchestrators.config` file on a server with the Keyfactor Command Service installed—once for the main logging section, once for the alert logging section, and once for the audit logging section. The default number of files to retain is 10 for the main log and the synchronization log but 14 for the alert log and audit log. The alert and audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.



Note: The default value for the `archiveAboveSize` setting was significantly larger in versions of Keyfactor Command prior to 7.5. In addition, the default `maxArchiveFiles` value was 2 for the main and CA synchronization logging sections. In environments where the logging level is consistently set at debug level or greater, this change may result in the generation of several log files per day.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs of the orchestrators file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files

```

<targets>
  <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command OrchestratorsAPI_Log.txt" layout="${longdate} ${logger} [{level}] - {message}"
    archiveFileName="c:\Keyfactor\logs\Command OrchestratorsAPI_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
  <target xsi:type="File" name="alertlogfile" fileName="C:\Keyfactor\logs\Command Alert_Log.txt" layout="${longdate} ${logger} [{level}] - {message}"
    archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command Audit_Log.txt" layout="${longdate} ${logger} [{level}] - {message}"
    archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="OutputDebugString" name="String" layout="${longdate} ${logger}:::{message}"/>
  <target xsi:type="Debugger" name="debugger" layout="${longdate} ${logger}:::{message}"/>
  <target xsi:type="Console" name="console" layout="${logger} {message}"/>
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor Command"
    eventId="{event-properties:item=eventID}" category="{event-properties:item=categoryID}" layout="{event-properties:item=message}" />
</targets>
<rules>
  <!-- Internal ASP.NET logging, off by default -->
  <logger name="CSS.CSS.Web.API.SimpleTracer" minlevel="Off" writeTo="logfile" final="true" />
  <logger name="*-EVENT*" minlevel="Info" writeTo="eventlog" final="true" />
  <logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
  <logger name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
</nlog>

```

Figure 51: Nlog_Orchestrators.config

NLog_Portal.config

The Portal log is for logging any activity to do with the Keyfactor Command web portal. The fields you may wish to edit are:

- fileName="C:\Keyfactor\logs\Command_Portal_Log.txt"

The path and file name of the active Keyfactor Command portal log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- archiveFileName="C:\Keyfactor\logs\Command_Portal_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command portal log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- fileName="C:\Keyfactor\logs\Command_Alert_Log.txt"

The path and file name of the active Keyfactor Command portal log file for alerting events. This entry is found on servers with the Keyfactor Command Service installed. Info level messages are written to this log whenever alerts (certificate expiration, pending certificate request, issued certificate, denied certificate request, or revocation monitoring) are run either as scheduled tasks or as tests. The log messages include the type of alert (e.g. expiration alert), the recipient of the alert (if an email was scheduled to be sent), and the alert subject line. You can change the level of logging in the log line that references writeTo="alertlogfile". These logs are generated separately from the general portal events to allow for separate tracking and log shipping of alerting events. By default, the alert log filename/location for all components is the same to allow for a central source for tracking alert events across the platform.

- archiveFileName="C:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command portal log files for alert events.

- fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"

The path and file name of the active Keyfactor Command portal log file for auditable events. These logs are generated separately from the general portal events to allow for separate tracking of auditable events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="C:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command portal log files for auditable events.

- `name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile"`

The level of log detail that should be generated for alert events and written to the alert logs.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the `NLog_Portal.-` config file on a server with the Keyfactor Command Service installed—once for the main logging section, once for the alert logging section, and once for the audit logging section. The default number of files to retain is 10 for the main log and the synchronization log but 14 for the alert log and audit log. The alert and audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs in the portal file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files

```

<targets>
  <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command Portal Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command Portal Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
  <target xsi:type="File" name="alertlogfile" fileName="C:\Keyfactor\logs\Command_Alert_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command_Audit_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="String" name="String" layout="{$(longdate)} ${logger} [${level}] - ${message}" />
  <target xsi:type="Debugger" name="debugger" layout="{$(longdate)} ${logger} [${level}] - ${message}" />
  <target xsi:type="Console" name="console" layout="{$(longdate)} ${logger} [${level}] - ${message}" />
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor Command"
    eventId="{$(event-properties:item=eventID)}" category="{$(event-properties:item=categoryID)}" layout="{$(event-properties:item=message)}" />
</targets>
<rules>
  <!-- Internal ASP.NET logging, off by default -->
  <logger name="CSS.CMS.Web.API.SimpleTracer" minlevel="Off" writeTo="logfile" final="true" />
  <logger name="*-EVENT*" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
  <logger name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
  <filters>
    <when condition="ends-with('${logger}', 'WebSecurityContext') and level <= LogLevel.Warn" action="Ignore" />
    <when condition="ends-with('${logger}', 'AlertsController') and level <= LogLevel.Warn" action="Ignore" />
    <when condition="ends-with('${logger}', 'CertStoreController') and level <= LogLevel.Warn" action="Ignore" />
  </filters>
</logger>
</rules>
</nlog>

```

Figure 52: Nlog_Portal.config

NLog_ClassicAPI.config

The ClassicAPI file logs activity related to invoking the ClassicAPI from Keyfactor Command. The fields you may wish to edit are:

- fileName="C:\Keyfactor\logs\Command_ClassicAPI_Log.txt"

The path and file name of the active Keyfactor Command classic API log file.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant both the service account under which the Keyfactor Command Service is running and the service account under which the IIS application pool for Keyfactor Command is running full control permissions on this directory. These roles may be served by the same service account.

- archiveFileName="C:\Keyfactor\logs\Command_ClassicAPI_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command classic API log files. Keyfactor Command rotates log files daily and names the previous files using this naming convention.

- fileName="C:\Keyfactor\logs\Command_Alert_Log.txt"

The path and file name of the active Keyfactor Command classic API log file for alerting events. This entry is found on servers with the Keyfactor Command Service installed. Info level messages are written to this log whenever alerts (certificate expiration, pending certificate request, issued certificate, denied certificate request, or revocation monitoring) are run either as scheduled tasks or as tests. The log messages include the type of alert (e.g. expiration alert), the recipient of the alert (if an email was scheduled to be sent), and the alert subject line. You can change the level of logging in the log line that references writeTo="alertlogfile". These logs are generated separately to allow for separate tracking and log shipping of alerting events. By default, the alert log filename/location for all components is the same to allow for a central source for tracking alert events.

- archiveFileName="C:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt"

The path and file name of previous days' Keyfactor Command classic API log files for alert events.

- `fileName="C:\Keyfactor\logs\Command_Audit_Log.txt"`

The path and file name of the active Keyfactor Command classic API log file for auditable events. These logs are generated separately from the general classic API events to allow for separate tracking auditable events. By default, the audit log filename/location for all components is the same to allow for a central source for tracking auditable events across the platform.

- `archiveFileName="C:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt"`

The path and file name of previous days' Keyfactor Command classic API log files for auditable events.

- `name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile"`

The level of log detail that should be generated for alert events and written to the alert logs.

- `maxArchiveFiles="10"`

The number of archive files to retain before deletion. This field is listed multiple times in the `Nlog_ClassicAPI.config` file —once for the main logging section, once for the alert logging section, and once for the audit logging section. The default number of files to retain is 10 for the main log and the synchronization log but 14 for the alert log and audit log. The alert and audit logs are small by default, and retaining a larger number of them shouldn't create a disk space issue.

- `archiveAboveSize="52428800"`

The maximum file size of each log file. Once a log file reaches this size (50 MB by default), it will be rotated to archive even if the end of the day has not been reached.

- `name="*" minlevel="Info" writeTo="logfile"`

The level of log detail that should be generated. This line applies to all the logs of the classicAPI file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF – No logging
- FATAL – Log severe errors that cause early termination
- ERROR – Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN – Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO – Log all of the above plus runtime events (startup/shutdown)
- DEBUG – Log all of the above plus detailed information on the flow through the system
- TRACE – Maximum log information—this option can generate VERY large log files


```

<targets>
  <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Command_ClassicAPI_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_ClassicAPI_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="10" archiveAboveSize="52428800"/>
  <target xsi:type="File" name="alertlogfile" fileName="C:\Keyfactor\logs\Command_Alert_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Alert_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="File" name="auditlogfile" fileName="C:\Keyfactor\logs\Command_Audit_Log.txt" layout="{$(longdate)} ${logger} [${level}] - ${message}"
    archiveFileName="c:\Keyfactor\logs\Command_Audit_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Date" archiveDateFormat="yyyyMMdd" maxArchiveFiles="14"/>
  <target xsi:type="OutputDebugString" name="String" layout="{$(longdate)} ${logger}::{$(message)}/>

  <target xsi:type="Debugger" name="debugger" layout="{$(longdate)} ${logger}::{$(message)}/>
  <target xsi:type="Console" name="console" layout="{$(longdate)} ${logger} [${level}] - ${message}"/>
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor Command"
    eventId="{event-properties:item=eventID}" category="{event-properties:item=categoryID}" layout="{$(event-properties:item=message)}/>
</targets>
<rules>
  <!-- Internal ASP.NET logging, off by default -->
  <logger name="CSS.CMS.Web.API.SimpleTracer" minlevel="Off" writeTo="logfile" final="true" />
  <logger name="*-EVENT*" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*-AUDIT*" minlevel="Info" writeTo="auditlogfile" final="true" />
  <logger name="EmailAlertLogger" minlevel="Info" writeTo="alertlogfile" final="true"/>
  <logger name="*" minlevel="Info" writeTo="logfile">
    <filters>
      <when condition="ends-with('{logger}', 'WebSecurityContext') and level <= LogLevel.Warn" action="Ignore" />
    </filters>
  </logger>
</rules>
</nlog>

```

Figure 53: Nlog_ClassicAPI.config

2.4.4.3 Configure CA Certificate Synchronization

The Keyfactor Command certificate management, notification and reporting features make use of a SQL database containing certificates from many locations, including:

- Certificates synchronized from domain-joined Microsoft CAs in your primary forest and forests with which the forest shares a trust
- Certificates synchronized from non-domain-joined EJBCA and Microsoft CAs
- Certificates synchronized from your domain-joined Microsoft CAs in non-trusted forests
- Certificates automatically imported based on SSL synchronization locations
- Certificates imported via Keyfactor CA Gateways from locations such as Entrust and Symantec clouds
- Manually imported certificates
- Certificates inventoried from certificate stores using Keyfactor Command Orchestrators

In order to get these certificates into the Keyfactor Command database so that you can begin using the management, notification and reporting features, you need to configure—at a minimum—CA synchronization. For more information:

- See *Certificate Authorities* in the *Keyfactor Command Reference Guide* for information on configuring CA synchronization for your Microsoft and EJBCA CAs.
- See *SSL Discovery* in the *Keyfactor Command Reference Guide* for information on configuring SSL discovery and monitoring.
- See the separate documentation for each type of CA gateway you have along with *Certificate Authorities* in the *Keyfactor Command Reference Guide* for information on configuring CA synchronization for your CA gateways.
- See *Add Certificate* in the *Keyfactor Command Reference Guide* for information on manually importing a certificate.
- See *Installing Orchestrators* in the *Keyfactor Orchestrators Installation and Configuration Guide* and *Orches-*

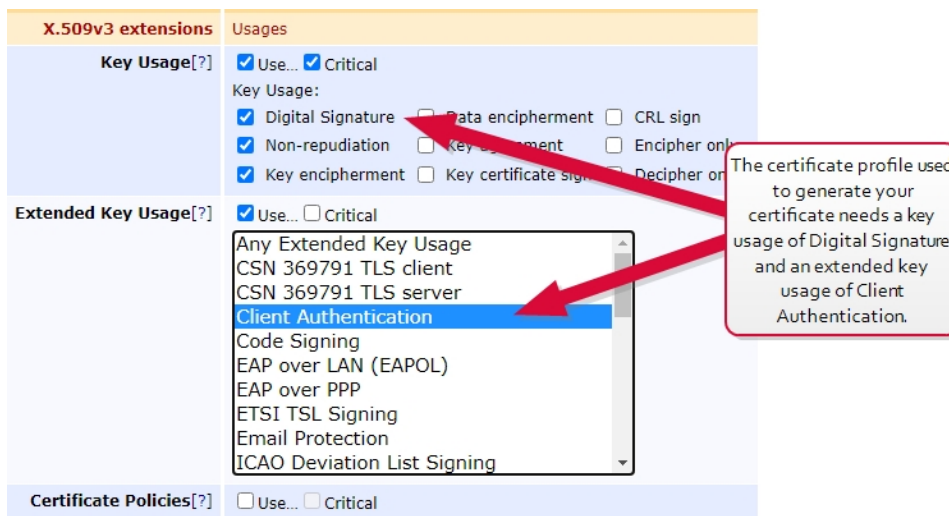
trators and Certificate Stores in the *Keyfactor Command Reference Guide* for information on inventorying certificates from certificate stores.

For information on using the Keyfactor Command Management Portal, see *Using the Management Portal* in the *Keyfactor Command Reference Guide*.

Acquire a Client Certificate for EJBCA CA Authentication

Keyfactor Command uses a client certificate to authenticate to the EJBCA certificate authority to support certificate synchronization, enrollment, and revocation. The certificate that Keyfactor Command uses for authentication needs:

- An extended key usage (EKU) of Client Authentication
- A key usage that includes Digital Signature



X.509v3 extensions	Usages
Key Usage[?]	<input checked="" type="checkbox"/> Use... <input checked="" type="checkbox"/> Critical Key Usage: <input checked="" type="checkbox"/> Digital Signature <input type="checkbox"/> Data encipherment <input type="checkbox"/> CRL sign <input checked="" type="checkbox"/> Non-repudiation <input type="checkbox"/> Key encipherment <input type="checkbox"/> Encipher only <input checked="" type="checkbox"/> Key encipherment <input type="checkbox"/> Key certificate sign <input type="checkbox"/> Decipher only
Extended Key Usage[?]	<input checked="" type="checkbox"/> Use... <input type="checkbox"/> Critical Any Extended Key Usage CSN 369791 TLS client CSN 369791 TLS server Client Authentication Code Signing EAP over LAN (EAPOL) EAP over PPP ETSI TSL Signing Email Protection ICAO Deviation List Signing
Certificate Policies[?]	<input type="checkbox"/> Use... <input type="checkbox"/> Critical

Figure 54: Certificate Profile for EJBCA Client Certificate

The certificate needs to be available as a PKCS#12 (*.pfx) file in order to import it into Keyfactor Command.



Confirm request

Issuer Distinguished Name: CN=ManagementCA,O=Key Example,C=US

Subject Distinguished Name: O=Key Example,L=Chicago,ST=Illinois,C=US

Public Key Specification: RSA_2048

Validity: 2y

[Show details](#)

Download JKS

Download PKCS#12

Download BCFKS

Download PEM

Figure 55: Certificate Download for EJBCA Client Certificate

The certificate needs to be granted appropriate access to the EJBCA CA to allow Keyfactor Command interactions with the CA to take place (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs below](#)).

Grant the Keyfactor Command Users and Service Account(s) Permissions on the CAs

In order for Keyfactor Command to be able to synchronize certificates from the CAs to the Keyfactor Command database, the service account under which Keyfactor Command makes a connection to the CA must have permissions to *read* the CA databases. For full Keyfactor Command functionality, additional permissions are needed. The permissions needed vary depending on the type of CA and the type of authorization you intend to configure to allow Keyfactor Command and users in Keyfactor Command to interact with the CA.

Microsoft CAs

When you configure Keyfactor Command access to a Microsoft CA, you have the option to enable the *Use Explicit Credentials* option. When this option is enabled, you enter a set of credentials that will be used specifically to access that Microsoft CA, and all management and enrollment tasks for that CA are done in the context of that service account. If you do not enable the *Use Explicit Credentials* option, management tasks (e.g. revocation, certificate synchronization) and enrollments are done in the context of the service account(s) you configure for the Keyfactor Command Service and the application pool for Keyfactor Command (which are the same service account in many implementations) and individual users. The exact combination of what happens in the context of who depends on the configuration of the delegation options (*Delegate management Operations* and *Delegate Enrollment*) on the CA when the *Use Explicit Credentials* option is not enabled. Delegation is supported for both Basic authentication and Kerberos authentication (see [Configure Kerberos Constrained Delegation \(Optional\) on page 75](#)). Use of explicit credentials is mutually exclusive of delegation.

The users and service account(s) you will be using to connect to your Microsoft CA(s) from Keyfactor Command need some set of the following permissions on the CA, based on the configuration of authorization for the CA:

- Read
To support CA synchronization
- Issue and Manage Certificates
To support certificate revocation and key recovery
- Manage CA
To support CRL publication following revocation
- Request Certificates
To support certificate enrollment through Keyfactor Command

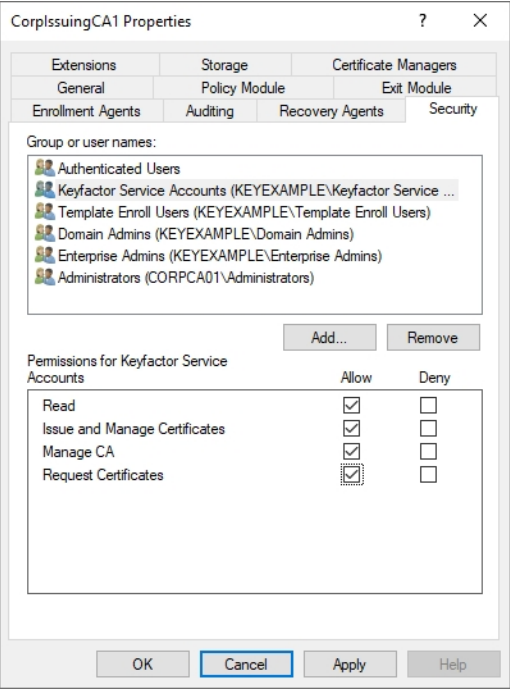


Figure 56: Microsoft CA Permissions

Table 9: [Microsoft CA Permission Matrix](#) provides information on what permissions are required based on possible authorization configurations.

Table 9: Microsoft CA Permission Matrix

	✔ Use Explicit Credentials	✘ Use Explicit Credentials	✘ Use Explicit Credentials	✘ Use Explicit Credentials	✘ Use Explicit Credentials
		✔ Delegate Management	✘ Delegate Management	✔ Delegate Management	✘ Delegate Management
		✔ Delegate Enrollment	✔ Delegate Enrollment	✘ Delegate Enrollment	✘ Delegate Enrollment
Explicit CA-Specific User	Read Issue & Manage Certificates Manage CA Request Certificates	n/a	n/a	n/a	n/a
Keyfactor	None	Read	Read	Read	Read

	<div> <div>✔ Use Explicit Credentials</div> <div>✘ Use Explicit Credentials</div> <div>✔ Delegate Management</div> <div>✘ Delegate Management</div> <div>✔ Delegate Enrollment</div> <div>✔ Delegate Enrollment</div> <div>✘ Delegate Enrollment</div> <div>✘ Delegate Enrollment</div> </div>				
Command Service Account		Request Certificates ¹	Request Certificates ²	Request Certificates ³	Request Certificates ⁴
Keyfactor Command Application Pool Account	None	Read Issue & Manage Certificates Manage CA Request Certificates ⁵	Read Issue & Manage Certificates Manage CA Request Certificates ⁶	Read Manage CA Request Certificates	Read Issue & Manage Certificates Manage CA Request Certificates
Individual Users	None	Read Issue & Manage Certificates Request Certificates	Read Request Certificates	Read Issue & Manage Certificates	None

In the management console for each CA that Keyfactor Command will be interacting with, open the properties for the CA and grant the users and service account(s) for Keyfactor Command the appropriate permissions for your environment before continuing.



Tip: In order to support PFX and CSR enrollment through the Management Portal, the user initiating the enrollment in the Management Portal must be granted "Request Certificates" permission in the CA if enroll-

¹To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

²To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

³To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

⁴To support certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

⁵To support tests of certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

⁶To support tests of certificate enrollment through Keyfactor Command that occurs in an automated fashion (e.g. certificate renewals initiated by the expiration alert renewal handler).

ment delegation is enabled. In many environments, all Authenticated Users are granted this permission, allowing the Management Portal users to inherit the permission.

EJBCA CAs

Management (e.g. revocation, certificate synchronization) and enrollment requests to an EJBCA CA are made in the context of the end entity associated with the client certificate selected in each CA configuration in the Keyfactor Command Management Portal to provide authentication to the EJBCA CA (see [Acquire a Client Certificate for EJBCA CA Authentication on page 95](#)). The access rule created or used for this needs to grant sufficient permissions to allow the end entity to synchronize certificates. For full functionality, it needs the following permissions:

- `/administrator/`
To support Keyfactor Command making API requests to the EJBCA CA
- `/ca/[your_ca_name]/`
To support Keyfactor Command access to your CA
- `/ca_functionality/create_certificate/`
To support certificate enrollment through Keyfactor Command
- `/ca_functionality/create_crl/`
To support CRL publication following revocation
- `/ca_functionality/view_ca/`
To support retrieval of CA information
- `/ca_functionality/view_certificate/`
To support CA synchronization
- `/ca_functionality/view_certificate_profiles/`
To support template import
- `/endentityprofilesrules/[your_end_entity_profile_name]/create_end_entity/`
To support creation of end entities (a new end entity is created for each Keyfactor Command certificate enrollment unless the *Enforce Unique DN* option is enabled and the new certificate's DN matches that of an existing certificate)
- `/endentityprofilesrules/[your_end_entity_profile_name]/edit_end_entity/`
To support certificate enrollment with the *Enforce Unique DN* option through Keyfactor Command and certificate renewal through Keyfactor Command
- `/endentityprofilesrules/[your_end_entity_profile_name]/revoke_end_entity/`
To support certificate revocation through Keyfactor Command
- `/endentityprofilesrules/[your_end_entity_profile_name]/view_end_entity/`
To support certificate enrollment through Keyfactor Command
- `/ra_functionality/create_end_entity`

To support creation of end entities (a new end entity is created for each Keyfactor Command certificate enrollment unless the *Enforce Unique DN* option is enabled and the new certificate's DN matches that of an existing certificate)

- /ra_functionality/edit_end_entity

To support certificate enrollment with the *Enforce Unique DN* option through Keyfactor Command and certificate renewal through Keyfactor Command

- /ra_functionality/revoke_end_entity

To support certificate revocation through Keyfactor Command

- /ra_functionality/view_end_entity

To support certificate enrollment through Keyfactor Command

- /system_functionality/view_administrator_privileges

To support overall functionality

Edit Access Rules[?]

Role : Keyfactor Role

Where "ManagementCA" is the name of your CA.

Resource	Rule
/administrator/	Allow
/ca/ManagementCA/	Allow
/ca_functionality/create_certificate/	Allow
/ca_functionality/create_crl/	Allow
/ca_functionality/view_ca/	Allow
/ca_functionality/view_certificate/	Allow
/ca_functionality/view_certificate_profiles/	Allow
/endentityprofilesrules,Sample:create_end_entity/	Allow
/endentityprofilesrules,Sample:edit_end_entity/	Allow
/endentityprofilesrules,Sample:revoke_end_entity/	Allow
/endentityprofilesrules,Sample:view_end_entity/	Allow
/ra_functionality/create_end_entity/	Allow
/ra_functionality/edit_end_entity/	Allow
/ra_functionality/revoke_end_entity/	Allow
/ra_functionality/view_end_entity/	Allow
/system_functionality/view_administrator_privileges/	Allow

Where "Sample" is the name of your end entity profile or profiles.

Figure 57: EJBCA Access Permissions

You may either create a new access rule that limits access to just these required permissions, or use an existing access rule. In either case, you need to add the certificate used to authenticate Keyfactor Command to the EJBCA CA as a member of that access rule.

Members

Role : Keyfactor Role

[Back to Roles Management](#)
[Edit Access Rules](#)

Match with	CA	Match	Operator	Match Value	Description	Action
X509: Certificate serial number (Recommended)	ManagementCA	-	Equal, case insens.	569B60F0BF65DF9EB473A4C8D3FF6F844D478C9F		Add
X509: Certificate serial number (Recommended)	ManagementCA	-	Equal, case insens.	5948057A4A5E6DAFE9157CF81C328A1FB67F1A54		Delete

Figure 58: Add Client Certificate as Member of EJBCA Access Rule

Enable and Start the Keyfactor Command Service

The Keyfactor Command Service runs on the Keyfactor Command server hosting the Services role and controls database synchronization, among other jobs. During the Keyfactor Command configuration process you configured the service account under which the Keyfactor Command Service will run and may have configured the service to start automatically at server boot time (see [Configure: Service on page 55](#)). To begin the CA synchronization, you just need to start the service (if it hasn't started automatically):

1. On the Keyfactor Command server hosting the Services role, open the Services MMC.
2. In the Services MMC confirm that the Keyfactor Command Service is set to a Startup Type of Automatic (if desired). If the service is not running, click the green arrow to start it.

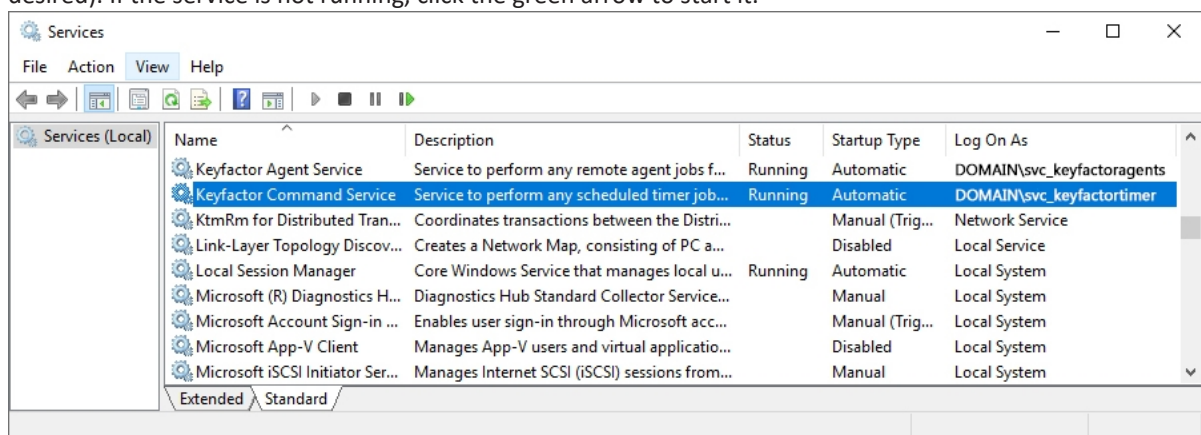
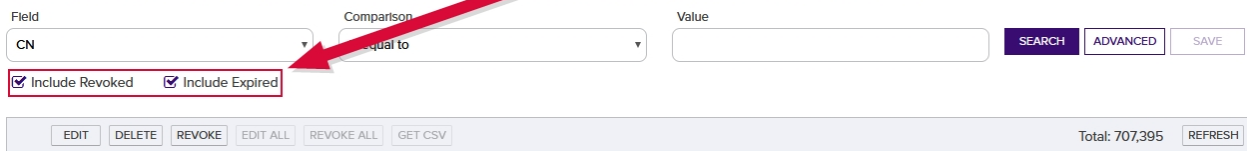


Figure 59: Keyfactor Command Service

The CA(s) will begin to synchronize when the first scheduled scan time is reached. Scans scheduled at intervals match to clock times, so a scan set at an interval of 15 minutes will run at 6:00, 6:15, 6:30, 6:45, etc. You can check the Keyfactor Command timer service log file on the Keyfactor Command Services server to confirm that synchronization is operating as expected. You can also use the Certificate Search feature in the Keyfactor Command Management Portal to confirm the certificates are appearing in the Keyfactor Command database. That database synchronization begins with the oldest certificates in the CA database, which may be expired or revoked. Be sure to toggle the Include Revoked and Include Expired options, see [Include Expired and Revoked Certificates in Certificate Search on the next page](#), when checking to see if synchronization is working. See [Certificate Search Page](#) in the *Keyfactor Command Reference Guide* for information on using the search.

Certificate Search⁹

Clicking "Advanced" allows you to build a query based on multiple criteria using AND/OR logic. Select a field and comparison operator in the drop-downs and then enter a comparison value. Click "Insert" to add the search criteria to the query field below the selection fields. Each time you click the "Insert" button, an AND is added between the previous search criteria and the newly added one. You can change the AND to an OR if desired. You can use parentheses around portions of the query along with AND/OR to change the query meaning.



Field: CN Comparison: Equal to Value:

☒ Include Revoked ☒ Include Expired

EDIT DELETE REVOKE EDIT ALL REVOKE ALL GET CSV

Total: 707,395 REFRESH

Figure 60: Include Expired and Revoked Certificates in Certificate Search

2.4.4.4 Create or Identify Certificate Templates for Enrollment

This step only needs to be completed if your Keyfactor Command license includes certificate enrollment and you plan to use this feature.



Note: Keyfactor Command and this documentation use the term *template* generically to refer to Microsoft certificate templates and EJBCA certificate templates. EJBCA templates are built from the EJBCA end entity profile and certificate profile and named using a naming scheme of <end entity profile name>_<certificate profile name> and <end entity profile name> (<certificate profile name>) for the template name and template display name.

The enrollment function in the Keyfactor Command Management Portal is generally used by administrators to request certificates for use on servers, network devices, and similar equipment. There's a good chance that certificate templates for these purposes already exist in your environment. To prepare for the Keyfactor Command installation, you need to gather a list of the CAs that will be used to issue certificates through the Keyfactor Command Management Portal and a list of the *template names* (vs template display names) of the templates that will be used for this (Microsoft CAs) or certificate profiles and end entity profiles (EJBCA CAs). If any new templates or profiles need to be created for this purpose, they should be created before completing the Keyfactor Command post-installation steps.

For Microsoft CAs, the security settings on your existing templates may need to be modified to allow users to enroll for certificates using them through the Keyfactor Command Management Portal, depending on how the templates have been used previously. For CAs in the local forest (the forest in which Keyfactor Command is installed) and forests in a two-way trust with the local forest, enrollment through the Keyfactor Command Management Portal is often done in the context of the user logged into the portal. This differs from enrolling for a certificate through the Microsoft certificates MMC, where requests for computer certificates (such as web server certificates) are done in the context of the machine account from which the certificate is requested, not the user account, and thus the machine account needs permissions, not the user. When using the Keyfactor Command Management Portal, each of the users who will use one of the enrollment functions needs **Read** and **Enroll** permissions on the templates they will be using through the portal (see [Grant the Keyfactor Command Users and Service Account\(s\) Permissions on the CAs on page 96](#) for more details).



Tip: Enrollment through the Keyfactor Command Management Portal against remote Microsoft CAs (CAs in a forest with either no trust with the forest in which Keyfactor Command is installed or a one-way trust) are done in the context of the service account configured on the Management Portal CA record for *Explicit Credentials* (see [Create Active Directory Service Accounts for Keyfactor Command on page 19](#)).

The Keyfactor Command Management Portal offers the option of using a different set of CAs and templates for each of the two different enrollment methods—PFX and CSR. As you collect your list of CAs and templates, you will need to decide whether you want to use the same CAs and templates for both types of enrollment or whether each type of enrollment will have a unique list of CAs and templates.

The list of templates used for enrollment in the Keyfactor Command Management Portal is configured through the Keyfactor Command Management Portal template management. Although in previous releases of Keyfactor Command, the templates and CAs for enrollment were configured during installation, this is now done as a post-install step in the Management Portal. See *Certificate Authorities* and *Configuring Template Options* in the *Keyfactor Command Reference Guide*.

2.4.4.5 Configure Renewal Handler Permission


The expiration renewal event handler allows you to execute a certificate renewal automatically for each expiring certificate that is found in a supported certificate store for each expiration alert when the alert task is triggered by the execution of the expiration alerts. In order for the renewal handler to execute successfully, the Active Directory service account under which the Keyfactor Command Service runs must have select permissions in the Keyfactor Command Management Portal. In addition, if you wish to test the execution of expiration alerts with renewal handlers and your IIS application pool runs in the context of a different Active Directory service account than the Keyfactor Command Service, the Active Directory service account for the IIS application pool must also be granted these permissions.



Note: If your Microsoft CA has been configured with the *Use Explicit Credentials* option, the permissions described here need to be granted to the user specified by the *Use Explicit Credentials* option, not either of the above-referenced service accounts.
If you're using an EJBCA CA, no further permissions need to be granted and this step may be skipped.

If you don't plan to use the expiration renewal handler, you can skip this step.

To configure permissions for the service account(s) to support use of the expiration renewal handler:

1. In the Keyfactor Command Management Portal, browse to *System Settings Icon*  > *Security Roles & Identities*.
2. On the Security Roles and Identities page on the Security Roles tab, click **Add** to create a new role to be used just to grant permissions to the service account(s) to support use of the expiration renewal handler.
3. On the Details tab, give it an appropriate name and description to reflect this usage.
4. On the Global Permissions tab, click the **Enroll PFX** toggle for *Certificate Enrollment* to enable it, click the **Read** and **Schedule** toggles for *Certificate Store Management* to enable them, and click the **Read** toggle for *Management Portal* to disable it, if enabled.

5. Click **Save** to save the role.

Add Security Role [X]

Details **Global Permissions** Collection Permissions Container Permissions Identities/Access

Select a Profile [v] [APPLY] [RESET] [CLEAR]

Certificate Store Management [^]

Certificates

Dashboard

Event Handler Registration

Mac Auto-Enroll Management

Management Portal

Monitoring

PKI Management

Privileged Access Management [v]

Read [on]

Schedule [on]

Modify [off]

[SAVE] [CANCEL]

Figure 61: Configure Expiration Renewal Handler

6. On the Security Roles and Identities page on the Security Identities tab, click **Add** to add a new security identity.
7. In the Security Identities dialog, enter the Active Directory user name of the service account under which the Keyfactor Command Service runs using DOMAIN\username format and click **Save**. If the account resolves correctly, the new identity will be saved and the dialog will close.

Add Security Identity [X]

Account Name

KEYEXAMPLE\svc_kycommand

[SAVE] [CANCEL]

Figure 62: Configure Expiration Renewal Handler: Add New Identity

8. If your IIS application pool runs as a different Active Directory service account from that used for the Keyfactor Command Service, repeat steps four and five for the IIS application pool service account.

9. In the Security Identity Editor section of the page, double-click the Keyfactor Command Service identity in the identity grid, right-click the row in the identity grid and choose **Edit Roles** from the right-click menu, or highlight the Keyfactor Command Service identity in the identity grid and click **Edit Roles** at the top of the identity grid.
10. In the Roles dialog, select the role you created for the expiration renewal handler in the Available Roles list and use the right arrow to move the role to the Current Roles list. Click **Save** to assign the role to the identity.

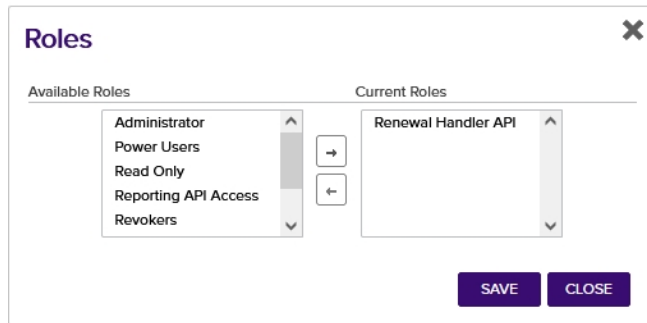


Figure 63: Configure Expiration Renewal Handler: Assign Role to Identity

11. If your IIS application pool runs as a different Active Directory service account from that used for the Keyfactor Command Service, repeat steps seven and eight for the IIS application pool service account.

2.4.4.6 Create a Certificate Template for Mac Auto-Enrollment

This step only needs to be completed if your Keyfactor Command license includes Mac auto-enrollment and you plan to use this feature.

To create the certificate template that will be used for Mac auto-enrollment:

1. On the CA that will issue the Mac auto-enrollment certificates, open the Certification Authority management tool.
2. In the Certification Authority management tool, drill down to locate the Certificate Templates folder. Right-click the **Certificate Templates** folder and choose **Manage**. This will open the Certificate Templates Console.
3. In the Certificate Templates Console, right-click the User template and choose **Duplicate Template**.
4. If prompted with a Duplicate Template dialog (some versions of Windows), choose Windows Server 2003 Enterprise and click **OK**.
5. General Tab: In the Properties of New Template dialog on the General tab, enter **Mac Auto-Enrollment** (or an alternate name of your choosing) in the **Template display name** field. The **Template name** will be auto-populated based on the text you enter in the **Template display name**. Select a **Validity period** for the certificate that's appropriate for your environment.
6. Extensions Tab: If you plan to use the certificates to authenticate to enterprise systems, you will need to ensure that **Client Authentication** is set as the only application policy in the certificate. To do this, in the **Extensions included in this template** section of the Extensions tab, highlight **Application Policies** and click the **Edit...**

button. In the Edit Application Policies Extensions dialog, remove the **Encrypting File System** and **Secure Email** policies and click **OK**.

7. **Security Tab:** In the Properties of New Template dialog on the Security tab, add the Active Directory group of users who will be allowed to auto-enroll from Macs and grant this group **Read**, **Enroll**, and **Autoenroll** permissions on the template.
8. Click **OK** to save the new template.
9. Back in the Certification Authority management tool, right-click the **Certificate Templates** folder and choose **New->Certificate Template to Issue**. Select the **Mac Auto-Enrollment** template from the list presented and click **OK**.

2.5 Keyfactor CA Policy Module

The Keyfactor CA Policy Module includes four certificate authority policy handlers that can be used to alter or restrict the functionality of a Microsoft certificate authority. The policy handlers are installed on the Microsoft CA and enabled through the Microsoft CA properties page. The available policy handlers are:

RFC 2818 Policy Handler

Automate inclusion of a DNS SAN matching the CN of the requested certificate in certificate enrollments for a defined set of CA templates.

SAN Attribute Policy Handler

Allow the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag.

vSCEP™ Policy Handler

Allow secure control of on-device key generation during certificate enrollment for iOS and Mac devices.

Whitelist Policy Handler

Enforce that certificate requests for a given template or templates can only be initiated from a given computer or set of computers.

2.5.1 System Requirements

The Keyfactor CA Policy Module is supported on Microsoft certificate authorities running on Windows Server 2016 or higher. It interoperates with Keyfactor Command versions 9.0 or greater.

The policy module requires the Microsoft .NET **Desktop** Runtime version 6.0 (x64). Version 6.0 is available for download from Microsoft:

<https://dotnet.microsoft.com/download/dotnet/6.0/runtime>

At the above link, this would be the **Download x64** option under the "Run desktop apps" heading.

You can use the following PowerShell command to check the .NET core version(s) installed on a server (if any):

```
dotnet --list-runtimes
```

Output from this command will look something like this if you have the correct 6.0 x64 version of the .NET Desktop Runtime installed (notice the paths are in C:\Program Files, not C:\Program Files (x86), indicating this is the x64 version):

```
Microsoft.NETCore.App 6.0.11 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
Microsoft.WindowsDesktop.App 6.0.11 [C:\Program
Files\dotnet\shared\Microsoft.WindowsDesktop.App]
```

2.5.2 Preparing for the Keyfactor CA Policy Module

The preparation steps necessary for the Keyfactor CA Policy Module vary depending on the policy handler(s) you intend to use. The policy handlers have the following preparation requirements:

RFC 2818 Policy Handler

During the configuration of the RFC 2818 Policy Handler, you will need to define the list of Microsoft certificate templates that will automatically be assigned a DNS SAN matching the certificate's CN when a certificate enrollment request reaches the CA. These templates are configured by selecting them from a list. You will need to have this list of templates ready.

SAN Attribute Policy Handler

During the configuration of the SAN Attribute Policy Handler, you will need to define the list of Microsoft certificate templates that will allow certificate enrollment requests via CSR to submit SANs outside of the CSR for inclusion in the final certificate, replacing any SANs originally in the CSR. These templates are configured by selecting them from a list. You will need to have this list of templates ready.

vSCEP™ Policy Handler

During the configuration of the vSCEP™ Policy Handler, you will need to enter the URL to the vSCEP service on your Keyfactor Command server and the username and password of a service account that the policy handler will use to make requests to the vSCEP API on the Keyfactor Command server. The user you enter here needs to be a member of the group you configure for *Allowed Users/Groups* on the vSCEP Service tab in the Keyfactor Command configuration wizard (see [vSCEP Services Tab on page 60](#)). The service account needs to be created in Active Directory prior to installation of the Keyfactor CA Policy Module software, and the person installing the Keyfactor CA Policy Module software needs to know the service account domain, username and password.

Whitelist Policy Handler

During the configuration of the SAN Attribute Policy Handler, you will need to define the list of Microsoft certificate templates that will be gated by the handler and the list of machines that will be allowed to use these templates. Any templates you include will be available for enrollment only from machines you include in the

allowed list. The purpose of this handler is to force all enrollments to be made from the Keyfactor Command server(s), so the list of machines should include your Keyfactor Command server(s). The templates for this policy handler are configured by typing in their certificate *template name* (short name), so you will need an exact list of the template names.

In addition, you will need to have your Keyfactor product license available for upload into the policy module once installed to activate it.

2.5.3 Installing the Keyfactor CA Policy Module Handlers

These steps only need to be completed if your Keyfactor Command license includes the Keyfactor CA Policy Module and you plan to use this feature and one or more of its policy handlers. Review the policy handlers to determine if one or more of them meets a need in your environment.



Important: For a CA Clustered solution, if the Keyfactor CA Policy Module is installed on a node then configured, then failed over to another node, this will corrupt the check point key. The module must be installed on BOTH nodes, configured on one node, then failed over to the other node.

The available policy handlers are:

RFC 2818 Policy Handler

Automate inclusion of a DNS SAN matching the CN of the requested certificate in certificate enrollments for a defined set of CA templates.

SAN Attribute Policy Handler

Allow the addition of SANs not included in the CSR when making a CSR enrollment request. The added SANs will overwrite any existing SANs in the CSR. This functionality is the same as that seen with the Microsoft default policy module for the CA as a whole when the CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag is set except the SAN Attribute Policy Handler provides the ability to control SAN addition on a template-by-template basis without the need to enable the Microsoft CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag.



Important: By default, Microsoft CAs do not support the addition of SANs not included in the CSR when making a request using a CSR enrollment method. To enable your CA to support requesting certificates with additional SANs, you must either install and configure the Keyfactor Command SAN Attribute Policy Handler on the CA(s) or enable the Microsoft CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag. There are security risks inherent in enabling either of these options on your CA. Keyfactor recommends that you do not enable these options unless it is an absolute requirement. With the SAN Attribute Policy Handler, you can limit the risk by limiting the exposure to just selected templates. Keyfactor further recommends that you:

- Use the SAN Attribute Policy Handler only with templates that require CA manager approval so that a manager will be required to review the request and the added SANs before the certificate is issued.
- Use the SAN Attribute Policy Handler in conjunction with the Whitelist Policy Handler to limit requests for the selected templates to being initiated only by the Keyfactor Command server(s).



- Configure server level monitoring with a product such as Microsoft's System Center Operations Manager (SCOM) to provide alerts for any changes relating to the CA(s) configured with the SAN Attribute Policy Handler so that, for example, changes to the templates configured to support SAN addition do not go unnoticed.

vSCEP™ Policy Handler

Allow secure control of on-device key generation during certificate enrollment for iOS and Mac devices.

Whitelist Policy Handler

Enforce that certificate requests for a given template or templates can only be initiated from a given computer or set of computers.



Note: The following Windows update affects how certificate requests are built when sent to a Microsoft CA and may cause enrollments done outside Keyfactor Command against a Microsoft CA configured with the Whitelist Policy Handler to fail.

<https://support.microsoft.com/en-us/topic/kb5014754-certificate-based-authentication-changes-on-windows-domain-controllers-ad2c23b0-15d8-4340-a468-4d4f3b188f16>

The processing order of the handlers currently available in the Keyfactor CA Policy Module, when used together on the same machine, is significant for some handlers and not others. Specifically, the processing order is not significant for the vSCEP™ Policy Handler and Machine Whitelist Policy handler. These handlers may be placed anywhere within the list of handlers. However, the processing order does matter for the SAN Attribute Policy Handler and the RFC 2818 Policy Handler. When these two handlers are used together, the SAN Attribute Policy Handler must be placed on the list above the RFC 2818 Policy Handler to allow the SAN Attribute Policy Handler to be processed before the RFC 2818 Policy Handler. This is because the SAN Attribute Policy Handler removes any existing SANs on the enrollment request and replaces them with those specified in the request outside of the CSR—such as those entered in the optional SAN section on the CSR page of the Keyfactor Command Management Portal. This includes any SANs added by the RFC 2818 Policy Handler.

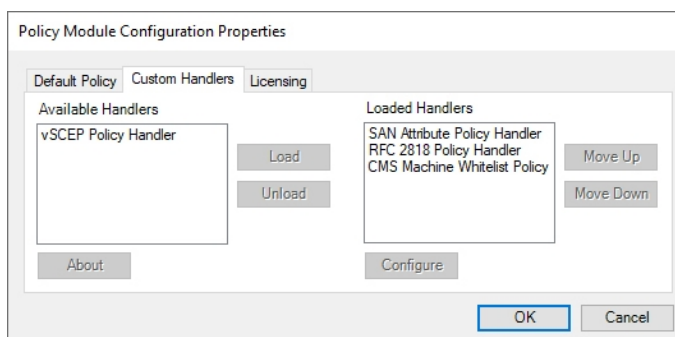


Figure 64: Keyfactor CA Policy Module Policy Module Handler Ordering

When the Keyfactor CA Policy Module is used, the policy module listed on the Default Policy tab of the Policy Module Configuration Properties dialog is run first when a request reaches the CA. This default policy might be the standard Windows default, as shown [Figure 65: Default Policy Module](#), or it might be another non-built-in policy

module, such as the Microsoft FIM CM Policy Module. After the default policy module runs, the Loaded Handlers on the Custom Handlers tab of the Policy Module Configuration Properties dialog are run in the order listed (top to bottom). After all the handlers have been run, the result (approved, denied, or marked as pending) is returned to the CA for processing.

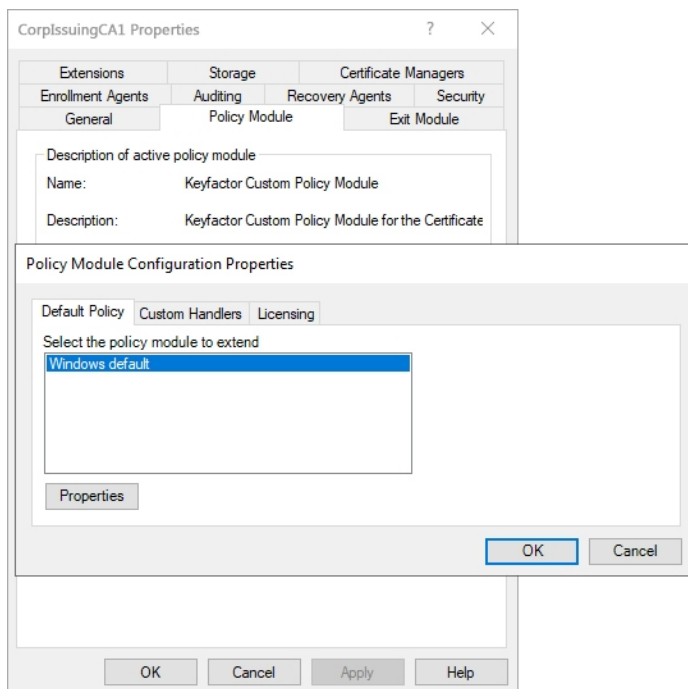


Figure 65: Default Policy Module



Tip: Once the installation is complete, the configuration options for the policy handlers can be found in the registry on the CA in the following paths (where CA_LOGICAL_NAME is the logical name of the local CA):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration\CA_
LOGICAL_NAME\PolicyModules\CMS_Custom.Policy\PolicyHandlers\RFC2818.PolicyHandler
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration\CA_
LOGICAL_NAME\PolicyModules\CMS_
Custom.Policy\PolicyHandlers\SANAttribute.PolicyHandler
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration\CA_
LOGICAL_NAME\PolicyModules\CMS_Custom.Policy\PolicyHandlers\vSCEP.PolicyHandler
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration\CA_
LOGICAL_NAME\PolicyModules\CMS_
Custom.Policy\PolicyHandlers\CMSWhitelist.PolicyHandler
```



Warning: These registry keys should not be modified without advice from Keyfactor support.

2.5.3.1 Install the Keyfactor RFC 2818 Policy Handler

To begin the RFC 2818 Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

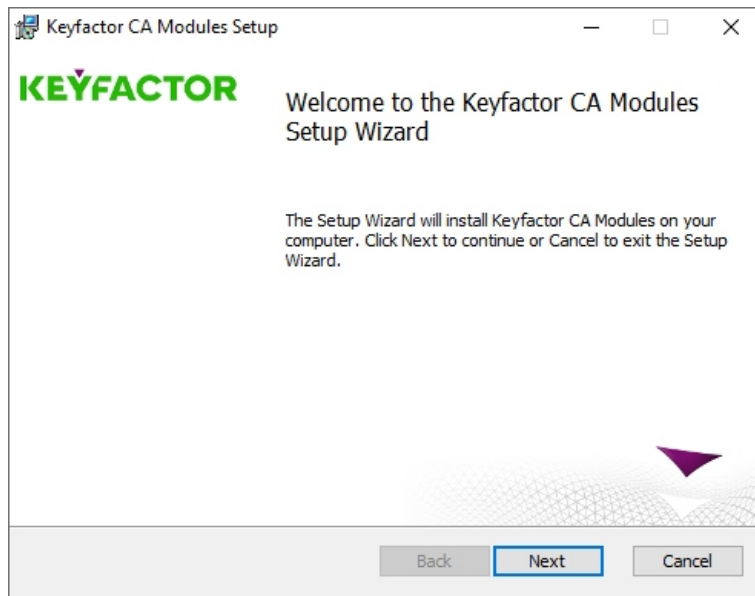


Figure 66: Install RFC 2818 Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the RFC 2818 Policy Handler, deselect all the components except the RFC 2818 Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

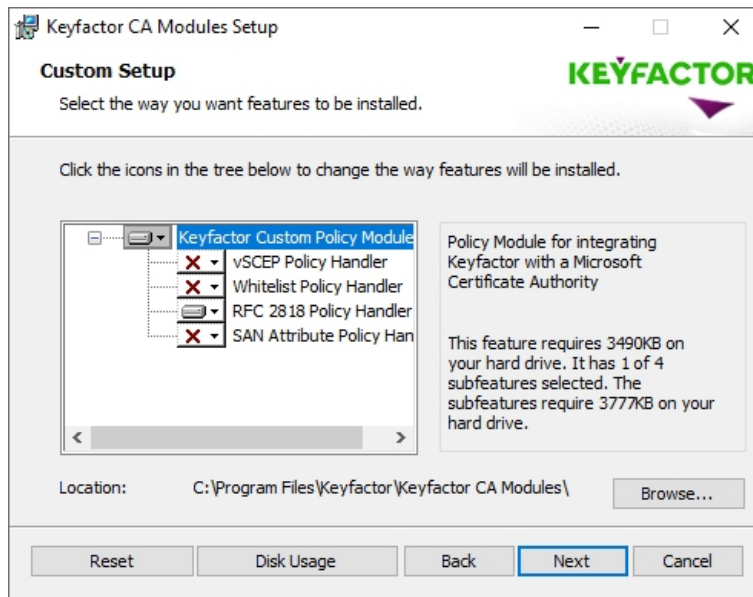


Figure 67: Install RFC 2818 Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the "Launch the CA MMC snap-in now" box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

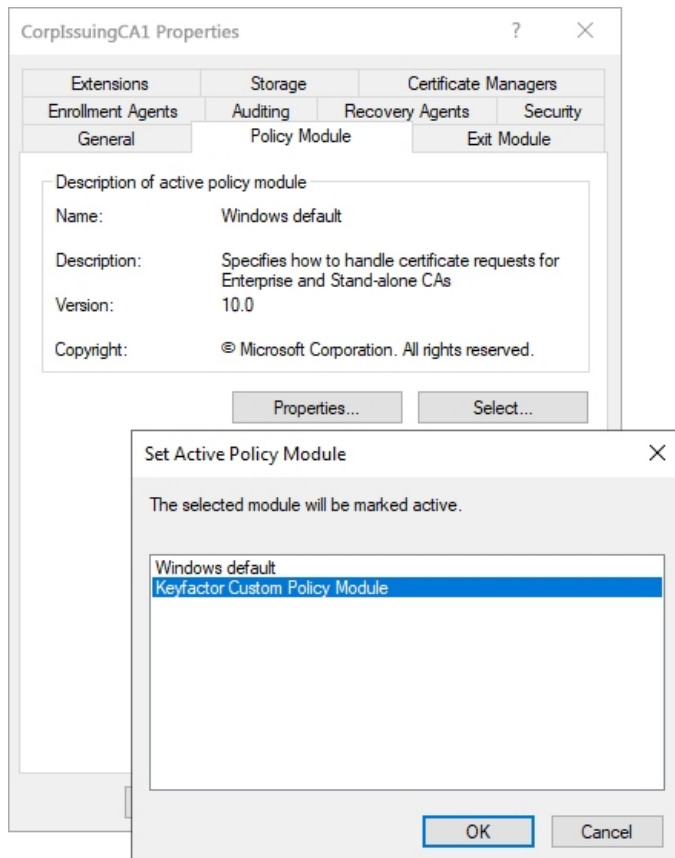


Figure 68: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

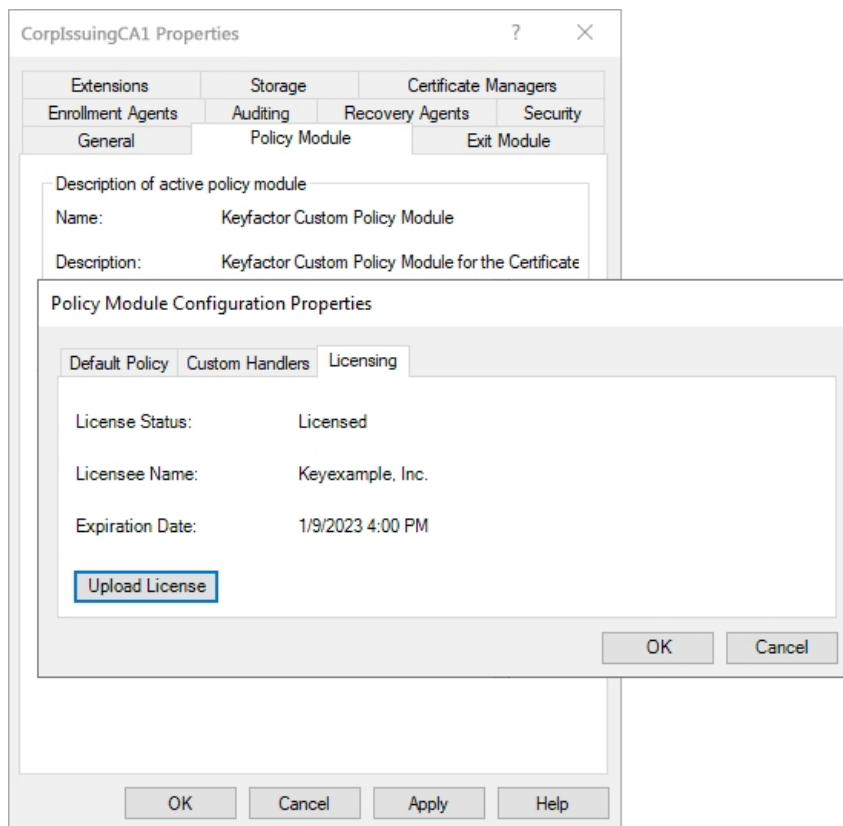


Figure 69: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **RFC 2818 Policy Handler** under Loaded Handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

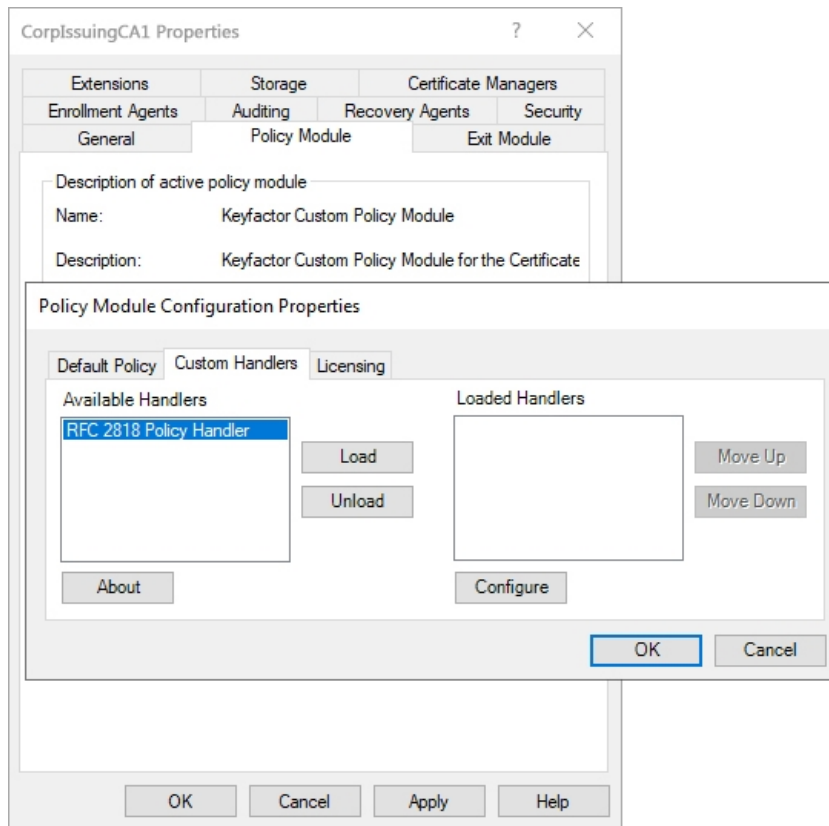


Figure 70: Enable the RFC 2818 Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the RFC 2818 Policy Handler under Loaded Handlers and click **Configure**.
12. On RFC 2818 Policy Handler configuration dialog, select the templates that should be under management by the RFC 2818 policy handler and click **Add**. Certificate enrollments from any source made using the templates selected here on the configured CA will automatically be assigned a DNS SAN matching the certificate's CN.

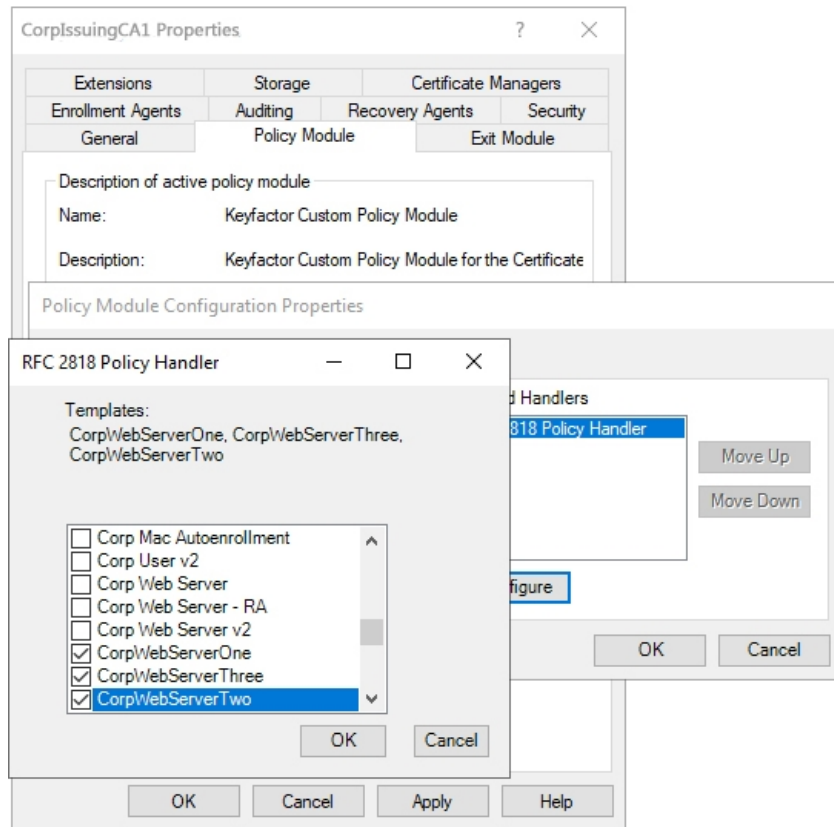


Figure 71: Add Templates for Management with the RFC 2818 Policy Handler

13. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

2.5.3.2 Install the Keyfactor SAN Attribute Policy Handler

To begin the SAN Attribute Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.



Figure 72: Install SAN Attribute Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the SAN Attribute Policy Handler, deselect all the components except the SAN Attribute Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

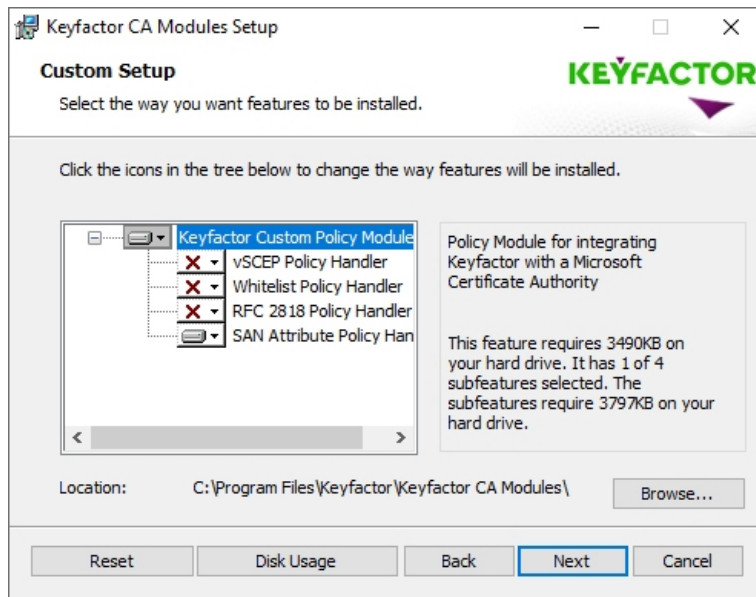


Figure 73: Install SAN Attribute Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the "Launch the CA MMC snap-in now" box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

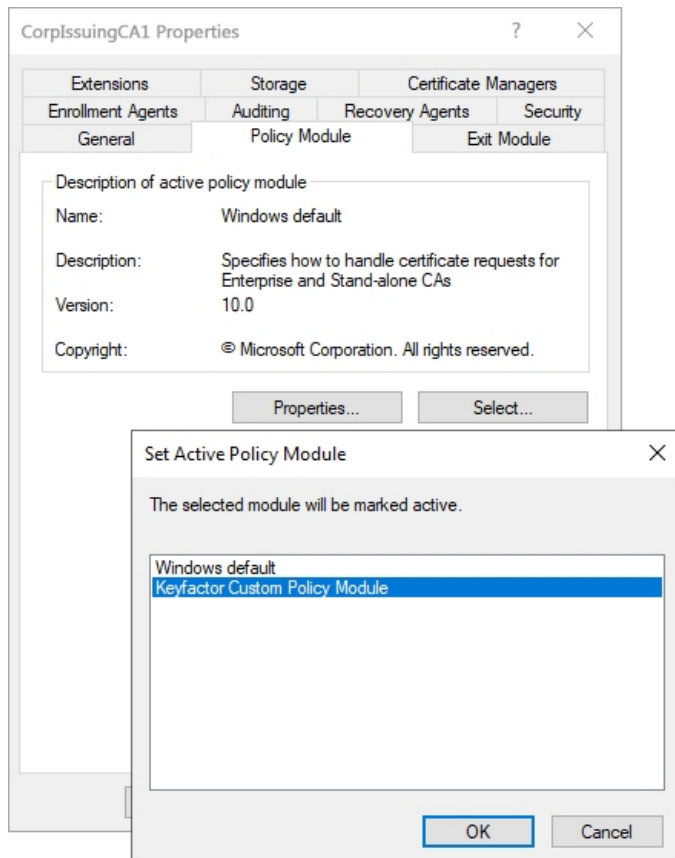


Figure 74: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

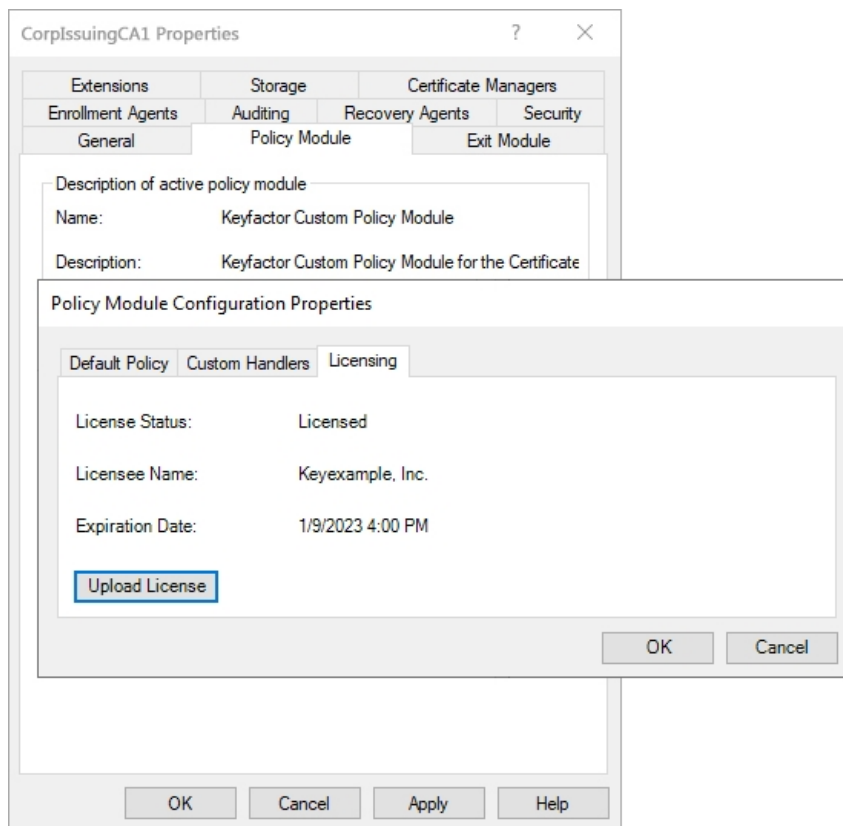


Figure 75: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **SAN Attribute Policy Handler** under Loaded Handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

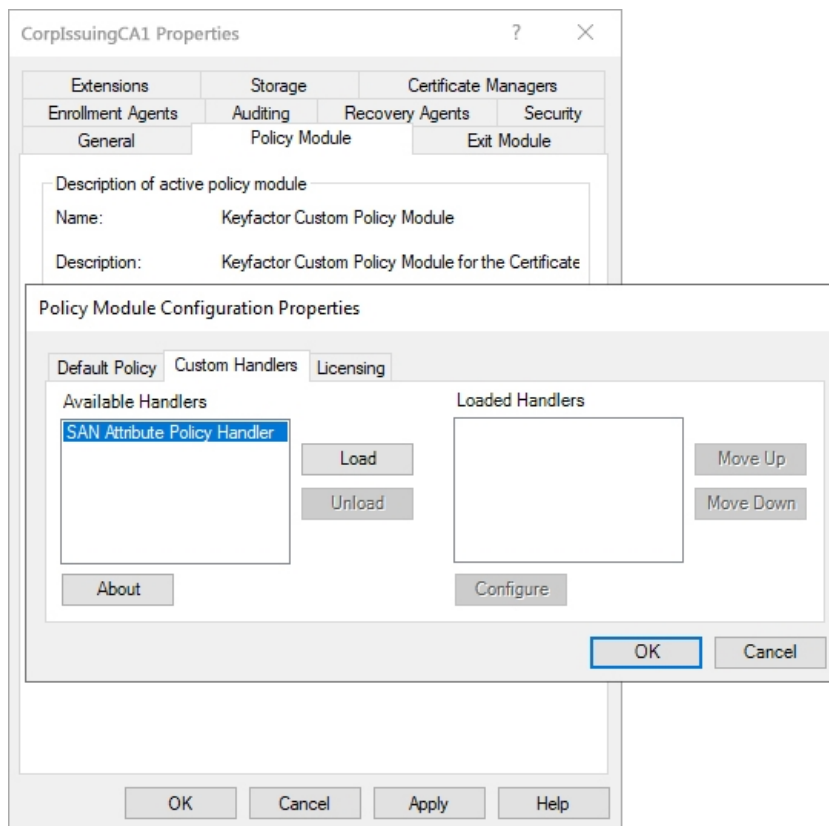


Figure 76: Enable the SAN Attribute Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the SAN Attribute Policy Handler under Loaded Handlers and click **Configure**.
12. On SAN Attribute Policy Handler configuration dialog, select the templates that should be under management by the SAN Attribute policy handler and click **Add**. Certificate enrollments from any source made using the templates selected here on the configured CA and a CSR enrollment method will allow the addition of SANs not included in the CSR and control the SAN addition functionality on a template-by-template basis without the need to enable the Microsoft CA EDITF_ATTRIBUTESUBJECTALTNAME2 flag.

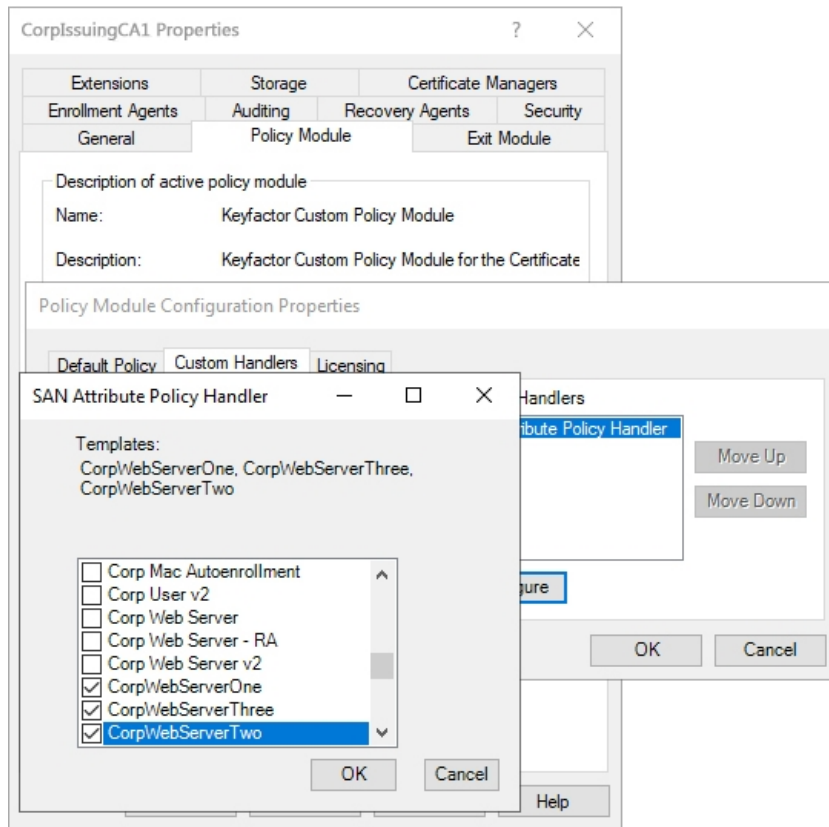


Figure 77: Add Templates for Management with the SAN Attribute Policy Handler

13. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

2.5.3.3 Install the Keyfactor vSCEP™ Policy Handler

To begin the vSCEP™ Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.

1. On the first installation page, click **Next** to begin the setup wizard.

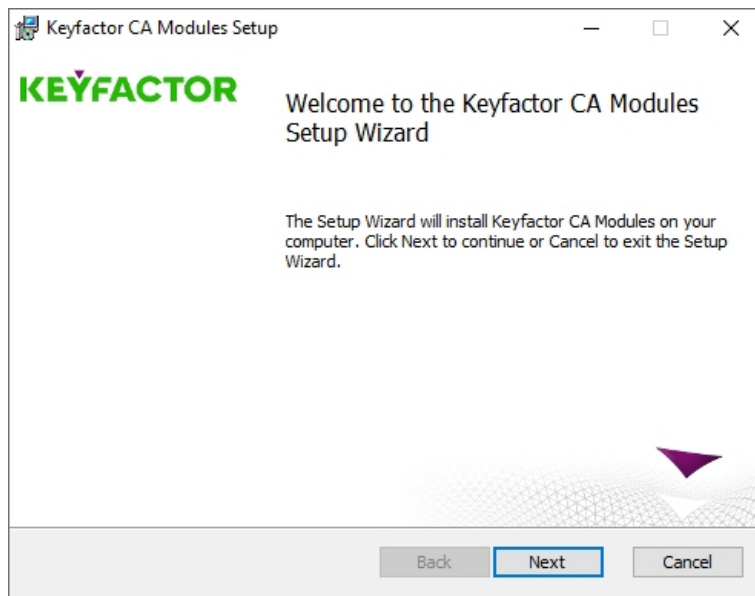


Figure 78: Install vSCEP Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the vSCEP™ Policy Handler, deselect all the components except the vSCEP™ Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

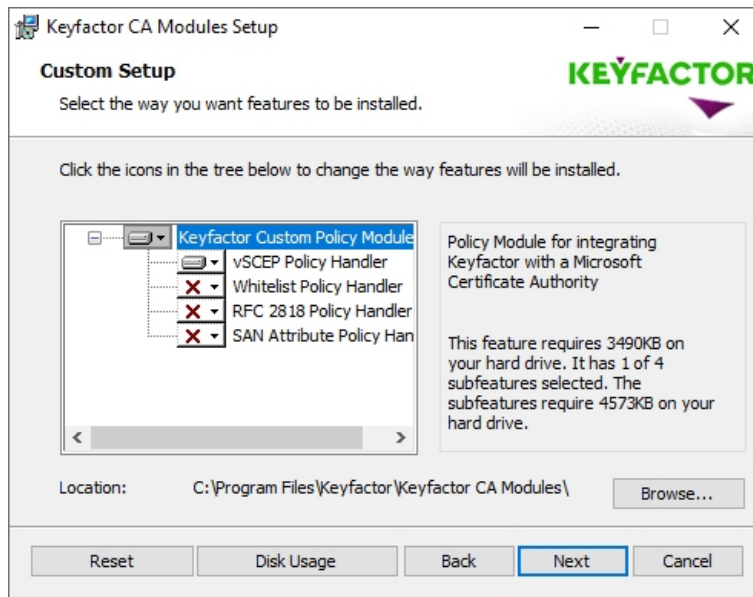


Figure 79: Install vSCEP Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the "Launch the CA MMC snap-in now" box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

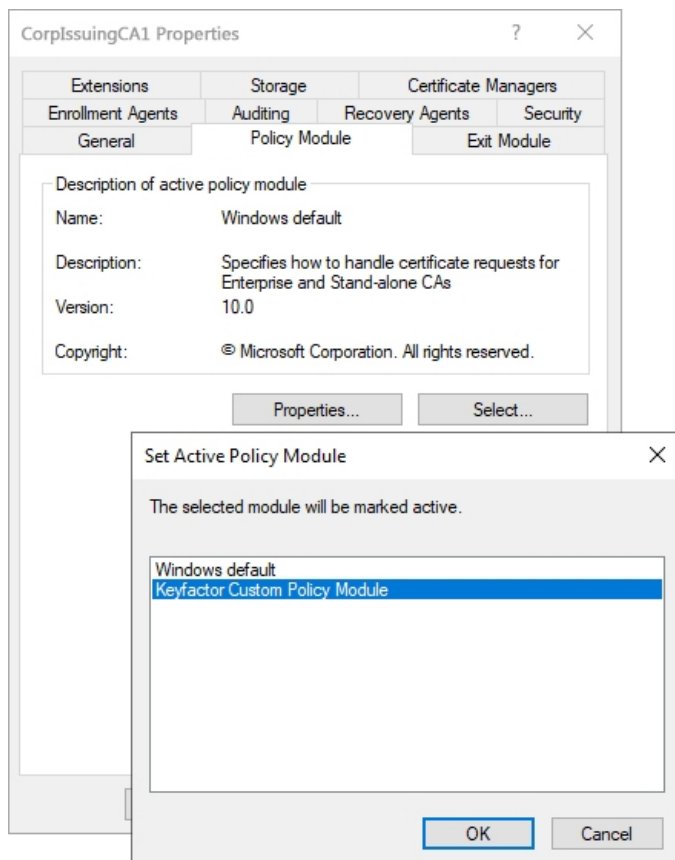


Figure 80: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

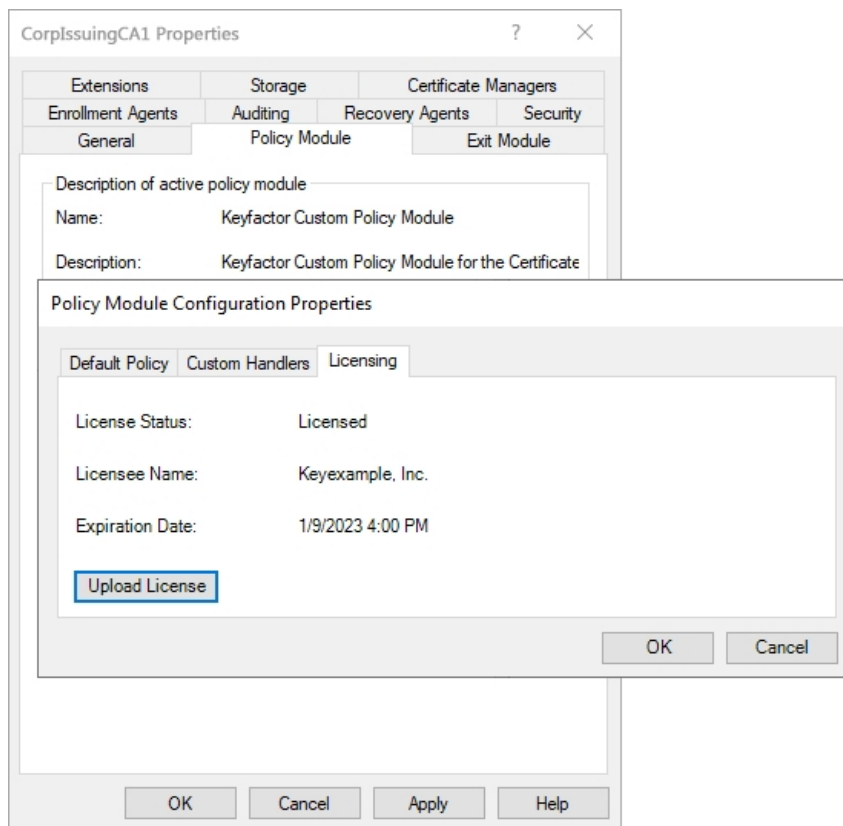


Figure 81: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **vSCEP Policy Handler** on the list of available handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

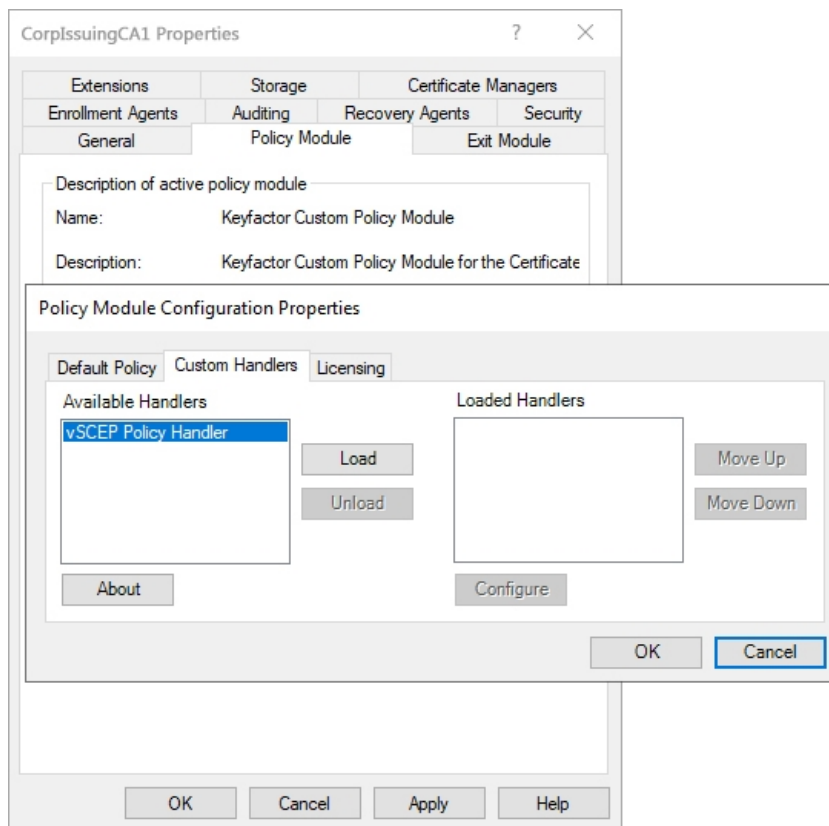


Figure 82: Enable the vSCEP™ Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the vSCEP™ Policy Handler under Loaded Handlers and click **Configure**.
12. On the vSCEP™ Policy Handler dialog, enter the vSCEP URL at the top of the page. This is the URL for the Keyfactor Command server where the vSCEP service is installed followed by the virtual directory name of the validation service. By default, the virtual directory name is *KeyfactorVSCEP*. Enter the username and password for the Active Directory service account used to authenticate to the vSCEP service on the Keyfactor Command server. Click the Verify button to confirm that the username and password entered are valid. Leave the **Verify Connection on Save** box checked and click **Save**. The URL entered will be tested to confirm that it can be reached and that authentication to it succeeds using the credentials provided.



Note: The user you enter here needs to be a member of the group you configure for *Allowed Users/Groups* on the vSCEP Service tab in the Keyfactor Command configuration wizard (see [vSCEP Services Tab on page 60](#)).

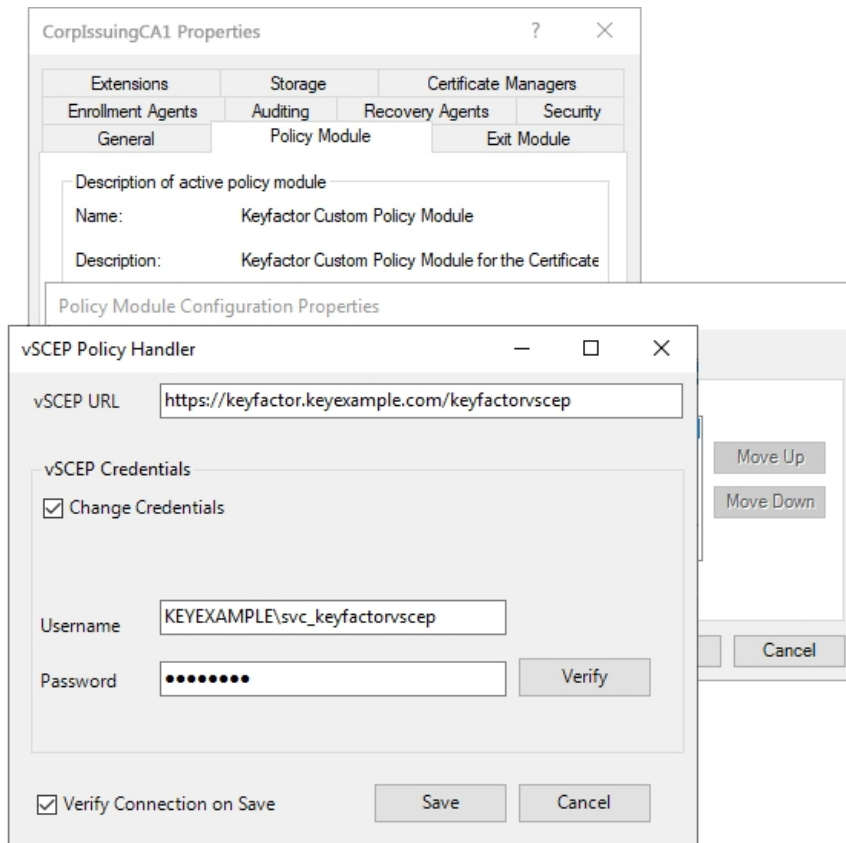


Figure 83: Configure Settings for the vSCEP™ Policy Handler

13. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

2.5.3.4 Install the Keyfactor Whitelist Policy Handler

To begin the Whitelist Policy Handler installation, execute the KeyfactorCAModuleInstaller.msi file from the Keyfactor installation media and install as follows.



Note: The following Windows update affects how certificate requests are built when sent to a Microsoft CA and may cause enrollments done outside Keyfactor Command against a Microsoft CA configured with the Whitelist Policy Handler to fail.

<https://support.microsoft.com/en-us/topic/kb5014754-certificate-based-authentication-changes-on-windows-domain-controllers-ad2c23b0-15d8-4340-a468-4d4f3b188f16>

1. On the first installation page, click **Next** to begin the setup wizard.



Figure 84: Install Whitelist Policy Handler: Begin Setup Wizard

2. On the next page, read and accept the license agreement and click **Next**.
3. On the next page, select the components to install. For the Whitelist Policy Handler, deselect all the components except the Whitelist Policy Handler component. If desired, you can highlight Keyfactor Custom Policy Module and click **Browse** to select an alternate installation location for the files. The default installation location is:

C:\Program Files\Keyfactor\Keyfactor CA Modules

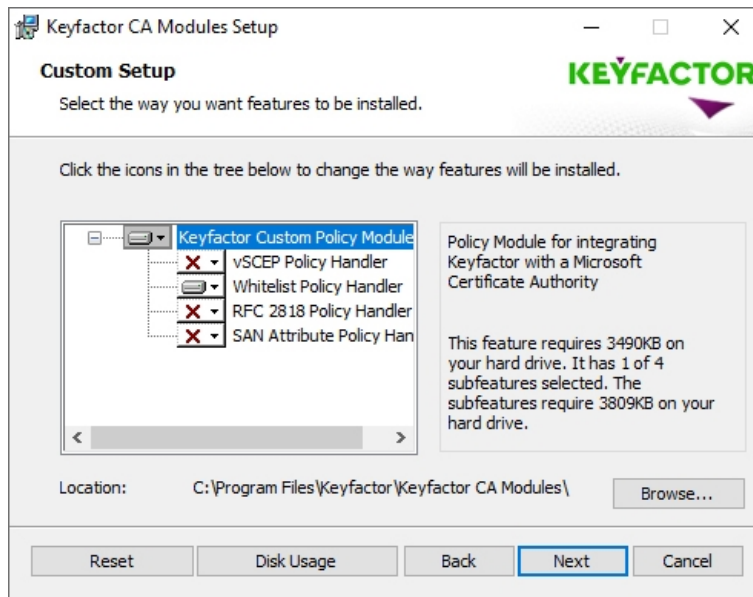


Figure 85: Install Whitelist Policy Handler: Select Components

4. On the next screen, click **Install**.
5. On the final installation wizard page, leave the "Launch the CA MMC snap-in now" box selected and click **Finish**. The Microsoft Certification Authority management tool should start automatically. This can take several seconds.
6. In the Certification Authority management tool, right-click the CA name at the top of the tree and choose **Properties**.
7. In the Properties dialog for the CA on the CA Policy Module tab, click **Select**, highlight the **Keyfactor Custom Policy Module** in the Set Active Policy Module dialog and click **OK**.

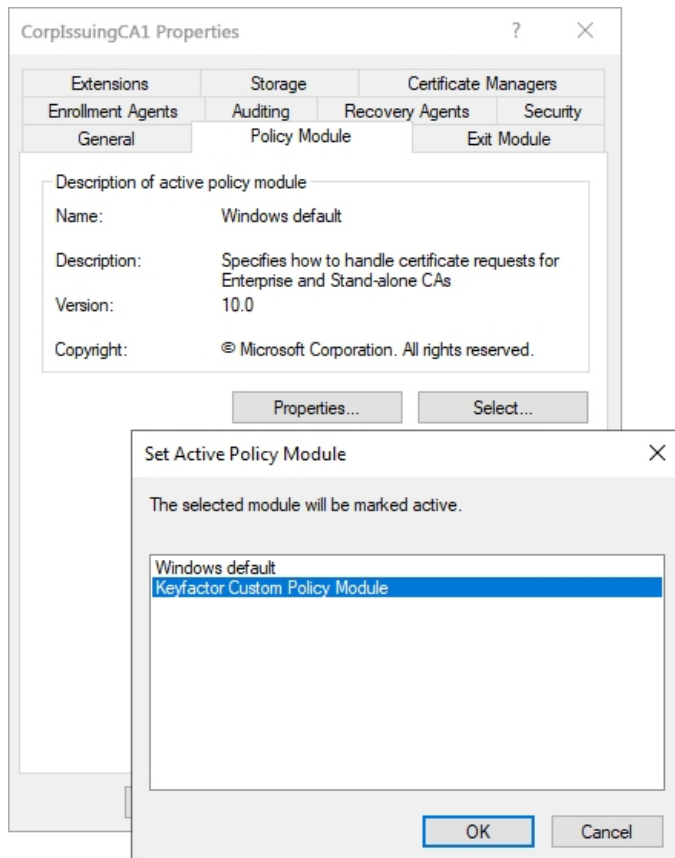


Figure 86: Enable the Keyfactor CA Policy Module

8. In the Properties dialog for the CA on the CA Policy Module tab, click **Properties**.
9. On the Licensing tab of the Policy Module Configuration Properties page, click **Upload License** and browse to locate the license file provided to you by Keyfactor. This file should have the extension CMSLICENSE.

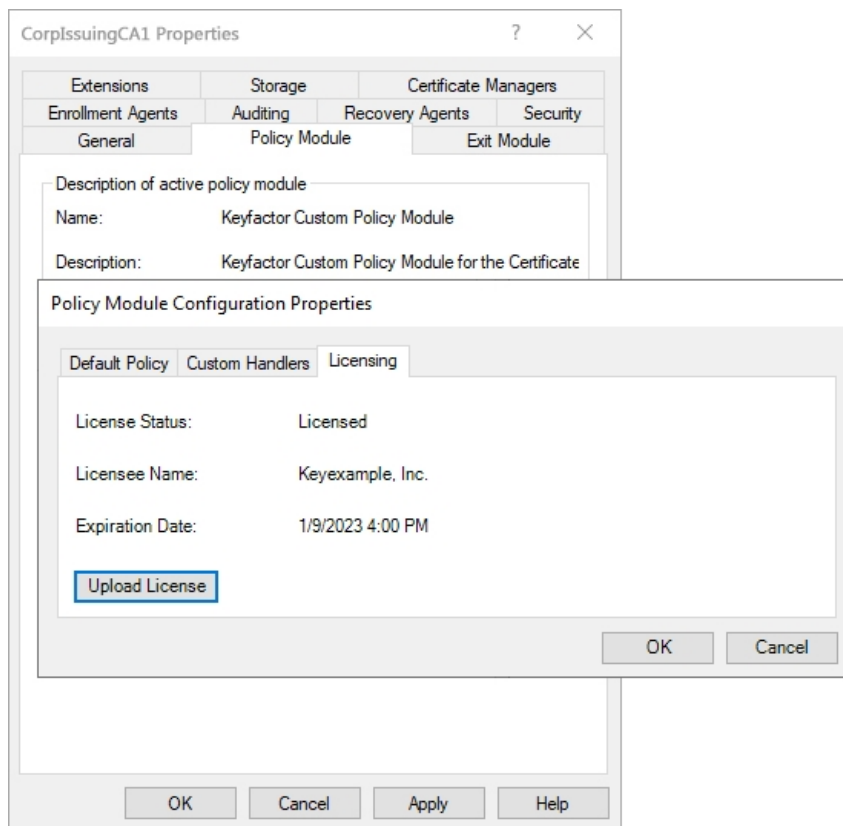


Figure 87: Upload the Keyfactor CA Policy Module License

10. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight the **CMS Machine Whitelist Policy** on the list of available handlers, click **Load** to move it over to the loaded handlers, and click **OK**.

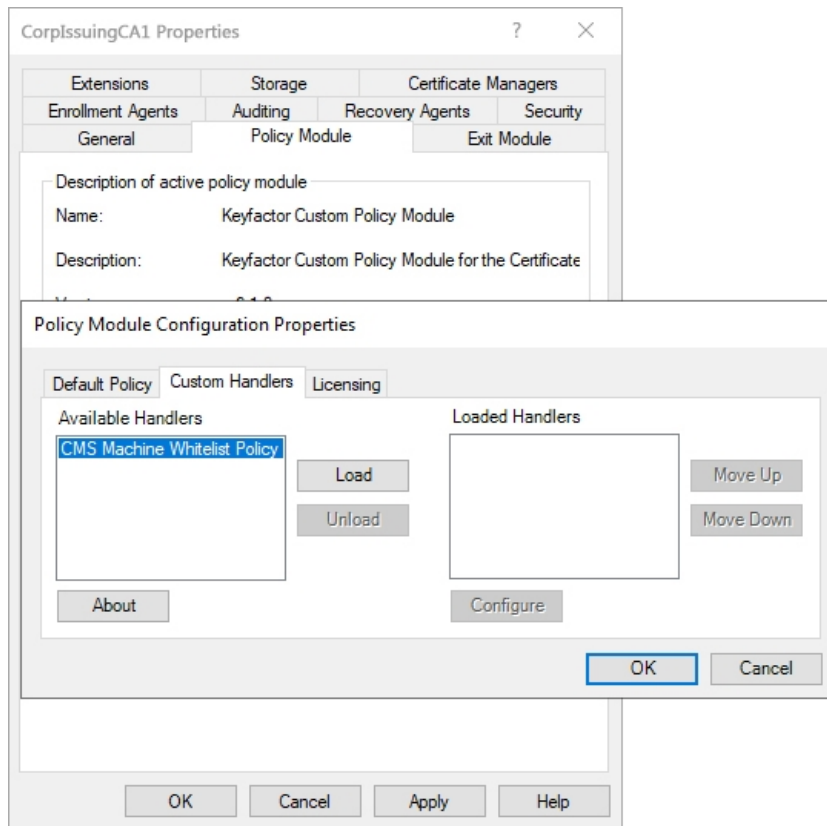


Figure 88: Enable the Whitelist Policy Handler

11. On the Custom Handlers tab of the Policy Module Configuration Properties page, highlight CMS Machine Whitelist Policy under Loaded Handlers and click **Configure**.
12. On the Template tab of the Policy Module Configuration dialog, enter the certificate *template names* (short names), not the template display names, one at a time, of the certificate template(s) you want to manage with the whitelist policy handler and click **Add**. In many cases, the template name is the same as the template display name with the spaces removed. Any templates entered here will be available for enrollment only from machines listed on the Machine Names tab.

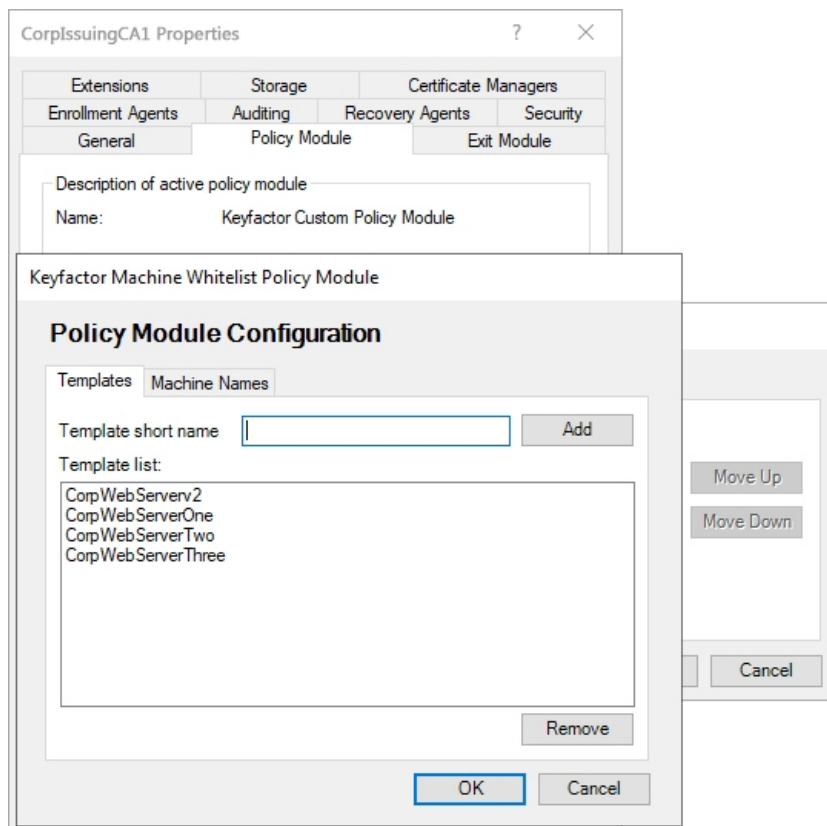


Figure 89: Add Templates for Management with the Whitelist Policy Handler

13. On the Machine Names tab of the Policy Module Configuration dialog, enter the machine names (FQDNs), one at a time, of the machines that you want to manage with the whitelist policy handler and click **Add**. Any machines entered here will be allowed to enroll for the templates listed on the Templates tab.

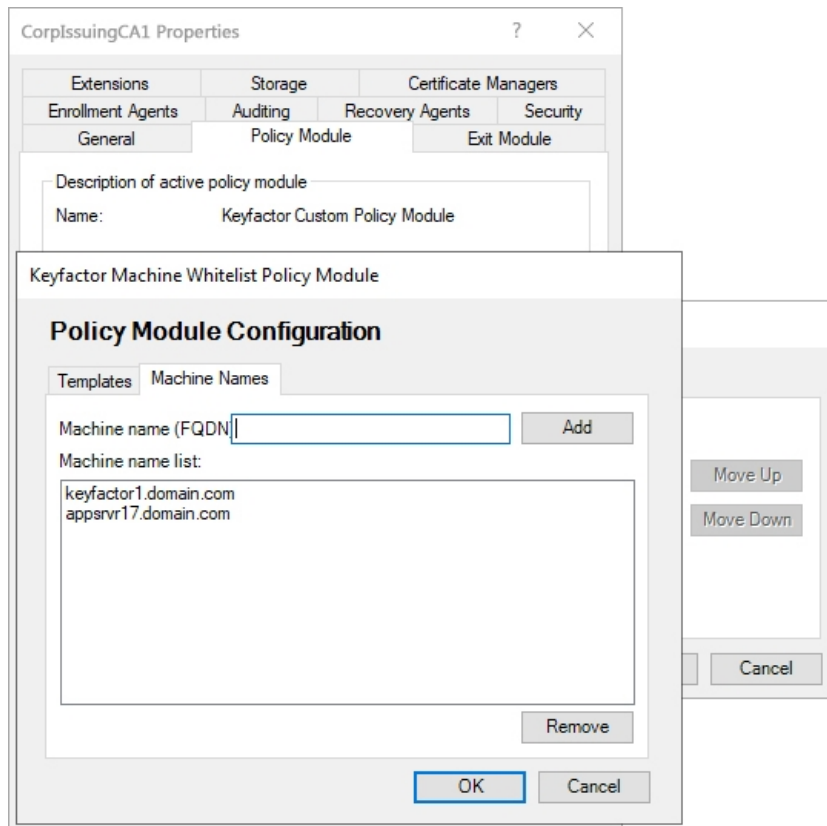


Figure 90: Add Machines for Management with the Whitelist Policy Handler

14. Click **OK** as many times as needed to close the configuration dialogs and save the configuration. You will be prompted to restart the CA services.

2.5.4 Configure Logging for the Keyfactor CA Policy Module

The Keyfactor CA Policy Module provides extensive logging for visibility and troubleshooting. By default, Keyfactor CA Policy Module places its log files in the C:\Keyfactor\logs directory, generates logs at the "Info" logging level and stores the primary logs for two days before deleting them.

To configure logging:

1. On the policy module server where you wish to adjust logging, open a text editor (e.g. Notepad) using the "Run as administrator" option.
2. In the text editor, browse to open the Nlog.config file for the Keyfactor CA Policy Module. The file is located in the installation directory for the product, which is the following by default:

C:\Program Files\Keyfactor\Keyfactor CA Modules\NLog.config

3. Your Nlog.config file may have a slightly different layout than shown here, but it will contain the five fields highlighted in [Figure 91: Keyfactor CA Policy Module NLog.config File](#). The fields you may wish to edit are:

- fileName="C:\Keyfactor\Logs\Keyfactor_CA_Log.txt"

The path and file name of the active policy module log file, referencing the logDirectory variable.



Important: If you choose to change the path for storage of the log files, you will need to create the new directory (e.g. D:\KeyfactorLogs) and grant the service account under which the Active Directory Certificate Services service is running full control permissions on this directory.

- archiveFileName="C:\Keyfactor\Logs\Keyfactor_CA_Log_Archive_{#}.txt"

The path and file name of previous days' orchestrator log files, referencing the logDirectory variable. The orchestrator rotates log files daily and names the previous files using this naming convention.

- maxArchiveFiles="2"

The number of archive files to retain before deletion.

- name="*" minlevel="Info" writeTo="logfile"

The level of log detail that should be generated and output to the log file. The default "Info" level logs error and some informational data but at a minimal level to avoid generating large log files. For troubleshooting, it may be desirable to set the logging level to "Debug" or "Trace". Available log levels (in order of increasing verbosity) are:

- OFF—No logging
- FATAL—Log severe errors that cause early termination
- ERROR—Log severe errors and other runtime errors or unexpected conditions that may not cause early termination
- WARN—Log errors and use of deprecated APIs, poor use of APIs, "almost" errors, and other runtime situations that are undesirable or unexpected but not necessarily "wrong"
- INFO—Log all of the above plus runtime events (startup/shutdown)
- DEBUG—Log all of the above plus detailed information on the flow through the system
- TRACE—Maximum log information—this option can generate VERY large log files

```
<targets>
  <target name="buffered_wrapper" xsi:type="BufferingWrapper" slidingTimeout="true" bufferSize="500" flushTimeout="500">
    <target xsi:type="File" name="logfile" fileName="C:\Keyfactor\logs\Keyfactor_CA_Log.txt" layout="${longdate} ${logger} [{level}] - ${message}"
      archiveFileName="C:\Keyfactor\logs\Keyfactor_CA_Log_Archive_{#}.txt" archiveEvery="Day" archiveNumbering="Rolling" maxArchiveFiles="2"/>
    </target>
  </target>
  <target xsi:type="OutputDebugString" name="String" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Debugger" name="debugger" layout="${longdate} ${logger}::${message}"/>
  <target xsi:type="Console" name="console" layout="${logger} ${message}"/>
  <target xsi:type="EventLog" name="eventLog" source="Keyfactor CA Modules"
    eventId="${event-properties:item=eventID}" category="${event-properties:item=categoryID}" layout="${event-properties:item=message}" />
</targets>
<rules>
  <!-- Don't write events to the log file (log file should contain different, more verbose, logging) -->
  <logger name="*-EVENT" minlevel="Info" writeTo="eventLog" final="true" />
  <logger name="*" minlevel="Info" writeTo="logfile" />
</rules>
```

Figure 91: Keyfactor CA Policy Module NLog.config File

2.5.5 Add Non-Keyfactor SCEP Servers to the Ignore List

This step only needs to be completed if you installed the Keyfactor CA Policy Module with the vSCEP™ Policy Handler.

If your CA for that issues certificates based on SCEP challenges is used by multiple SCEP servers, you will need to add SCEP servers not used for Keyfactor Command vSCEP API requests to the ignore list on the CA running the vSCEP™ Policy Handler. This will allow the vSCEP™ Policy Handler to ignore requests (passing them through to the CA) from the listed SCEP servers. Without this feature, SCEP challenges from non-Keyfactor Command servers would be denied because no data exists against which to verify the certificate details.

The vSCEP™ Policy Handler reads the ignore list in the registry string value **CCMBlacklist**. This field contains a semi-colon-delimited list of SCEP server host names from which all certificate requests should be ignored by the Keyfactor CA Policy Module and passed through to the CA. The **CCMBlacklist** setting can be found in the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Certified Security Solutions\vSCEP\Configuration
```

The **CCMBlacklist** registry value does not exist by default. Use the Registry Editor (regedit) to create the **CCMBlacklist** value as a DWORD and populate it with the SCEP server FQDNs for any SCEP servers whose requests should bypass the vSCEP™ Policy Handler.

2.6 Appendices

- [Appendix - Troubleshooting Logi Log Files below](#)
- [Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup on the next page](#)
- [Appendix - Configuration Wizard Errors in the Logs on page 140](#)

2.6.1 Appendix - Troubleshooting Logi Log Files

When troubleshooting Logi, the first thing to try is setting the *Debug Embedded Reports* application setting to **True** (see *Application Settings: Console Tab* in the *Keyfactor Command Reference Guide*). This allows the reports to output errors with debug level information if they generate errors. If this does not generate the information necessary to resolve the problem, it can sometimes be helpful to modify the Keyfactor Analysis web.config file to allow IIS to show the actual error the application is experiencing at a lower level. To configure this:

1. Browse to the *Logi* directory under the installed directory for your Keyfactor Command implementation. By default, this is:
C:\Program Files\Keyfactor\Keyfactor Platform\Logi
2. Using a text editor opened with the "Run as administrator" option, open the web.config file for editing.
3. Find the <customErrors mode="RemoteOnly"/> section and change this to <customErrors mode="On"/>.
4. Look for the debug output in the *Logi\rdDownload* directory under the installed directory for your Keyfactor Command implementation. By default, this is:
C:\Program Files\Keyfactor\Keyfactor Platform\Logi\rdDownload



Tip: If you do not find debug output when running a report manually in the Management Portal, try scheduling a report for delivery via email or saving to disk using the Report Manager. Debugging operates differently for these two modes of running a report.



Note: When the portal experiences a 500 error, Logi logs will not be written to the usual output directory.

```
<system.web>
  <!-- DYNAMIC DEBUG COMPILATION
  Set compilation debug="true" to insert debugging symbols (.pdb information)
  into the compiled page. Because this creates a larger file that executes
  more slowly, you should set this value to true only when debugging and to
  false at all other times. For more information, refer to the documentation about
  debugging ASP.NET files.
  -->
  <compilation defaultLanguage="vb" debug="true" />

  <!-- CUSTOM ERROR MESSAGES
  Set customErrors mode="On" or "RemoteOnly" to enable custom error messages, "Off" to disable.
  Add <error> tags for each of the errors you want to handle.
  -->
  <customErrors mode="RemoteOnly" />
  <!-- AUTHENTICATION
  This section sets the authentication policies of the application. Possible modes are "Windows",
  "Forms", "Passport" and "None"
  -->

  <authentication mode="Windows" />
```

Figure 92: Logi web.config

2.6.2 Appendix - Logi Load Balancing: Keyfactor Command Configuration Wizard Setup

In order for the Keyfactor Command Management Portal Dashboard and Reports to load when using a load balancer, the Keyfactor Command Configuration Wizard should have the following configuration on each of the application servers:

- On the Keyfactor Command Portal Tab, the Host Name must be the Load Balanced URL.

The screenshot shows the 'Keyfactor Configuration Wizard' window. On the left is a navigation pane with tabs: Application Pools, Database, Service, Email, Keyfactor Portal (selected), Dashboard and Reports, vSCEP Service, Orchestrators, API, and Audit Configuration. The main area displays configuration settings for the 'Keyfactor Portal' tab. The 'Host Name' field is highlighted with a red rectangle and contains the text 'keyfactor.keyexample.com'. To its right is a checked checkbox for 'Use SSL'. Below these are fields for 'Web Site' (Default Web Site), 'Virtual Directory' (KeyfactorPortal), and 'Application Pool' (CMS). The 'Administrative Users' field contains 'KEYEXAMPLE\Keyfactor Administrator'. The 'Certificate Subject Format' field contains a complex string: 'CN={CN},E={E},O={O},OU={OU},L={L},ST={ST},C={C}'. The 'Downloads Folder' field contains 'C:\CMS\AdminEnroll\'. The 'PFX Enrollment' section is expanded, showing 'Enabled' checked, 'PFX Password Type' set to 'Auto-Generated' (with 'Domain' also available), and 'Alphanumeric Password Characters' checked. At the bottom, there are 'Verify Configuration' and 'Cancel' buttons. The status bar at the very bottom shows 'Server: sqlsrvr1.keyexample.com', 'Database Name: Keyfactor', and 'Credential Type: Windows'.

Figure 93: Logi Configuration Settings—Keyfactor Command Portal Tab

- Dashboard and Reports Tab:
 - The Host Name must be the Load Balanced URL. This is the host name that the Management Portal server uses to connect to the Logi Analytics Platform, and it therefore needs to be the name used on the internal side of the network.
 - The Keyfactor Command Site IP Address(es) must contain:
 - The application machine IPv4 addresses for all of the servers that will be load balanced.
 - The application machine IPv6 addresses for all of the servers that will be load balanced if IPv6 is enabled.
 - The internal IPv4 addresses of the load balancer that the load balancer will use to connect to the application servers.

Keyfactor Configuration Wizard

File

Validation Errors and Warnings

Application Pools: Host Name: ☒ Use SSL

Database: Web Site:

Service: Virtual Directory:

Email: Application Pool:

Keyfactor Portal: Keyfactor Site IP Address(es):

☐ Use Basic Authentication

vSCEP Service: Keyfactor API User:

Orchestrators: Keyfactor API Password:

API

Audit Configuration

Verify Configuration Cancel

Server: sqlsrvr1.keyexample.com Database Name: Keyfactor Credential Type: Windows

Figure 94: Logi Configuration Settings—Keyfactor Command Dashboards and Reports Tab



Tip: All IP Addresses that could be used internally to connect to the Logi application must be in the Dashboard and Reports configuration section in the configuration wizard. This includes the application host IPs and the load balancer IPs. It is also recommended that the host file is modified to map the load balancer URL to the local IP address.

- Load Balancer

On the load balancer, a new rewrite rule needs to be made that changes the outbound URL from the application servers. Logi sends the `HostName.domain.com/KeyfactorAnalysis` URL back to the browser instead of the `LoadBalancer.URL.com/KeyfactorAnalysis` URL that the browser needs to complete the Logi authorization. In short, an Outbound rewrite rule needs to be created on the Load Balancer that does the following: `HostName_URL/KeyfactorAnalysis` needs to be converted to `LoadBalancer_URL/keyfactorAnalysis`

2.6.3 Appendix - Configuration Wizard Errors in the Logs

If incoming web request runs before the configuration is fully completed, you may encounter the following errors in the portal log file after upgrading. These errors are not something to be worried about. They just indicate that the web request was still looking at an old version of the database prior to it being completely upgraded.

```
2021-11-15 12:41:36.5155 Keyfactor.EF.KeyfactorExecutionStrategy [Error] - SqlException
with number 207 occurred, not attempting to retry the connection.
2021-11-15 12:41:36.5780 Keyfactor.EF.KeyfactorExecutionStrategy [Error] - Invalid column
name 'Immutable'.
```

Invalid column name 'SubscriberTerms'.

2021-11-15 12:41:36.6249 ASP.global_asax [Error] - An uncaught application error occurred:
An error occurred while executing the command definition. See the inner exception for
details.

2021-11-15 12:41:37.3281 ASP.global_asax [Error] - An uncaught application error occurred:
An item with the same key has already been added.

at System.ThrowHelper.ThrowArgumentException(ExceptionResource resource)

A

AIA

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

AnyAgent

The AnyAgent, one of Keyfactor's suite of orchestrators, is used to allow management of certificates regardless of source or location by allowing customers to implement custom agent functionality via an API.

AnyGateway

The Keyfactor AnyGateway is a generic third party CA gateway framework that allows existing CA gateways and custom CA connections to share the same overall product framework.

API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

Argument

A parameter or argument is a value that is passed into a function in an application.

Authority Information Access

The authority information access (AIA) is included in a certificate--if configured--and identifies a location from which the chain certificates for that certificate may be retrieved.

B

Bash Orchestrator

The Bash Orchestrator, one of Keyfactor's suite of orchestrators, is used to discover and manage SSH keys across an enterprise.

Blueprint

A snapshot of the certificate stores and scheduled jobs on one orchestrator, which can be used to create matching certificate stores and jobs on another orchestrator with just a few clicks.

C

CA

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

Certificate Authority

A certificate authority (CA) is an entity that issues digital certificates. Within Keyfactor Command, a CA may be a Microsoft CA or a Keyfactor gateway to a cloud-based or remote CA.

Certificate Revocation List

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

Certificate Signing Request

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When you generate a CSR within Keyfactor Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

CN

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. server-name.keyexample.com or www.keyexample.com).

Collection

The certificate search function allows you to query the Keyfactor Command database for certificates from any available source based on any criteria of the certificates and save the results as a collection that will be available in other places in the Management Portal (e.g. expiration alerts and certain reports).

Common Name

A common name (CN) is the component of a distinguished name (DN) that represents the primary name of the object. The value varies depending on the type of object. For a user object, this would be the user's name (e.g. CN=John Smith). For SSL certificates, the CN is typically the fully qualified domain name (FQDN) of the host where the SSL certificate will reside (e.g. server-name.keyexample.com or www.keyexample.com).

Configuration Tenant

A grouping of CAs. The Microsoft concept of forests is not used in EJBCA so to accommodate the new EJBCA functionality, and to avoid confusion, the term forest needed to be renamed. The new name is configuration tenant. For EJBCA, there would be one configuration tenant per EJBCA server install. For Microsoft, there would be one per forest. Note that configuration tenants cannot be mixed, so Microsoft and EJBCA cannot exist on the same configuration tenant.

CRL

A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted.

CSR

A CSR or certificate signing request is a block of encoded text that is submitted to a CA when enrolling for a certificate. When you generate a CSR within Keyfactor Command, the matching private key for it is stored in Keyfactor Command in encrypted format and will be married with the certificate once returned from the CA.

D

DER

A DER format certificate file is a DER-encoded binary certificate. It contains a single certificate and does not support storage of private keys. It sometimes has an extension of .der but is often seen with .cer or .crt.

Distinguished Name

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs, separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

DN

A distinguished name (DN) is the name that uniquely identifies an object in a directory. In the context of Keyfactor Command, this directory is generally Active Directory. A DN is made up of attribute=value pairs, separated by commas. Any of the attributes defined in the directory schema can be used to make up a DN.

DNS

The Domain Name System is a service that translates names into IP addresses.

E

ECC

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

Endpoint

An endpoint is a URL that enables the API to gain access to resources on a server.

Enrollment

Certificate enrollment refers to the process by which a user requests a digital certificate. The user must submit the request to a certificate authority (CA).

EOBO

A user with an enrollment agent certificate can enroll for a certificate on behalf of another user. This is often used when provisioning technology such as smart cards.

F

Forest

An Active Directory forest (AD forest) is the top most logical container in an Active Directory configuration that contains domains, and objects such as users and computers.

G

Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

H

Host Name

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g. server-name.keyexample.com) and sometimes just as a short name (e.g. servername).

Hosted Config Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor Gateway Connector and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command. The portal is Azure-hosted and managed by Keyfactor.

Hosted Configuration Portal

The Keyfactor Hosted Configuration Portal is used to configure connections between on-premise instances of the Keyfactor Gateway Connector and on-premise CAs to make them available to Azure-hosted instance of Keyfactor Command. The portal is Azure-hosted and managed by Keyfactor.

Hostname

The unique identifier that serves as name of a computer. It is sometimes presented as a fully qualified domain name (e.g.

servername.keyexample.com) and sometimes just as a short name (e.g. servername).

J

Java Agent

The Java Agent, one of Keyfactor's suite of orchestrators, is used to perform discovery of Java keystores and PEM certificate stores, to inventory discovered stores, and to push certificates out to stores as needed.

Java Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

JKS

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

K

Key Length

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

Key Pair

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

Key Size

The key size or key length is the number of bits in a key used by a cryptographic algorithm.

Key Type

The key type identifies the type of key to create when creating a symmetric or asymmetric key. It references the signing algorithm and often key size (e.g. AES-256, RSA-2048, Ed25519).

Keyfactor CA Management Gateway

The Keyfactor CA Management Gateway is made up of the Keyfactor Gateway Connector, installed in the customer forest to provide a connection to the local CA, and the Azure-hosted and Keyfactor managed Hosted Configuration Portal. The solution is used to provide a connection between a customer's on-premise CA and an Azure-hosted instance of Keyfactor Command for synchronization, enrollment, and management of certificates.

Keyfactor Gateway Connector

The Keyfactor Gateway Connector is installed in the customer forest to provide a connection between the on-premise CA and the Azure-hosted, Keyfactor managed Hosted Configuration Portal to provide support for synchronization, enrollment and management of certificates through the Azure-hosted instance of Keyfactor Command for the on-premise CA. It is supported on both Windows and Linux.

Keyfactor Universal Orchestrator

The Keyfactor Universal Orchestrator, one of Keyfactor's suite of orchestrators, is used to interact with Windows servers (a.k.a. IIS certificate stores) and FTP capable devices for certificate management, run SSL discovery and management tasks, and manage synchronization of certificate authorities in remote forests. With the addition of custom extensions, it can run custom jobs to provide certificate management capabilities on a variety of platforms and devices (e.g. F5 devices, NetScaler devices, Amazon Web Services (AWS) resources) and execute tasks outside the standard list of certificate management functions. It runs on either Windows or Linux.

Keystore

A Java KeyStore (JKS) is a file containing security certificates with matching private keys. They are often used by Java-based applications for authentication and encryption.

L

Logical Name

The logical name of a CA is the common name given to the CA at the time it is created. For Microsoft CAs, this name can be seen at the top of the Certificate Authority MMC snap-in. It is part of the FQDN\Logical Name string that is used to refer to CAs when using command-line tools and in some Keyfactor Command configuration settings (e.g. ca2.keyexample.com\Corp Issuing CA Two).

M

MAC Agent

The MAC Agent, one of Keyfactor's suite of orchestrators, is used to manage certificates on any keychains on the Mac on which the Keyfactor MAC Agent is installed.

Metadata

Metadata provides information about a piece of data. It is used to summarize basic information about data, which can make working with the data easier. In the context of Keyfactor Command, the certificate metadata feature allows you to create custom metadata fields that allow you to tag certificates with tracking information about certificates.

O

Object Identifier

Object identifiers or OIDs are a standardized system for identifying any object, concept, or

"thing" with a globally unambiguous persistent name.

OID

Object identifiers or OIDs are a standardized system for identifying any object, concept, or "thing" with a globally unambiguous persistent name.

Orchestrator

Keyfactor orchestrators perform a variety of functions, including managing certificate stores and SSH key stores.

P

P12

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

P7B

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ---- END CERTIFICATE----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

P7C

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like ---- BEGIN CERTIFICATE---- and ---- END CERTIFICATE----. Unlike PEM files, PKCS #7

files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

Parameter

A parameter or argument is a value that is passed into a function in an application.

PEM

A PEM format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PEM certificates always begin and end with entries like -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----. PEM certificates can contain a single certificate or a full certificate chain and may contain a private key. Usually, extensions of .cer and .crt are certificate files with no private key, .key is a separate private key file, and .pem is both a certificate and private key.

PFX

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

PKCS #7

A PKCS #7 format certificate file is a base64-encoded certificate. Since it's presented in ASCII, you can open it in any text editor. PKCS #7 certificates always begin and end with entries that look something like -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----. Unlike PEM files, PKCS #7 files can contain only a certificate and its certificate chain but NOT its private key. Extensions of .p7b or .p7c are usually seen on certificate files of this format.

PKCS#12

A PFX file (personal information exchange format), also known as a PKCS#12 archive, is a single, password-protected certificate archive that contains both the public and matching private key and, optionally, the certificate chain. It is a common format for Windows servers.

PKI

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

Private Key

Private keys are used in cryptography (symmetric and asymmetric) to encrypt or sign content. In asymmetric cryptography, they are used together in a key pair with a public key. The private or secret key is retained by the key's creator, making it highly secure.

Public Key

In asymmetric cryptography, public keys are used together in a key pair with a private key. The private key is retained by the key's creator while the public key is widely distributed to any user or target needing to interact with the holder of the private key.

Public Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

R

Rogue Key

A rogue key, in the context of Keyfactor Command, is an SSH public key that appears in an

authorized_keys file on a server managed by the SSH orchestrator without authorization.

Root of Trust

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

RoT

A root of trust (RoT) is a source within a cryptographic system that can always be trusted. It is typically a hardened hardware module. HSMs (hardware security modules) and TPMs (trusted platform modules) are examples of RoTs.

RPC

Remote procedure call (RPC) allows one program to call a function from a program located on another computer on a network without specifying network details. In the context of Keyfactor Command, RPC errors often indicate Kerberos authentication or delegation issues.

rsyslog

Rsyslog is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network.

S

SAN

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of SAN formats are supported, with DNS name being the most common.

server name indication

Server name indication (SNI) is an extension to TLS that provides for including the hostname of the target server in the initial handshake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

SMTP

Short for simple mail transfer protocol, SMTP is a protocol for sending email messages between servers.

SNI

Server name indication (SNI) is an extension to TLS that provides for including the hostname of the target server in the initial handshake request to allow the server to respond with the correct SSL certificate or allow a proxy to forward the request to the appropriate target.

SSH

The SSH (secure shell) protocol provides for secure connections between computers. It provides several options for authentication, including public key, and protects the communications with strong encryption.

SSL

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

Subject Alternative Name

The subject alternative name (SAN) is an extension to the X.509 specification that allows you to specify additional values when enrolling for a digital certificate. A variety of SAN formats are supported, with DNS name being the most common.

T

Template

A certificate template defines the policies and rules that a CA uses when a request for a certificate is received.

TLS

TLS (Transport Layer Security) and its predecessor SSL (Secure Sockets Layer) are protocols for establishing authenticated and encrypted links between networked computers.

Trusted CA

A certificate authority in the forest in which Keyfactor Command is installed or in a forest in a two-way trust with the forest in which Keyfactor Command is installed.

U

Untrusted CA

A certificate authority in a forest in a one-way trust with the forest in which Keyfactor Command is installed or in a forest that is untrusted by the forest in which Keyfactor Command is installed. Non-domain-joined standalone CAs also fall into this category.

W

Web API

A set of functions to allow creation of applications. Keyfactor offers the Keyfactor API, which allows third-party software to integrate with the advanced certificate enrollment and management features of Keyfactor Command.

Windows Orchestrator

The Windows Orchestrator, one of Keyfactor's suite of orchestrators, is used to manage

synchronization of certificate authorities in remote forests, run SSL discovery and management tasks, and interact with Windows servers as well as F5 devices, NetScaler devices, Amazon Web Services (AWS) resources, and FTP capable devices, for certificate management. In addition, the AnyAgent capability of the Windows Orchestrator allows it to be extended to create custom certificate store types and management capabilities regardless of source platform or location.

Workflow

A workflow is a series of steps necessary to complete a process. In the context of Keyfactor Command, it refers to the workflow builder, which allows you automate event-driven tasks when a certificate is requested or revoked.

X

x.509

In cryptography, X.509 is a standard defining the format of public key certificates. An X.509 certificate contains a public key and an identity (e.g. a host name or an organization or individual name), and is either signed by a certificate authority or self-signed. When a certificate is signed by a trusted certificate authority it can be used to establish trusted secure communications with the owner of the corresponding private key. It can also be used to verify digitally signed documents and emails.

4.0 Copyright Notice

User guides and related documentation from Keyfactor are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. This documentation may not be disclosed, transferred, modified, or reproduced in any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of Keyfactor and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by Keyfactor. Keyfactor assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from Keyfactor.