[MS-CAESO]: Certificate Autoenrollment System Overview

Intellectual Property Rights Notice for Protocol Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Preliminary Documentation. This Open Specification is preliminary documentation for this technology. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Abstract

Microsoft networks and protocols often require the use of digital certificates for encryption and authentication. Enrolling these certificates is normally the task of the system administrator. Implementing the Computer Certificate Autoenrollment Task will enable a system to enroll and re-enroll certificates automatically. Autoenrollment handles all of the details for enrollment and re-enrollment of certificates, leaving the system administrator free to perform other tasks. This document describes the functionality of autoenrollment and how it uses certificate enrollment protocols. It provides examples of some of the common usage scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that this task uses.

Contents

1 Introduction

A "Defined Task" is a logical procedure that uses one or more Protocols or Systems to accomplish a specific goal. This Defined Task System Document describes the Computer Certificate Autoenrollment Task.

In conjunction with Protocol Technical Documents, which are primarily intended to cover Protocols, this Defined Task System Document presents and covers the rules for information exchange relevant to the Tasks, and the Protocols they use, that are used to interoperate or communicate with a Windows client operating systems and selected Windows Server scenarios (those covered in published TDs) in its various environments.

The Certificate Autoenrollment System Overview (CAESO) describes the task of automatically enrolling and re-enrolling digital certificates that systems and protocols require to operate. System administrators usually perform this task manually, and as demand for certificates increases, they can become overwhelmed. Autoenrollment automatically handles certificate enrollment and the re-enrollment of expired certificates, which relieves the administrator from this task.

Autoenrollment serves a central role in client and server relationships that rely on certificate enrollment. By instituting autoenrollment, the system administrator can concentrate on other tasks. Autoenrollment determines what policies are available for certificate enrollment, the set of certificates specified through these policies, and what certificates can be issued based on the templates in these policies.

This document provides the framework necessary to understand the relationships among the protocols that implement this functionality, and presents an overview of the working process of autoenrollment. It does not attempt to provide detailed information that is not needed to implement this protocol.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Active Directory
Active Directory domain
Basic Encoding Rules (BER)
certificate
certification authority (CA)
certificate revocation lists (CRL)
certificate template
digital certificate
Distributed Component Object Model (DCOM)
domain controller (DC)
domain member
exchange certificate
Group Policy
key exchange
Lightweight Directory Access Protocol (LDAP)
private key
public key
public key infrastructure (PKI)
registration authority (RA)
root certificate

The following terms are defined in [MS-ADTS]:

forest root domain NC

The following terms are defined in [MS-GPSO]:

Group Policy Client (GP Client)

The following terms are defined in [MS-WCCE]:

issuance

The following terms are defined in [MS-XCEP]:

certificate enrollment policy

The following terms are specific to this document:

CA certificates: CA certificates are **certificates** that are issued by one CA to another CA. These CA certificates become a part of the certificate trust hierarchy, the certificate path from end entity certificates to the trusted root CA certificate.

CEP: Certificate enrollment policy as defined in [MS-XCEP].

certificate enrollment: Certificate enrollment is the process of acquiring a **digital certificate** from a certification authority. This certificate and its associated **private key** establish a trusted identity for an entity using the **public key**-based services and applications.

LDAP: In this document the term LDAP always refers to the Lightweight Directory Access Protocol (LDAP) profile specified in [MS-ADTS] section 3.1.1.3.

policy server end point: A collection of information about a policy server, such as the protocol it supports, its Uniform Resource Identifier (URI), and authentication to be used when accessing the server.

The following protocol abbreviations are used in this document:

CRTD: Certificate Templates Structure Specification

DCOM: Distributed Component Object Model (DCOM) Remote Protocol Specification

GPREG: Group Policy: Registry Extension Encoding

HTTP: Hypertext Transfer Protocol

WCCE: Windows Client Certificate Enrollment Protocol Specification.

WSTEP: WS-Trust Enrollment Extensions

XCEP: X.509 Certificate Enrollment Policy Protocol Specification

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. Note that in [RFC2119] terms, most of these specifications should be imperative, to ensure interoperability. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

Any specification that does not explicitly use one of these terms is mandatory, exactly as if it used MUST.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-ADSO] Microsoft Corporation, "Active Directory System Overview", July 2009.

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification", June 2007.

[MS-AUTHSO] Microsoft Corporation, "Windows Authentication Services System", to be published 2009.

[MS-CRTD] Microsoft Corporation, "Certificate Templates Structure Specification", March 2007.

[MS-DISO] Microsoft Corporation, "Domain Interactions System Overview", to be published 2009.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-GPREG] Microsoft Corporation, "Group Policy: Registry Extension Encoding", August 2007.

[MS-GPSO] Microsoft Corporation, "Group Policy System Overview", February 2009.

[MS-WCCE] Microsoft Corporation, "Windows Client Certificate Enrollment Protocol Specification", June 2007.

[MS-WSTEP] Microsoft Corporation, "WS-Trust X.509v3 Token Enrollment Extensions", December 2008.

[MS-XCEP] Microsoft Corporation, "X.509 Certificate Enrollment Policy Protocol Specification", December 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC3852] Housley, R. "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, http://www.ietf.org/rfc/rfc3852.txt

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.2.2 Informative References

[MS-CASO] Microsoft Corporation, "Certification Authority System Overview", June 2008.

[CRYPTO] Menezes, A., Vanstone, S., and Oorschot, P., "Handbook of Applied Cryptography", 1997, http://www.cacr.math.uwaterloo.ca/hac/

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, http://www.unicode.org/

[MS-WSO] Microsoft Corporation, "Windows Protocols System Overview", to be published 2009.

2 Overview

Section 1, "Introduction" primarily describes this Defined Task System Document per se. This section introduces the Tasks that are being documented.

2.1 Summary

Many systems and protocols they implement require digital certificates to operate. Those systems usually do not specify how their certificates are obtained. As long as a valid certificate for enrollment is available the server will use that certificate. Certificates have a certain lifetime and will eventually face expiration. Obtaining or renewing certificates is a burden on the server administrator. Computer certificate autoenrollment takes this burden away from the server administrator by automating **certificate enrollment** and renewal for server certificates.

Autoenrollment uses local configuration and **Group Policy** settings to determine what **certificate enrollment policies (CEPs)** are available. Each CEP specifies a set of **certificate templates** and issuers that can issue certificates based on those templates. Autoenrollment examines local certificate storage and renews or enrolls for new certificates as needed, based on a pre-defined policy, encoded in the form of CEPs.

2.2 List of Tasks

This document describes the following task:

Computer Certificate Autoenrollment: This task is responsible for enrolling and renewing computer certificates automatically.

2.3 Relevant Standards

The task uses the following standard to allow interoperability with other external systems.

Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, as specified in [RFC5280]. This specification is one part of a family of standards for the X.509 Public Key Infrastructure (PKI) for the Internet. This specification profiles the format and semantics of certificates and **certificate revocation lists (CRLs)** for the Internet **public key infrastructure (PKI)**.

3 Background Knowledge and System-Specific Concepts

This section identifies the theoretical and practical information needed to understand this document and the Tasks in this System, and summarizes:

- Background knowledge that is required to understand this document.
- Concepts that are specific to the Tasks in this System.

Familiarity with public key infrastructure (PKI) concepts such as asymmetric and symmetric cryptography, digital certificates, and cryptographic **key exchange** is required for a complete understanding of this specification. In addition, a comprehensive understanding of the [X509] standard is required for a complete understanding of the task and its usage. For a comprehensive introduction to cryptography and PKI concepts, see [CRYPTO]. PKI basics and certificate concepts are as specified in [X509] and [RFC5280].

In order to understand automatic certificate enrollment, it is required to understand certificate enrollment in general as described in this section. At the very abstract level and as illustrated in the following diagram, the administrator enters a policy as a machine-readable certificate enrollment policy (CEP) stored in a policy server.

The CEP is made available from the policy server to certificate enrollment clients, which consume this CEP to determine which certificates the client is supposed to have and which issuers are available to provide those certificates. Clients then create certificate requests and submit them to issuers that issue certificates back to the clients.

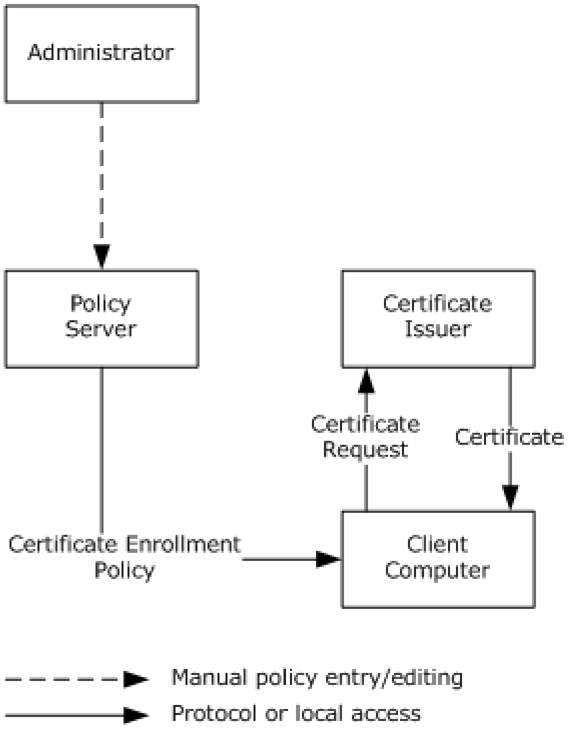

Figure 1: Certificate enrollment

There are two protocol stacks that implement certificate enrollment. The first stack (see the following figure) uses **WCCE** for certificate requests. It uses the **LDAP** profile specified in [MS-

ADTS] section 3.1.1.3 to obtain a CEP from a **domain controller (DC)**. Finally, the CEP is expressed via certificate templates that are data structures specified in [MS-CRTD] and **certification authority (CA)** information that is specified in [MS-WCCE] section 2.2.2.9.2.

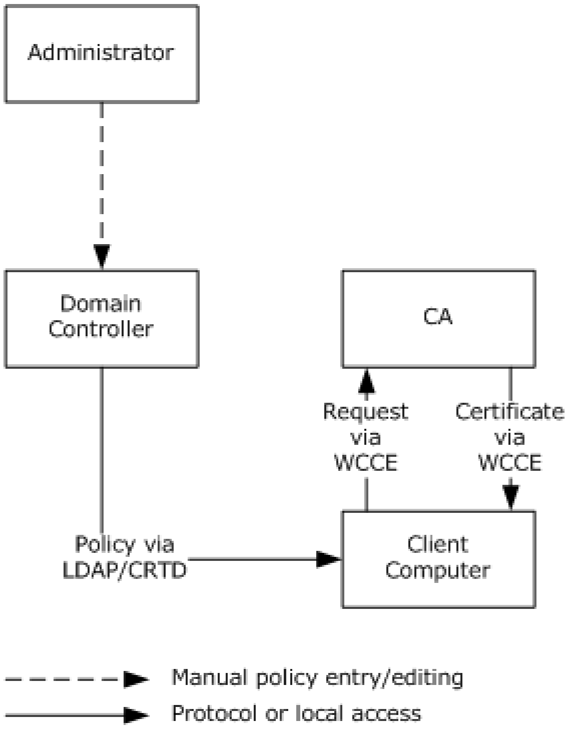

Figure 2: Certificate requests using WCCE and Active Directory/CRTD

The second stack of certificate enrollment protocols (as shown in the following figure) uses the **WSTEP** protocol for certificate requests. It uses **XCEP** to retrieve the CEP.

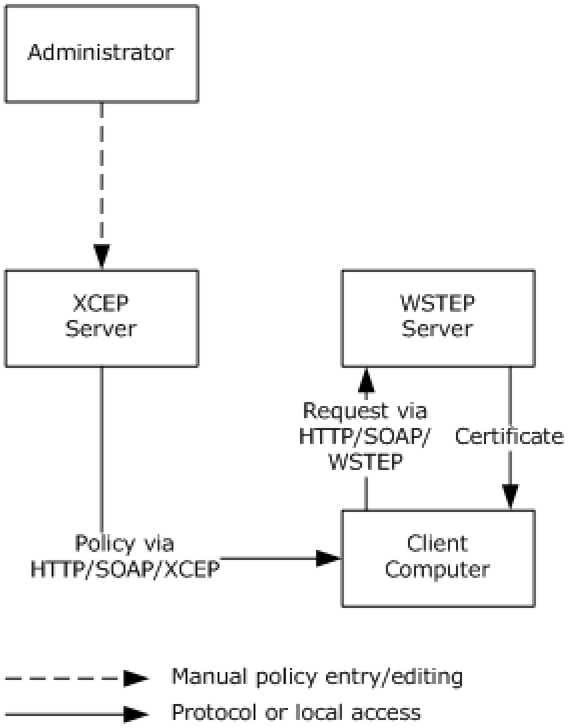

Figure 3: Certificate requests using WSTEP and XCEP

As illustrated in the following figure, it is possible to use both protocol stacks to enroll for certificates based on the same company policy.

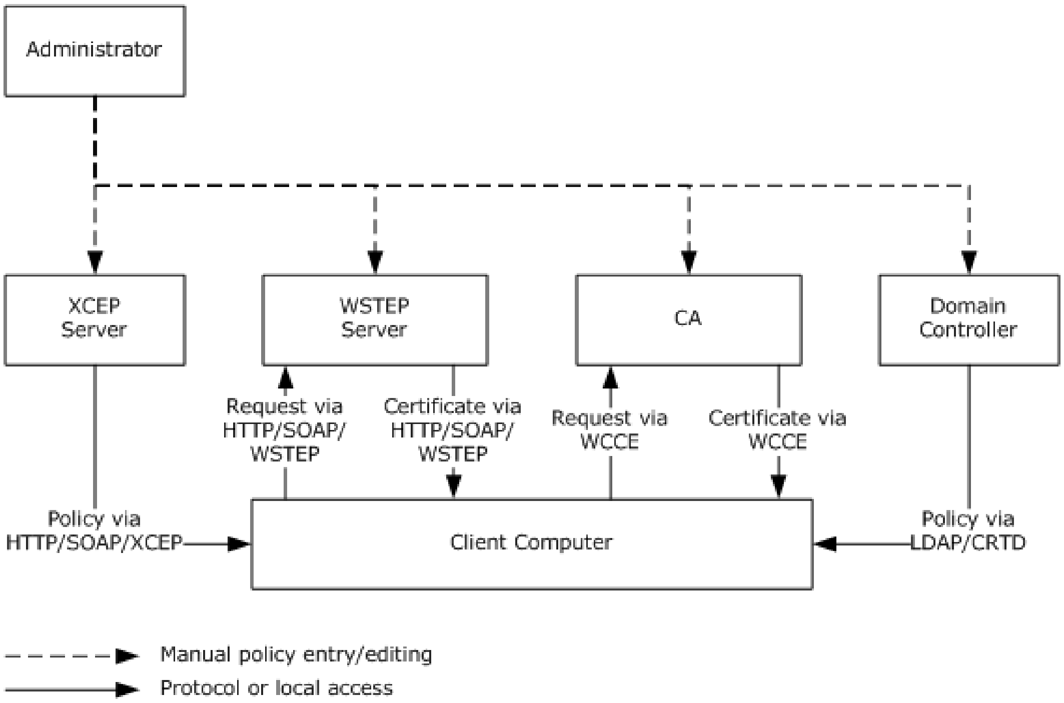

Figure 4: Certificate enrollment with WCCE/LDAP and WSTEP/LDAP

A client computer starts by discovering a policy server. With WCCE/CRTD, the policy server is always a domain controller, discovered as specified in [MS-ADTS] section 7.3. For the use of XCEP/WSTEP, the Web service address has to be configured out of band (for example, manually or by Group Policy).

Certificate enrollment clients can use Group Policy, specifically the **GPREG** protocol, to obtain **policy server end points** that were configured by the administrator in the enterprise. Clients can also use a local configuration store that contains policy server end points specific to a particular client. The following figure illustrates this concept.

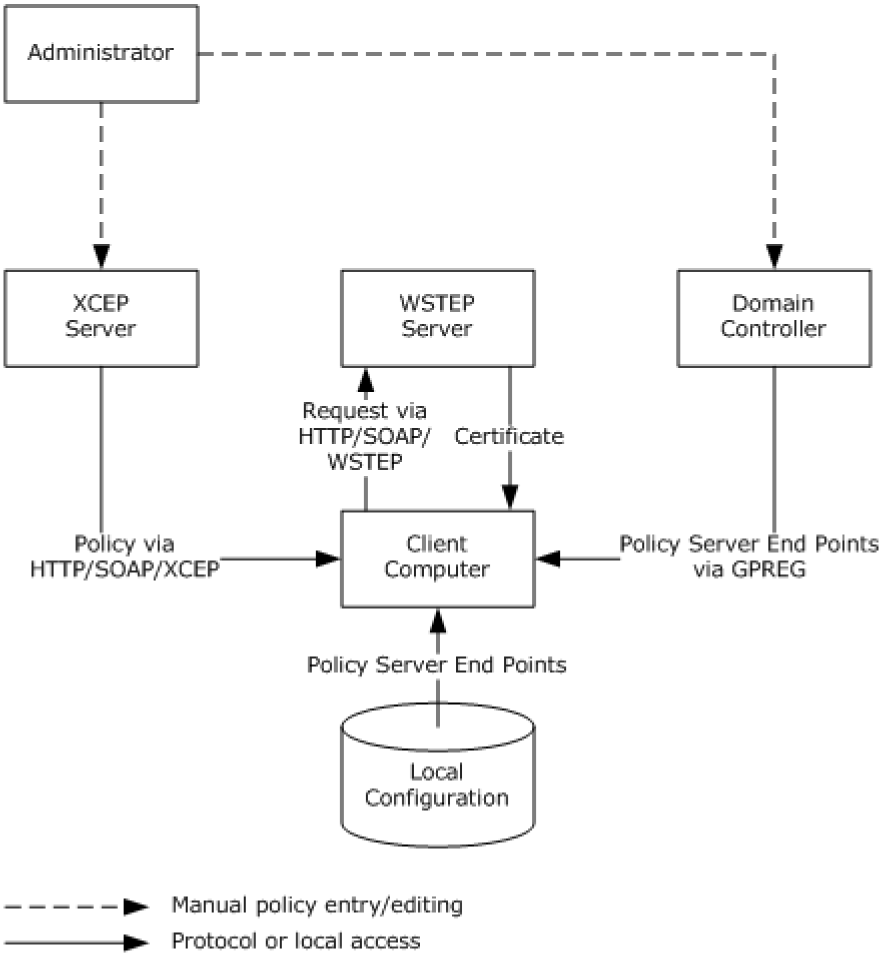

Figure 5: Certificate enrollment using Group Policy

The following diagram shows an example of one possible deployment. In this case, the client computer is a member of some domain where a PKI administrator has configured a CEP by defining some templates and installing an enterprise CA, XCEP server, and WSTEP server. The client computer discovers available CEP servers through Group Policy. Also, the administrator of the client computer itself needs to obtain a certificate for this computer from a third party so the computer can be configured with the policy server end point of the third party server. The client computer can now request certificates based on both policies.

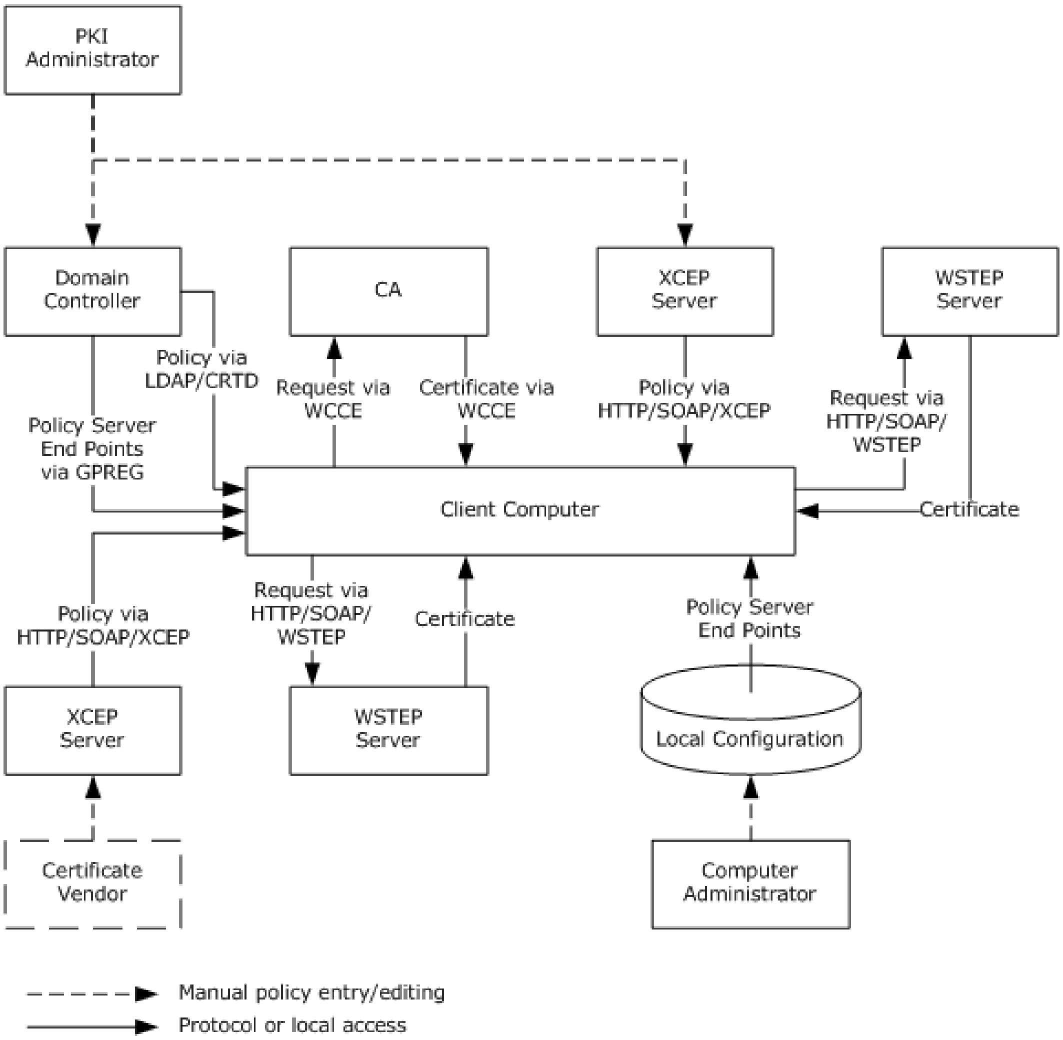

Figure 6: Domain member client requesting certificate enrollment through a Group Policy deployment

Considering that any client can be configured to work with multiple CEPs that have multiple policy server end points, can define multiple certificate templates, and are used by multiple issuers, it is clear that enrolling for certificates manually can be a difficult task. The job of autoenrollment is to traverse all of the CEPs and enroll for certificates as needed.

4 Computer Certificate Autoenrollment Task

This section describes the Computer Certificate Autoenrollment Task. Any system using digital certificates needs to be provisioned with those certificates. This provisioning can be manual, using custom software. Autoenrollment achieves provisioning driven by policy and performs as automatically as possible. It is therefore preferred in an environment where certificate provisioning policy is established at a central source and where there are enough computers to make the cost of manual enrollment prohibitive.

4.1 Task Overview

4.1.1 Task Purpose

Protocols that make use of certificates usually do not specify how certificates are obtained so the job of certificate enrollment and renewal falls to the administrator. The efforts to obtain certificates manually can be great. This task minimizes the administrative burden by automatically enrolling for certificates and renewing them as they get close to their expiration, based on a machine-readable CEP. Certificates and their associated private keys that result from this task are stored in the implementation-specific local storage (see section 4.2.1.1). Server applications running on the computer can then obtain and use the certificates and private keys in their protocol operations.

4.1.2 Task Applicability

This task is appropriate in environments where the work of provisioning certificates is enough of a burden to warrant automation. Also, the task's environment includes servers responsible for certificate issuance that implement protocols listed in section 4.2.2.2.

4.1.3 Task Use Cases

4.1.3.1 Stakeholders and Interests Summary

PKI Administrators: PKI Administrators are responsible for implementing the company's policy by defining CEPs and setting up servers that provide certificates to clients.

Computers: Computers are end entities for certificates that are enrolled by this task. Computers are responsible for persisting certificates and their associated keys and providing them to the Server Applications stakeholders.

Server Applications: Applications that are executing on Computers stakeholders. These applications use certificates and their associated keys.

Server Administrators: Server Administrators are responsible for deploying servers that need certificates for their operations and configuring them to use specific CEPs.

4.1.3.2 Supporting Actors and Task Interests Summary

Issuer: A WCCE or WSTEP server that issues certificates based on requests submitted by this task.

Policy Server: A domain controller (as documented in [MS-ADSO]) or an XCEP server that provides CEP to this task.

Group Policy Client (GP Client):A client (as documented in [MS-GPSO]) that retrieves domain policy specifying autoenrollment options and available CEP servers.

4.1.3.3 Use Case Diagrams

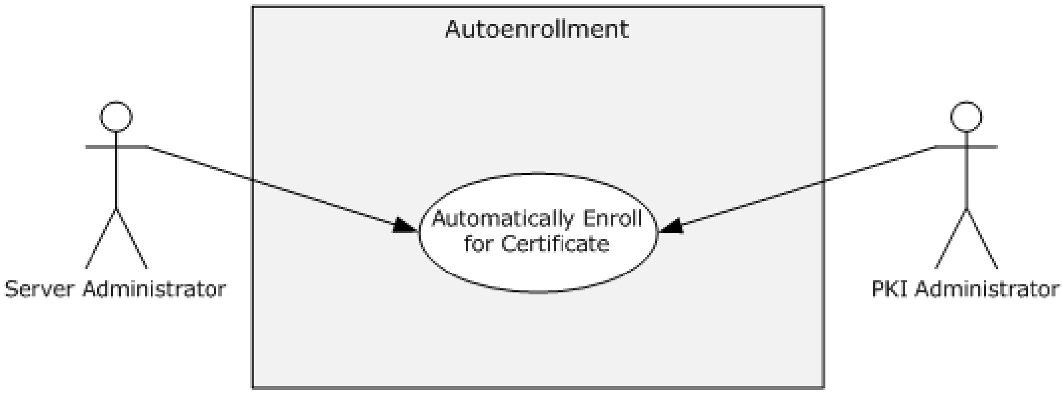

Figure 7: Computer Certificate Autoenrollment use case diagram

4.1.3.4 Use Case - Automatically Enroll and Renew Certificates

Goal: To automatically enroll and renew certificates for a computer based on the administrator-defined CEPs.

Context of Use: Automatic certificate enrollment is done occasionally or in response to system events and is constrained by a CEP. For this use case the Issuer and CEP server have been deployed and a local configuration has been configured with autoenrollment options and policy server end point information. For the case when the computer is joined to the domain, Group Policy has been configured.

Direct Actor: The Server Administrator is a direct actor of this use case. The Server Administrator enables autoenrollment either by joining the computer to a domain that has autoenrollment enabled and CEPs defined or by configuring the local configuration with some CEPs.

Primary Actor: The PKI Administrator is the primary actor. The PKI Administrator sets up the environment consisting of servers as specified in section 4.1.3.2. That environment allows the Server Administrator to enable autoenrollment for servers managed by the Server Administrator.

Supporting Actors: All supporting actors specified in section 4.1.3.2.

Stakeholders and Interests: All stakeholders specified in section 4.1.3.1.

Precondition: None.

Main Success Scenario:

1. One of the triggers documented in section 4.3.6 initiates execution of this task.
2. Obtain and evaluate execution options.
3. Obtain domain Issuer certificates (domain clients only).
4. Obtain CEPs.
5. Retrieve pending requests.
6. Enroll for new certificates or renew certificates close to expiration.

7. Update local certificate storage with new certificates and delete old certificates if required.

8. Quiesce.

Extension:

Domain Unjoin: If the task parameters indicate domain unjoin, autoenrollment deletes all Issuer certificates obtained in step 3 of the main success scenario from the local storage.

4.2 Task Context

This section describes the relationship between this Task and its environment.

4.2.1 Task Environment

Autoenrollment depends on the following external entities:

- Local certificate storage
- Local private key storage

4.2.1.1 Local Certificate Storage

Autoenrollment requires the computer on which it is executing to provide some implementation-specific persisted local certificate storage that is logically organized into the following groups of certificates:

Computer Certificates: This group includes the end entity certificates for the computer. While autoenrollment manages certificates in this store by storing newly enrolled certificates and deleting other certificates, applications running on the same computers actually use these certificates and private keys associated with them for cryptographic operations.

CA Certificates: This group includes intermediate certification authority (CA) certificates that are used to validate end entity certificates as specified in section 4.4.6.3.

Root Certificates: This group includes the **root certificates** used to validate end entity certificates as specified in section 4.4.6.3.

Key Archival CA Certificates: This group stores the certificates of CAs that are allowed to do key archival. These certificates are used to validate the CA **exchange certificate** as specified in section 4.4.6.4.

4.2.1.2 Local Private Key Storage

Autoenrollment requires that the computer on which it is executing provide some implementation-specific persisted local private key storage where it could store private keys associated with the certificates it is requesting.

4.2.2 Task Relationships

This section specifies the task's relationships with other entities.

4.2.2.1 Black Box Relationship Diagrams

When the computer on which this task is executing is not a **domain member** the following diagram represents the black box relationship for this task.

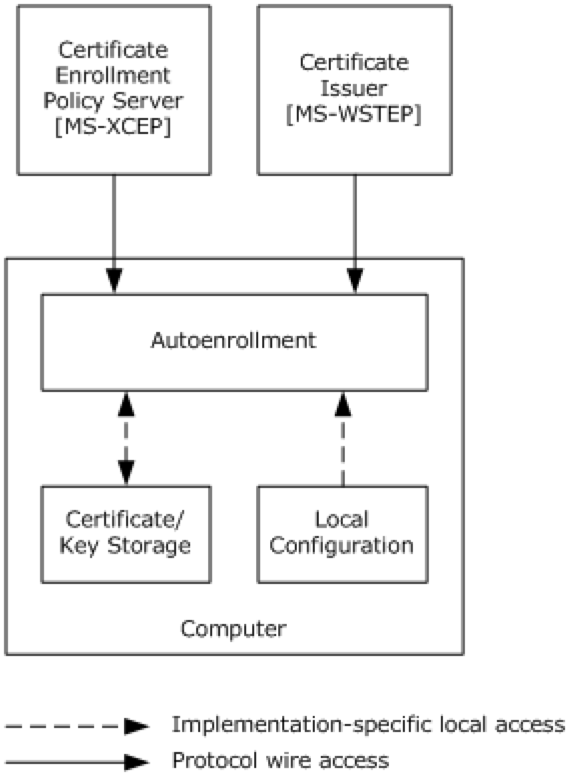
Certificate Enrollment Policy Server [MS-XCEP]
Certificate Issuer [MS-WSTEP]
Autoenrollment
Certificate/ Key Storage
Local Configuration
Computer
Implementation-specific local access
Protocol wire access

Figure 8: Non-domain-joined autoenrollment

When this task is executing on a computer that is a domain member, the following diagram represents the black box relationship for this task. Note that this is a superset of the non-domain-joined case shown previously.

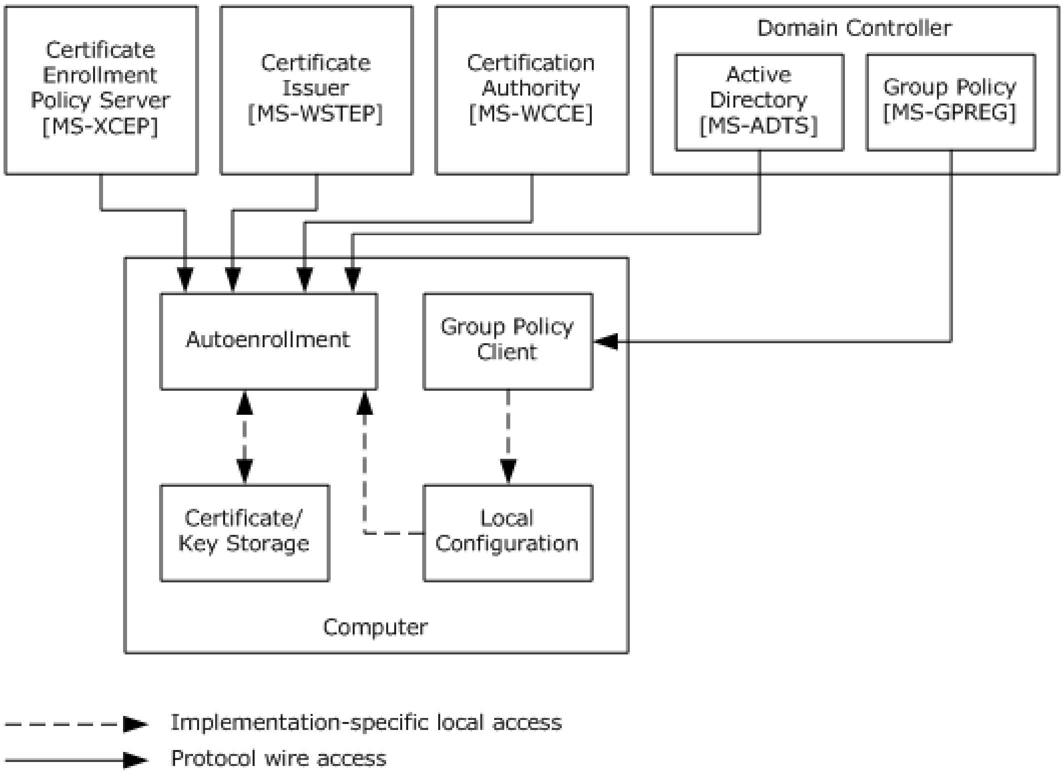
Certificate Enrollment Policy Server [MS-XCEP]
Certificate Issuer [MS-WSTEP]
Certification Authority [MS-WCCE]
Domain Controller
Active Directory [MS-ADTS]
Group Policy [MS-GPREG]
Autoenrollment
Group Policy Client
Certificate/ Key Storage
Local Configuration
Computer
Implementation-specific local access
Protocol wire access

Figure 9: Domain member autoenrollment

4.2.2.2 Task Dependencies

This task depends on the following protocols:

- **Certificate enrollment policy (CEP) protocols**: The following protocols/data structures are used for the CEP that determines which certificates this task enrolls for and which issuers can process requests.
  - X.509 Certificate Enrollment Policy Protocol as defined in [MS-XCEP] with **HTTP** as transport.
  - Certificate template structures defined in [MS-CRTD] and pKIEnrollmentService objects specified in [MS-WCCE] section 2.2.2.9.2 with LDAP profile and extension, as documented in [MS-ADTS] section 3.1.1.3, as transport.
- **Certificate enrollment protocols**: The following protocols allow this task to request certificates from an issuer:
  - Windows Client Certificate Enrollment Protocol (as specified in [MS-WCCE]) with the **Distributed Component Object Model (DCOM)** protocol as transport.
  - WS-Trust Enrollment Extensions [MS-WSTEP] with HTTP as transport.

This task depends on the following systems and defined tasks:

- **Active Directory domains**: When this task executes on the domain member computer, domain controllers in the computer's environment provide important services like authentication, directory services, and Group Policy.

- Active Directory System Overview (described in [MS-ADSO]) provides information about **Active Directory** as a system.

- Domain Integration System Overview (described in [MS-DISO]) specifies tasks that are performed by domain members.

- Windows Authentication Services System (described in [MS-AUTHSO]) specifies authentication tasks performed in the domain environment.

4.2.2.3 Task Influences

Configuration options of this task and information about CEP servers that this task can use are distributed through Group Policy when this task executes on the computer that is a domain member. This task SHOULD rely on a separate Group Policy Client (GP Client), as specified in [MS-GPSO], to retrieve configuration options needed for this task; however, such architectural design is not required and an implementation of this task MAY read required Group Policy data as part of this task processing by using the Registry Extension Encoding protocol, as specified in [MS-GPREG].

4.2.3 Task Assumptions and Preconditions

Autoenrollment makes these assumptions about policies it consumes:

- When autoenrollment processes CEP end point information via Group Policy (as specified in section 4.4.5.3.2) it assumes that:<1>
  - The Group Policy data that initializes EndPoint.Flags (as specified in section 4.3.2.2) does not differ between separate Uniform Resource Identifiers (URIs) that belong to the same policy.
  - The Group Policy data does not contain instructions that identify more than one CEP as the default.
- Autoenrollment assumes that no single CEP defines more than one template that has the same CommonName (as specified in section 4.3.2.1.2.1).<2>

4.2.4 Task Versioning and Capability Negotiation

This task does not have versioning or capability negotiation.

4.3 Task Architecture

4.3.1 Task Architectural Constraints

There are no constraints for this task.

4.3.2 Task Abstract Data Model

This section describes state established, used, and maintained by processing rules of this Task. State could be volatile or persisted. State could pertain to one, some, or all instances of the Task. The Task's state consists of the values of the named data elements (also called state variables) presented in this section. The overall organization of the data elements, with their names, is the Abstract Data Model. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules could depend upon associations established by the structure of its Abstract Data Model, such association can be achieved in other ways. Implementations could depart from this model so long as their external behavior remains consistent with that described in this document.

4.3.2.1 AutoEnrollmentPolicy

The AutoEnrollmentPolicy datum is a structure that contains the options for the Computer Certificate Autoenrollment Task and a set of the CEPs. The abstract data model (ADM) of this task contains only one instance of this structure. The logical representation of the structure is as follows:

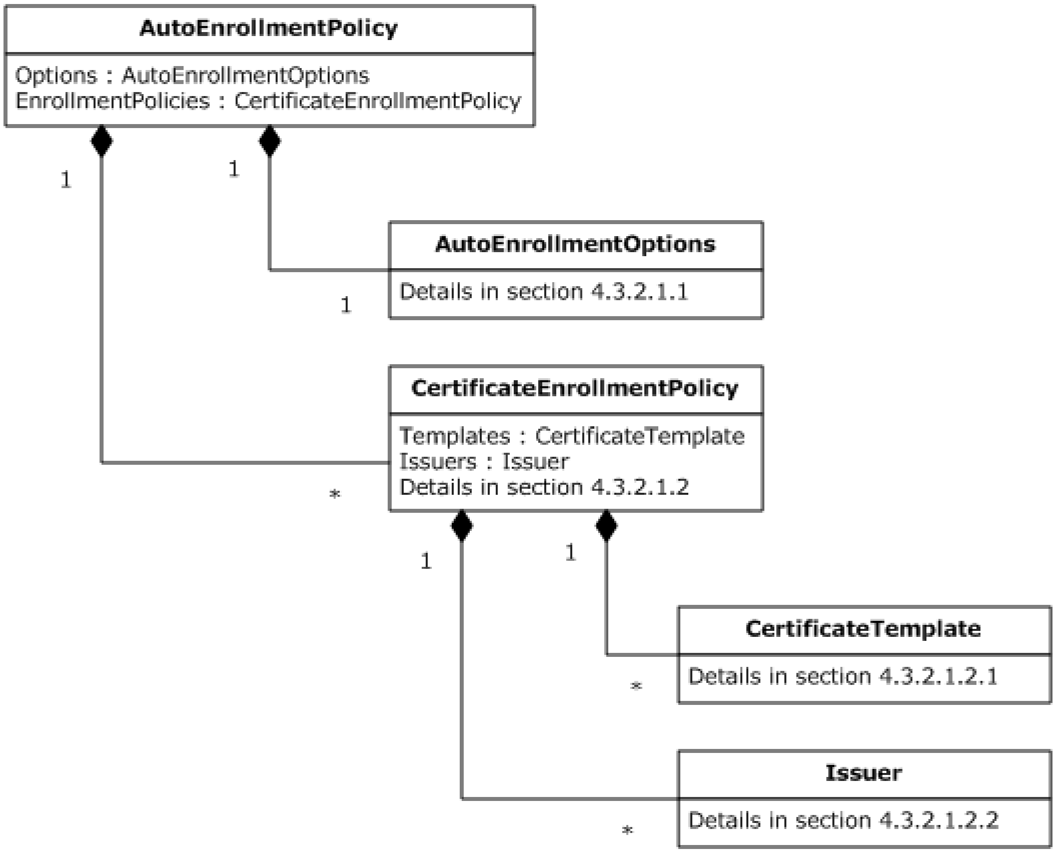
AutoEnrollmentPolicy
Options : AutoEnrollmentOptions
EnrollmentPolicies : CertificateEnrollmentPolicy
1
1
AutoEnrollmentOptions
Details in section 4.3.2.1.1
1
CertificateEnrollmentPolicy
Templates : CertificateTemplate
Issuers : Issuer
Details in section 4.3.2.1.2
*
1
1
CertificateTemplate
Details in section 4.3.2.1.2.1
*
Issuer
Details in section 4.3.2.1.2.2
*

Figure 10: AutoEnrollmentPolicy structure

The individual fields of this structure are specified in the following subsections.

4.3.2.1.1 AutoEnrollmentOptions

The AutoEnrollmentOptions structure specifies options that control autoenrollment behavior and contains the fields described as follows.

AutoEnrollmentOptionFlags: Flags that determine the autoenrollment mode of execution that can be one of the following values:

- AutoEnrollmentOptionFlags.NotDefined - no options have been defined.
- AutoEnrollmentOptionFlags.Disabled - autoenrollment has been turned off (do not execute).
- AutoEnrollmentOptionFlags.Enabled - autoenrollment is enabled.
- A combination of AutoEnrollmentOptionFlags.Enabled and any of the following values:

- AutoEnrollmentOptionFlags.Enroll - enroll and renew certificates based on certificate templates that have been set up for autoenrollment.
- AutoEnrollmentOptionFlags.Manage - renew certificates when the certificate's templates are not set up for autoenrollment.
- AutoEnrollmentOptionFlags.RetrievePending - retrieve pending requests.

4.3.2.1.2 CertificateEnrollmentPolicy

The CertificateEnrollmentPolicy datum provides the information about certificate templates and issuers. As illustrated in the AutoEnrollmentPolicy structure figure, section 4.3.2.1, the AutoEnrollmentPolicy datum has a list of zero or more instances of the CertificateEnrollmentPolicy data. The CertificateEnrollmentPolicy defines these fields:

EnrollmentProtocol: Protocol used for the certificate enrollment. Valid values are WCCE (specified in [MS-WCCE]) and WSTEP (specified in [MS-WSTEP]).

PolicyId: The ID of the CEP. The PolicyId allows the grouping of policy server end points that serve the same CEP together. It is also used to record which CEP contained a certificate template on which a particular certificate was based.

IsDefault: Boolean flag used to identify default CEP. A default CEP is used to renew certificates for which the original PolicyId is unknown.

IsUntrustedIssuerAllowed: When set to true, autoenrollment does not require an issuer signing certificates, or end entity certificates that it enrolls for, to chain up to a trusted root.

Issuers: A list of zero or more instances of Issuer data specifying available issuers for this CEP.

Templates: A list of zero or more instances of CertificateTemplate data specifying available certificate templates for this CEP.

The CertificateTemplate and Issuer are also structures and are described in the following subsections.

4.3.2.1.2.1 CertificateTemplate

The CertificateTemplate datum is a selection of the information in a certificate template as defined in [MS-CRTD]. The following information is needed for the task processing rules:

OID: Object Identifier (OID) that identifies a template.

CommonName: The name of a template.

SchemaVersion: The schema version of a template.

MajorVersion: The major version of the template.

OverlapPeriod: The time before a certificate expires, during which autoenrollment sends a renewal certificate request.

AutoEnrollmentEnabled: Set to true if this template is allowed for autoenrollment.

SupersededTemplates: A list of names (CertificateTemplate.CommonName) that identifies templates that were superseded by this template.

SubjectNameFlags: Flags that control the format of the subject name and alternative subject name for the certificate based on this template. Possible values are combinations of the following:

- EnrolleeSuppliesSubject - the end entity is responsible for supplying the subject name for the certificate.
- EnrolleeSuppliesSubjectAltName - the end entity is responsible for supplying the alternative subject name for the certificate.
- OldCertSuppliesSubjectAndAltName - autoenrollment can reuse subject and alternative subject name information from an existing certificate to submit a renewal request for that certificate.

EnrollmentFlags: Flags that determine autoenrollment behavior when enrolling for this template. Possible values are combinations of the following:

- PreviousApprovalValidateReenrollment - indicates that for a renewal request based on a certificate template that requires a **registration authority (RA)** signature, the RA signature requirement can be fulfilled by a signature with a valid existing certificate based on the same template.
- HumanInteractionRequired - indicates that user consent or input is required before enrolling for a certificate based on this certificate template and therefore is not appropriate for automatic enrollment.
- RemoveInvalidCertificateFromComputerStore - instructs autoenrollment to delete certificates based on this template when autoenrollment enrolls for a certificate that is more recent.

GeneralFlags: Other template flags. Possible values are combinations of the following:

- IsMachineType -indicates that this template is for certificates where the end entity is a computer.
- IsCA - indicates that this template is for certificates where the end entity is a CA.
- IsCrossCA - indicates that this template is for the CA cross-certification.

PrivateKeyAttributes.Archive: A Boolean flag indicating if autoenrollment ought to send the private key for archival to the issuer.

RARequirements.*: Data that describes additional signature requirements for the certificate request as follows:

- RARequirements.SignatureCount - the number of additional signatures required for the certificate request.
- RARequirements.EKUs - a list of extended key usage extension OIDs that are required to be part of at least one certificate that was used for additional signatures on the certificate request.
- RARequirements.IssuancePolicies - a list of issuance policy OIDs that are required to be part of at least one certificate that was used for additional signatures on the certificates request.

4.3.2.1.2.2 Issuer

The Issuer datum is the information about the server that can fulfill a certificate request. The fields of the Issuer structure are as follows:

Name: Name of the issuer.

EndPoint: The DNS name of the CA or the HTTP URI of the WSTEP server (if the value of the CertificateEnrollmentPolicy is the WCCE, a DNS name is used; otherwise for the WSTEP protocol an HTTP URI is used).

Templates: A list of template names (CertificateTemplate.CommonName) supported by this issuer.

IsRenewalOnly: Indicates that an issuer supports renewal requests only.

Priority: Relative priority of the issuer in respect to other issuers that belong to the same CertificateEnrollmentPolicy instance. Issuers with higher priority values are chosen first during enrollment.

Certificate: Certificate for the issuer's signing key.

Authentication: Specifies authentication used to communicate with an issuer. This can be one of the following values:

- Anonymous - anonymous authentication
- Kerberos - transport Kerberos authentication
- Password - message username and password authentication
- Certificate - message X.509 certificate authentication

4.3.2.2 EndPointsInformation

The EndPointsInformation datum is a structure that contains information about the policy server end points used to obtain information about CEPs. This structure is initialized from Group Policy and local configuration information to facilitate communication with CEP servers.

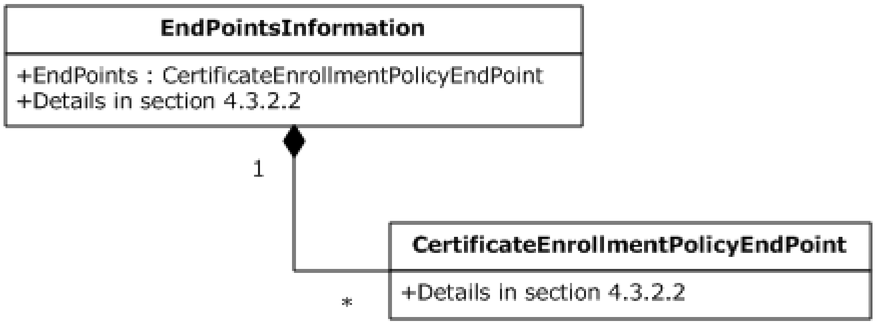
EndPointsInformation
+EndPoints : CertificateEnrollmentPolicyEndPoint
+Details in section 4.3.2.2
1
*
CertificateEnrollmentPolicyEndPoint
+Details in section 4.3.2.2

Figure 11: EndPointsInformation structure

The EndPointsInformation defines the following fields:

DefaultPolicyId: Contains the ID of the default CEP.

IsGroupPolicyConfigurationEnabled: Set to true if end points defined in Group Policy ought to be used.

IsLocalConfigurationEnabled: Set to true if end points configured locally ought to be used.

The CertificateEnrollmentPolicyEndPoint datum is a single record containing the information that describes a point of access for some specific CEP. The EndPointInformation datum contains a table of these records. The columns for this table are defined as follows:

EndPoint.URI: The URI used to access this policy server end point.

EndPoint.PolicyId: The ID of the CEP.

EndPoint.Flags: Flags describing options for this CEP. Possible values are combinations of the following values:

- AutoEnrollmentEnabled: Indicates that autoenrollment is allowed for this CEP.
- AllowUntrustedIssuer: Same semantics as CertificateEnrollmentPolicy.IsUntrustedIssuerAllowed (see section 4.3.2.1.2).

EndPoint.Authentication: The authentication type to be used when accessing the end point. The value can be one of the following:

- Anonymous - anonymous authentication
- Kerberos - transport Kerberos authentication
- Password - message username and password authentication
- Certificate - message X.509 certificate authentication

Note that autoenrollment uses these values only to sort different policy server end points that belong to the same CEP. The actual authentication considerations are specified by the protocols that autoenrollment uses for certificate enrollment.

EndPoint.Cost: Relative cost to access the particular URI. Both units of this field and their values are determined by administrators.

4.3.2.3 Local Information

The Local Information data contains information about the local state of the system that helps autoenrollment determine certificates to be enrolled, existing certificates to be renewed, and pending requests to be retrieved.

4.3.2.3.1 Certificates

The Certificates data contains lists of certificates used by autoenrollment. The Certificates data defines these lists:

Certificates.CurrentCertificates: A list of the current end entity certificates.

Certificates.ToBeAdded: A list of certificates to be added to the local certificate storage.

Certificates.ToBeDeleted: A list of certificates to be deleted from the local certificate storage.

Certificates.Roots: A list of certificates that holds certificates from the Root Certificates group specified in section 4.2.1.1.

Certificates.CAs: A list of certificates that holds certificates from the CA Certificates group specified in section 4.2.1.1.

Certificates.KeyArchivalCAs: A list of certificates that holds certificates from the Key Archival CA Certificates group specified in section 4.2.1.1.

4.3.2.3.2 PendingRequests

The PendingRequests datum is a table that contains information about pending certificate requests. Autoenrollment uses this table to retrieve issued certificates for pending requests. The table has the following columns:

PendingRequests.Protocol: Protocol that was used to submit the request. Possible values are "WCCE" or "WSTEP".

PendingRequests.Issuer: Information about the issuer that the request was submitted to, typically the value of the Issuer.EndPoint datum.

PendingRequests.RequestId: Request ID that was assigned to the request by the issuer.

PendingRequests.RequestTime: Time when request was originally submitted.

PendingRequests.PolicyId: CEP ID associated with the certificate template referenced in the request.

PendingRequests.TemplateName: Name of the certificate template referenced in the request.

PendingRequests.TemplateOID: OID of the certificate template referenced in the request.

PendingRequests.MajorVersion: Version of the certificate template referenced in the request.

Autoenrollment SHOULD<3> persist this table between different executions in an implementation specific way.

4.3.2.3.3 ProcessedEnrollments

The ProcessedEnrollments datum is a table that contains information about certificates that autoenrollment has successfully enrolled. It has the following columns:

ProcessedEnrollments.PolicyId: ID of the CEP from which the certificate has been enrolled.

ProcessedEnrollments.CertificateId: Implementation specific identifier of the certificate that autoenrollment has enrolled. The easiest to implement identifier would be the certificate itself.

Autoenrollment SHOULD<4> persist this table between different executions in an implementation specific way.

4.3.3 Task Abstract Parameters

This section describes data passed to an instance of this task at the time it is invoked or triggered. The parameters consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Parameter. It is intended to facilitate the reader's conceptual understanding of the specification. While a task's processing rules could depend upon associations established by the structure of its Abstract Parameters, such association can be achieved in other ways. Implementations could depart from this abstraction so long as their external behavior remains consistent with that described in this document.

The following are the parameters for this task:

IsUnjoiningDomain: Boolean indicating if the computer is being removed from the domain as described in [MS-DISO] section 9. If not specified, the parameter defaults to false.

4.3.4 Task Abstract Results

This section describes data returned by an instance of this task to its caller. The results consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Result. It is intended to facilitate the reader's conceptual understanding of the specification. While a task's processing rules could depend upon associations established by the structure of its Abstract Results, such association can be achieved in other ways. Implementations could depart from this abstraction so long as their external behavior remains consistent with that described in this document.

The Computer Certificate Autoenrollment Task never returns any data to its callers. Instead the effects of this task are realized through local certificate storage that this task modifies.

4.3.5 White-Box Relationships

The Computer Certificate Autoenrollment Task is a procedure that does not have separate internal components interacting with each other in a non-sequential manner. Diagrams in this section illustrate how external entities influence the task's ADM.

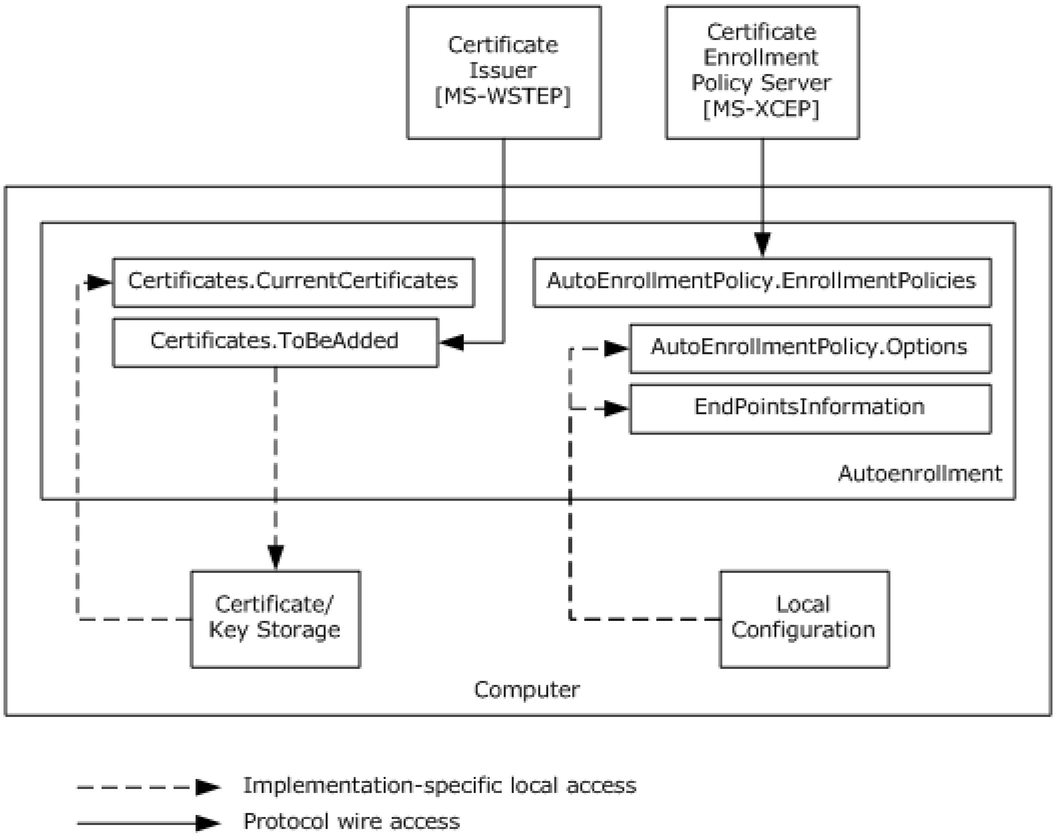
Certificate Issuer [MS-WSTEP]
Certificate Enrollment Policy Server [MS-XCEP]
Certificates.CurrentCertificates
Certificates.ToBeAdded
AutoEnrollmentPolicy.EnrollmentPolicies
AutoEnrollmentPolicy.Options
EndPointsInformation
Autoenrollment
Certificate/ Key Storage
Local Configuration
Computer
Implementation-specific local access
Protocol wire access

Figure 12: Autoenrollment white-box diagram for computers not joined to a domain

As shown in the preceding figure, autoenrollment reads its configuration options and policy server end point information from the local configuration. Based on that, it retrieves a CEP from some policy server. Depending on the available CEP and certificates currently present on the system, autoenrollment submits requests and persists newly enrolled or renewed certificates in the local certificate storage.

The task's ADM is not shared with other systems, tasks, or instances of the Computer Certificate Autoenrollment Task. The local certificate and key storage can be read or modified by other systems in an implementation-specific way, but autoenrollment makes no assumptions on how or even if this happens. Local configuration is modified by the computer administrator through the use of administration tools.

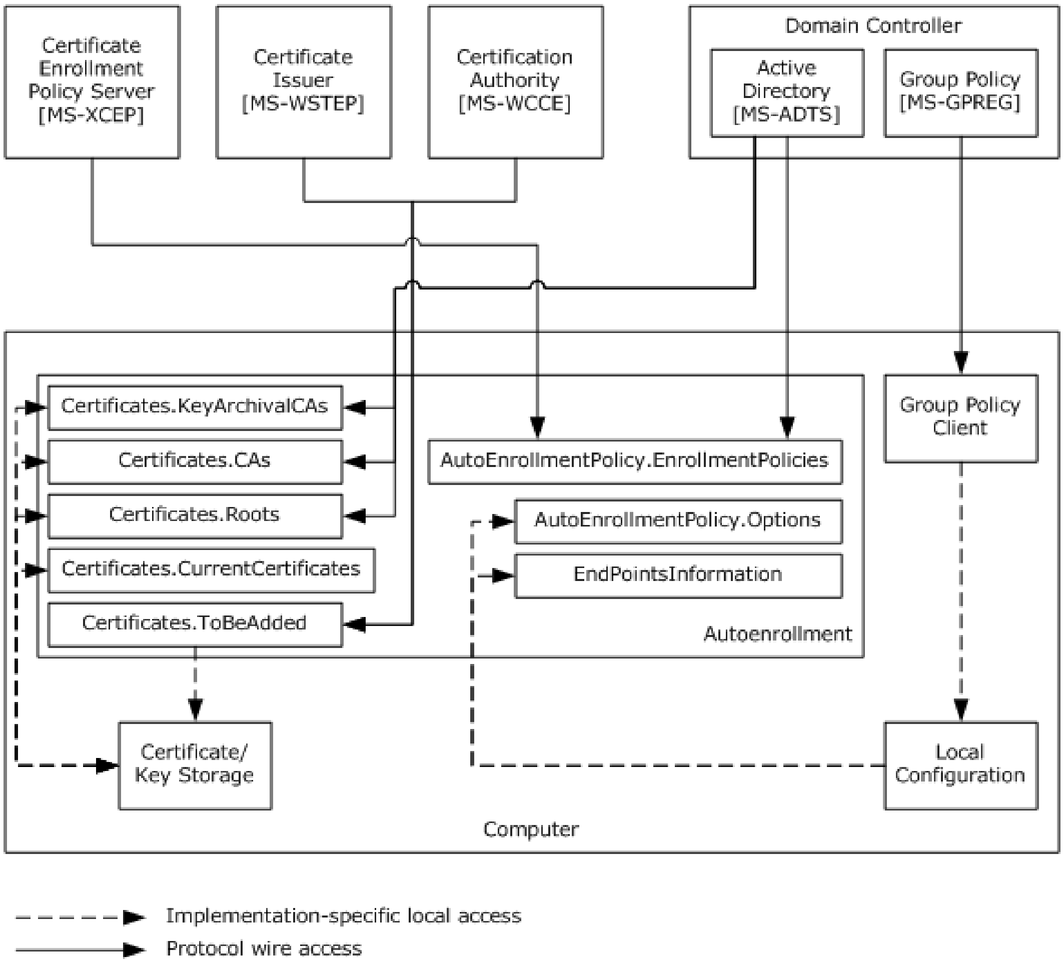
Certificate Enrollment Policy Server [MS-XCEP]
Certificate Issuer [MS-WSTEP]
Certification Authority [MS-WCCE]
Domain Controller
Active Directory [MS-ADTS]
Group Policy [MS-GPREG]
Certificates.KeyArchivalCAs
Certificates.CAs
Certificates.Roots
Certificates.CurrentCertificates
Certificates.ToBeAdded
AutoEnrollmentPolicy.EnrollmentPolicies
AutoEnrollmentPolicy.Options
EndPointsInformation
Autoenrollment
Group Policy Client
Certificate/ Key Storage
Local Configuration
Computer
Implementation-specific local access
Protocol wire access

Figure 13: Autoenrollment white-box diagram for computer joined to a domain

Interactions illustrated in the preceding figure are a superset of those illustrated in the diagram for computers not joined to a domain. In this case an optional Group Policy client can also modify the local configuration with the settings from the domain's Group Policy (as noted in section 4.2.2.3, an autoenrollment implementation can choose to read Group Policy settings directly). Also, there are issuer certificates that can exist in Active Directory that autoenrollment persists in the local certificate storage.

4.3.6 Task Events

4.3.6.1 Task Timers

Autoenrollment SHOULD <5> have a timer that allows it to periodically execute to keep the local certificate storage current. Timers' details are specified in section 4.4.3.1.

4.3.6.2 Task Non-Timer Events

The task's non-timer events are as follows:

- Autoenrollment MUST execute after computer startup.
- Autoenrollment MAY<6> execute when new Group Policy is applied.
- This task is explicitly called by Join a Domain Using a Predefined Account, Join a Domain by Creating an Account via SAM, Join a Domain by Creating an Account via LDAP, and the Remove a Domain Member tasks as described in [MS-DISO] in sections 6, 7, 8 and 9, respectively.

Events details are specified in section 4.4.3.2.

4.3.7 Task Architecture and Communication

Autoenrollment follows a straightforward procedure when executing and does not have multiple internal components that execute concurrently. To accurately represent this design, all diagrams in this section depict autoenrollment as a single box.

As shown in the following diagram, for the case where the computer on which autoenrollment executes is not a joined to a domain, autoenrollment access two local data stores (local certificate and key storage documented in section 4.2.1 and local configuration) and communicates with two types of external entities (XCEP and WSTEP servers) to get a CEP and enroll for certificates. Local configuration is modified by the server administrator by using a system-specific tool. Remote communications are synchronous.

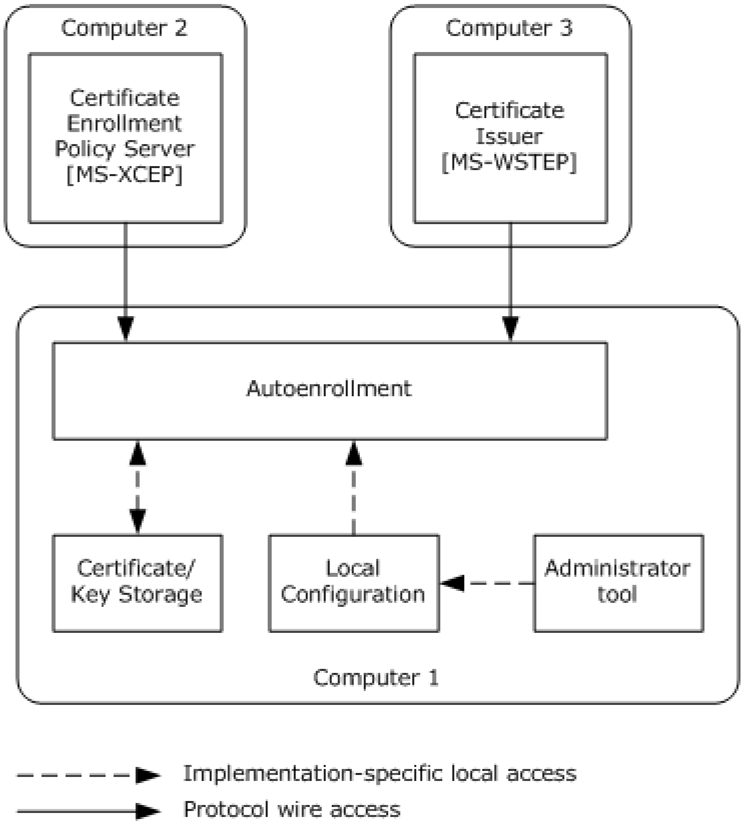

Figure 14: Autoenrollment executing on a computer that is not joined to a domain

The following diagram documents the case where autoenrollment executes on a computer that is joined to a domain. In this case, there is an additional internal (local) entity, Group Policy Client (GP Client), which modifies the local configuration with settings configured by the domain administrator. Also, there is an external entity, domain controller, which provides Group Policy settings as well as certificate template and CA information. Finally, another external entity is a CA (WCCE server) from which autoenrollment can request certificates.

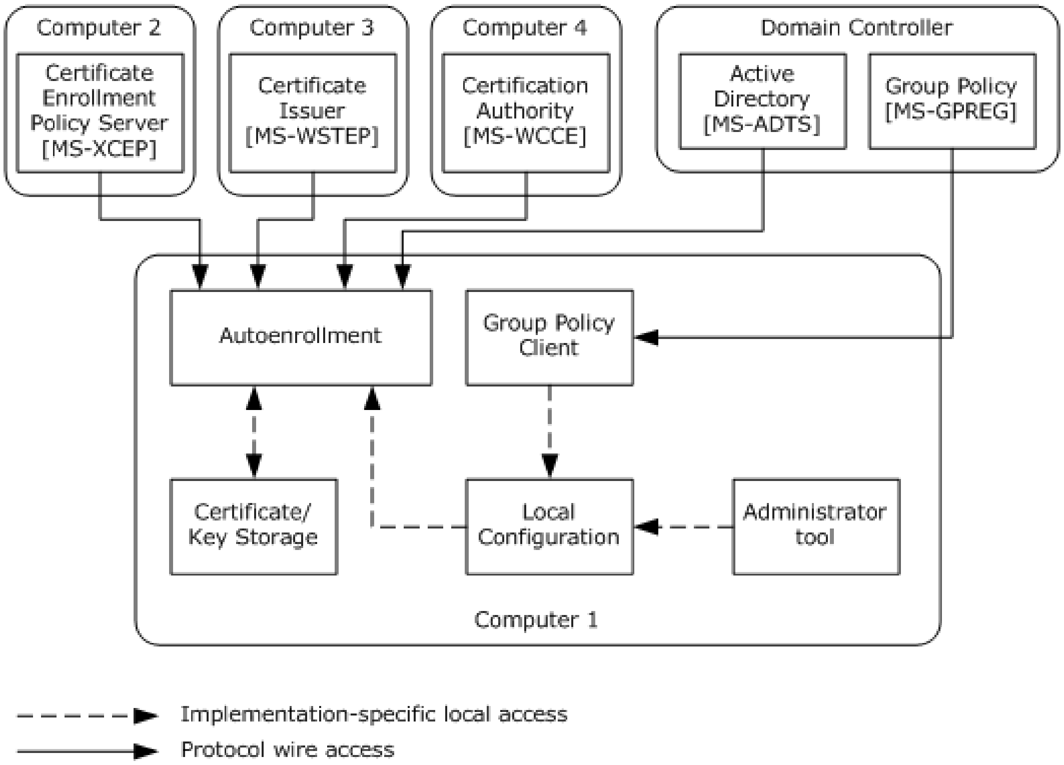

Figure 15: Autoenrollment executing on a computer that is joined to a domain

4.3.8 Task Processing Rules

This section outlines the task's high-level steps. See section 4.4.5 for the details on each step.

Abstract Parameters: See section 4.3.3.

Preconditions: None.

Main Success Scenario:

1. Initialize AutoEnrollmentPolicy.AutoEnrollmentOptions (section 4.4.5.1).
2. Update issuer stores (section 4.4.5.2).
3. Initialize AutoenrollmentPolicy.EnrollmentPolicies (section 4.4.5.3).
4. Initialize local information (section 4.4.5.4).

5. Retrieve pending requests (section 4.4.5.5).

6. Autoenroll based on certificate templates. For each CertificateEnrollmentPolicy instance in AutoenrollmentPolicy.EnrollmentPolicies (section 4.4.5.6):

7. Determine which templates to enroll for.

8. Enroll or renew based on the identified templates.

9. Renew manually enrolled certificates (section 4.4.5.7).

10. Update local storage (section 4.4.5.8).

11. Terminate autoenrollment (section 4.4.5.9).

Extensions:

If the IsUnjoiningDomain parameter to the task (see section 4.3.3) is true, remove all issuer certificates from local certificate storage (section 4.4.5.10).

4.3.9 Task Failure Scenarios

Autoenrollment is a task meant to execute occasionally in anticipation of any real deadline. The most severe failure causes it to terminate for this time, with the expectation that it will run again at the next occasion. If it does execute, it consists of nested loops and each of those might terminate by continuing with the next item in the loop. The different types of failures are as follows:

- Fatal errors terminate this execution of the task. The fatal errors are:
  - Failure to initialize the AutoEnrollmentPolicy.AutoEnrollmentOptions datum.
  - Failure to read or write to local certificate storage.
- Failures initializing an instance of the CertificateEnrollmentPolicy datum. Any request covered by that datum will be ignored for this execution.
- Failures requesting a certificate based on a particular certificate template do not affect processing of other certificate templates associated with the same CEP.

For details on when failures occur and how they affect the flow of the Computer Certificate Autoenrollment Task, see section 4.4.5.

4.4 Task Details

This section contains the details that complete the descriptions in earlier sections of the document. These are needed to understand and implement this Task.

4.4.1 Task Precondition Details

Not Applicable.

4.4.2 Task Initialization of External Entities

Not Applicable.

4.4.3 Task Event Details

4.4.3.1 Task Timer Details

Autoenrollment SHOULD<7> execute at least twice a day.

4.4.3.2 Task Non-Timer Event Details

Execution after computer startup: In an implementation specific way, the Computer Certificate Autoenrollment Task SHOULD be triggered after the computer has started. Depending on the operating system capabilities, autoenrollment can subscribe to some type of notification or be added to some type of list of tasks that execute at startup.

Execution after Group Policy updates: If the system on which autoenrollment executes has a GP Client that executes periodically, autoenrollment MAY<8> execute when new Group Policy is applied.

Execution after Domain Join Task: This task is triggered by the tasks described in [MS-DISO] sections 6, 7, 8, and 9.. When the computer is removed from the domain, autoenrollment executes as specified in the Main Success Scenario in section 4.3.8. When the computer is being disjoined from the domain, the IsUnjoiningDomain parameter is set to true and autoenrollment executes as specified in Extensions in section 4.3.8.

4.4.4 Task Architectural Details

This section provides the following examples of task execution:

- Autoenrollment in the standalone environment with XCEP/WSTEP protocols
- Autoenrollment in the domain environment with CRTD/WCCE protocols

These are representations of typical message flows in common task usage scenarios.

4.4.4.1 Autoenrollment in the Standalone Environment with XCEP/WSTEP Protocols

The goal of this example (as detailed in section 4.1.3.4) is to enroll for a certificate in an automated way. In this example, the computer on which the task executes is not joined to any domain. It has also been configured with policy server end point information in its local configuration. After a trigger, the following messages will be exchanged as shown in the following figure.

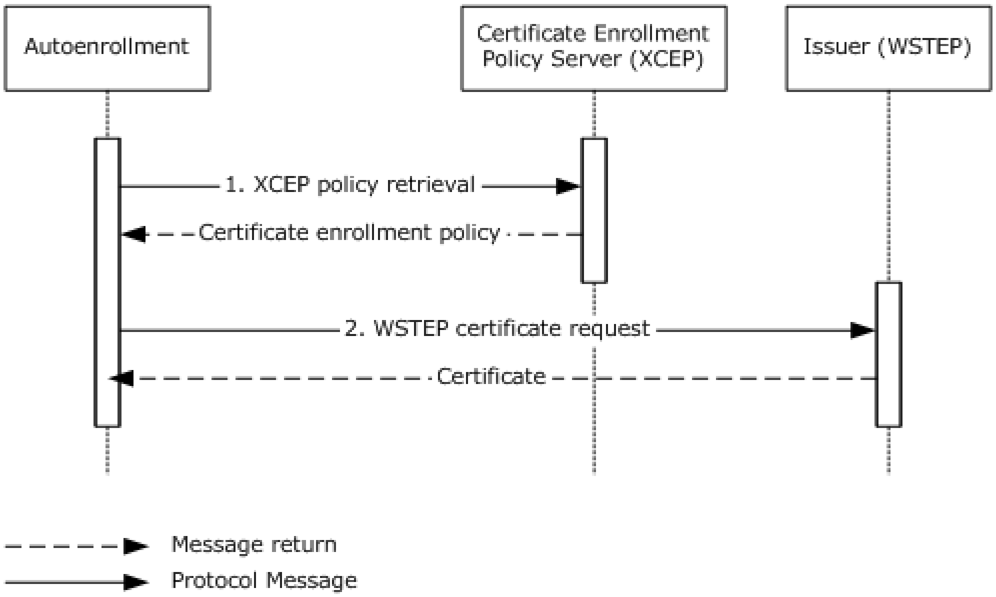
Autoenrollment
Certificate Enrollment Policy Server (XCEP)
Issuer (WSTEP)
1. XCEP policy retrieval
Certificate enrollment policy
2. WSTEP certificate request
Certificate
Message return
Protocol Message

Figure 16: Message exchange on a computer not joined to a domain

1. After autoenrollment has processed the local configuration, it sends a GetPolicies message to an XCEP server and receives a GetPoliciesResponse message in return.
2. Based on the CEP received from the XCEP server and the state of the local certificate storage, the task sends a certificate request using the WSTEP protocol to the issuer and receives a newly issued certificate that is then stored in the local certificate storage.

4.4.4.2 Autoenrollment in the Domain environment with CRTD/WCCE Protocols

The goal of this example (as detailed in section 4.1.3.4) is to enroll for a certificate in an automated way. In this example, the computer on which the task executes is joined to a domain. Also, a CA (WCCE server) exists in the same domain. Finally, a certificate template and Group Policy enabling autoenrollment have been configured by the domain administrator. After the GP Client caches Group Policy information and after a trigger, the messages illustrated in the following figure will be exchanged.

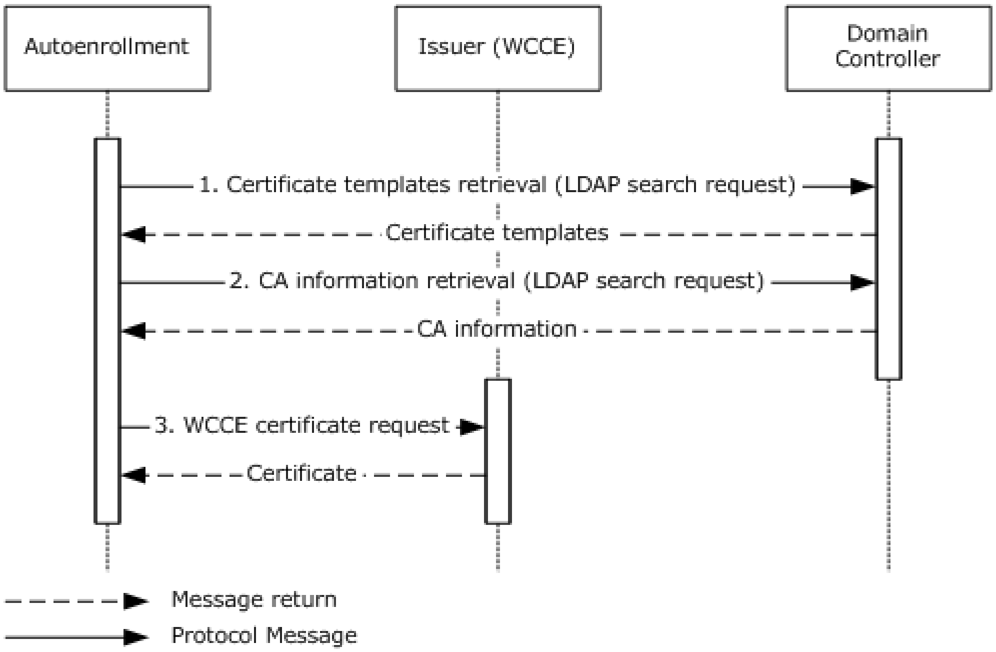
Autoenrollment
Issuer (WCCE)
Domain Controller
1. Certificate templates retrieval (LDAP search request)
Certificate templates
2. CA information retrieval (LDAP search request)
CA information
3. WCCE certificate request
Certificate
Message return
Protocol Message

Figure 17: Message exchange on a computer joined to a domain

1. After autoenrollment has processed local configuration, it sends an LDAP search request to obtain certificate templates and CA information from Active Directory.
2. Autoenrollment then sends LDAP search requests to obtain CA information from Active Directory.
3. Based on the information read from Active Directory and the state of the local certificate storage, the task sends a certificate request using the WCCE protocol to the CA and receives a newly issued certificate that is then stored in the local certificate storage.

Note that in this example there were no new CA certificates added to Active Directory since the last time the task executed, therefore no messages to read them are shown.

4.4.5 Task Processing Rule Details

This section describes details for the steps identified in section 4.3.8. The details are the pseudo-code for the Computer Certificate Autoenrollment Task that starts in section 4.4.5.1 and continues through section 4.4.5.9. Unless otherwise specified, the processing falls through from one section to the next.

As specified in section 4.2.2.3, an implementation of autoenrollment can read Group Policy information directly or rely on a separate GP Client to cache this data locally. In this section and its subsections any reference to reading Group Policy information refers to any of those implementation choices.

4.4.5.1 Initialize AutoEnrollmentPolicy.AutoEnrollmentOptions

In this step, autoenrollment initializes its options to determine it's specified function.

1. If a computer on which autoenrollment is running is not a domain member as specified in [MS-DISO] section 4, the task executes one of the following steps:

- If the local configuration has the settings to initialize AutoEnrollmentPolicy.Options datum, autoenrollment SHOULD<9> initialize the options based on that configuration. The format of this local configuration is implementation-specific.

-- Or --

- If the local configuration does not have the settings to initialize AutoEnrollmentPolicy.Options datum, autoenrollment SHOULD set AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags to a logical OR of AutoEnrollmentOptionFlags.Enabled, AutoEnrollmentOptionFlags.Enroll, AutoEnrollmentOptionFlags.RetrivePending, and AutoEnrollmentOptionFlags.Manage options. Or, the autoenrollment MAY set AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags to AutoEnrollmentOptionsFlags.Disabled.<10>

2. If a computer on which autoenrollment is running is a domain member as specified in [MS-DISO] section 4, the task executes the following steps:

   - Autoenrollment SHOULD read the following Group Policy instruction under the computer-scoped Group Policy Object (GPO) path as specified in [MS-GPREG] section 2.2.1.<11>

     - **Key:** SOFTWARE\Policies\Microsoft\Cryptography\Autoenrollment
     - **Value:** AEPolicy
     - **Type:** REG_WORD
     - **Size:** 4

     The mapping of the bits in **Value** is as follows.

     - If the instruction previously listed does not exist, the AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags is set to AutoEnrollmentOptionFlags.NotDefined.
     - If the data in the instruction previously listed has the 0x8000 bit set, AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags is set to AutoEnrollmentOptionFlags.Disabled.
     - In all other cases AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags is set to a logical OR of the AutoEnrollmentOptionFlags.Enabled and other options if their corresponding bits are set in the data of the instruction. The mapping of the bits and options is shown in the following table.

Bit in the data	AutoEnrollmentOptionFlags.*
0x1	AutoEnrollmentOptionFlags.Enroll
0x2	AutoEnrollmentOptionFlags.Manage
0x4	AutoEnrollmentOptionFlags.RetrivePending
0x20	The task MUST ignore this bit.

   - The task MUST ignore the following Group Policy instruction under the computer-scoped GPO path as specified in [MS-GPREG] section 2.2.1.

     - **Key:** SOFTWARE\Policies\Microsoft\Cryptography\Autoenrollment
     - **Value:** OfflineExpirationPercent

- **Type:** REG_DWORD
- **Size:** 4

3. If AutoEnrollmentPolicy.Options.AutoEnrollmentOptionsFlags is set to AutoEnrollmentOptionsFlags.Disabled, continue processing from section 4.4.5.9.

4.4.5.2 Update Issuer Stores

At this step, autoenrollment updates CA Certificates, Root Certificates, and Key Archival CA Certificates groups (as specified in section 4.2.1.1) with the certificates from Active Directory. If autoenrollment encounters any errors while executing steps specified in this section, it continues with the steps in the next section.

If a computer on which autoenrollment is running is a domain member as specified in [MS-DISO] section 4, execute the following steps, otherwise continue with the steps in the next section.

1. Read certificates from Active Directory as specified in the following table.

Local Certificate Storage Group	Corresponding Active Directory location and format
CA Certificates	Container: "CN=AIA, CN=Public Key Services, CN=Services, CN=Configuration, DC=..." Object: all objects of type certificationAuthority Attribute: cACertificate Format: multi-valued OCTET string attribute that contains the DER encoded CA certificates.
Root Certificates	Container: "CN=Certification Authorities, CN=Public Key Services, CN=Services, CN=Configuration, DC=..." Object: all objects of type certificationAuthority Attribute: cACertificate Format: multi-valued OCTET string attribute that contains the DER encoded root CA certificates.
Key Archival CA Certificates	Container: "CN=Public Key Services, CN=Services, CN=Configuration, DC=..." Object: an object of type certificationAuthority and CN=NTAuthCertificates Attribute: cACertificate Format: multi-valued OCTET string attribute that contains the DER encoded CA certificates.

Note The "CN=Configuration,DC=.." part in the LDAP paths in the table is replaced with the value of the configurationNamingContext attribute (specified in [MS-ADTS] section 3.1.1.3.2.1) of the rootDSE object.

2. Make each local group of certificates a copy of its corresponding Active Directory group.

4.4.5.3 Initialize AutoenrollmentPolicy.EnrollmentPolicies

There are two ways in which autoenrollment can initialize AutoenrollmentPolicy.EnrollmentPolicies datum: the Basic Initialization (see section 4.4.5.3.1) in which only Active Directory is used, or the Advanced Initialization (see section 4.4.5.3.2) where Group Policy and local configuration is used to discover CEP servers in addition to using Active Directory. Autoenrollment SHOULD implementAdvanced Initialization <12>. If it does not, it MAY only implement Basic Initialization. <13> Autoenrollment MUST process the automatic certificate request setting specified in section 4.4.5.3.3 after it completes Basic or Advanced initialization.

4.4.5.3.1 Basic Initialization

In this mode autoenrollment initializes only one instance of the CertificateEnrollmentPolicy in the AutoenrollmentPolicy.EnrollmentPolicies list. The instance is initialized from the certificate templates and CA information in Active Directory.

If and only if the computer on which autoenrollment is running is a domain member as specified in [MS-DISO] section 4, execute the following steps:

1. Create an instance of the CertificateEnrollmentPolicy and add it to the AutoEnrollmentPolicy.EnrollmentPolicy list.

2. Set CertificateEnrollmentPolicy.EnrollmentProtocol to WCCE.

3. Set CertificateEnrollmentPolicy.IsDefault to true.

4. Set CertificateEnrollmentPolicy.Flags to AutoEnrollmentEnabled.

5. Initialize the CertificateEnrollmentPolicy.Templates list as specified in section 4.4.5.3.1.1.

6. Initialize the CertificateEnrollmentPolicy.Issuers list as specified in section 4.4.5.3.1.2.

4.4.5.3.1.1 Initializing Certificate Templates

This section specifies how the CertificateEnrollmentPolicy.Templates list is initialized from the Active Directory, as follows:

1. Perform an LDAP search for certificate template (pKICertificateTemplate) objects (specified in [MS-CRTD]) under the following container:

   ```
   "CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=..."
   ```

   where "CN=Configuration,DC=..." is replaced with the value of the **configurationNamingContext** attribute (specified in [MS-ADTS] section 3.1.1.3.2.1) of the rootDSE object. If the search fails for any reason, the CertificateEnrollmentPolicy.Templates list is initialized to an empty list.

2. For each object in the search result:

   1. If ntSecurityDescriptor attribute of the object does not have Enroll permission or has Enroll permission denied (specified in [MS-CRTD] section 2.5) for the current computer, continue with the next object.

   2. Autoenrollment SHOULD<14> create an instance of a CertificateTemplate datum and map the attributes (defined in [MS-CRTD]) to the ADM defined in section 4.3.2.1.2.1 as specified in the following table.

Attribute name	Attribute values	ADM datum	ADM values
cn	All	CommonName	All
flags		GeneralFlags	
	CT_FLAG_MACHINE_TYPE		IsMachineType
	CT_FLAG_IS_CA		IsCA
	CT_FLAG_IS_CROSS_CA		IsCrossCA
ntSecurity Descriptor		No direct mapping; see specific values for details.	
	AutoEnroll allowed and not denied as specified in [MS-CRTD] section 2.5.		The autoenrollment SHOULD<15> set AutoEnrollmentEnabled to true.
	AutoEnroll denied or not allowed as specified in [MS-CRTD] section 2.5.		The autoenrollment SHOULD<16> set AutoEnrollmentEnabled to false.
revision	All	MajorVersion	All
pKIOverlap Period	All	OverlapPeriod	All
msPKI-Template-Schema-Version	All	SchemaVersion	All
msPKI-RA-Signature	All	RARequirements. SignatureCount	All
msPKI-RA-Policies	All	RARequirements.IssuancePolicies	All
msPKI-RA-Application-Policies	All values when msPKI-Template-Schema-Version equals 1 or 2. If msPKI-Template-Schema-Version equals 3, the pair with the name msPKI-RA-Application-Policies.	RARequirements.EKUs	All
msPKI-Private-Key-Flag	CT_FLAG_REQUIRE_PRIVATE_KEY_ARCHIVAL	PrivateKeyAttributes.Archive	All
msPKI-Cert-Template-OID	All	OID	All
msPKI-	All	SupersededTempl	All

Attribute name	Attribute values	ADM datum	ADM values
Supersede-Templates		ates	
msPKI-Enrollment-Flag		EnrollmentFlags	
	CT_FLAG_PREVIOUS_APPROVAL_VALIDATE_REENROLLMENT		PreviousApprovalValidateReenrollment
	The computer autoenrollment MUST ignore CT_FLAG_PUBLISH_TO_DS and CT_FLAG_AUTO_ENROLLMENT_CHECK_USER_DS_CERTIFICATE flags.		
	CT_FLAG_USER_INTERACTION_REQUIRED		HumanInteractionRequired
	CT_FLAG_REMOVE_INVALID_CERTIFICATE_FROM_PERSONAL_STORE		RemoveInvalidCertificateFromComputerStore
	CT_FLAG_AUTO_ENROLLMENT		If this flag is not set, the autoenrollment MAY<17> set the AutoEnrollmentEnabled to false.
msPKI-Certificate-Name		SubjectNameFlags	
	CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT		EnrolleeSuppliesSubject
	CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT_ALT_NAME		EnrolleeSuppliesSubjectAltName
	CT_FLAG_OLD_CERT_SUPPLIES_SUBJECT_AND_ALT_NAME		OldCertSuppliesSubjectAndAltName

Attributes or specific attribute values (such as flags) that are defined in [MS-CRTD], but are not listed in the previous table MUST be ignored by autoenrollment.

4.4.5.3.1.2 Initializing CAs

The following section specifies how the CertificateEnrollmentPolicy.Issuers list is initialized from Active Directory.

1. Autoenrollment MUST do an LDAP search for the CA information (pKIEnrollmentService) objects (specified in [MS-WCCE] section 2.2.2.9.2) under the following container:

   ```
   "CN=Enrollment Services,CN=Public Key Services,CN=Services,CN=Configuration,DC=..."
   ```

   where "CN=Configuration,DC=..." is replaced with the value of the **configurationNamingContext** attribute (specified in [MS-ADTS] section 3.1.1.3.2.1) of the rootDSE object. If the search fails for any reason, initialize CertificateEnrollmentPolicy.Issuers to an empty list.

2. For each object in the search result:

    1. If the ntSecurityDescriptor attribute of the object does not have Enroll permission or has Enroll permission denied (specified in [MS-CRTD] section 2.5) for the current computer, continue with the next object.

    2. Create an instance of the Issuer datum.

    3. Set Issuer.IsRenewalOnly to false.

    4. Set Issuer.Priority to zero.

    5. Set Authentication to Kerberos.

    6. Map the attributes of the object to the ADM defined in section 4.3.2.1.2.2 as specified in the following table.

Attribute name	ADM datum
CN	Issuer.Name
dNSHostName	Issuer.EndPoint
certificateTemplates	Issuer.Templates
cACertificate	Issuer.Certificate

    7. Verify that the certificate in Issuer.Certificate is valid and trusted as specified in section 4.4.6.3. If verification succeeds, add the Issuer datum to the CertificateEnrollmentPolicy.Issuers list.

4.4.5.3.2 Advanced Initialization

In this mode autoenrollment initializes instances of the CertificateEnrollmentPolicy based on its Group Policy and local configuration. At the highest level autoenrollment executes the following steps:

1. Initialize configuration options (see section 4.4.5.3.2.1).

2. Obtain policy server end point information from Group Policy and local configuration (see section 4.4.5.3.2.2).

3. Combine policy server end point information by CEP ID and sort it within each group to determine the order in which they ought to be used when accessing the CEP data (see section 4.4.5.3.2.3).

4. Read CEP data from policy server end points and initialize CertificateEnrollmentPolicy instances with the retrieved data (see section 4.4.5.3.2.4).

The details for these steps are described in the following subsections.

4.4.5.3.2.1 Initializing Configuration Options

In this step autoenrollment initializes options of the EndPointsInformation that determine how autoenrollment populates the CertificateEnrollmentPolicyEndPoints table.

1. If autoenrollment is running on a computer that is not a domain client as specified in [MS-DISO] section 4:
    - Set EndPointsInformation.IsGroupPolicyConfigurationEnabled to false.
    - Set EndPointsInformation.IsLocalConfigurationEnabled to true.
2. If autoenrollment is running on a computer that is a domain client as specified in [MS-DISO] section 4:

    Autoenrollment MUST read the following Group Policy instruction under the computer-scoped GPO path as specified in [MS-GPREG] section 2.2.1:

    - **Key:** SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers
    - **Value:** Flags
    - **Type:** REG_WORD
    - **Size:** 4

      The mapping of the bits in **Value** is as follows:

        - If the 0x2 bit of the data in the previous instruction is set, set the EndPointsInformation.IsGroupPolicyConfigurationEnabled to true, otherwise set it to false.
        - If the 0x4 bit of the data in the previous instruction is set, set the EndPointsInformation.IsLocalConfigurationEnabled to true, otherwise set it to false.
        - If this instruction does not exist, set the EndPointsInformation.IsGroupPolicyConfigurationEnabled and EndPointsInformation.IsLocalConfigurationEnabled to false.

4.4.5.3.2.2 Obtaining End Point Information

In this step autoenrollment initializes the CertificateEnrollmentPolicyEndPoints table.

1. If EndPointsInformation.IsGroupPolicyConfigurationEnabled is false, go to step 3 of this section.
2. Read the following Group Policy instructions under the computer-scoped GPO path (as specified in [MS-GPREG] section 2.2.1) to initialize rows in the CertificateEnrollmentPolicyEndPoints table.

    In the following instructions the value "<KeyName>" could be any string. When processing Group Policy information, instructions with Keys, where the <KeyName> is the same, are considered to be for the same row in the EndPoints table of the EndPointInformation datum and are related to the same policy server end point. For each value of "<KeyName>" a new row in the EndPoints table is created and initialized with default values. For columns that do not specify a default value, an instruction in the Group Policy file has to exist. If it does not exist, the whole row is considered invalid and MUST be ignored in later processing.

    URL:

    - **Instruction Key**: SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers\<KeyName>

- **Instruction Value**: URI
- **Instruction Type**: REG_SZ
- **Instruction Size**: Depends on the data
- **Column**: Endpoint.URI
- **Default**: Not defined

PolicyID:

- **Instruction Key**: SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers\<KeyName>
- **Instruction Value**: PolicyId
- **Instruction Type**: REG_SZ
- **Instruction Size**: Depends on the data
- **Column**: EndPoint.PolicyId
- **Default**: Not defined

Flags:

- **Instruction Key**: SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers\<KeyName>
- **Instruction Value**: Flags
- **Instruction Type**: REG_DWORD
- **Instruction Size**: 4
- **Column**: EndPoint.flags
    - If the data of this instruction has the 0x10 bits set, set the AutoEnrollmentEnabled flag in this column.
    - If the data of this instruction has the 0x20 bits set, set the AllowUntrustedIssuer flag in this column.
- **Default:** Neither AutoEnrollmentEnabled nor AllowUntrustedIssuer flags are set.

AuthFlags:

- **Instruction Key:** SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers\<KeyName>
- **Instruction Value:** AuthFlags
- **Instruction Type:** REG_DWORD
- **Instruction Size: 4**
- **Column:** EndPoint.Authentication

  Use the following mapping of the data of this instruction to the ADM:

  0x1 - Anonymous

0x2 - Kerberos

0x3 - Password

0x8 - Certificate

- **Default:** Not defined

Cost:

- **Instruction Key:** SOFTWARE\Policies\Microsoft\Cryptography\PolicyServers\<KeyName>
- **Instruction Value:** Cost
- **Instruction Type:** REG_DWORD
- **Instruction Size: 4**
- **Column:** EndPoint.Cost
- **Default:** 0x7FFFFFFD

3. If EndPointsInformation.IsLocalConfigurationEnabled is true, read information from implementation-specific local storage that contains information which can be used to initialize rows in the EndPoints table.

4.4.5.3.2.3 Group and Sort End Point Information

In this step autoenrollment processes the end point information by grouping it by CEP ID and sorting in the order with which it will use the end point to access the CEP information.

1. Create groups of the CertificateEnrollmentPolicyEndPoint instances that were created in section 4.4.5.3.2.2 and have the same value of the EndPoint.PolicyId datum.
2. Sort each group by following these rules:
   - Sort the CertificateEnrollmentPolicyEndPoint instances in ascending order based on the EndPoint.Cost value.
   - For instances that have the same EndPoint.Cost:
     1. Sort those that have EndPoint.Authentication equal to Kerberos first.
     2. Then sort those that have EndPoint.Authentication equal to Anonymous.
     3. The rest of the CertificateEnrollmentPolicyEndPoint instances follow in an arbitrary order.

4.4.5.3.2.4 Read CEP Data

In this step autoenrollment initializes instances of the CertificateEnrollmentPolicy by accessing end points associated with CEP groups created in the previous step.

For each group created in the previous step (section 4.4.5.3.2.3):

1. Pick an arbitrary instance of the CertificateEnrollmentPolicyEndPoint from the group (see section 4.2.3 for an assumption on why any instance is equivalent for this step). If this instance does not have the AutoEnrollmentEnabled flag set in the EndPoint.Flags, continue with the next group.

2. If the current group contains a CertificateEnrollmentPolicyEndPoint instance with EndPoint.URI equal to "LDAP":

    1. Perform an LDAP search to read the value of the objectGuid attribute of the root object of the **forest root domain NC**. If any errors are encountered, continue with the next group.

    2. Compare the value read in the previous step to the EndPoint.PolicyId datum CertificateEnrollmentPolicyEndPoint instance. If the values do not match, continue with the next group.

3. For each CertificateEnrollmentPolicyEndPoint instance for that group:

    1. If EndPoint.URI equals "LDAP":

        1. Follow the steps specified section 4.4.5.3.1 to initialize an instance of the CertificateEnrollmentPolicy. If there are errors performing LDAP searches, continue with the next CertificateEnrollmentPolicyEndPoint instance. If no errors are encountered, add CertificateEnrollmentPolicy instance to the AutoEnrollmentPolicy.EnrollmentPolicies list.

        2. If EndPointsInformation.DefaultPolicyId does not equal the current value of EndPoint.PolicyId, set the IsDefault field of the CertificateEnrollmentPolicy datum created in the previous step to false.

        3. Continue with the next group.

    2. If EndPoint.URI starts with "HTTPS//":

        1. Use the XCEP protocol (specified in [MS-XCEP]) to obtain CEP information returned via a GetPoliciesResponse message from a server identified in the EndPoint.URI datum.

        2. If autoenrollment was unable to retrieve a GetPoliciesResponse message from a XCEP server, continue with the next CertificateEnrollmentPolicyEndPoint instance.

        3. If autoenrollment was able to retrieve a GetPoliciesResponse message:

            1. Create an instance CertificateEnrollmentPolicy datum.

            2. Set CertificateEnrollmentPolicy.EnrollmentProtocol to WSTEP.

            3. Set CertificateEnrollmentPolicy.PolicyID to the value of the EndPoint.PolicyId.

            4. If EndPointsInformation.DefaultPolicyId is equal to the value of EndPoint.PolicyId, set the CertificateEnrollmentPolicy.IsDefault to true, otherwise set it to false.

            5. If the last instance of the CertificateEnrollmentPolicyEndPoint in the group has the AllowUntrustedIssuer flag set in the EndPoint.Flags, set CertificateEnrollmentPolicy.IsUntrustedIssuerAllowed to true, otherwise set it to false.

            6. Initialize the CertificateEnrollmentPolicy.Issuers list as specified in section 4.4.5.3.2.4.1.

            7. Initialize the CertificateEnrollmentPolicy.Templates list as specified in section 4.4.5.3.2.4.2.

            8. Add the newly created CertificateEnrollmentPolicy instance to the AutoEnrollmentPolicy.EnrollmentPolicies list.

            9. Continue with the next group.

3. For any other value of the EndPoint.URI, ignore this instance of the CertificateEnrollmentPolicyEndPoint.

4.4.5.3.2.4.1 Initialize Issuers

This section describes how, given an instance of the CertificateEnrollmentPolicy and a GetPoliciesResponse message of the XCEP protocol, autoenrollment initializes the CertificateEnrollmentPolicy.Issuers list.

For each <cA> element (<CA> type) in the <cACollection> (<CACollection> type) of the GetPoliciesResponse message:

1. If enrollPermission of the <cA> element is false, continue with the next <cA> element.

2. Verify that the certificate in <certificate> element is valid and trusted as specified in section 4.4.6.3. If verification does not succeed, continue with the next <cA> element.

3. For each <cAURI> element (<CAURI type) in the <uris> collection of the current <cA> element:

    1. Create an instance of the Issuer datum (specified in section 4.3.2.1.2.2).

    2. Set Issuer.EndPoint to the value of the <uri> element.

    3. Set Issuer.IsRenewalOnly to the value of the <renewalOnly> element.

    4. Set Issuer.Priority to the value of the <priority> element.

    5. Set Issuer.Authentication using the following table:

<clientAuthentication> element value	ADM value
1	Anonymous
2	Kerberos
4	Password
8	Certificate

    6. Set Issuer.Certificate to the value of the <certificate> element of the parent <cA> element.

4.4.5.3.2.4.2 Initializing Certificate Templates

This section describes how, given an instance of the CertificateEnrollmentPolicy and a GetPoliciesResponse message of the XCEP protocol, autoenrollment initializes the CertificateEnrollmentPolicy.Templates list and updates the CertificateEnrollmentPolicy.Issuers list with templates that each Issuer supports.

For each <policy> element in the <policyCollection> element (<policyCollection> type defined in [MS-XCEP] section 3.1.4.1.3.20) of the GetPoliciesResponse message:

1. If <attributes.permission.enroll> element is false, continue with the next element.

2. Create an instance of CertificateTemplate datum.

3. Use the following table to initialize the new CertificateTemplate instance.

XML element	ADM datum	ADM values
attributes.commonName	CommonName	All
attributes.generalFlags	GeneralFlags	Use the same mapping of individual bits (as specified in section 4.4.5.3.1) for the flags attribute and GeneralFlags ADM.
attributes.permissions.autoenroll	AutoEnrollmentEnabled	All
attributes.revision.majorRevision	MajorVersion	All
attributes.certificateValidity.renewalPeriodSeconds	OverlapPeriod	All
attributes.policySchema	SchemaVersion	All
attributes.rARequirements.rASignatures	RARequirements.SignatureCount	All
attributes.rARequirements.rAPolicies	RARequirements.IssuancePolicies	All
attributes.rARequirements.rAEKUs	RARequirements.EKUs	All
attributes.privateKeyFlags	Multiple	Same as the mapping for the msPKI-Private-Key-Flag in section 4.4.5.3.1.
Use attributes.policyOIDReference element to find an OID element in the GetPoliciesResponseMessage.oIDs collection that has a matching oIDReference element. The value element will contain the OID of the template.	OID	All
attributes.supersededPolicies	SupersededTemplates	All
attributes.enrollmentFlags	EnrollmentFlags	Use the same mapping of individual bits (as specified in section 4.4.5.3.1) for the flags attribute and EnrollmentFlags ADM.
attributes.subjectNameFlags	SubjectNameFlags	Use the same mapping of individual bits (as specified in section 4.4.5.3.1) for the flags

XML element	ADM datum	ADM values
		attribute and SubjectNameFlags ADM.

4. Add the datum to the CertificateEnrollmentPolicy.Templates list.

5. For each <cAReference> in the <cAs> collection of the current <policy> XML element (<CertificateEnrollmentPolicy> type):

    1. Find a corresponding <CA> object in the <cAs> collection of the GetPoliciesResponse message by matching its <cAReferenceID> element.

    2. For each <cAURI> element in the <uris> element (type <CAURICollection>) collection of the <CA> object located in the previous step:

        1. Find an Issuer ADM instance in the current CertificateEnrollmentPolicy.Issuers list that has Issuer.EndPoint equal to the <cAURI.uri>.

        2. Add the current CertificateTemplate.CommonName to the Issuer.Templates list.

4.4.5.3.3 Initializing Automatic Certificate Request Settings

The following section specifies how to process automatic certificate request settings in Group Policy. Group Policy contains entries that specify which templates to autoenroll for. This is in addition to the autoenroll permission specified in [MS-CRTD] section 2.5.

1. If the computer on which autoenrollment is running is not a domain member (as specified in [MS-DISO] section 4), the processing for this section is complete.

2. If there are no instances of the CertificateEnrollmentPolicyDatum in the AutoEnrollmentPolicy.EnrollmentPolicies list OR there is no instance of the CertificateEnrollmentPolicy datum in the AutoEnrollmentPolicy.EnrollmentPolicies list that has IsDefault field set to true (this specific CertificateEnrollmentPolicy instance will be referred to as the default CEP in this section), the processing for this section is complete.

3. Read the following Group Policy instructions under the computer-scoped GPO path as specified in [MS-GPREG] section 2.2.1:

    - **Key:** SOFTWARE\Policies\Microsoft\SystemCertificates\ACRS\CTLs\<HashOfData> where <HashOfData> is a SHA1 hash of the instruction data. Autoenrollment MUST verify that hash is in fact correct for the data in this instruction and MUST ignore the instruction if it is not.

    - **Value:** Blob

    - **Type:** REG_BINARY

    - **Size:** Depends on the data

    The data in this instruction is a **Basic Encoding Rules (BER)** encoded unsigned Cryptographic Message Syntax (CMS) message (as specified in [RFC3852]) with contentType equal to 1.3.6.1.4.1.311.10.1. The content of the message is an ASN.1 structure defined as follows:

```
CertificateTrustList ::= SEQUENCE {
    version                 CTLVersion DEFAULT v1,
    subjectUsage            SubjectUsage,
    listIdentifier          ListIdentifier OPTIONAL,
    sequenceNumber          HUGEINTEGER OPTIONAL,
    ctlThisUpdate           ChoiceOfTime,
    ctlNextUpdate           ChoiceOfTime OPTIONAL,
    subjectAlgorithm        AlgorithmIdentifier,
    trustedSubjects         TrustedSubjects OPTIONAL,
    ctlExtensions           [0] EXPLICIT Extensions OPTIONAL
}

CTLVersion ::= INTEGER {v1(0)}

SubjectUsage ::= EnhancedKeyUsage

ListIdentifier ::= OCTETSTRING

TrustedSubjects ::= SEQUENCE OF TrustedSubject

TrustedSubject ::= SEQUENCE{
    subjectIdentifier       SubjectIdentifier,
        subjectAttributes       Attributes OPTIONAL
}

SubjectIdentifier ::= OCTETSTRING
```

Note the autoenrollment SHOULD<18> ignore the fields of the CertificateTrustList structure unless explicitly used in this section.

4. For each instruction previously identified:

    1. If the data is not a BER encoded unsigned CMS message (as specified in [RFC3852]) with contentType equal to 1.3.6.1.4.1.311.10.1, continue with the next instruction.

    2. If the subjectUsage field of the CertificateTrustList structure is not 1.3.6.1.4.1.311.20.1, continue with the next instruction.

    3. If the listIdentifier is not included, continue with the next instruction.

    4. The listIdentifier is a [UNICODE] string that contains the name of the template to autoenroll for. If the string contains a "|" character (binary 0x007C), the template name is the string starting right after that character up to the null character (0x0000) or to the end of the string. If the string does not have the character, the whole string is the name of the template.

    5. If there exists an instance of the CertificateTemplate datum that has a CommonName equal to the one identified in the previous step AND is part of the CertificateEnrollmentPolicy.Templates list of the default CEP (see step 2), autoenrollment MUST set the AutoEnrollmentEnabled field of that datum to true.

4.4.5.4 Initialize Local Information

In this step, autoenrollment initializes all of the ADM elements that are specified in section 4.3.2.2 by executing the following steps:

1. Read certificates from the local certificate storage, Computer Certificates group (specified in section 4.2.1.1) to initialize the Certificates.CurrentCertificates list. If autoenrollment fails to read certificates from local storage, this is fatal error and autoenrollment MUST terminate (section 4.4.5.9).

2. Initialize Certficates.ToBeAdded and Certificates.ToBeDeleted to empty lists.

3. Read certificates from the local certificate storage, CA Certificates group (specified in section 4.2.1.1) to initialize the Certificates.CAs list. If autoenrollment fails to read certificates from local storage, initialize the Certificates.CAs list to empty.

4. Read certificates from the local certificate storage, Root Certificates group (specified in section 4.2.1.1) to initialize the Certificates.Roots list. If autoenrollment fails to read certificates from local storage, initialize the Certificates.Roots list to empty.

5. Read certificates from the local certificate storage, Key Archival CA Certificates group (specified in section 4.2.1.1) to initialize the Certificates.KeyArchivalCAs list. If autoenrollment fails to read certificates from local storage, initialize the Certificates.KeyArchivalCAs list to empty.

6. Read pending request information from the implementation specific local storage and initialize the PendingRequests table. If autoenrollment fails to read from the local storage, initialize the PendingRequests table as empty.

7. Read successful enrollment information from the implementation specific local storage and initialize the ProcessedEnrollments table. If autoenrollment fails to read from the local storage, initialize the ProcessedEnrollments table as empty.

4.4.5.5 Retrieve pending requests

If AutoEnrollmentPolicy.Options.AutoEnrollmentOptionFlags has the AutoEnrollmentOptionFlags.RetrivePending flag set, autoenrollment SHOULD<19> retrieve pending requests.

For each row in the PendingRequests table:

1. If the request is older than sixty days (based on the PendingRequests.RequestTime), delete the record from the table and continue with the next row.

2. Find a corresponding CertificateTemplate datum for this pending request using the FindCertificateTemplate algorithm specified in section 4.4.6.1. Use PendingRequests.PolicyId, PendingRequests.TemplateOID, and PendingRequests.TemplateName to initialize variables of the algorithm instead of using the certificate as input. If no CertificateTemplate instance was found, continue with the next record.

3. Retrieve pending certificate request as follows:

    1. If the value of the PendingRequests.Protocol is "WCCE", use information stored in the PendingRequests.RequestId column to retrieve the certificate request from the issuer stored in the PendingRequests.Issuer datum as specified in [MS-WCCE] section 3.1.1.5.2.5.

    2. If the value of the PendingRequests.Protocol is "WSTEP", use information stored in the PendingRequests.RequestId column to retrieve the certificate request from the issuer stored in the PendingRequests.Issuer datum as specified in [MS-WSTEP] section 3.1.7.3.

4. If retrieval resulted in a certificate being issued:

    1. Add the new certificate to the Certificates.CurrentCertificates and Certificates.ToBeAdded lists.

    2. Autoenrollment SHOULD add a row to the ProcessedEnrollment table to record the successful certificate enrollment.<20>

    3. If the CertificateTemplate instance identified in step 2 has the RemoveInvalidCertificateFromComputerStore bit set in the EnrollmentFlags field, autoenrollment SHOULD add all certificates from the Certificates.CurrentCertificates list based on the same certificate template to the Certificates.ToBeDeleted list.<21>

    4. Delete the current row from the PendingRequests table and continue with the next row.

5. If retrieval resulted in certificate still pending, continue with the next row.

6. If retrieval resulted in the request being denied by the issuer, remove the current row from the table and continue with the next row.

7. If retrieval resulted in an error, continue with the next row.

4.4.5.6 Autoenroll Based on Certificate Templates

In this step, autoenrollment processes all certificate templates available from each CEP by examining each template's validity for autoenrollment. Autoenrollment checks certificate storage to see if an acceptable certificate based on each template already exists, and enrolls for new certificates as needed. The processing rules are as follows:

1. If AutoEnrollmentPolicy.Options.AutoEnrollmentOptionFlags does not have the AutoEnrollmentOptionFlags.Enroll flag set, processing for this section is complete.

2. For each CertificateEnrollmentPolicy instance in AutoenrollmentPolicy.EnrollmentPolicies:

    - For each instance of the CertificateTemplate in CertificateEnrollmentPolicy.Templates:

        1. If the current instance of the CertificateTemplate is not valid for autoenrollment (see section 4.4.5.6.1), continue with the next instance.

        2. Determine the action required based on the certificates currently in storage (see section 4.4.5.6.2).

        3. If the action is to submit a new request, submit the request (see section 4.4.5.6.3) and continue with the next instance.

        4. If the action is to submit a renewal request, submit the renewal request (see section 4.4.5.6.4) and continue with the next instance.

The following subsections provide details for the preceding steps.

4.4.5.6.1 Determine if a CertificateTemplate Instance is Valid for Autoenrollment

If any conditions in the following list are true, autoenrollment SHOULD NOT process a new enrollment for the specific CertificateTemplate instance<22>:

- CertificateTemplate.AutoEnrollmentEnabled is set to false.
- CertificateTemplate.GeneralFlags does not have the IsMachineType, IsCA, or IsCrossCA flags set.
- CertificateTemplate.EnrollmentFlags has the HumanInteractionRequired flag set.
- CertificateTemplate.SubjectNameFlags has the EnrolleeSuppliesSubjectName flag set.
- CertificateTemplate.SubjectNameFlags has the EnrolleeSuppliesSubjectAltName flag set.
- CertificateTemplate.RARequirements.SignatureCount is greater than 1.

- There exists an instance of CertificateTemplate in the CertificateEnrollmentPolicy.Templates list whose CertificateTemplate.SupersededTemplates list contains a value equal to the current CertificateTemplate.CommonName.

4.4.5.6.2 Determine Action Required Based on the Certificates Currently in Storage

The following section specifies whether enrollment or renewal is required for a certificate based on some specific template.

To understand the processing rules in this section, several terms to classify certificates need to be defined. First, this section specifies the common requirements to the classes of certificates followed by the class definitions themselves. Finally the section provides the processing rules.

Common Certificate Requirements: The following list defines common certificate requirements:

- The certificate is based on a given certificate template as specified in the FindCertificateTemplate algorithm in section 4.4.6.1.
- The certificate is valid as specified in section 4.4.6.3.
- If the certificate subject name or subject alternative name contains a FQDN name, the FQDN matches the name of the computer on which autoenrollment is running.

Note that Autoenrollment SHOULD<23> keep track of how many times it has tried to enroll for new certificates based on the DNS name change and SHOULD<24> terminate those attempts after some implementation-dependent threshold.

Classes of certificates are as follows:

Acceptable Certificate: A certificate that satisfies all of the following criteria:

- All of the Common Certificate Requirements specified earlier in this section.
- The certificate is within 80% of its lifetime OR the certificate has not reached the CertificateTemplate.OverlapPeriod of its template.
- If the certificate has the Certificate Template OID extension (1.3.6.1.4.1.311.21.7), as specified in [MS-WCCE] section 2.2.2.5.5.2, the templateMajorVersion field of that extension equals CertificateTemplate.MajorVersion.

Certificate Close To Expire: A certificate that satisfies all of the following criteria:

- All of the Common Certificate Requirements specified earlier in this section.
- The certificate has passed 80% of its validity period AND the certificate reached the CertificateTemplate.OverlapPeriod of its template.
- If the certificate has the Certificate Template OID extension (1.3.6.1.4.1.311.21.7), as specified in [MS-WCCE] section 2.2.2.5.5.2, the templateMajorVersion field of that extension equals CertificateTemplate.MajorVersion.

Special Case Certificate: A certificate for which the template's version has changed, but it still can be renewed. This certificate has the following criteria:

- All of the Common Certificate Requirements specified earlier in this section.

- Certificate has the Certificate Template OID extension (1.3.6.1.4.1.311.21.7), as specified in [MS-WCCE] section 2.2.2.5.5.2, and the templateMajorVersion field of that extension is less than CertificateTemplate.MajorVersion.
- CertificateTemplate.EnrollmentFlags has the PreviousApprovalValidateReenrollment flag set.

There are three possible actions that autoenrollment can take for a certificate template based on the certificates that exist on the system. Those actions are "no action," "new request," or "renewal." The steps to determine the appropriate action are as follows:

1. If the Certificates.CurrentCertificates list has an Acceptable Certificate, no action is required.
2. If the Certificates.CurrentCertificates list has a Certificate Close To Expire, the action is to renew that certificate.
3. If the Certificates.CurrentCertificates list has a Special Case Certificate, the action is to renew that certificate.
4. In all other cases the action is to request a new certificate.

4.4.5.6.3 Submitting a New Request

This section specifies how autoenrollment submits new certificate requests based on a specific template that is part of some CertificateEnrollmentPolicy. The steps are as follows:

1. Get a list of issuers that support certificate templates being used for this request by following the SelectAndOrderIssuers algorithm specified in section 4.4.6.2. The input for the algorithm is the current CertificateTemplate instance being processed, the CertificateEnrollmentPolicy that the template belongs to, and the IsRenewalOnlyAllowed parameter set to false. If there are no issuers returned by the algorithm, the request cannot be submitted.
2. If CertificateTemplate.RARequirements.SignatureCount equals 1, autoenrollment SHOULD attempt to find a private key to co-sign the certificate request as follows:<25>
    1. Search the Certificates.CurrentCertificates list for a certificate that has the following:
        - OIDs identified in the CertificateTemplate.RARequirements.EKUs list as a part of the Extended Key Usage extension (as specified in [RFC5280] section 4.2.1.12).
        - OIDs identified in the CertificateTemplate.RARequirements.IssuancePolicies list as part of the Certificate Policies extension (as specified in [RFC5280] section 4.2.1.4).
    2. If no certificate has been found, the certificate request cannot be processed.
    3. If a certificate has been found and has an associated private key, use its private key to co-sign a request that is created in steps 3 or 4.
3. If CertificateTemplate.PrivateKeyAttributes.Archive is true, autoenrollment SHOULD validate the CA exchange certificate as specified in section 4.4.6.4. <26> If certificate validation fails, the certificate request cannot be processed.
4. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WCCE, create a certificate request as specified in [MS-WCCE] section 3.1.2.
5. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WSTEP, create a certificate request as specified in [MS-WSTEP].

    Note that when creating a private key in steps 4 and 5, persist the key in the local private key storage specified in section 4.2.1.2.

6. For each issuer identified in step 1:

    1. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WCCE, submit a certificate request as specified in [MS-WCCE] section 3.1.2. When submitting the request, make a remote DCOM call to the computer specified in the Issuer.EndPoint datum and use the value of the Issuer.Name datum for the pwszAuthority parameter of the ICertRequestD methods.

    2. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WSTEP, submit a certificate request as specified in [MS-WSTEP].

    3. If a certificate has been issued by the issuer:

        1. If CertificateEnrollmentPolicy.IsUntrustedIssuerAllowed is false, autoenrollment SHOULD validate the certificate as specified in section 4.4.6.3.<27> If the certificate is not valid, the certificate cannot be accepted and MUST not be added to local certificate storage.

        2. Add the certificate in the Certificates.ToBeAdded and Certificates.CurrentCertificates lists.

        3. Autoenrollment SHOULD add an entry to the ProcessedEnrollments table<28>.

        4. If the CertificateTemplate.SupersededTemplates property is not empty, the autoenrollment SHOULD eliminate obsolete certificates as follows<29>:

            1. Find all certificates based on those templates in the Certificates.CurrentCertificates list.

            2. For each certificate, if its corresponding CertificateTemplate instance has the RemoveInvalidCertificateFromComputerStore bit set in the EnrollmentFlags field, add that certificate to the Certificates.ToBeDeleted list.

        5. The processing for this section is complete.

    4. If the certificate has been set to pending by the issuer, autoenrollment SHOULD add an entry to the PendingRequests table<30>. The processing for this section is complete.

    5. If there was an error while submitting a request, continue with the next issuer.

4.4.5.6.4 Submitting a Renewal Request

This section specifies how autoenrollment submits renewal certificate requests based on a template that is part of some CertificateEnrollmentPolicy. The steps are as follows:

1. Get a list of issuers that support certificate templates being used for this request by following the SelectAndOrderIssuers algorithm specified in section 4.4.6.2. The input for the algorithm is the current CertificateTemplate instance being processed, the CertificateEnrollmentPolicy that the template belongs to, and the IsRenewalOnlyAllowed parameter is set to true. If there are no issuers returned by the algorithm, the request cannot be submitted.

2. If CertificateTemplate.RARequirements.SignatureCount is equal to 1 AND CertificateTemplate.EnrollmentFlags.PreviousApprovalValidateReenrollment is set to true, autoenrollment SHOULD sign the request with the key whose certificate is being renewed.<31>

3. If CertificateTemplate.PrivateKeyAttributes.Archive is set to true, autoenrollment SHOULD validate the CA exchange certificate as specified in section 4.4.6.4.<32> If certificate validation fails, the certificate request cannot be processed.

4. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WCCE, create a renewal request as specified in [MS-WCCE] section 3.1.2. When submitting the request, make a remote DCOM call to the computer specified in the Issuer.EndPoint datum and use the value of the Issuer.Name datum for the pwszAuthority parameter of the ICertRequestD methods.

5. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WSTEP, create a renewal request as specified in [MS-XCEP].

   Note that when creating a private key in steps 4 and 5, persist the key in the local private key storage specified in section 4.2.1.2.

6. For each issuer identified in step 1:

   1. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WCCE, submit a certificate request as specified in [MS-WCCE] section 3.1.2.

   2. If CertificateEnrollmentPolicy.EnrollmentProtocol is equal to WSTEP, submit a certificate request as specified in [MS-WSTEP].

   3. If a certificate has been issued by the issuer:

      1. If CertificateEnrollmentPolicy.IsUntrustedIssuerAllowed is set to false, autoenrollment SHOULD validate the certificate as specified in section 4.4.6.3.<33> If the certificate is not valid, the certificate cannot be accepted and MUST not be added to local certificate storage.

      2. Add the certificate in the Certificates.ToBeAdded and Certificates.CurrentCertificates lists.

      3. If CertificateTemplate.EnrollmentFlags has the RemoveInvalidCertificateFromComputerStore flag set, autoenrollment SHOULD add all of the certificates from the Certificates.CurrentCertificates list that are based on the same template to the Certificates.ToBeDeleted list.<34>

      4. Autoenrollment SHOULD add an entry to the ProcessedEnrollments table.<35>

      5. If the CertificateTemplate.SupersededTemplates property is not empty, the autoenrollment SHOULD eliminate obsolete certificates as follows<36>:

         1. Find all certificates based on those templates in the Certificates.CurrentCertificates list.

         2. For each certificate, if its corresponding CertificateTemplate instance has the RemoveInvalidCertificateFromComputerStore bit set in the EnrollmentFlags field, add that certificate to the Certificates.ToBeDeleted list.

      6. The processing for this section is complete.

   4. If the certificate has been set to pending by the issuer, autoenrollment SHOULD add an entry to the PendingRequests table.<37> The processing for this section is complete.

   5. If there was an error while submitting a request, continue with the next issuer.

7. If autoenrollment was unable to renew the certificate from any of the issuers identified in step 1, attempt submitting a new request based on the same template as specified in section 4.4.5.6.3.

4.4.5.7 Renew Manually Enrolled Certificates

Some certificates require manual initial enrollment, but later can be automatically renewed. In this step, autoenrollment SHOULD examine current certificates to determine if any such certificates exist and SHOULD attempt to renew them.<38> The steps are as follows:

1. If AutoEnrollmentPolicy.Options.AutoEnrollmentOptionFlags does not have the AutoEnrollmentOptionFlags.Manage flag set, the processing for this section is complete.

2. For each certificate in the Certificates.CurrentCertificates list:

    1. If a certificate is a Close to Expire Certificate (see section 4.4.5.6.2):

        1. If autoenrollment has already attempted to renew this certificate in the step documented in section 4.4.5.6.4, continue with the next certificate.

        2. Find an instance of the CertificateTemplate for this certificate by using the FindCertificateTemplate algorithm as specified in section 4.4.6.1. If no instance is found, continue with next certificate.

        3. If there is an Acceptable Certificate in the Certificates.CurrentCertificates list based on the same certificate template, continue with the next certificate in the list.

        4. If CertificateTemplate.GeneralFlags does not have the IsMachineType, IsCA, or IsCrossCA flags set, continue with next certificate.

        5. If CertificateTemplate.EnrollmentFlags has the HumanInteractionRequired flag set, autoenrollment SHOULD continue with next certificate.<39>

        6. If CertificateTemplate.SubjectNameFlags has the EnrolleeSuppliesSubjectName or the EnrolleeSuppliesSubjectAltName flags set, the autoenrollment SHOULD<40>:

            1. If the OldCertSuppliesSubjectAndAltName is not set, continue with the next certificate.

            2. Else, when submitting a renewal request, autoenrollment SHOULD copy the subject and alternative subject name information from the certificate being renewed into the request.<41>

        7. If CertificateTemplate.RARequirements.SignatureCount is greater than 1, autoenrollment SHOULD continue with the next certificate.<42>

        8. The autoenrollment SHOULD iterate through the CertificateEnrollmentPolicy.Templates list and check if there are any certificate templates whose CertificateTemplate.SupersededTemplates property contains a string matching the CertificateTemplate.CommonName of the current template.<43> If such a template exists, continue with the next certificate.

        9. Submit renewal request as specified in section 4.4.5.6.4.

    2. Otherwise, continue with the next certificate in the list.

4.4.5.8 Update Local Storage

At this step autoenrollment persists local information that has been modified after it was initialized in section 4.4.5.4. The steps are as follows:

1. Add newly enrolled certificates that are part of the Certificates.ToBeAdded list to the local certificate storage.

2. Delete any certificates from the local storage if they are also part of the Certificates.ToBeDeleted list.

3. Write PendingRequests and ProcessedEnrollments tables back to their implementation specific local storage.

4.4.5.9 Autoenrollment Termination

Autoenrollment does not trigger any other task when it completes. It also does not return any status directly to any of its callers.

4.4.5.10 Processing Rules when Leaving a Domain

This section specifies an alternative flow for autoenrollment that is executed when the task's parameter IsUnjoiningDomain is set to true. In this case autoenrollment does not attempt any certificate enrollment or renewal. The only step in this case is to remove all of the certificates from the Root Certificates, CA Certificates, and Key Archival CA Certificates groups from the local certificate storage specified in section 4.2.1.1.

4.4.6 Task Algorithms

This section contains algorithms that are used in multiple steps of the task.

4.4.6.1 FindCertificateTemplate Algorithm

This algorithm locates an instance of the CertificateTemplate datum in the AutoEnrollmentPolicy datum based on some search criteria.

Input Parameters:

- **Certificate:** A certificate for which a certificate template needs to be located.

Output:

If a certificate template is found, returns an instance of the CertificateTemplate datum. Otherwise returns a logical nothing.

Processing

1. Initialize these variables:

   PolicyId: If an entry exists in the ProcessedEnrollments table corresponding to the certificate, set the PolicyId to the value of the ProcessedEnrollments.PolicyId column. Otherwise, set to empty.

   TemplateOID: If the certificate has the Certificate Template OID extension (1.3.6.1.4.1.311.21.7), as specified in [MS-WCCE] section 2.2.2.5.5.2, the autoenrollment SHOULD set the TemplateOID to the value of that extension<44>. Otherwise, set to empty.

   TemplateName: If the certificate has the Certificate Template Common Name Extension (1.3.6.1.4.1.311.20.2), as specified in [MS-WCCE] section 2.2.2.5.5.1, set TemplateName to the value of that extension. Otherwise set to empty.

2. If TemplateOID and TemplateName are both empty, return nothing.

3. Find a CertificateEnrollmentPolicy instance in the AutoEnrollmentPolicy.EnrollmentPolicies list by using the PolicyId variable. If PolicyId is empty, find a CEP that has CertificateEnrollmentPolicy.IsDefault set to true. If no matching CertificateEnrollmentPolicy is found, return nothing.

4. Within the CertificateEnrollmentPolicy instance identified in the previous step, search for the CertificateTemplate instance in CertificateEnrollmentPolicy.Templates as follows:

    1. If TemplateOID is not empty, search for a CertificateTemplate instance that has SchemaVersion greater than 1 and the OID field equal to the TemplateOID variable. If found return the instance of the CertificateTemplate. Otherwise return nothing.

    2. If TemplateOID is empty, search for a CertificateTemplate instance that has SchemaVersion equal to 1 and CommonName equal to the TemplateName variable. If found return the instance of the CertificateTemplate. Otherwise return nothing.

4.4.6.2 SelectAndOrderIssuers Algorithm

This algorithm specifies how to identify and order a subset of issuers in order to submit a certificate request for a specific certificate template.

Input Parameters:

- **CertificateTemplate:** An instance of the certificate template used for the certificate request.

- **CertificateEnrollmentPolicy:** An instance of the CertificateEnrollmentPolicy to which the CertificateTemplate instance belongs.

- **IsRenewalOnlyAllowed**: A boolean parameter indicating if the Issuer instances that have the Issuer.IsRenewalOnly field set to true are to be included in the output.

Output:

List of Issuer data (specified in section 4.3.2.1.2.2) that supports issuing specific certificate templates. The list is ordered in a specific way for autoenrollment to use when submitting a certificate request.

Assumptions:

The processing rules specified in sections 4.4.5.3.1 and 4.4.5.3.2.4 guarantee that all instances in the CertificateEnrollmentPolicy.Issuers list share the same value in the Issuer.Protocol field. This allows autoenrollment to order Issuers of WCCE from WSTEP.

Processing

1. For each Issuer instance in the CertificateEnrollmentPolicy.Issuers list:

    1. If the Issuer.Templates list does not contain a string that matches the value of CertificateTemplate.CommonName, continue with the next Issuer.

    2. If Issuer.IsRenewalOnly is set to true and IsRenewalOnlyAllowed is set to false, continue with the next Issuer.

    3. Add this Issuer instance to the output list of issuers.

2. If there is more than one instance of the Issuer in the output list:

    1. If Issuers use WSTEP (see assumptions in section 4.2.3):

        1. Order the list based on the Issuer.Priority field in descending order.

        2. For the instances with the same Issuer.Priority, put those that have Issuer.Authentication equal to Kerberos first, then those that have Issuer.Authentication equal to Anonymous.

        3. For the remaining instances, order them as specified in the following step.

2. If Issuers use WCCE (see assumptions):

    1. Select an arbitrary number R between 1 and N where N is the number if Issuer instances in the output list. Note that the arbitrary number here is NOT REQUIRED to be cryptographically random. It need only be uniformly distributed. The intent of selecting an arbitrary number in this step is if there is more than one issuer issuing certificates based on the same template, they receive roughly the same number of requests from autoenrollment clients they work with.

    2. Order the output list such that Issuer instances indexed R though N appear first in the list and Issuer instances 1 through R-1 follow.

3. Return the list to the caller.

4.4.6.3 Verifying Certificates During Enrollment

Verifying certificates during enrollment is in accordance with [RFC5280]. In this case, a root MUST be one of the certificates in the Certificates.Roots list defined in section 4.3.2.3.1, or MUST be one of the roots that the system might already trust in some implementation-specific way. Intermediate CA certificates can be one of the certificates cached in the Certificates.CAs list defined in section 4.3.2.3.1.

4.4.6.4 Verifying CA Exchange Certificates

Verifying CA exchange certificates is a more restrictive case of the Verifying Certificates during Enrollment specified in section 4.4.6.3. In addition to the requirements specified in that section, the immediate issuer of the certificate being validated MUST be one of the certificates in the Certificates. Certificates.KeyArchivalCAs list defined in section 4.3.2.3.1.

4.5 Task Security

There are no task-specific security considerations. Please refer to the Security section of this specification and the Security sections of the referenced protocol Technical Documents.

5 Security

There are no task-specific security considerations. Please refer to the Security sections of the referenced protocol Technical Documents.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft Windows® 2000 operating system
- Microsoft Windows® 2000 Server operating system
- Microsoft Windows® 2000 Advanced Server operating system
- Windows® XP operating system
- Windows® XP operating system Service Pack 1 (SP1)
- Windows® XP operating system Service Pack 2 (SP2)
- Windows® XP operating system Service Pack 3 (SP3)
- Windows Server® 2003 operating system
- Windows Server® 2003 R2 operating system
- Windows Server® 2003 operating system with Service Pack 1 (SP1)
- Windows Server® 2003 operating system with Service Pack 2 (SP2)
- Windows Server® 2003 operating system with Service Pack 3 (SP3)
- Windows Vista® operating system
- Windows Vista® operating system with Service Pack 1 (SP1)
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

<1> Section 4.2.3: The Windows Group Policy user interface that configures CEPs ensures this assumption is valid.

<2> Section 4.2.3: Since certificate templates in Microsoft implementations are always stored in Active Directory in a single container, this assumption is always valid as there cannot be more than one object with the same CommonName attribute in the same container.

<3> Section 4.3.2.3.2: Only Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 retrieve pending requests.

<4> Section 4.3.2.3.3: Only Windows 7 and Windows Server 2008 R2 persist this information.

<5> Section 4.3.6.1: Only Windows Vista, Windows 7, and Windows Server 2008 R2 execute based on this timer.

<6> Section 4.3.6.2: Only on Windows 2000, Windows XP, and Windows Server 2003 is autoenrollment triggered by Group Policy.

<7> Section 4.4.3.1: Only Windows Vista, Windows 7, and Windows Server 2008 R2 execute based on this timer.

<8> Section 4.4.3.2: Only on Windows 2000, Windows XP, and Windows Server 2003 is autoenrollment triggered by Group Policy.

<9> Section 4.4.5.1: Only Windows 7 and Windows Server 2008 R2 support autoenrollment on the computers not joined to the domain.

<10> Section 4.4.5.1: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 always set AutoEnrollmentOptionsFlags to AutoEnrollmentOptionsFlags.Disabled.

<11> Section 4.4.5.1: Windows 2000 does not process this Group Policy and initializes AutoEnrollmentOptionsFlags to the logical OR of the AutoEnrollmentOptionFlags.Enabled and AutoEnrollmentOptionFlags.Enroll.

<12> Section 4.4.5.3: Only Windows 7 and Windows Server 2008 R2 implement Advanced Initialization.

<13> Section 4.4.5.3: Only Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 implement basic initialization.

<14> Section 4.4.5.3.1.1: Windows 2000 ignores templates whose msPKI-Template-Schema-Version attribute is greater than 1. Windows XP and Windows Server 2003 ignore templates whose msPKI-Template-Schema-Version attribute is greater than 2.

<15> Section 4.4.5.3.1.1: Windows 2000 always sets this datum to false at this step.

<16> Section 4.4.5.3.1.1: Windows 2000 always sets this datum to false at this step.

<17> Section 4.4.5.3.1.1: Only Windows XP and Windows Server 2003 set AutoEnrollmentEnabled to false in this case.

<18> Section 4.4.5.3.3: Windows 2000 will only submit requests to the CA whose signing certificate thumbprint matches the subjectIdentifier field of the first TrustedSubject in the trustedSubjects field.

<19> Section 4.4.5.5: Only Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 retrieve pending requests.

<20> Section 4.4.5.5: Only Windows 7 and Windows Server 2008 R2 persist this information.

<21> Section 4.4.5.5: Windows 2000 does not support the RemoveInvalidCertificateFromComputerStore flag.

<22> Section 4.4.5.6.1: Windows 2000 does not process any certificate template when the template's CertificateTemplate.SchemaVersion is greater than 1.

Windows XP and Windows Server 2003 do not process any certificate template when the template's CertificateTemplate.SchemaVersion is greater than 2.

Windows 2000 ignores the HumanInteractionRequired, EnrolleeSuppliesSubjectName, and EnrolleeSuppliesSubjectAltName flags, the CertificateTemplate.RARequirements.SignatureCount field, and the CertificateTeplate.SupersededTemplates list.

<23> Section 4.4.5.6.2: Only Windows XP SP2, Windows XP SP3, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 monitor DNS name changes.

<24> Section 4.4.5.6.2: Only Windows XP SP2 and later, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 terminate after 3 attempts. Other versions of Windows do not monitor DNS names changes.

<25> Section 4.4.5.6.3: Windows 2000 does not support co-signing certificate requests to satisfy RA signing requirements.

<26> Section 4.4.5.6.3: Windows 2000 does not support key archival.

<27> Section 4.4.5.6.3: Only Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 perform certificate validation in this step.

<28> Section 4.4.5.6.3: Only Windows 7 persists information specified in the ProccessedEnrollments table.

<29> Section 4.4.5.6.3: Windows 2000 ignores the CertificateTemplate.SupersededTemplates list.

<30> Section 4.4.5.6.3: Only Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 retrieve pending requests.

<31> Section 4.4.5.6.4: Windows 2000 does not support co-signing certificate requests to satisfy RA signing requirements.

<32> Section 4.4.5.6.4: Windows 2000 does not support key archival.

<33> Section 4.4.5.6.4: Only Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 perform certificate validation in this step.

<34> Section 4.4.5.6.4: Windows 2000 does not support the RemoveInvalidCertificateFromComputerStore flag.

<35> Section 4.4.5.6.4: Only Windows 7 persists information specified in the ProccessedEnrollments table.

<36> Section 4.4.5.6.4: Windows 2000 ignores the CertificateTemplate.SupersededTemplates list.

<37> Section 4.4.5.6.4: Only Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 retrieve pending requests.

<38> Section 4.4.5.7: Only Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 implement this step.

<39> Section 4.4.5.7: Windows 2000 does not support the HumanInteractionRequired flag.

<40> Section 4.4.5.7: Windows 2000 ignores the EnrolleeSuppliesSubjectName and EnrolleeSuppliesSubjectAltName flags.

<41> Section 4.4.5.7: Only Windows 7 and Windows Server 2008 R2 support using the old certificate to populate subject information in this case.

<42> Section 4.4.5.7: Windows 2000 does not support co-signing certificate requests to satisfy RA signing requirements.

<43> Section 4.4.5.7: Windows 2000 ignores the CertificateTemplate.SupersededTemplates list.

<44> Section 4.4.6.1: Windows 2000 does not process this extension and sets TemplateOID to empty.

7 Change Tracking

This section identifies changes made to [MS-CAESO] protocol documentation prior to October 2008. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

Major and minor changes can be described further using the following revision types:

- New content added.
- Content update.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
4.4.4.2 Autoenrollment in the Domain environment with CRTD/WCCE Protocols	Replaced figure for message exchange on a computer joined to a domain. Added a new step two for sequence of steps following the new figure.	N	New content added.

8 Index