[MS-GPOD]:
Group Policy Protocols Overview

This document provides an overview of the Group Policy Protocols Overview Protocol Family. It is intended for use in conjunction with the Microsoft Protocol Technical Documents, publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Protocol System Document does not require the use of Microsoft programming tools or programming environments in order to implement the Protocols in the System. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

Abstract

The Group Policy system consists of a set of core protocols that are used to create, read, update, and remove Group Policy Objects (section 1.1.3). The Group Policy system enables the Group Policy client to retrieve policy settings from a Group Policy server and enables an Administrative tool to retrieve, create, update, and delete policy settings on a Group Policy server. The base functionality of the Group Policy system, as described in [MS-GPOL], can be extended through client-side extensions that implement application-specific policy settings, and through Administrative tool extensions that implement authored configuration settings. These extensions to the Group Policy: Core Protocol [MS-GPOL] consist of the protocols specified in [MS-GPAC], [MS-GPDPC], [MS-GPEF], [MS-GPFAS], [MS-GPFR], [MS-GPIE], [MS-GPIPSEC], [MS-GPNAP], [MS-GPNRPT], [MS-GPPREF], [MS-GPREG], [MS-GPSB], [MS-GPSCR], [MS-GPSI], and [MS-GPWL].

This document describes the intended functionality of the Group Policy system and how the protocols in this system interact with each other. It provides examples of some of the common use cases. 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 make up this system.

Revision Summary

Date	Revision History	Revision Class	Comments
09/23/2011	1.0	New	Released new document.
12/16/2011	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	2.0	Major	Significantly changed the technical content.
07/12/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	3.0	Major	Significantly changed the technical content.

Contents

1 Introduction

Organizations face increasingly complex challenges in managing their IT infrastructures. They must deliver and maintain customized desktop configurations for many types of workers, including mobile users, information workers, and others assigned to strictly defined tasks, such as data entry. Changes to standard operating system images might be required on an ongoing basis. Security settings and updates must be delivered efficiently to all the computers and devices in the organization. New users need to be productive quickly without costly training. In the event of a computer failure or disaster, service must be restored with a minimum of data loss and interruption.

Typically, IT departments must respond to various factors that require changes in the IT environment. These changes might consist of requirements such as the following:

- Installation of new operating systems and applications.
- Updates to operating systems and applications.
- Installation of new hardware.
- Configuration changes to support new business needs.
- Management of centralized control of resources.
- Configuration changes that enhance security.
- Addition of new users and computers in the **domain**.

Group Policy enables IT departments to efficiently respond to requirements such as these, by providing the necessary framework to deliver computer configuration and **policy setting** changes that target specific computers and users. These policy settings are specified by a **Group Policy Administrator (GP Administrator)**.

1.1 Conceptual Overview

Group Policy provides the infrastructure used to deliver and apply one or more desired configurations or policy settings to a set of targeted users and computers within a **directory service (DS)** environment. Policy settings are administrative directives that define computer-wide and user-specific setting configurations. Administrators can define policy settings once and rely on the Windows operating system to enforce that policy. This section provides a conceptual overview of the major components and processes of the **Group Policy System (GP System)**, which includes the following:

- Group Policy core protocol
- Group Policy settings
- Group Policy Objects
- Group Policy extensions
- Group Policy data storage
- Group Policy administration
- Group Policy application

- Group Policy SOM
- Group Policy management
- Group Policy structure
- GPO configuration model

1.1.1 Group Policy Core Protocol

The Group Policy: Core Protocol [MS-GPOL] is a client/server protocol that enables a **Group Policy Client (GP Client)** to discover and retrieve policy settings that are created by a GP Administrator (a domain administrator) and stored as a **Group Policy Object (GPO)** in the **Active Directory** directory service [MS-ADTS]. A GP Administrator creates policy settings to control GP Client behavior and capabilities. The Group Policy: Core Protocol then facilitates the communication of the administrator-defined policies from the **Group Policy (GP) Server** to domain members such as a GP Client or a user interactively logged on to the GP Client computer.

For example, a GP Administrator might want to target the firewall configuration of a group of client computers to open a specific port on each one. The GP Administrator can use the GP System to create a policy setting that specifies the firewall configuration, and the Group Policy: Core Protocol enables it to be delivered to GP Clients.

The Group Policy: Core Protocol has the following primary modes of operation:

- **Policy administration** — the policy administration mode is driven by the GP Administrator, where the **Administrative tool** is used to create or modify behavior and capability settings of computers and users.
- **Policy application** — the **policy application** mode is driven by the GP Client, where the GP Client retrieves administrator-specified behavior and capability settings from the GP Server, with the assistance of the Group Policy: Core Protocol.

The Group Policy: Core Protocol of itself does not define policy settings. The Group Policy: Core Protocol is implemented by the **core Group Policy engine (core GP engine)**, which issues the network requests that constitute the policy application sequence. The Group Policy: Core Protocol is the actual network traffic for the associated message sequences. Some of the major tasks that are handled by the core GP engine on behalf of the Group Policy: Core Protocol are described as follows:

- **Applying policy** — the core GP engine is responsible for the application of Group Policy at regular refresh intervals; this process is called background policy application. It also applies Group Policy each time that a GP Client computer starts up or shuts down, or a user logs on or logs off the GP Client computer; this process is called foreground policy application.
- **Locating GPOs** — the core GP engine locates GPOs from the appropriate domain, **site**, and **organizational unit (OU)** containers in Active Directory, using the **gpLink** attribute of a **scope of management (SOM)** container object (section 1.1.8) that specifies the **distinguished names (DNs)** of applicable GPOs.
- **Filtering and ordering GPOs** — the core GP engine determines whether the GP Administrator specified that certain GPOs should be filtered out or whether a GPO application order was configured.
- **Invoking execution of CSEs under specified conditions** — the core GP engine can run **client-side extensions (CSEs)** under specific conditions, as configured in the **registry**.

- **Maintaining CSE version numbers and history** — the core GP engine maintains a list of version numbers for CSEs and also keeps a registry-based history that records when policy settings were last applied by a CSE and whether that application was successful.
- **Calling CSEs** — upon determining that a CSE should be executed, the core GP engine loads the CSE's dynamic link library (DLL) and accesses its execution entry point for execution.
- **Providing notification of policy changes** — following policy application, the core GP engine fires the **PolicyChange** event to indicate that a policy has changed. Applications can subscribe to this event and receive notification of policy application.

Note that the core GP engine is installed on all GP Clients.

1.1.2 Group Policy Settings

There are two types of policy settings, as follows:

- **User policy settings** — specify capabilities and behaviors for interactively logged-on users. These settings can also affect different users who are logged on to the same computer. Examples of such settings include the user's default location for saving documents, or the desktop background image for a user.

  Some settings will affect the user regardless of the computer they log on to. For example, policy source mode, as described in [MS-GPOL] section 3.2.1.2, can override user policy settings by causing computer policy settings to be applied to the user.
- **Computer policy settings** — specify capabilities and behaviors for individual computers (even when no users are logged on). Computer policy settings can also globally affect every user who logs on to the computer. Examples include policy settings that enable a computer to host a web server, schedule automated disk backups of the computer, or specify a standard web home page for all users of the computer.

The Group Policy: Core Protocol enables GP Clients to discover and retrieve these policy settings. The policy settings that are applied to the GP Client depend on the filtered GPO list, which is derived and prioritized by the core GP engine on the GP Client. The filtered GPO list is a set of GPOs that have passed various test criteria to verify whether they are permitted or denied applicability on the GP Client, as described in [MS-GPOL] section 3.2.1.5.

The application of Group Policy settings to the GP Client is discussed further in section 1.1.7 and an example with message sequences is provided in section 3.2.

1.1.3 Group Policy Objects

The GP System utilizes several protocols to create, read, update, and remove GPOs. Group Policy uses a document-centric approach to create, store, and associate policy settings. Group Policy settings are contained in GPOs to maintain various sets of behavior specifications. A GPO is a virtual object that stores policy-setting information with two components:

- **Directory service** — GPOs and their attributes are stored in a directory service, such as Active Directory [MS-ADTS].<1>
- **File share** — GPOs also store policy settings information on a local or remote file **share**, such as the **Group Policy file share (GP FS)**.<2>

Both of these storage components can reside on the GP Server. Through the hierarchical modeling of Active Directory, GPOs can be linked to site, domain, and organizational unit (OU) containers to

enable policy settings to be applied to target users and computers associated with these containers. This infrastructure provides a high degree of flexibility that enables the GP Administrator to customize configurations, such as delivering a specific piece of software to specialized users based on their membership in an OU.

A GPO is uniquely identified on the system by a **globally unique identifier (GUID)**. GPO settings are evaluated by the GP Client through the hierarchical nature of Active Directory and by interpreting the extension policy file data on the GP FS. The processes for creating a GPO are described in section 2.1.3.2.1.

1.1.4 Group Policy Extensions

GP System functionality can be enhanced through the implementation of **Group Policy Extension (GP Extension)** functionalities. GP Extensions consist of client-side extensions (CSEs) and **Administrative tool extensions**. Most GP Extensions have these two extension implementation pairs; a CSE that applies policy settings, and an associated administrative-side extension that plugs into the Administrative tool to define and set policy settings. GP Extensions are invoked by the Administrative tool when creating or updating policy settings. GP Extensions are also invoked by the core GP engine when applying policy on a **policy target** such as a GP Client.

A few GP Extensions have only an administrative-side, as shown in the diagram of section 2.1.2.2 and as described in section 2.2. In most cases, these depend on another CSE to perform client-side functions. For GP Extensions that implement both a client-side and administrative-side, the Extension list stored in a GPO specifies a list of GUID pairs. The first GUID of each pair is the **CSE GUID**, and the second GUID of each pair is an **Administrative tool extension GUID**. Extension lists are maintained by the **gPCMachineExtensionNames** and **gPCUserExtensionNames** attributes of a GPO, the former of which contains GP Extension GUID pairs that apply to computer policy settings, and the latter of which contain GP Extension GUID pairs that apply to user policy settings.

CSEs and Administrative tool extensions function in the following manner:

- CSEs — enable the application of explicit functionality to various subsystems on GP Client computers. This is accomplished by implementing application-specific policy settings, such as the client security policies specified in [MS-GPSB], on GP Client computers.

  The CSEs that apply to a set of policy targets are designated by the Extension list of a GPO. Each CSE in the GPO Extension list is represented as a GUID that is associated with a CSE protocol, sometimes referred to as a client-side plug-in, residing on the GP Client computer. The GUID enables the core GP engine on the GP Client to locate and invoke the CSE protocol, which in turn applies policy settings to the policy target. These settings are all defined by the GPO, which includes the extension policy files that reside on the GP FS.

- CSE protocols depend on the execution of the core GP engine on the GP Client for the following:

  - To identify GPOs that a CSE should query to obtain the stored settings for that extension.

  - To provide the message sequences for retrieving the CSE settings stored in the logical part of a GPO.

  - To invoke a **remote file access (RFA)** protocol to retrieve extension-related policy settings in the extension policy files on the GP FS.

- Administrative tool extensions — facilitate authoring and modification of specific administrative settings related to extended functionality, such as the security-based settings specified in [MS-GPIPSEC].

The Administrative tool extensions that apply to a set of policy targets are designated by the Extension list of a GPO. Each Administrative tool extension in the GPO Extension list is represented as a GUID that is associated with an administrative-side extension protocol, sometimes referred to as an administrative plug-in, residing on the computer hosting the Administrative tool. This GUID enables the Administrative tool to locate the extension for administering the GPO settings related to that particular extension. Settings for such extensions, for example, those specified in [MS-GPSB], are typically stored in Active Directory via the **Lightweight Directory Access Protocol (LDAP)** [RFC2251] and in the GP FS via a remote file access protocol.

Administrative tool extension protocols depend on the Administrative tool for the following:

- To identify GPOs that the administrative-side extension should query to obtain the stored settings for that extension.
- To provide the message sequences for updating the administrative-side extension settings stored in the logical part of a GPO.
- To invoke a remote file access protocol to retrieve or store extension-related policy settings in the extension policy files on the GP FS.

Policy settings for a given class of extension functionality are communicated by a CSE protocol itself and not directly by the core GP engine. The behavior of a given protocol extension is specified in the documentation for that extension. For example, the behavior of the Group Policy: IP Security (IPsec) Protocol is documented in [MS-GPIPSEC].

The extension protocols that are native to the GP System are specified in section 2.2. However, vendors can extend the functionality of the GP System by implementing custom GP Extensions, as described in [MS-GPOL] section 1.8.

1.1.5 Group Policy Data Storage

The GP System writes and reads policy information to and from the **Group Policy data store (GP DS)**, which contains the following components:

- **Active Directory data store** — this store is part of **AD DS** implemented on the GP Server and serves as a repository for GPOs. GPOs are maintained in Active Directory as type *groupPolicyContainer* objects within a Group Policy Objects container and are accessed via LDAP calls. A GPO maintains policy configuration settings that apply to policy targets, such as a user interactively logged on to a GP Client.

  Some policy configuration settings that are stored in GPOs can be regarded as Group Policy metadata because this information (section 1.1.7.3), embedded in the attributes of Active Directory objects, is used to identify Group Policy configurations such as SOM, extension applicability, and the policy file location, rather than the actual policy settings that are applied to GP Clients. For example, a GPO contains attributes that specify a user extension list and computer extension list that are specific to that particular GPO configuration. These lists specify the extension protocols that apply to target users and computers, for which the GPO is configured. The actual settings for these extensions are stored in the GP FS and comprise the actual policy settings that are to be applied by CSEs on the GP Client. However, it is a GPO attribute in Active Directory that holds the pointer to the file share location where the CSE policy settings reside.

- **GP FS data store** — this store persists user and computer policy settings and also maintains a file that specifies GPO version information. If a GPO has registry settings, the GP FS data store will contain the file registry.pol, which stores the registry settings that are generated by

configuring **Administrative template** items with a management tool such as the **Group Policy Management Console (GPMC)**. The GP FS store can exist locally on the GP Server or remotely on a file share, where policy data is retrieved via a remote file access protocol.<3>

Policy settings for GP Extensions are persisted in extension policy files on the GP FS and/or in a GPO. These settings are retrieved for the application of extension policy settings on the GP Client. For more information about how extension settings are applied to a GP Client, refer to section 1.1.7.4.

1.1.6 Group Policy Administration

Group Policy administration consists of creating new GPOs, deleting GPOs, and editing existing policy settings, as described in section 2.1.3.2. In policy administration mode, the GP Administrator uses the Administrative tool to locate the GP Server and interact with the same Active Directory objects as occurs during policy application by the GP Client. However, the Administrative tool does not directly apply policy settings to the GP Client. Rather, it only enables the GP Administrator to create, update, or delete policy settings, and then update the GP Server with those configurations via LDAP. Thereafter, following a Group Policy trigger, the GP Client accesses those updated or new objects and associated settings during the policy application process.

Policy administration also applies to modifying and authoring GP Extension settings, in addition to authoring Administrative template settings:

- **Modifying extension settings** — GPOs that contain classes of settings for a specific Administrative tool extension are identified by an Administrative tool extension GUID, which is used to invoke the extension protocol that can retrieve the associated settings from a GPO for updating. The retrieval process is facilitated by the Administrative tool, which invokes LDAP and a remote file access protocol to access the settings. After extension settings are edited, the Administrative tool sends an LDAP **modifyRequest** to update the logical component of a GPO and a remote file access open/write request to update the GP FS location where the extension policy files reside.
- **Authoring extension settings** — when authoring new extension settings for a new GPO, the GP Administrator must first create the new GPO by following the processes outlined in section 2.1.3.2.1. Thereafter, the GP Administrator can use the Administrative tool to author settings for an Administrative tool extension. When this occurs, the Administrative tool sends an LDAP **addRequest** to Active Directory to write the Administrative tool extension GUID and client-side extension GUID (CSE GUID) to the Extension lists of the GPO. These attributes enable the GP Client to determine which GP Extensions should apply their settings to the GP Client during the policy application process.
- **Configuring administrative template settings** — policy administration includes the configuration of Administrative template settings that are accessible from a management tool such as the GPMC. The Administrative template policy configurations generate registry settings that are stored in the file registry.pol, which is located on the GP FS. During policy application, this file is read by Group Policy: Registry Extension Encoding protocol [MS-GPREG], and its settings are applied to the GP Client registry.

1.1.7 Group Policy Application

The policy application process utilizes a pull model when retrieving Group Policy data that is to be applied to the GP Client. For example, when retrieving policy settings, the GP Client polls the GP Server to check for new policy settings specified by the GP Administrator that affect either the client computer itself or a domain user that is interactively logged on to the client computer.

To accommodate these requirements, the application of Group Policy is specified in two modes. The first is computer policy mode, which affects the client computer and all users logging on to the client computer; the second is user policy mode, which only affects the users who log on to the client computer. For user policy mode, the policy target is a domain user account, for which policy settings are retrieved. For computer policy mode, the policy target is a domain computer account, for which policy settings are retrieved.

The application of Group Policy is triggered by specific events, such as a user logon or computer startup, as described in section 1.1.7.1. The following is a conceptual summary of the processes that occur whenever Group Policy is applied. The specified actions of the GP Client are carried out by the core GP engine running on the GP Client:

- **DC discovery** — the GP Client searches for a **domain controller (DC)** and connects to Active Directory. The communication details for this process are described in section 2.1.3.1.1.
- **DN discovery** — the GP Client attempts to discover the DN of the policy target, which is used in querying for applicable GPOs, as described in [MS-GPOL] section 3.2.5.1.2.
- **Domain SOM search** — the GP Client queries the GP Server for any GPOs that are linked to the domain, which therefore will apply to the GP Client policy target account. The communication details for this process are described in section 2.1.3.1.2.

  SOM defines hierarchical levels from which GPOs apply to policy targets; these levels include the domain, site, and OU levels. For example, a domain SOM search returns the DNs of all GPOs that are linked to the domain container, which holds one or more policy targets to which the GPOs will apply. For more information about SOM, refer to section 1.1.8.
- **Site SOM search** — the GP Client queries the GP Server for any GPOs that are linked to the site container, which therefore will apply to the GP Client policy target account. The communication details for this process are described in section 2.1.3.1.3.
- **GPO search** — the GP Client queries the collection of GPOs defined by the SOM, to obtain various information sets that include the GPO security descriptor, the GPO file system path, GPO version number, the GUIDs of extensions that apply to the GP Client, and other GPO metadata, as described in section 1.1.7.3. Communication details for this process are described in section 2.1.3.1.4.
- **GPO filter evaluation** — the GP Client processes each GPO to check its functionality version, disabled/enabled status, empty status, and security rights. These checks determine whether the GPO is allowed or denied applicability on the GP Client, as described in [MS-GPOL] section 3.2.5.1.6.
- **WMI filter evaluation** — the GP Client queries the GP Server for any WMI filters that limit the set of GPOs that are to be used by GP Extensions. The communication details for this process are described in section 2.1.3.1.5.
- **Link speed discovery** — the GP Client attempts to estimate the network speed of its connection to the GP Server, as described in section 2.1.3.1.6.
- **Extension protocol sequences** — the GP Client determines which CSEs apply to it for user policy mode and computer policy mode, and then invokes a protocol sequence that causes each CSE to apply its settings to the GP Client, as described in section 1.1.7.4.
- **Policy change event** — the GP Client raises a local PolicyChange event at the end of policy application to indicate that a policy has changed, as described in section 2.8.2.

The programmatic details for these processes are specified in [MS-GPOL] section 3.2.5.1. Formats for the messages associated with these processes are specified in [MS-GPOL] section 2.2.

1.1.7.1 Triggering Group Policy Application

The application of Group Policy is triggered by certain events that occur, at which time the core GP engine is invoked to initiate the application process. The following describes the events that trigger the application of Group Policy in computer policy mode and user policy mode.

Computer policy mode — the following events trigger the application of Group Policy to the GP Client computer:

- Computer startup
- Computer shutdown
- Periodic refresh timer

User policy mode — the following events trigger the application of Group Policy to the user on the GP Client computer:

- User logon
- User logoff
- Periodic refresh timer

Note that the periodic refresh timer can be superseded to apply Group Policy at any time, as described in section 2.8.2.

The application of Group Policy in either computer policy mode or user policy mode involves the application of both Administrative template settings and extension settings. However, before this can occur, it is necessary to discover the domain controller that contains the GPOs that apply to the policy targets, as described in the sections that follow.

1.1.7.2 Discovering the Server and Applicable GPOs

Policy application starts with an initial discovery step by the GP Client to locate a domain controller, as described in [MS-ADOD] (section 3.1.1). This step is necessary to identify the domain controller that contains the Group Policy Objects container for the domain in which the GP Client resides. After locating a domain controller, the core GP engine on the GP Client performs a set of LDAP queries to Active Directory on the GP Server.

The initial queries determine which GPOs were assigned to the policy target accounts by the GP Administrator, which include the domain computer account and the account of the user logged on to the GP Client. The remaining queries assemble the logical GPO from its component parts, which include the components stored in Active Directory and in the file system (GP FS), as described in sections 1.1.7.3 and 1.1.7.4.

To discover the GPOs that apply to the policy target account, the initial queries perform a search on the Active Directory hierarchy containing the policy target accounts. This hierarchy typically contains a domain root container that has OU containers within it, which in turn contain domain account objects. GPOs can be associated with any of these containers, to define the scope of Group Policy applicability, and therefore apply to any domain accounts that exist within them.

Essentially, the initial queries locate the Group Policy Objects container for the domain to discover the GPOs contained within it, along with the SOM container objects (domain, sites, and/or OUs) to

which the GPOs are linked, so that a **Resultant Set of Policy (RSoP)** can be achieved on the GP Client.

1.1.7.3 Retrieving GPO Attributes

Using information obtained from the initial queries, the GP Client uses another set of queries to assemble the logical GPO from its component parts that exist in Active Directory and on the GP FS. These queries utilize LDAP to return GPO attributes that are associated with the policy target accounts, as follows:

- **Extension list** — provides a list of GUIDs, contained within a GPO, that identify classes of settings (associated with extension protocols) to be applied to the GP Client.
- **Filtering** — enables specified policy target accounts to be excluded from association with a GPO.
- **GPO path directories** — provides the location of extension policy files and the GPO version information file (gpt.ini) stored on the GP FS.
- **GPO security descriptor** — determines whether a GPO is allowed or denied, based on an **access control entry (ACE)** right that applies to the Active Directory security group in which the policy target account is a member.
- **Precedence** — enables resolution of conflicts between settings of different GPOs.
- **Version** — specifies the version of a GPO, for use in determining whether a policy target requires updating.

Using the **GPO path** directory information, the core GP engine on the GP Client invokes a remote file access protocol to query the GP FS to locate the file containing the GPO version information and the directories containing the extension policy files.

The GP Client uses all of the previous information to compute a list of the GPOs that apply to it, along with the GUIDs that identify the extensions whose settings are to be applied in the next and final steps of policy application.

1.1.7.4 Retrieving and Applying Extension Settings

The last steps of policy application involve the actual retrieval and application of extension settings. The GP Client uses its computed list of GPOs with different classes of settings to begin the process. For each class of settings in the list, the GP Client uses a CSE GUID to identify a CSE (a GP Extension), such as the Group Policy: Registry Extension Encoding protocol [MS-GPREG]), that contains corresponding extension settings. The core GP engine on the GP Client invokes a protocol sequence that uses the CSE GUID to locate the settings associated with the CSEs that are stored in the GPO on the GP Server. The CSE retrieves the associated settings stored in the GPO by using LDAP to access the Active Directory-based component of the GPO and by using a remote file access protocol to access the GP FS-based component of the GPO. When the settings are successfully retrieved, the CSE on the GP Client interprets the settings and enforces the behaviors they specify. The GP Client of itself cannot interpret and enforce settings because it does not recognize the internal details of GP Extension.

The following summary provides some additional context to the preceding discussion by further clarifying the retrieval and application of extension policy settings to a GP Client via a CSE protocol.

- Prior to the Group Policy trigger, the GP Administrator will have configured extension settings with the Administrative tool for a policy target.

This creates an extension policy file, which is then associated with a GPO in Active Directory and stored on the GP FS. For some extensions, settings are stored on the GP FS and/or in the GPO itself.

- A Group Policy trigger causes the GP Client to invoke the core GP engine to initiate the retrieval of attributes and policy settings from a GPO (or set of GPOs) that apply to the GP Client and that specify the applicable CSEs.
- The core GP engine initiates an LDAP call that reads the GUID of the CSE protocol from a GPO that applies to the GP Client and then invokes the CSE protocol for policy application.
- The CSE protocol reads and parses the settings of the extension policy file on the GP FS and/or reads the extension settings stored in the GPO itself, and then applies them to the appropriate GP Client subsystem.

1.1.8 Group Policy SOM

The collection of GPOs that apply to a set of policy targets is considered the scope of management. SOM tells the core GP engine which site-, domain-, or OU-level GPOs apply to a policy target. During policy application, the core GP engine searches for GPOs in the Group Policy Objects container (section 1.1.9) in Active Directory and then determines the SOM by inquiring which site, domain, and OU containers the GPOs are linked to, along with the order of precedence in which they apply to the policy target.

SOM is not an object itself but rather a construct that describes how Group Policy is applied to policy targets from Active Directory hierarchical levels using GPOs. SOM associates GPOs with policy targets that exist within a site, domain, or OU container object, in accordance with the GPOs that are linked to such objects. This association is established, in order of GPO precedence, within a list of GPO DNs that is contained by the **gpLink** attribute of the site, domain, or OU container object. For example, there might be GPOs at the domain and OU level that apply to a particular set of policy targets, and the order of precedence might be that the OU-level GPO overrides a GPO at the domain-level in terms of certain policy settings that have priority. The GPO applicability and precedence configuration is resolved through various filtering evaluations that result in a final computed list of GPOs whose settings will be applied to one or more policy targets.

The following attributes must be maintained by all SOM containers:

- **SOM DN** — the DN of the SOM container, such as a domain container.
- **gpLink** — a directory string value for the **gpLink** attribute of the SOM container.
- **gpOptions** — an integer value that is used to set the Group Policy inheritance configuration among hierarchical SOM containers. See [MS-GPOL] section 2.2.2 for more information.
- **SOM object type** — specifies the type of Active Directory container that the SOM represents; one of the following values is assigned to this attribute:
  - GPLinkOrganizationalUnit: the SOM container object represents an OU.
  - GPLinkDomain: the SOM container object represents a domain.
  - GPLinkSite: the SOM container object represents a site.

An Active Directory container in the GP System comes into scope of management when one or more GPOs are linked to it.

1.1.9 Group Policy Management

Group Policy can be managed from an interface such as the GPMC, a custom application, or a command line tool. GPOs exist within a Group Policy Objects container in Active Directory, as represented in the following diagram, and can be managed by a GP Administrator:

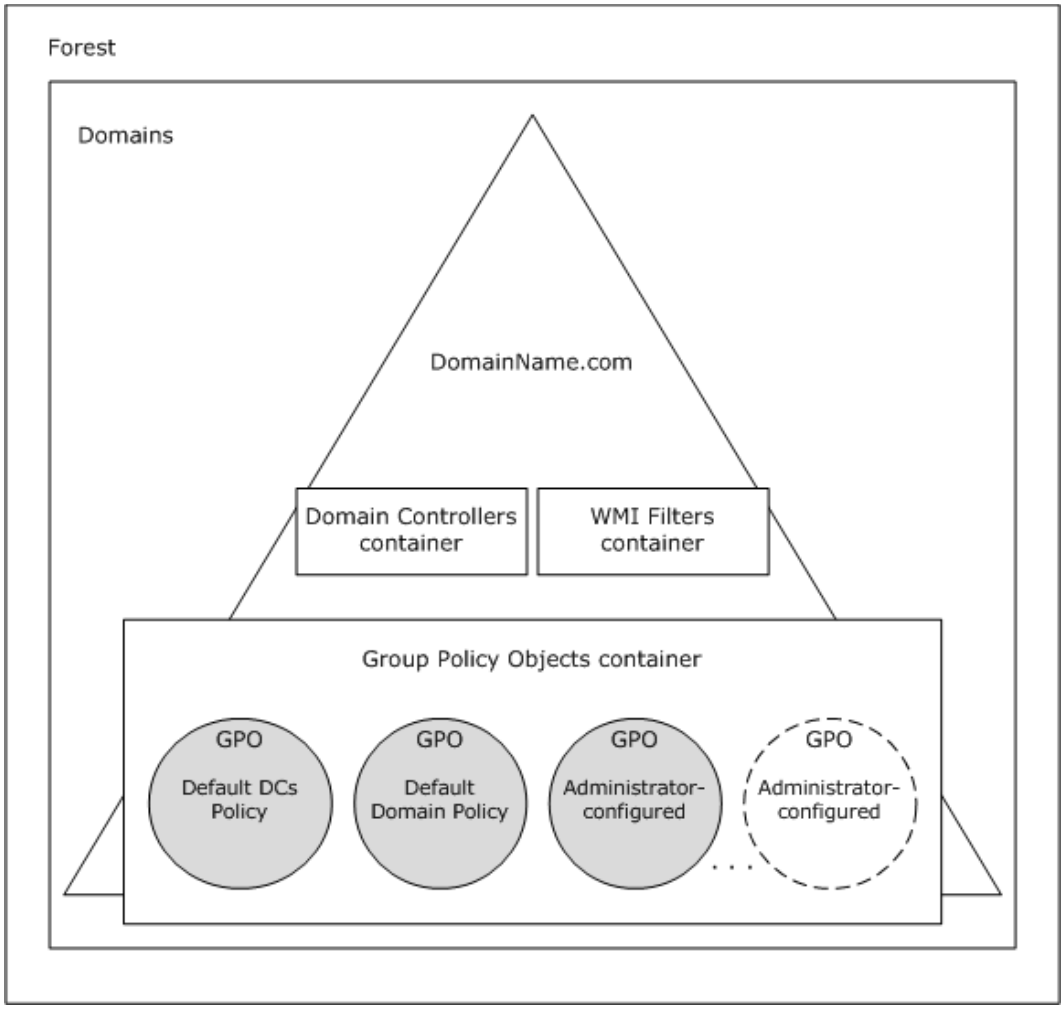

Figure 1: GPO location in Active Directory

The GP Administrator uses the Active Directory container objects for the domain shown in the diagram to manage Group Policy. When GP Administrators need to manage GPOs, they can create a new GPO, delete a GPO, or edit an existing one. They can also manage policy settings via other default GPOs for the domain. The default objects and containers that can be accessed in a domain for management purposes are described as follows:

- **Domain Controllers container** — a default container that is automatically created when a server is promoted to a domain controller. It is linked to the domain controller's OU and manages security settings for all domain controllers in a domain.
- **WMI Filters container** — a default container that is automatically created when a server is promoted to a domain controller. It holds WMI filter objects, created by the GP Administrator,

that are linked to GPOs to exempt specific GP Clients from the extension policy settings that they hold. For information about evaluating WMI filters, refer to [MS-GPOL] section 3.2.5.1.7.

- **Group Policy Objects container** — a default container that is automatically created when a server is promoted to a domain controller. It provides a hierarchical repository for GPOs that are created by the GP Administrator with the use of the Administrative tool. For more information about how GPOs are created, refer to section 2.1.3.2.1.
- **Default Domain Controllers Policy** — a default GPO that is automatically created and linked to the domain whenever a server is promoted to a domain controller. This GPO represents the default policy that is applied to all domain controllers in the Domain Controllers container.
- **Default Domain Policy** — a default GPO that is automatically created and linked to the domain whenever a server is promoted to a domain controller. It has the highest precedence of all GPOs linked to the domain, and it applies to all users and computers in the domain. The Default Domain Policy GPO is generally used to manage default account settings, although there are exceptions to this practice. For other areas of policy management, new GPOs should be created; however, some policy settings are best configured at the domain level, and there are no restrictions against doing so.
- **Administrator-configured** — a GPO that is created by the GP Administrator to generate custom Group Policy settings for policy targets such as a GP Client computer.

1.1.10 Group Policy Structure

Group Policy structure is modeled after Active Directory structure, in that it has both physical and logical components. At the core of Active Directory's physical architecture is an extensible storage engine that reads and writes information to the Active Directory data store. This engine makes use of the logical, object-based hierarchy that represents data store information.

Group Policy structure is similar to that of Active Directory, because it maintains both a logical and physical representation of GPOs, as follows:

- **Logical component** — consists of a Group Policy container (GPC) object, which is stored in the Group Policy Objects container of Active Directory. The GPC object contains attributes that specify basic GPO information, such as the following:
  - GPO display name
  - **GPO path** to the extension policy and group policy template (GPT) files.
  - GPO version number
  - GPO status
  - **Access control list (ACL)**
  - GUID-references to the CSEs that are to be invoked when the core GP engine on the GP Client processes the GPO.

  When the GP Administrator creates a GPO, Active Directory creates a GPC object for that GPO, as described in section 2.1.3.2.1. This GPC is a container object of the *groupPolicyContainer* class and is named with a GUID that identifies the GPO. The GPC is stored under the CN=Policies,CN=System container within the domain. The Administrative tool and the GP Client locate this container according to its DN, which is the exact path to the GPC object in the Active Directory data store.

- **Physical component** — consists of the GP FS component that stores GPT and GP Extension settings on a domain controller or other server.

  The physical component of a GPO is represented through a series of files containing Administrative template and extension policy settings that are stored on disk. These files contain numerous policy settings along with the state of these settings. These files are stored in Machine and User subdirectories along with the associated GPO version file gpt.ini, in the following path, which is also known as the GPO path: <dns domain name>\<GP FS-name>\<dns domain name>\Policies\<guid>\.

  Whenever the GP Administrator creates a new GPO, the <guid> folder in this path is automatically created and named with the GUID of the GPO. Within the <guid> folder are Machine and User subdirectories that contain extension policy settings and Administrative template configuration items. During policy administration, when the GP Administrator creates or modifies GP Extension or Administrative template settings, the Administrative tool locates the policy files according to the <guid> in the GPO path. During policy application, the GP Client locates the policy files in the same manner.

1.1.11 GPO Configuration Model

The GPO configuration model accommodates settings for users and computers, and includes Software, Windows, and Administrative Templates settings for both user and computer configurations. Software settings enable the GP Administrator to specify software applications to be installed on GP Client computers; Windows settings hold the extension configurations; and Administrative Templates represents GP Client subsystems for which registry settings can be configured.

Policy targets in Active Directory are individual user and computer accounts that exist within domain, site, or OU containers. Each site, domain, and OU has a **gpLink** attribute that associates it with one or more GPC objects, which represent GPOs in Active Directory. Each GPO contains various attributes that are associated with users and computers; this includes an attribute that specifies the GPO path to policy files that store user and computer policy settings. The file system component of a GPO itself is configured with directories that hold policy data for users and computers. Therefore, when the GP Administrator views a GPO in a management interface such as the GPMC, two different sets of configuration settings are provided, as indicated in the diagram of section 2.1.3.2.2:

- **User Configuration** — contains all information related to user policies that GP Clients retrieve during policy application in user policy mode, which includes data for the applicable CSEs. These CSEs store all server state for policy settings within the user configuration, in a format that is described in corresponding extension specifications.

- **Computer Configuration** — contains all information related to computer policies that GP Clients retrieve during policy application in computer policy mode, which includes data for the applicable CSEs. These CSEs store all server state for policy settings within the computer configuration, in a format that is described in corresponding extension specifications.

The logical component of each GPO contains a user Extension list and a computer Extension list that specifies the GUIDs of CSEs that apply to users and computers, respectively. The actual settings for these extensions are stored in the physical (file system) component of the GPO, as described in section 1.1.10. The extension settings for the user and computer configuration are configurable from the Administrative tool. When the GP Administrator creates or modifies extension settings, they are sent to the GP DS. For example, any modifications to GPO attributes are communicated to Active Directory on the GP Server via LDAP [RFC2251], while the actual extension policy settings are communicated to the GP FS via a remote file access protocol, both of which protocols are invoked by the Administrative tool.

1.2 Glossary

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

access control entry (ACE)
access control list (ACL)
Active Directory
directory
directory service (DS)
distinguished name (DN)
domain
domain controller (DC)
Domain Name System (DNS)
domain naming context (domain NC)
Encrypting File System (EFS)
forest
globally unique identifier (GUID)
Group Policy
Group Policy Object (GPO)
Group Policy Object (GPO) GUID
Group Policy Object (GPO) path
Kerberos
Lightweight Directory Access Protocol (LDAP)
Netlogon
NT LAN Manager (NTLM) Authentication Protocol
policy application
policy setting
policy target
print server
registry
scope of management (SOM)
Server Message Block (SMB)
share
site
system volume (SYSVOL)
UncPath

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

Active Directory Domain Services (AD DS)
configuration naming context (config NC)
domain controller (DC)

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

Group Policy (GP) Server

The following terms are specific to this document:

Administrative templates: A series of **Group Policy** master templates that extend the **Group Policy** management functionalities that can be applied to a **policy target** such as a **GP Client**, the settings for which are accessible from a management interface such as the **GPMC**.The **Administrative templates** provide an extensive collection of **policy settings** for applications and operating system components, which are applied through **registry** modifications on **GP Clients**. For this reason, **Administrative template policy settings** are also referred to as **registry**-based policy.

Administrative tool: A tool that allows administrators to read and write **policy settings** from and to a **GPO** and the policy files. The **Administrative tool** uses the Extension list of a **GPO** to determine which **Administrative tool extensions** are required to read settings from and write settings to the logical and physical components of the **GPO**.

Administrative tool extension: A **GP Extension** protocol that is identified by an **Administrative tool extension GUID** and invoked by a management entity such as the **GPMC**. The **Administrative tool** enables the **GP Administrator** to administer **policy settings** associated with the specific context provided by the extension.

Administrative tool extension GUID: A **GUID** that enables a specific **Administrative tool extension** to be associated with settings that are stored in a **GPO** on the **GP Server**, for that particular extension. The **GUID** enables the **Administrative tool** to identify the extension protocol for which settings are to be administered.

client-side extension (CSE): A **GP Extension** protocol that resides locally on the **GP Client** computer and is identified by a **CSE GUID**.

client-side extension GUID (CSE GUID): A **GUID** that enables a specific **CSE** on the **GP Client** to be associated with policy data that is stored in the logical and physical components of a **GPO** on the **GP Server** for that particular extension.

core Group Policy engine (core GP engine): The software entity that implements the Group Policy: Core Protocol [MS-GPOL]. The **core GP engine** issues the message sequences that result in core protocol network traffic during **policy application** on **GP Clients**. The engine handles functions on behalf of the core protocol such as the **Group Policy** refresh interval, **GPO** and policy file access, **GPO** filtering and ordering, invoking transport protocols for retrieving and storing **policy settings**, and the loopback configurations described in [MS-GPOL] section 3.2.1.3.

Group Policy Administrator (GP Administrator): A **domain** administrator who is responsible for defining **policy settings** and managing the **Group Policy** infrastructure of a **domain**.

Group Policy Client (GP Client): A client computer that receives and applies settings of a **GPO**. A **GP Client** also contains **CSEs** that extend the functionality of the **GP System**. In addition, a **GP Client** can contain **Administrative tool extensions**, sometimes called server-side extensions, if Remote Server Administration Tools [MSDN-RSATW7] are installed on it.

Group Policy data store (GP DS): A data store that consists of two types of stores. One is a physical (file system) data store on the **GP FS** that contains **policy settings** (extension and **administrative template** data), which can be locally or remotely accessed depending on location. The other is a logical data store that is part of **Active Directory** and serves as a repository for **GPOs** that are accessible via LDAP.

Group Policy Extension (GP Extension): A protocol that extends the functionality of the **GP System**. **GP Extensions** consist of **CSEs** and **Administrative tool extensions**. They provide settings and other **Group Policy** information that can be read from and written to **GP DS** components. **GP Extension** protocols depend on the Group Policy: Core Protocol [MS-GPOL], via the **core GP engine**, to identify **GPOs** containing a list of extensions that apply to a particular **GP Client**.

Group Policy file share (GP FS): A file system storage location that contains **policy settings** that include extension settings and **Group Policy** template settings for **GPOs**. The latter settings consist of security and **registry** settings, script files, and application installation information.

Group Policy Management Console (GPMC): An implementation-specific **Administrative tool** that provides an integrated interface to create, view, and manage **GPOs** and **policy settings** in multiple **forests**, **domains**, and **sites**.

Group Policy System (GP System): The collection of protocols that facilitate **Group Policy** processing and administration.

organizational unit (OU): An **Active Directory** object contained within a **domain**, into which users, groups, computers, and other organizational units can be placed.

PolicyChange: A local event that indicates that a policy has changed.

remote file access (RFA): A protocol that provides methods for accessing, reading, writing, and closing policy files on a remote file share such as the **GP FS**.

Resultant Set of Policy (RSoP): The cumulative effect of **GPO** inheritance and processing on an individual computer or a specific user. When the **policy application** process is initiated, the **core GP engine** looks at local **registry** and WMI settings, and then the **RSoP**, to determine whether a **policy target** requires a **Group Policy** update. **RSoP** data is stored, along with WMI data, in a local WMI database.

The following protocol abbreviations are used in this document:

ICMP: Internet Control Message Protocol, as defined in [RFC792].

SMB: Server Message Block Protocol, as defined in [MS-SMB].

LDAP: Lightweight Directory Access Protocol, as defined in [RFC2251].

DNS: Domain Naming System, as defined in [RFC1034] and [RFC1035].

WMI: Windows Management Instrumentation Remote Protocol, as defined in [MS-WMI].

1.3 References

We conduct frequent surveys of informative references to assure their continued availability. If you have any issue with finding an informative 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-ADOD] Microsoft Corporation, "Active Directory Protocols Overview".

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

[MS-AUTHSOD] Microsoft Corporation, "Authentication Services Protocols Overview".

[MS-CERSOD] Microsoft Corporation, "Certificate Services Protocols Overview".

[MS-FASOD] Microsoft Corporation, "File Access Services Protocols Overview".

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

[MS-GPAC] Microsoft Corporation, "Group Policy: Audit Configuration Extension".

[MS-GPCAP] Microsoft Corporation, "Group Policy: Central Access Policies Protocol Extension".

[MS-GPDPC] Microsoft Corporation, "Group Policy: Deployed Printer Connections Extension".

[MS-GPEF] Microsoft Corporation, "Group Policy: Encrypting File System Extension".

[MS-GPFAS] Microsoft Corporation, "Group Policy: Firewall and Advanced Security Data Structure".

[MS-GPFR] Microsoft Corporation, "Group Policy: Folder Redirection Protocol Extension".

[MS-GPIE] Microsoft Corporation, "Group Policy: Internet Explorer Maintenance Extension".

[MS-GPIPSEC] Microsoft Corporation, "Group Policy: IP Security (IPsec) Protocol Extension".

[MS-GPNAP] Microsoft Corporation, "Group Policy: Network Access Protection (NAP) Extension".

[MS-GPOL] Microsoft Corporation, "Group Policy: Core Protocol".

[MS-GPPREF] Microsoft Corporation, "Group Policy: Preferences Extension Data Structure".

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

[MS-GPSB] Microsoft Corporation, "Group Policy: Security Protocol Extension".

[MS-GPSCR] Microsoft Corporation, "Group Policy: Scripts Extension Encoding".

[MS-GPSI] Microsoft Corporation, "Group Policy: Software Installation Protocol Extension".

[MS-GPWL] Microsoft Corporation, "Group Policy: Wireless/Wired Protocol Extension".

[MS-KILE] Microsoft Corporation, "Kerberos Protocol Extensions".

[MS-NAPOD] Microsoft Corporation, "Network Access Protection Protocols Overview".

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol".

[MS-NRPC] Microsoft Corporation, "Netlogon Remote Protocol".

[MS-PRSOD] Microsoft Corporation, "Print Services Protocols Overview".

[MS-SMB] Microsoft Corporation, "Server Message Block (SMB) Protocol".

[MS-SPNG] Microsoft Corporation, "Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extension".

[MS-WMI] Microsoft Corporation, "Windows Management Instrumentation Remote Protocol".

[MS-WSUSOD] Microsoft Corporation, "Windows Server Update Services Protocols Overview".

[MS-WUSP] Microsoft Corporation, "Windows Update Services: Client-Server Protocol".

[MSDN-GroupPolicy] Microsoft Corporation, "Group Policy API", http://msdn.microsoft.com/en-us/library/aa374177(VS.85).aspx

If you have any trouble finding [MSDN-GroupPolicy], please check here.

[MSDN-RSATW7] Microsoft Corporation, "Remote Server Administration Tools for Windows 7", http://msdn.microsoft.com/en-us/library/ee449475(WS.10).aspx

[RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981, http://www.ietf.org/rfc/rfc792.txt

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987, http://www.ietf.org/rfc/rfc1034.txt

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, http://www.ietf.org/rfc/rfc2251.txt

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, http://www.ietf.org/rfc/rfc4120.txt

2 Functional Architecture

2.1 Overview

The GP System enables a GP Administrator to maintain standard operating environments for specific groups of users. As policies, software, and environments change over time, administrators can use Group Policy to update an already-deployed operating environment. Group Policy can also enforce rules that restrict the programs that can be run on company computers. To manage such environments, the GP System utilizes an architectural model that embraces a dual approach consisting of policy administration and policy application features.

The policy administration feature makes use of an Administrative tool, Administrative tool extensions, a GP data store (GP DS) containing GPOs and data, and a GP Server that provides directory service-based access to Group Policy metadata (sections 1.1.5 and 1.1.7.3) and file access to policy settings.

The policy application feature makes use of the GP Client, client-side extensions, and the GP DS on the GP Server, from where GPO metadata and policy settings are obtained by the GP Client for the policy application process (section 1.1.7).

The basic architecture of the GP System is shown in the following diagram. Note that the Administrative tool in this architecture is an implementation-specific interface that the GP Administrator uses to manage Group Policy.

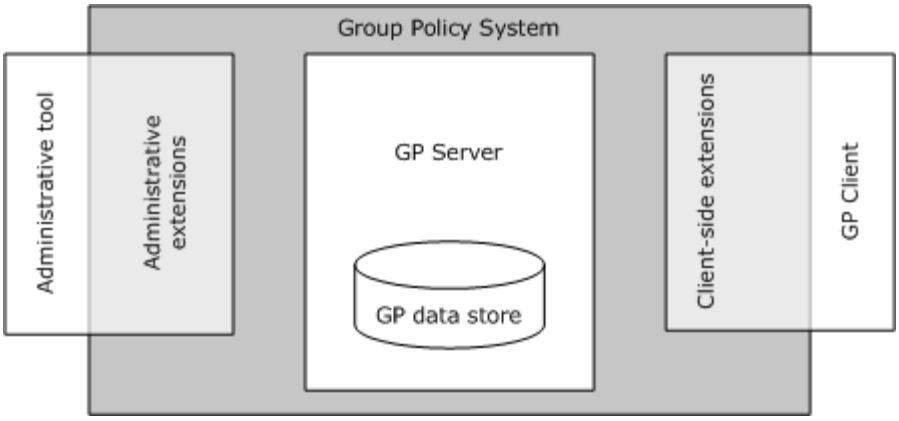

Figure 2: Group Policy System architecture

The main components of the GP System are described in section 2.1.2.

GP System components are typically installed in a distributed environment. The diagram that follows depicts a basic deployment of GP System components in a distributed environment consisting of three computers.

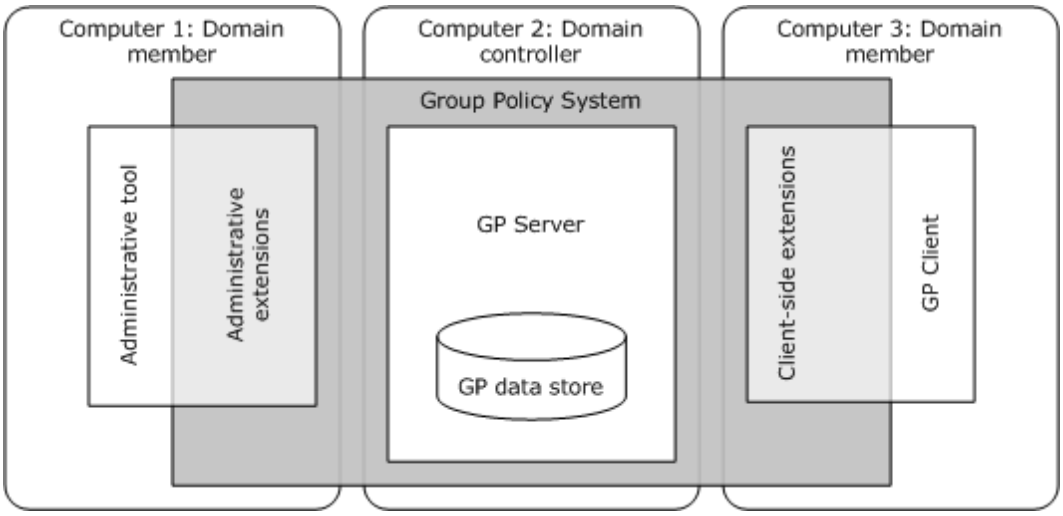

Figure 3: Group Policy System distributed environment

2.1.1 System Purpose

System administrators are required to provide consistency among groups of computers and/or users, with respect to such things as OS versions, sets of applications, and the general user experience. Group Policy enables a remote administrator to ensure that groups of computers conform to standards, and that specific users are provided with a consistent experience regardless of the computer they use.

As the enabling technology in Windows, Group Policy allows programs and administrators to use Active Directory as an infrastructure to centralize network administration, centrally define management policy, and delegate administrative authority. Users, computers, devices, and resources are represented as objects in Active Directory. With Group Policy, administrators can target policy settings on everything from users and computers to individual objects throughout the Active Directory hierarchy.

Group Policy depends on a domain-joined environment, as described in section 2.4. In this environment, the GP System enables a GP Client to retrieve GPO metadata and policy settings from a GP Server, and it enables the Administrative tool to create, retrieve, update, and delete policy settings. The protocol that provides the core functionality of the GP System is the Group Policy: Core Protocol [MS-GPOL], as described in section 1.1.1. The GP System is extensible on both the client-side (policy application) and the administrative-side (policy administration) of the functionality.

2.1.1.1 Core Protocol

The Group Policy: Core Protocol [MS-GPOL] is the main protocol in the GP System. It is a client/server protocol that allows clients to discover and retrieve policy settings created by GP Administrators. Policy settings are the directives that GP Administrators employ to control client behavior. For example, a GP Administrator might want to configure every computer in a group of computers to open a specific firewall port. The administrator could use Group Policy to implement such a directive and communicate it to clients through the Group Policy: Core Protocol. Various extensions to the core protocol are also provided to enable more granular control over different aspects of client systems.

2.1.1.2 Extensible Architecture

Group Policy has an extensible architecture consisting of the Group Policy: Core Protocol and the extension protocols that are described in section 2.2. The Group Policy: Core Protocol is fully implemented by the core GP engine. The core GP engine provides the functionality that determines which policies apply to a policy target such as a GP Client, whereas an extension, based on the determined policy applicability, is responsible for the actual policy application. The core GP engine itself does not apply actual policy settings to a GP Client; rather, it makes the LDAP or remote file access calls and extension invocations through which extension and Administrative template settings are applied.

Note that failure of a particular protocol extension sequence does not cause policy application to fail. Failure simply means that GP Clients are not able to enforce settings that are associated with a specific extension or Administrative template configuration item.

2.1.1.3 Scriptable Policy Settings

The GP System applies policy settings to GP Clients when specific events occur, such as computer startup, computer shutdown, user logon, and user logoff, as described in section 1.1.7.1. These events provide the GP Administrator with the opportunity to run scripts that apply additional policy configurations to the GP Client. These scripts can be stored on any server that contains a GP FS, which includes the GP Server. This share must be accessible by users and computers.

For more information about applying policy settings during the events mentioned in this section, refer to the documentation for the Group Policy: Scripts Extension Encoding protocol [MS-GPSCR].

2.1.2 System Components

The main components of the GP System, as shown in section 2.1, are described as follows:

- Administrative tool — an implementation-specific management entity, such as the GPMC, that enables a GP Administrator to create, modify, and delete GPOs and policy settings (Administrative templates and extension settings). The Administrative tool manages policy settings that are specific to the GP Client implementation. Policy settings and other GP System functions are managed through the following administrative tasks:
  - Authoring or editing GPOs via write access to Active Directory, to facilitate configuration of GPOs with specific policy directives or settings.
  - Updating policy files on the GP FS via remote file access write operations.
  - Configuring core aspects of the GP System, such as SOM and GPO precedence.

  The Administrative tool, along with its associated extensions, can be located and run on any computer that is a member of the domain, including the GP Server. Note that all GP Server SKUs, and GP Clients with Remote Server Administration Tools [MSDN-RSATW7] installed, will have the Administrative tool and extensions.
- GP Client — the client computer on which Group Policy settings are applied by invoking the core GP engine and the CSEs. The GP Client communicates with GP DS components, which includes the Active Directory and GP FS data stores, via the Group Policy: Core Protocol [MS-GPOL], as implemented by the core GP engine on the client computer.
- GP Extensions — consist of CSE and Administrative tool extension protocols that enhance the base functionality of the GP System. Extension data is typically read from and written to GP DS components.

- GP data store — consists of an Active Directory data store that provides storage and access to GPOs containing Group Policy metadata. It also contains a GP FS data store that serves as a file system repository for user and computer extension policy settings, GPO version information, and administrative template policy settings.

  The Group Policy administrative templates can be used to configure registry-based settings for a GPO, which can include security settings, script files for custom policy configurations, and software installation information. Administrative template settings are stored on the GP FS, however, note that administrative templates are not a requirement for a GPO.

- GP Server — a domain controller (DC) that implements the Active Directory directory service, from which a GP Client retrieves GPO metadata via LDAP and policy settings via a remote file access protocol.

Note that the terms "domain controller" and "GP Server" are used interchangeably throughout this document.

Although the GP System extends Active Directory functionality to support Group Policy operations, Active Directory<4> is not officially part of the GP System. Implementers are free to choose Active Directory or any LDAP-accessed directory service with which the GP System is compatible, to support Group Policy operations. However, for purposes of discussion herein, this document assumes that Active Directory is the LDAP-accessed directory service for the GP System. Note that the directory service chosen by the implementer MUST support **forests**.

The following sections describe the GP System components and the interrelationships among their parts, consumers, and dependencies. In particular, the following communication and process functionalities of the GP System are covered in the discussions, along with applicable standards:

- Protocol communications between system components
- Relationships between internal components
- Communication architecture and message flows
- Policy application and administration processes
- Applicability and interoperability standards

2.1.2.1 System Component Protocol Communications

The following diagram depicts the GP System along with the protocols that facilitate communication between its components.

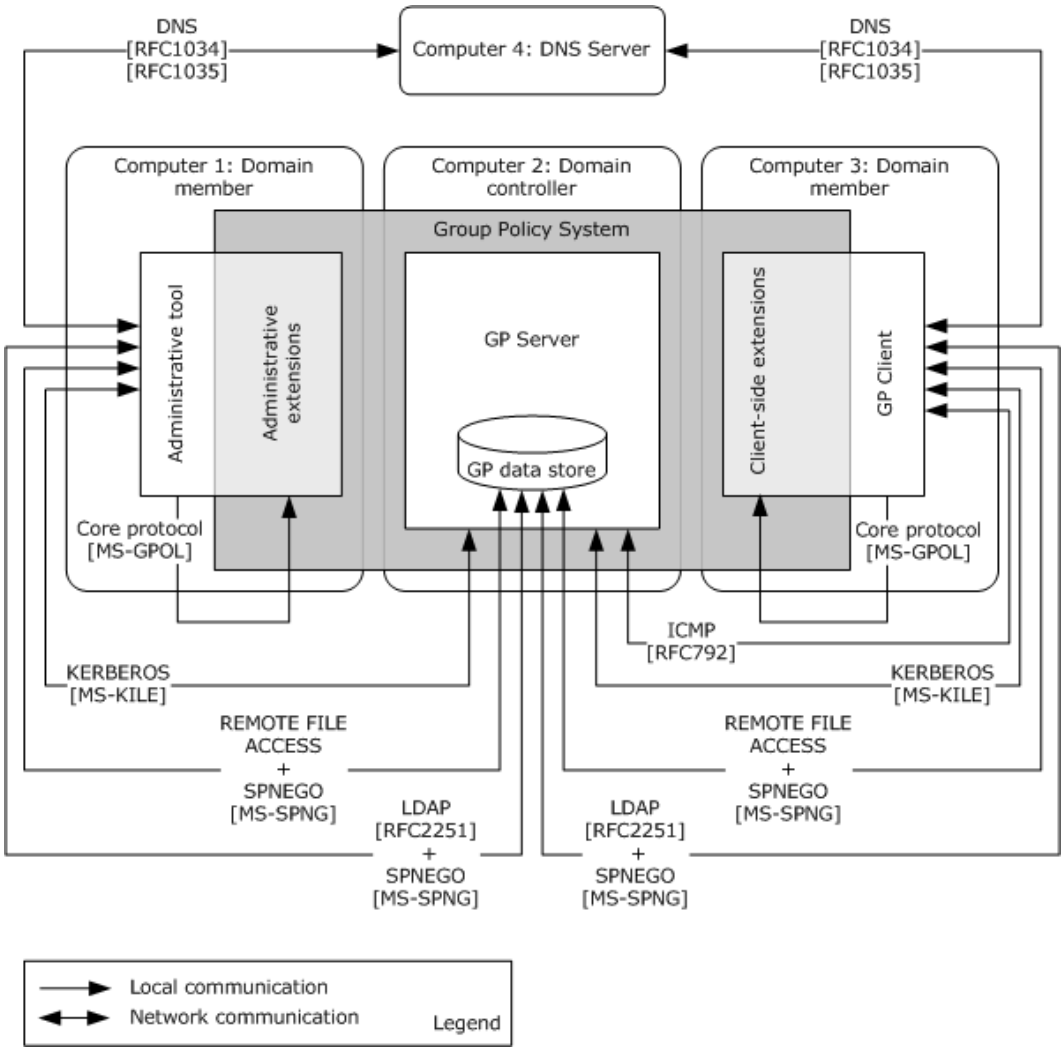

Figure 4: System component protocol communications

The GP System makes use of several protocols to facilitate communications among its components, as illustrated in the preceding diagram:

Administrative Tool Communication Protocols

The communication protocols used by the Administrative tool include:

- LDAP [RFC2251] and a remote file access protocol for accessing GP data store components, which includes the Active Directory data store on the GP Server and the GP FS data store.
- **Domain Name System (DNS)**, as described in [MS-ADOD] (section 3.1.1), for locating a domain controller.
- **Kerberos** [MS-KILE] or **NT LAN Manager (NTLM) Authentication Protocol** [MS-NLMP], as described in [MS-SPNG], for authenticating to the GP Server.

- Group Policy: Core Protocol, as described in [MS-GPOL], for invoking and processing Administrative tool extensions via the Administrative tool.

GP Client Communication Protocols

The communication protocols used by the GP Client include:

- LDAP [RFC2251] and a remote file access protocol, for accessing GP data store components, which include the Active Directory data store on the GP Server and the GP FS data store.
- DNS, as described in [MS-ADOD] (section 3.1.1), for locating a domain controller.
- Kerberos [MS-KILE] or NTLM [MS-NLMP], as described in [MS-SPNG], for authenticating to the GP Server.
- Group Policy: Core Protocol, as described in [MS-GPOL], for invoking and processing CSEs via the core GP engine.

GP Extension Communication Protocols

The communication protocols used by the GP Extensions, which include Administrative tool extensions and CSEs, are as follows:

- LDAP [RFC2251] and a remote file access protocol, for communicating with Active Directory and the GP FS.

  In policy administration mode, Administrative tool extensions make direct writes against Active Directory via LDAP and against policy files via a remote file access protocol. In policy application mode, CSEs use LDAP and a remote file access protocol to query the GP Server and the GP FS data store, respectively, for the retrieval and application of policy settings.

GP Server Communication Protocols

The communication protocols used by the GP Server include:

- LDAP [RFC2251], when accepting access to GPOs in Active Directory.
- Remote file access protocol, for accepting local access to user and computer policy files, that is, when the GP FS data store is located on the GP Server.

  Note that the core GP engine on the GP Client chooses the appropriate protocol to invoke whenever access to Active Directory or the GP FS is required by the GP Client. Likewise, the Administrative tool chooses the appropriate protocol to invoke when it needs access to Active Directory or the GP FS.

GP Data Store Communication Protocols

The communication protocols used by the GP DS include:

- LDAP [RFC2251], when access is required for the storage and retrieval of GPOs in Active Directory.
- Remote file access protocol, when access is required for updating and retrieving user and computer policy settings, and GPO version information, on the GP FS.

The protocols, systems, or services that enable these communications between GP System components are described as follows:

- **Authentication protocols** — authentication services [MS-AUTHSOD] are provided by NTLM [MS-NLMP] or Kerberos [RFC4120] [MS-KILE] to secure communications within the GP System. This system also provides authentication services that support the client-to-server communication within and outside the GP System. This includes the use of the Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Protocol Extensions, as described in [MS-SPNG], which facilitate a secure environment while negotiating which authentication protocol the GP System will use: either NTLM [MS-NLMP] or Kerberos [RFC4120], as described in [MS-SPNG], section 1.5.
- **DNS Server** — **DNS** [RFC1034] [RFC1035] is used by both the GP Client and the Administrative tool to discover the location of the GP Server.
- **Internet Control Message Protocol** — in some instances, **ICMP** [RFC792] is used by the GP Client to determine the network speed of the link to the domain controller, to ensure that bandwidth-intensive protocol extension sequences will be sufficiently supported.<5>
- **Lightweight Directory Access Protocol** — LDAP [RFC2251] is invoked by the Group Policy: Core Protocol and may be invoked by GP extensions, to read and update various policy attributes stored in GPOs within the Active Directory hierarchy on the GP Server.
- Remote file access — A file access protocol, such as **Server Message Block (SMB)** or **SMB2**, that is invoked to read and update policy files on the GP FS and to transmit policy settings and other data between the GP Server and GP Client.

2.1.2.2 System Component Functionality

The diagram that follows illustrates the internal components and protocol connections for the GP System.

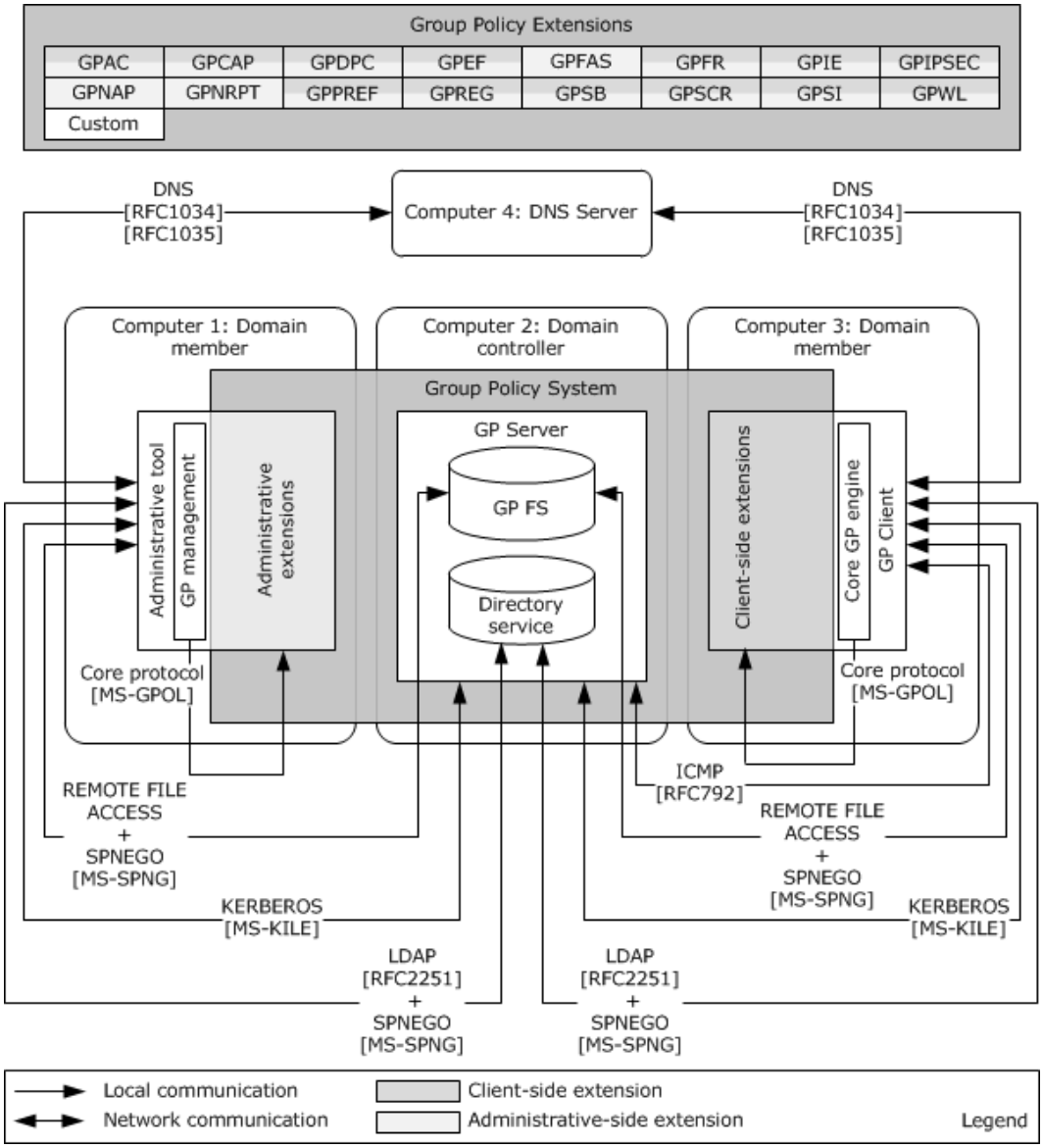

Figure 5: Internal component functions

The general functions of GP System components are described as follows:

- **Core GP engine**: coordinates the application and processing of Group Policy by handling tasks such as:
  - Applying Group Policy at regular intervals
  - Accessing GPOs and retrieving GPO extension lists from Active Directory
  - Accessing policy settings on the GP FS
  - Filtering and ordering GPOs

- Providing notification of Group Policy changes.

- **Extension protocols**: consists of CSE and Administrative tool extension protocols that extend Group Policy application functionality. Note that implementers can create their own custom extension protocols, as described in [MS-GPOL], section 1.8.

  In the preceding diagram, the color-code scheme indicates that most GP Extension protocols implement both an administrative-side and a client-side. However, the GPFAS, and GPNAP protocols implement only an administrative-side. For additional information about administrative-side and client-side extensions, see sections 1.1.4 and 2.2.

- **GP file share:** an implementation-specific version of a file share location. The GP FS location and its internal **directory** structure are shared with all GP Clients and can be replicated to other peers in a multi-master topology.

- **GP management:** the Administrative tool provides facilities for locating, retrieving, creating, modifying, and deleting group policies. These management functions can be accomplished from an interface such as the GPMC, a custom application, or a command line tool.

- **Directory service:** an implementation-specific version of an LDAP-accessible directory service, such as Active Directory, for the storage of GPOs.

2.1.2.3 System Component Tasks

The diagram that follows provides a high-level depiction of the major tasks performed by GP System components. The sections following the diagram provide details about the messaging and GP System component functions that enable these tasks to be carried out.

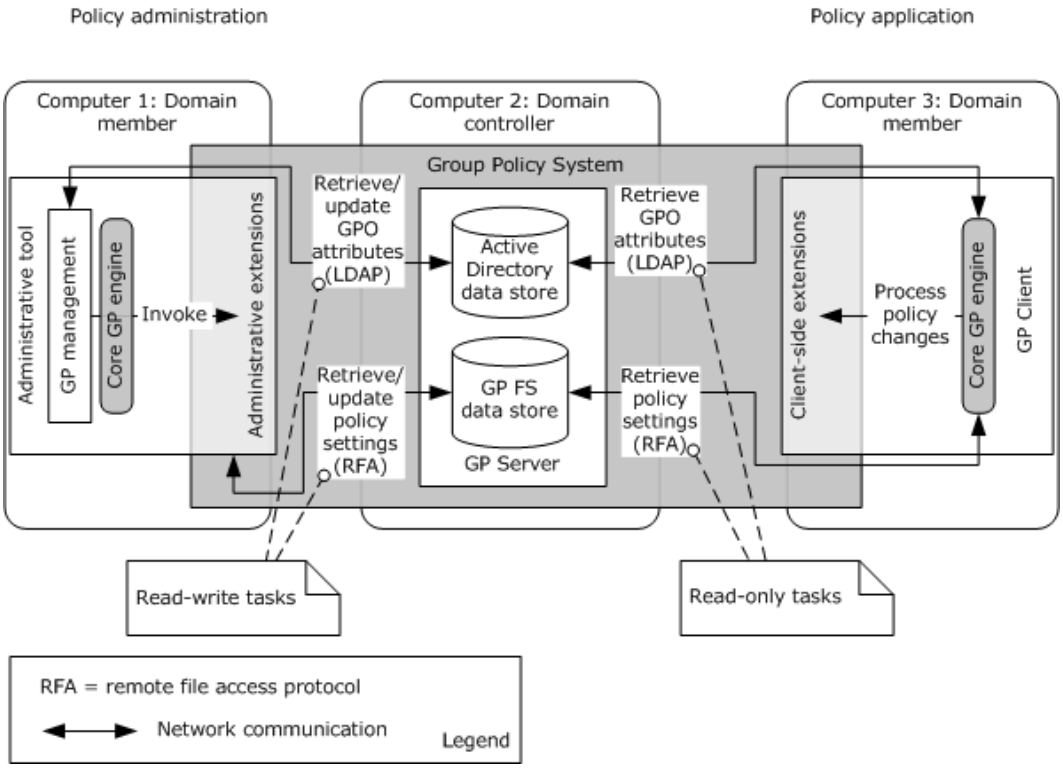

Figure 6: System communications architecture

2.1.2.3.1 Group Policy Server

The GP Server is a domain controller that implements Active Directory Domain Services (AD DS). The GP Server of itself has no knowledge of Group Policy. It is simply a server that provides storage for managed generic objects (GPOs) that are used to maintain policy information.

The GP Server maintains state via two GP DS components, which consist of the following:

- **Active Directory data store** — a hierarchical directory service that stores the logical component of GPOs that are accessible through LDAP.
- **GP FS data store** — a domain-based file share that stores GP Extension and Group Policy template settings and is accessible through a remote file access protocol. Note that the GP FS data store can be located on a remote file server or on the GP Server itself.

These data stores are modified as a result of changes made when authoring or modifying policy settings with the Administrative tool. In addition, GP Clients use these repositories as read-only stores during the policy application process.

For more information about the GP Server, including how GPOs are structured, see [MS-GPOL] section 3.1.

2.1.2.3.2 Group Policy Client

The GP Client contains the core GP engine and the CSEs that extend the GP System. The CSEs that extend the GP System are described in section 2.2.

The core GP engine has the task of managing various functionalities on GP Clients and across CSEs, which includes the following:

- Applying Group Policy at regular intervals, as described in sections 2.8.1 and 2.8.2.
- Accessing GPO attribute information from the appropriate locations in Active Directory and accessing policy settings on the GP FS.
- Handling special cases that affect all CSEs, such as loopback mode, are described in [MS-GPOL] section 3.2.1.3.
- Appropriately filtering and ordering GPOs, as described in [MS-GPOL] sections 3.2.5.1.6 and 3.2.5.1.7.
- Invoking extension protocol sequences, as described in [MS-GPOL] section 3.2.5.1.10.
- Maintaining version numbers and histories for all CSEs.
- Invoking CSEs for the policy application process.
- Notifying various components of changes made by Group Policy. The core GP engine is responsible for this activity after the completion of policy processing.

The basic communication flow associated with the GP Client consists of the following:

1. The GP Client locates a domain controller (GP Server), as described in [MS-ADOD] (section 3.1.1).
2. The GP Client uses LDAP to query the GP Server for a list of GPOs, as described in [MS-GPOL] section 3.2.5.1.5.

3. For each object in the GPO list, the GP Client queries the GP Server for the GPO's attributes, using LDAP and a remote file access protocol, as described in [MS-GPOL] sections 3.2.5.1.5, 3.2.5.1.6, and 3.2.5.1.7.

4. Based on the GUIDs in the Extension list of GPOs, the core GP engine on the GP Client invokes the appropriate CSEs ([MS-GPOL] section 3.2.5.1.10).

5. In turn, each CSE uses LDAP and a remote file access protocol to query the GP Server and GP FS, respectively, for the retrieval of GPO attributes and policy settings, as described in [MS-GPOL] section 1.3.3.3.

2.1.2.3.3 Group Policy Administrative Tool

The Administrative tool facilitates the creation, deletion, and modification of Group Policy settings. It also enables the GP Administrator to define the manner in which policy settings are to be applied, by creating the SOM configuration and GPO precedence order.

The Administrative tool uses the same set of protocols to discover the GP Server and the same extensions when authoring policy as the GP Client uses to discover the GP Server and apply policy settings. An overview of communication and authoring processes is provided in section 2.1.3.2.1.

The basic communication flow associated with the Administrative tool consists of the following:

1. The Administrative tool locates the domain controller (GP Server) as specified in [MS-ADOD] (section 3.1.1).

2. The Administrative tool uses LDAP to query Active Directory on the GP Server for the retrieval of GPO attributes.

3. The core GP engine on the computer hosting the Administrative tool invokes an Administrative tool extension, via a GUID specified in the GPO Extension list.

4. The Administrative tool extension retrieves Group Policy attributes from the logical component of a GPO by using LDAP to query Active Directory on the GP Server, as described in section 1.1.6.

5. The Administrative tool extension retrieves policy settings from the file system component of the GPO by using a remote file access protocol to query the appropriate GP FS directory locations.

6. The extension uses LDAP or a remote file access protocol to update Group Policy attributes in Active Directory on the GP Server and extension and template setting changes on the GP FS, respectively.

7. The Administrative tool uses LDAP to update version information for the GPO in Active Directory and uses a remote file access protocol to update version information in the gpt.ini file on the GP FS. This is described in detail in [MS-GPOL] section 3.3.4.1.

2.1.3 System Communication Process Details

This section describes the protocol communications, interactions, and transports upon which the GP System relies. Although the related protocols have been noted earlier, the details of the actual communication process have not. The two communication processes of interest involve interactions between the following:

- GP Client and the GP Server
- Administrative tool and the GP Server

The communication discussions that follow assume that AD DS is implemented on the GP Server.

2.1.3.1 Protocol Communication Between a Group Policy Client and Group Policy Server

The GP System uses LDAP and a remote file access protocol to transport Group Policy-specific information sent between the GP Client and the GP Server. The sections that follow describe the communication that occurs between a GP Client and a GP Server via policy application messages, to enable the client to read and apply Group Policy.

The following diagram summarizes the communication between various GP System components during policy application by the GP Client. The communications illustrated in the diagram map to the discussions in the sections that follow.

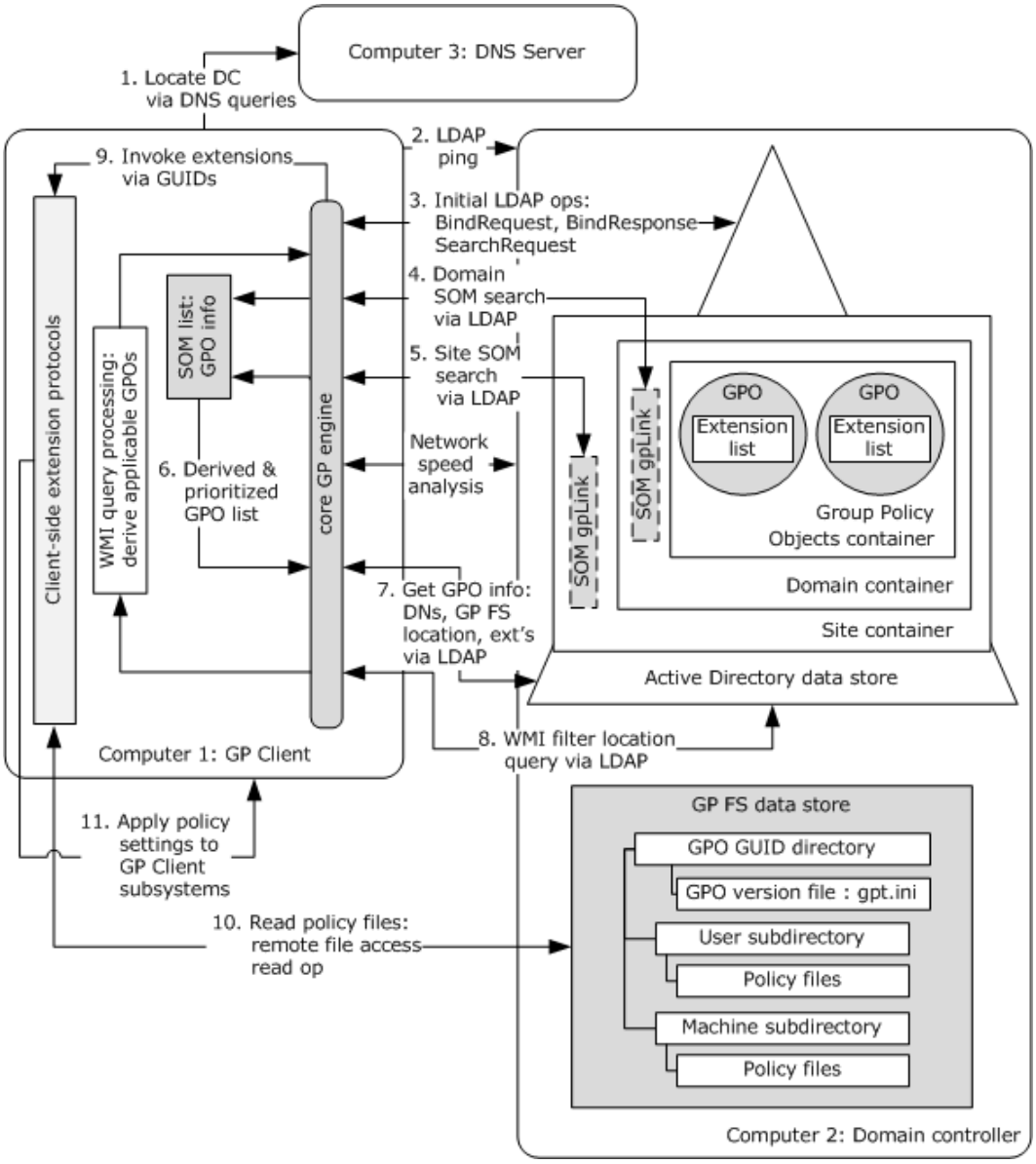

Figure 7: Policy application process

2.1.3.1.1 Locating a GP Server

The GP Client locates the GP Server by discovering the location where the Active Directory data store resides, and through an associated LDAP lookup, locates the file system share where the extension policy files reside. In the Microsoft implementation, both the Active Directory data store and file system share (**SYSVOL**) are located on the GP Server, which is the domain controller.

The process of locating a domain controller (GP Server) is specified in [MS-ADOD] (section 2.5) and [MS-ADOD] (section 3.1.1).

2.1.3.1.2 Domain SOM Search and Response

SOM is associated with an Active Directory container, such as a domain, site, or OU, that holds user and computer accounts that are managed through Group Policy. The GP Client must access the SOM container to obtain attribute information. To initiate this process, the GP Client sends an LDAP **BindRequest**, and the GP Server sends an LDAP **BindResponse** in reply. After the GP Client has successfully received a **BindResponse** from the GP Server, it sends an LDAP **SearchRequest** to the GP Server, with the LDAP information about its directory location. The GP Client then queries for the **gpLink** and **gpOptions** attributes that hold information about the GPOs in the SOM container for the **configuration naming context (config NC)**, which stores configuration information in Active Directory, as described in [MS-ADTS] sections 3.1.1.1.5 and 6.1.1.1.2.

The GP Server processes the information provided as part of the request for the domain SOM and returns an object with **gpLink** and **gpOptions** attribute information to the GP Client along with the DN to which it applies.

The **gpLink** attribute retrieved from the domain container in Active Directory holds LDAP DNs for GPOs that are associated with domain-level SOM. This information enables the policy application process to determine GPO names, the policy file location on the GP FS, and any extensions specified in the GPO extension lists, all of which apply to domain-level SOM. For information about the corresponding **gpLink** and **gpOptions** ADM elements, see [MS-GPOL] section 3.2.1.6.

The domain SOM data is added to an **SOM list** maintained by the GP Client. For information about the **SOM list** ADM element, see [MS-GPOL] section 3.2.1.6.

2.1.3.1.3 Site SOM Search and Response

After the GP Client has determined its domain SOM, it then uses a site search message, as described in [MS-GPOL] sections 2.2.3 and 3.2.5.1.4, to determine the site to which the computer belongs. The name of the site to which the GP Client computer belongs is maintained by the client **site name** ADM element, as described in [MS-ADOD] (section 3.1.1). Because the site can change based on the GP Client's location, the **site name** ADM element must be maintained as part of policy processing.

After the GP Client has the site to which it belongs, it makes an LDAP query for the same attributes that a domain SOM search does; these are the **gpLink** and **gpOptions** attributes, although the GP Client also passes the site name that it has discovered in this LDAP query. The GP Server returns the **gpLink** and **gpOptions** attribute values that apply to the GP Client for processing.

The **gpLink** attribute retrieved from the site container in Active Directory holds LDAP DNs for GPOs that are associated with site-level SOM. Similar to the domain-level SOM, this information enables the policy application process to determine GPO names, the policy file location on the GP FS, and any extensions specified in the GPO extension lists, all of which apply to site-level SOM. The site DN and the **gpLink** and **gpOptions** ADM element values must be appended to the end of the **SOM list**. For more information about the **SOM list** ADM element, see [MS-GPOL] section 3.2.1.6.

If the site search message specified in [MS-GPOL] section 2.2.3 is invalid in any way, the entire Group Policy: Core Protocol policy application sequence must be terminated.

2.1.3.1.4 GPO Search and Reply

After the GP Client has computed the domain SOM and configured the **SOM list**, the GP Client searches for the GPOs that apply to it.

The search for GPOs involves the GP Client creating a prioritized list of GPOs, as described in [MS-GPOL] sections 3.2.5.1.5, 3.2.5.1.6, and 3.2.5.1.7, and sending an LDAP query containing this list to the GP Server. The GP Server returns an LDAP reply with further attribute information about each queried GPO, as described in [MS-GPOL] section 2.2.4. These attributes describe the GPO display name, the location of the policy file on the GP FS, extensions used in that policy file, a security descriptor, an enabled flag, denial status, and any WMI filters that may apply to the GPO.

This LDAP query message requires the success of all previous messages that have retrieved SOM data and a **gpLink** attribute associated with each SOM, and this information must be stored in the **SOM list**. If this message is invalid, the entire policy application sequence must be terminated, and the GP Client must not generate further policy application messages for this GPO processing sequence.

For each GPO successfully retrieved in each search, the following remote file access protocol sequences must be generated by the GP Client:

File open — The version file gpt.ini typically exists in the <gpo path> directory on a remote GP FS or a local SYSVOL share. The policy files typically exist in subdirectories of the <gpo path> directory. As part of file open operations, authentication must occur in accordance with SPNEGO [MS-SPNG] for user policy mode, and in accordance with **Kerberos** [RFC4120] for computer policy mode. The directory <gpo path> corresponds to the file system path retrieved from the GPO in the **gPCFileSysPath** attribute of the search.

File read — A series of file reads must occur until either the entire contents of the opened file are read or an error in reading occurs.

File close — A file close operation must then be issued.

2.1.3.1.5 WMI Filter Processing

When the GP Client has processed the GPO attributes returned by the GP Server and has determined that a policy object has a WMI query that applies to a GPO, the GP Client will also have the location of that WMI filter in Active Directory. The GP Client then uses LDAP to query the GP Server for the WMI query, passing into the query the required location and attributes, as described in [MS-GPOL] section 2.2.5.

The GP Server replies with an LDAP response that returns the necessary attribute information, as described in [MS-GPOL] section 2.2.5. The GP Client processes the WMI query to determine which GPOs apply to it, as indicated by the WMI query.

If the WMI query cannot be evaluated due to a local GP Client error, the entire policy application mode sequence must be terminated. If the WMI query returns no results, the GPO must be denied; otherwise, the GPO must be allowed, as described in [MS-GPOL] section 3.2.5.1.7.

2.1.3.1.6 Link Speed Determination

The GP Client should estimate the link speed of the network between the GP Client and the GP System by implementation-specific means.<6> The implementation can send a message to

determine link speed by using ICMP as a transport, but it must support at least 500-byte packets, as described in [RFC792]. If the determined link speed ([MS-GPOL] section 3.2.5.1.9) is below an implementation-defined threshold, the implementation should not invoke any bandwidth-intensive protocol extension sequence.<7>

2.1.3.1.7 Policy File Read Operation

When the GP Client has all the GPO attribute information that applies to the GP Client, has evaluated WMI filters, and has determined the link state, it is ready to read the extension information from the policy files.

Using the specific extensions relevant to the GPO, the GP Client makes a remote file access protocol request to the file system location indicated by the attributes returned in the LDAP queries specified in section 2.1.3.1.4. It then reads the specific extension settings from the policy files.

2.1.3.2 Protocol Communication Between the Administrative Tool and Group Policy Server

Group Policy is managed with an Administrative tool that uses the same protocols (LDAP and a remote file access protocol), and in several instances, the same protocol sequence methods used by the GP Client. The protocol steps differ for the following Group Policy management operations:

- Creating new policies
- Editing existing policies
- Deleting policies

2.1.3.2.1 Creating Group Policy Objects

When authoring new GPOs with the Administrative tool, the GP Administrator will follow the same initial steps of the protocol sequence that occurs during GP Client operations, as follows:

1. Locate a GP Server, as specified in section 2.1.3.1.1 and [MS-ADOD] (section 3.1.1).
2. Initiate an LDAP **BindRequest** and **BindResponse**, as specified in section 2.1.3.1.2.

Thereafter, to complete the GPO configuration, the Active Directory containers and file system components of the GPO will need to be created, and various GPO attributes will need to be set.

2.1.3.2.1.1 Creating the Active Directory Containers

To construct a GPO after the preceding initial protocol sequence, it is necessary to create a Group Policy container (GPC) object for the GPO in Active Directory on the GP Server. The GPC for a GPO is an object of the *groupPolicyContainer* class. The GPC is typically created in the *Group Policy Objects* container within the domain; it is then linked to the domain container. Following creation of the GPC object, GPO *User* and *Machine* subcontainers will need to be created to complete the Active Directory components of the GPO.

Creating the GPC for a GPO is accomplished by sending LDAP messages from the Administrative tool to the GP Server. The first message is an LDAP **addRequest** that follows the format specified in [MS-GPOL] section 2.2.8.1.4, to create a Policies container. Additional LDAP messages, as specified in [MS-GPOL] sections 2.2.8.1.5, 2.2.8.1.6, and 2.2.8.1.7, are then required for each of the following:

- GPO **addRequest**

- GPO *User* subcontainer **addRequest**
- GPO *Machine* subcontainer **addRequest**

When creating the new GPO, the Administrative tool must also send an LDAP **SearchRequest** to return the security descriptor for the new GPO. The Administrative tool must also create a unique GUID for the GPO DN. Further details on the process of creating a GPO and the associated hierarchical containers are specified in [MS-GPOL] section 3.3.5.1.

For each of the LDAP **addRequest** messages, the GP Server will reply to the Administrative tool with **addResponse** messages, as defined in [RFC2251] section 4.7. The value of the **resultCode** field of the **addResponse** messages determines message success or failure; the value zero indicates success, while any other value indicates failure.

2.1.3.2.1.2 Creating the GPO File System Components

To create the file system components of the GPO, it is necessary to create an associated set of directories on the GP FS, to which the GPO will point, for storing and locating user and computer policy files, in addition to GPO version and GPT information.

After the preceding LDAP messages are successfully processed, the required set of directories on the GP FS are created with the operations that follow. These processes will utilize the Group Policy Object (GPO) path to create a *User* subdirectory and a *Machine* subdirectory. The GPO path is a **UncPath** of the form: "\\<dns domain name>\<GP FS-name>\<dns domain name>\policies\<gpo guid>", where <dns domain name> is the DNS domain name, and <gpo guid> is a **Group Policy Object (GPO) GUID**.

The steps that follow create the GPO path directory and gpt.ini file on the GP FS via the file and directory operations of a remote file access protocol:

1. Send a **File Status** request for the GPO path, using SPNEGO (as described in [MS-SPNG]) for authentication.
2. Send a **Create Directory** request to create a new directory named by the GPO GUID of the GPO DN, using SPNEGO for authentication, as described in [MS-SPNG].
3. Send a **Close** request, using SPNEGO for authentication, as described in [MS-SPNG].
4. Send an **Open** request for the GPO path, using SPNEGO for authentication, as described in [MS-SPNG].
5. Send a **Create File** request to create a file named gpt.ini, using SPNEGO for authentication, as described in [MS-SPNG].
6. Send a **Write File** request to write contents to the gpt.ini file (as described in [MS-GPOL] section 2.2.4), which should contain the required section named "General"; the key "Version" under the General section; and the value of the key "Version" set to "65537" for the first version (the decimal equivalent of hexadecimal value 0x00010001). The Write File request should use SPNEGO for authentication, as described in [MS-SPNG].

   Sample content for a gpt.ini file is described in [MS-GPOL] section 4.10.
7. Send a **Close** request, using SPNEGO for authentication, as described in [MS-SPNG].

The steps that follow create directories named with the user-scoped GPO path and the computer-scoped GPO path via the directory operations of a remote file access protocol:

1. Send an **Open** request for the GPO path, using SPNEGO for authentication, as described in [MS-SPNG].
2. Send a **Create Directory** request for the directory that is named with the user-scoped GPO path, using SPNEGO for authentication, as described in [MS-SPNG].
3. Send a **Close** request, using SPNEGO for authentication, as described in [MS-SPNG].
4. Send an **Open** request for the GPO path, using SPNEGO for authentication, as described in [MS-SPNG].
5. Send a **Create Directory** request for the directory that is named with the computer-scoped GPO path, using SPNEGO for authentication, as described in [MS-SPNG].
6. Send a **Close** request, using SPNEGO for authentication, as described in [MS-SPNG].

Any failures from these remote file access protocol operations means that the overall message that creates the GPO is invalid, and as a result, the protocol sequence must be terminated.

2.1.3.2.1.3 Completing the GPO Configuration

GPOs store various information sets in the form of attributes, which support Group Policy processes. Some of these attributes are automatically generated when the GPO is created and some are configured by the GP Administrator, such as the Extension lists. When the GP Administrator is finished creating and configuring the GPO, it will contain the following key attributes:

- **createTimeStamp** — stores the date and time that the *groupPolicyContainer* object was created.
- **displayName** — stores the friendly name of the GPO specified by the GP Administrator.
- **DistinguishedName** — stores the full DN of the *groupPolicyContainer* object.
- **Flags** — stores the state of the GPO:
  - Flags=0; the GPO is enabled
  - Flags=1; the user configuration portion of the GPO is disabled
  - Flags=2; the computer configuration portion of GPO is disabled
  - Flags=3; the GPO is disabled
- **gPCFileSysPath** — stores the GP FS path to the GPO's gpt.ini file.
- **gPCMachineExtensionNames** — stores a list of GUIDs that correspond to computer-specific CSEs that are implemented in this GPO.
- **gPCUserExtensionNames** — stores a list of GUIDs that correspond to user-specific CSEs that are implemented in this GPO.
- **versionNumber** — stores the current version number for the *groupPolicyContainer* of the GPO. Versioning is used to determine how many changes have been made to the GPO and whether the changes synchronize with the version specified by the gpt.ini file in the GPO path.

After a GPO is successfully created, it can be edited in the same manner as an existing policy is edited, as described in section 2.1.3.2.2.

2.1.3.2.2 Editing Existing Policies

Before the administrator can use the Administrative tool to edit policy objects, a connection to Active Directory is required to lookup LDAP objects. This involves the same two steps used in policy application:

- Locate a GP Server, as described in section 2.1.3.1.1 and [MS-ADOD] (section 3.1.1).
- Initiate an LDAP **BindRequest** and **BindResponse**, as described in section 2.1.3.1.2.

After the Administrative tool discovers a writable GP Server and makes a successful connection to Active Directory, the administrator can select a policy to be edited.

The diagram that follows summarizes the communication between various GP System components during the policy administration editing process, as facilitated by the Administrative tool.

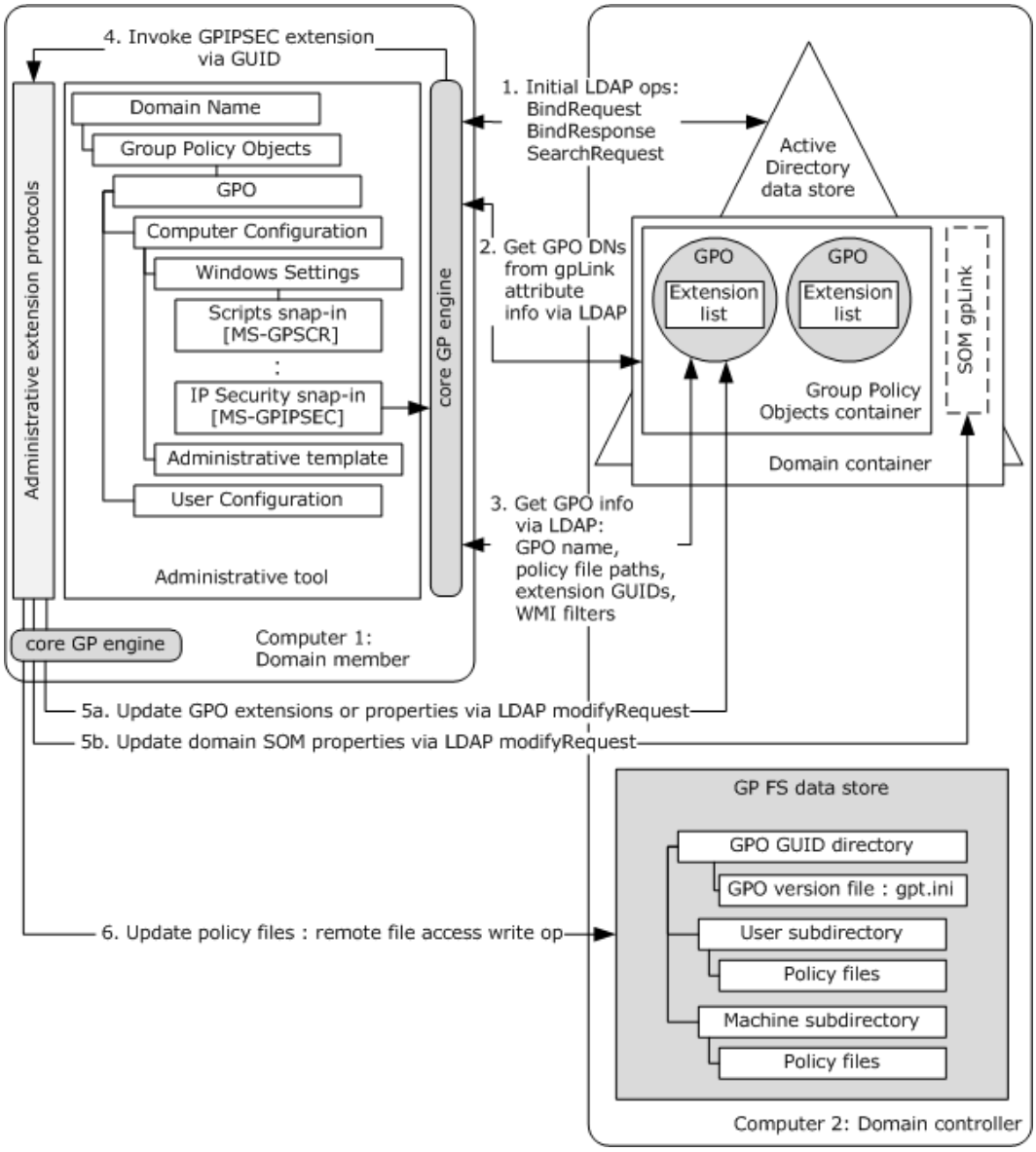

Figure 8: Policy administration editing process

The sections that follow describe the processes that occur when editing GPOs and policy files.

2.1.3.2.2.1 Modifying Extension Settings

When the administrator uses the Administrative tool to update the configuration of an administrative-side extension, the tool invokes the administrative extension via a GUID that is referenced in the GPO Extension list. To apply updates, the extensions make direct writes against Active Directory by using LDAP, and against the policy settings files via a remote file access protocol.

Whenever the Administrative tool invokes an extension protocol specified by a GPO and that extension modifies the GPO, the extension invokes a GPO extension update sequence, which in turn

generates a GPO extension update message. This message must be an LDAP **modifyRequest** message with specific parameters passed, as described in [MS-GPOL] section 2.2.8.2.

The extension receives a **modifyResponse** message in reply. This message provides a return value that indicates success or failure of the **modifyRequest** message. A value equal to the integer zero indicates success, whereas any other value indicates failure.

The Administrative tool then uses a remote file access protocol to update the gpt.ini file and any applicable policy settings in the GPO path and receives responses that confirm success or failure. See [MS-GPOL] section 3.3.5.2 for additional details about updating the gpt.ini file.

2.1.3.2.2.2 Updating GPO Properties

Whenever the administrator uses the Administrative tool to modify GPO properties, the tool generates a GPO property update message. This message must be an LDAP **modifyRequest** message with specific passed parameters, as described in [MS-GPOL] section 2.2.8.3. The Administrative tool receives a **modifyResponse** message in reply. This message provides a return value that indicates success or failure of the modify request. A value equal to the integer zero indicates success, whereas any other value indicates failure.

The following tasks are also required after updating GPO properties:

1. Opening the policy file on the GP FS, using SPNEGO for authentication, as described in [MS-SPNG].
2. Modifying the directory security descriptor.
3. Closing the policy file.

2.1.3.2.2.3 Updating SOM

Whenever the administrator uses the Administrative tool to modify SOM properties, the tool generates a SOM property update message. This message must be an LDAP **modifyRequest** message with specific passed parameters, as described in [MS-GPOL] section 2.2.8.4. The Administrative tool receives a **modifyResponse** message in reply. This message provides a return value that indicates success or failure of the modify request. A value equal to the integer zero indicates success, whereas any other value indicates failure.

2.1.3.2.3 Deleting Group Policy Objects

To delete a GPO, it is necessary to delete all Active Directory objects associated with the GPO on the GP Server and to delete corresponding directories on the GP FS that contain user and computer settings, to which the GPO links. To delete the Active Directory objects for a GPO, it is necessary to send an LDAP **delRequest** message, as described [MS-GPOL] section 2.2.8.5 and [RFC2251] section 4.8, from the Administrative tool to the GP Server.

The GP Server replies to the **delRequest** message with a **delResponse** message, as defined in [RFC2251] section 4.8. The value of the **resultCode** field in the **delResponse** message determines whether the delete operation succeeded or failed; success is indicated by a **resultCode** field value of zero, while all other values indicate failure.

A GPO is an Active Directory container; therefore, an LDAP **delRequest** message is first sent for all Active Directory objects contained in the GPO, and then an LDAP **delRequest** is sent recursively for each subcontainer and all Active Directory objects contained in the subcontainer. To begin the sequence, an LDAP **SearchRequest** ([RFC2251] section 4.5.1) containing the parameters specified in [MS-GPOL] section 3.3.5.6 is sent to the GP Server to retrieve the GPOs.

To delete GP FS files and directories, it is necessary to recursively delete the files and directories in the <gpo path> via a remote file access protocol. All I/O operations that fail should be logged.

For further details about deleting GPOs, see [MS-GPOL] section 3.3.5.6.

2.1.3.3 Transport Requirements

The GP Client and the Administrative tool use the following protocols for data transport:

- LDAP and a remote file access protocol to transmit policy settings and to transmit instructions between the GP Client and the GP Server.
- Kerberos [RFC4120] and SPNEGO [MS-SPNG] for authentication in computer policy application mode.
- SPNEGO [MS-SPNG] for authentication in user policy application mode.

See section 2.3.2 for other protocols upon which the GP Server relies.

2.1.4 Applicability

The GP System is primarily applicable in scenarios where centralized administration of users and computers is desired.

2.1.5 Relevant Standards

The GP System uses the following communication standards to allow interoperability with other external systems:

DNS — as described in [RFC1034] [RFC1035]. Used for locating the GP Server and determining site membership.

Lightweight Directory Access Protocol (LDAP)— as described in [RFC2251]. Used for communication with the GP Server to obtain GPO attribute data.

File access protocol — as described in [MS-FASOD]. The Windows platform chooses an SMB file access protocol to remotely access the GP FS and obtain user policy information, computer policy information, and GPO version data.

SPNEGO — as described in [MS-SPNG]. Used for authentication and authorization. See [MS-GPOL] section 1.4 for the authentication protocols supported by the GP System.

2.2 Protocol Summary

This section describes the member protocols that accomplish the goals of the GP System. The GP System is organized into the following main protocol groups:

- Group Policy core — consists of the Group Policy: Core Protocol [MS-GPOL]. The core protocol is implemented fully by the core GP engine, which enables the processing and application of Group Policy.
- Group Policy extensions — consists of the extension protocols specified in [MS-GPAC], [MS-GPCAP], [MS-GPDPC], [MS-GPEF], [MS-GPFAS], [MS-GPFR], [MS-GPIE], [MS-GPIPSEC], [MS-GPNAP], [MS-GPPREF], [MS-GPREG], [MS-GPSB], [MS-GPSCR], [MS-GPSI], and [MS-GPWL].

The following table provides a comprehensive list and functional description of the member protocols specified in the preceding groups, and which constitute the framework of the Group Policy System.

Group Policy member protocols

Protocol Name	Functional Description	Short Name
Group Policy: Core Protocol	Enables discovery and connection to a domain controller, discovery and retrieval of GPOs, support for the authoring of policies and extension settings, and communication of administrator-defined policies from the GP Server to the GP Client. The Group Policy: Core Protocol is fully implemented by the core GP engine.	[MS-GPOL]
Group Policy: Audit Configuration Extension	Enables advanced audit policies to be distributed to multiple client systems where they are enforced in accordance with administrative intent. The policy settings for this extension enable the underlying audit subsystem to determine the activities to be monitored and logged in the security event log. The GPAC extension has both client-side and administrative-side implementations. The administrative-side extension enables the GP Administrator to author audit policies, store them on the GP FS, and update a GPO with the path to the policy files on the GP FS. The client-side extension is invoked by the core GP engine on the GP Client to locate GPO(s) containing audit configuration settings (as indicated by the GPAC GUID appearing in the GPO Extension list), transfer the policy files to the GP Client computer via a remote file access protocol, and then configure the advanced audit policy, audit options, and global object access auditing settings on the GP Client computer.	[MS-GPAC]
Group Policy: Central Access Policies Extension	Provides the means of configuring central access policies on GP Client computers for centralized control of user access to resources. This protocol extension also contains the mechanisms that enable GP Administrators to retrieve policy files and configure central access policy information that is stored in the GP DS. The administrative-side extension participates in authoring settings for central access policies via GPO configuration. The administrative-side extension of this protocol invokes LDAP to write or retrieve GPO information and invokes an RFA protocol to write or read extension-specific data in central access policy files that are stored on the GP FS. Central access policy settings are created or modified by the Administrative tool. The client-side extension retrieves policy settings from the file system component of one or more GPOs. These settings consist of one or more DNs of central access policy objects that reside in Active Directory. The CSE binds to these objects and pulls down central access policy configuration data from the object attributes. The CSE uses this data to populate local data elements on the GP Client, typically a file server, to maintain state that is later applied by an administrator to enforce the central access policies that authorize user access to resources on the file server.<8>	[MS-GPCAP]
Group Policy: Deployed Printer Connections Extension	Supports the management of printer connections that are hosted by print servers and shared by multiple users. The GPDPC extension has both client-side and administrative-side implementations.	[MS-GPDPC]

Protocol Name	Functional Description	Short Name
	The administrative-side extension enables the GP Administrator to configure printer connections by updating settings in a GPO that applies to GP Clients. The client-side extension is invoked by the core GP engine on the GP Client to enable users to discover the printer connections that were configured by the GP Administrator and to apply them to the GP Client computer.	
Group Policy: Encrypting File System Extension	Enables remote administrative configuration of the **Encrypting File System (EFS)**. The GPEF extension has both client-side and administrative-side implementations. The administrative-side extension enables the GP Administrator to retrieve and edit EFS configuration settings that are stored in a registry-based policy file on the GP FS, for later application to the registry of GP Clients that are affected by GPO(s) that specify those settings. The client-side extension is invoked by the core GP engine on the GP Client to parse the registry policy file settings and copy them to the GP Clientregistry. The EFS extension then reads those registry settings and applies them to the EFS subsystem on the GP Client computer.	[MS-GPEF]
Group Policy: Firewall and Advanced Security Data Structure Extension	Enables administrators to use Group Policy to control firewall and advanced security behavior on a GP Client with the use of the GPREG protocol. The GPFAS extension is invoked by the Administrative tool and is responsible for loading and updating the firewall and advanced security settings specified by a GPO. GPFAS reads registry values that are copied to the GP Clientregistry by the Group Policy: Registry Extension Encoding protocol [MS-GPREG] and applies them to the local Firewall and Advanced Security Protocol server. Because this extension relies on the CSE implementation of GPREG, GPFAS is implemented as an administrative-side extension only.	[MS-GPFAS]
Group Policy: Folder Redirection Protocol Extension	Enables the GP Administrator to redirect the path of certain file system folders to a new location. The new location can be a folder on the local computer or a shared directory on a network. This enables users to work with documents on a remote server share, as if the documents were located on the hard disk of their local computer. This extension has both client-side and administrative-side implementations. The administrative-side extension enables the GP Administrator to establish and configure folder locations for user folders and to store them on the GP FS. The client-side extension is invoked by the core GP engine on the GP Client to retrieve GPFR configuration data from the GP FS and to apply it to the GP Client computer.	[MS-GPFR]
Group Policy: Internet Explorer Maintenance Extension	Enables a GP Administrator to manage policy settings to configure Internet Explorer in the following ways: ▪ Customize the browser appearance.	[MS-GPIE]

Protocol Name	Functional Description	Short Name
	▪ Preset and manage browser connection settings. ▪ Set the default URLs displayed by the browser. ▪ Set the default programs used for each Internet service. ▪ Preset the security zone content rating, certification authority, and authenticode settings. This extension has both client-side and administrative-side implementations. The administrative-side extension enables the GP Administrator to specify Internet Explorer configuration information in specific files which are then stored on the GP FS. The client-side extension is invoked by the core GP engine on the GP Client to retrieve these files and copy them to the GP Client, where they are later processed by Internet Explorer components.	
Group Policy: IPsec Protocol Extension	Enables centralized configuration of the IPsec component on multiple client systems to provide basic traffic filtering, data integrity, and optional data encryption, for IP traffic. The GP Administrator assigns an IPsec policy to a group of managed client computers using a GPO. This extension has both client-side and administrative-side implementations. The administrative-side extension enables the GP Administrator to create one or more IPsec policies and store them in policy files on the GP FS. The client-side extension is invoked by the core GP engine on the GP Client to retrieve the associated policy settings stored in the policy files and to apply them to the GP Client computer.	[MS-GPIPSEC]
Group Policy: Network Access Protection (NAP) Extension	Enables control of client computer access to network resources. Access can be granted or restricted on a per-client computer basis, according to a client computer's identity and its degree of compliance with corporate governance policy. For non-compliant client computers, the NAP extension specifies methods to automatically reinstate compliance and dynamically upgrade access to network resources. The GP Administrator uses the Administrative tool to manage the NAP client configuration settings through the NAP extension protocol. This extension reads and updates the registry using the GPREG extension protocol. Because it relies on the CSE implementation of GPREG, GPNAP is implemented as an administrative-side extension only.	[MS-GPNAP]
Group Policy: Preferences Extension Data Structure	Enables the GP Administrator to manage and deploy preferences. GPPREF extension settings are specified by using an XML file. This extension has both administrative-side and client-side implementations. The administrative-side extension enables the GP Administrator to invoke the GPPREF extension on his or her computer to define, maintain, and associate extension-specific settings with a GPO. The client-side extension is invoked by the core GP engine on	[MS-GPPREF]

Protocol Name	Functional Description	Short Name
	the GP Client to read the XML preferences file specified by the GPO and apply its preferences configuration to the GP Client computer. GPPREF supports both computer and use policy modes. Policy application in computer policy mode applies to the GP Client computer and all users who log on to it, whereas user policy mode applies to specific users who log on to the GP Client computer.	
Group Policy: Registry Extension Encoding	Provides the mechanism for an GP Administrator to control any behavior on a GP Client that depends on registry-based settings. This extension has both administrative-side and client-side implementations. The administrative-side extension enables the GP Administrator to use Administrative template settings to write out a registry policy file and associate it with a GPO. The client-side is extension invoked by the core GP engine on the GP Client to read the registry policy file specified by a GPO and apply its contents to the registry of the GP Client computer.	[MS-GPREG]
Group Policy: Security Protocol Extension	Enables the GP Administrator to distribute and apply group security policies to multiple client systems. This extension has both administrative-side and client-side implementations. The administrative-side extension enables the GP Administrator to author security policies as .inf files and save them to the GP FS. The GP Administrator assigns security policies by specifying a reference, within the logical structure of a GPO, to the GP FS network location where the security policy files reside. The client-side extension is invoked by the core GP engine on the GP Client to process GPOs that have a reference to security policies. The client-side extracts the GP FS network location from the GPO, transfers the security policy files to the GP Client computer by using a remote file access protocol, and then utilizes the retrieved security policy files to configure the security settings of the applicable subsystems on the GP Client computer.	[MS-GPSB]
Group Policy: Scripts Extension Encoding	Provides a mechanism for the GP Administrator to configure the execution of administrator-specified code on specific policy targets at computer start, computer shut-down, user logon, or user logoff. The code executed by specified policy targets is contained in a command-line tool or batch-processing script that resides in the file system of the GP Client computer or at a network file system location. This extension has both administrative-side and client-side implementations. The administrative-side extension enables the GP Administrator to store and retrieve GPO metadata that specifies a directive for running a command at computer startup or shutdown that affects the configuration of a GP Client subsystem. The client-side extension is invoked by the core GP engine on the GP Client to identify the directive that runs the administrator-specified command and to configure a command execution subsystem in the GP Client operating system with this directive, such that it executes the command at computer	[MS-GPSCR]

Protocol Name	Functional Description	Short Name
	startup or shutdown.	
Group Policy: Software Installation Protocol Extension	Enables a GP Administrator to install, update, and remove software applications on GP Client computers. This extension has both administrative-side and client-side implementations. The administrative-side extension enables the GP Administrator to specify applications to be installed on GP Client computers and to control the manner in which they are installed, for example, with minimum user interaction. The related settings are stored on the GP FS and the metadata that specifies the path to the settings is stored in the logical structure of a GPO. The client-side extension is invoked by the core GP engine on the GP Client to locate the GPO(s) containing software installation settings, retrieve those settings from the appropriate GP FS location, and apply them on the GP Client computer.	[MS-GPSI]
Group Policy: Wireless/Wired Protocol Extension	Enables a GP Administrator to create, update, and store GPWL data in a GPO. This extension has both administrative-side and client-side implementations. The administrative-side extension is used by the GP Administrator to read and edit wireless or wired policy settings through a user interface, and to store the settings within the logical structure of a GPO via LDAP. The client-side extension is invoked by the core GP engine on the GP Client to retrieve the wireless or wired policy settings from the specified location via LDAP, and to apply them on the GP Client computer.	[MS-GPWL]

Sections 2.1.2 and 2.1.3 describe the major functions and interactions of these protocol groups.

The subsections that follow provide additional technical details about these protocol groups.

2.2.1 Core Protocol Group

The Group Policy: Core Protocol is required for successful Group Policy processing via the core GP engine. The core GP engine enables clients to discover and retrieve data from GPOs created by GP Administrators.

In policy application mode, the core GP engine is responsible for invoking the message sequences that discover the GP Server and obtain a list of GPOs that apply to a policy target, such as a GP Client computer or interactively logged-on user. The retrieved GPOs specify policy settings that are to be applied to a policy target by one or more extensions. The core GP engine is also responsible for invoking the extensions so that their settings can be applied to the policy target. The core GP engine of itself does not recognize the internal details of specific extensions or the settings that it applies.

In the policy administration mode, the Administrative tool makes use of Group Policy: Core protocol messaging when authoring and modifying extension-specific settings.

For additional information about the Group Policy: Core Protocol [MS-GPOL], refer to section 1.1.

2.2.2 Group Policy Extension Protocol Group

The GP System is extended through CSE functionality. The GP System supports CSEs for the application of specific client functionality, such as the client security policies specified in [MS-GPSB], and supports Administrative tool extensions for authoring extension-specific settings, such as the security settings specified in [MS-GPIPSEC].

CSEs are used for implementing application-specific policy settings on GP Client computers. CSE protocols depend on the core GP engine to execute on the GP Client to identify GPOs that the extension should query for policy application.

GPOs with settings for a particular extension are identified with an Administrative tool extension GUID, to enable the Administrative tool to identify the extension and administer its settings. Such extensions, for example, those specified in [MS-GPSB], typically use LDAP to store and retrieve GPO attributes in Active Directory and use a remote file access protocol to store and retrieve policy settings that reside in policy files on the GP FS.

Policy settings for a given class of functionality are communicated by the extension protocol and not directly by the core GP engine. If an extension is not present or policy settings related to an extension are not present, then that specific extension is ignored by the core GP engine.

The presence of extensions is not required for the GP System to function. For additional information about GP Extensions, refer to section 1.1.4.

2.3 Environment

Group Policy depends on a number of prerequisites to facilitate the configuration, application, and utilization of Group Policy by GP Client computers. There are core networking protocols and services that must be open, running, and configured to handle the query and response messages that facilitate the application of Group Policy. For example, the network must be capable of supporting TCP/IP traffic for protocol communications such as DNS, LDAP, and a remote file access protocol, to support the lookup, transport, and transfer of services and policy data. The network must also support Netlogon (with Kerberos v5 [RFC4120]) authentication and authorization traffic. In addition, firewalls residing on clients and servers must have open TCP ports for all services that support Group Policy.

An example of a supporting service is the Domain Name System (DNS), which facilitates the correlation of service names to IP addresses during the GP Server discovery process.

A GP Server utilizing the LDAP protocol is required to store GPO attributes. After discovering the location of the GP Server through DNS, the core GP engine on the GP Client queries the GP Server to discover and calculate which policies apply to it and where to find the necessary policy files for application. It also uses the GP Server to discover WMI filters that determine whether a particular policy applies to the GP Client. In a large business or government network, it is common to have a number of GP Servers in the network for redundancy and performance, each with a copy of the LDAP-accessible database for replication and data consistency. For more information about Group Policy in replication scenarios, see section 2.7.1.5.

The GP System uses a GP FS that supports communications via a remote file access protocol to store policy data in a specific service location, to which the GP Client must have full read access. The GP FS can be co-located on the GP Server along with the Active Directory data store, or can be hosted in a remote network location.

2.3.1 Dependencies on This System

Windows components and subsystems that require configuration and change management depend on the GP System. As a result, Group Policy influences a large number of systems and protocols. The most prominent examples of protocols and systems that have a dependency on the GP System are as follows:

Certificate Services [MS-CERSOD] — provides a set of customizable services for issuing certificates to requestors, managing certificate lifetime and renewals, and revoking certificates. Certificates are used in software security systems that utilize public key technologies, to bind the identity of a person, device, or service to an associated private key.

The Certificate Services System depends on the GP System for the following:

- Group Policy store: The Certificate Authority server depends on a Policy Server to store policy end point information that can be obtained through the Group Policy: Registry Extension Encoding [MS-GPREG] protocol.
- Policy Server discovery: The Certificate Authority server depends on Group Policy to enable enrollment clients to discover available certificate Policy Servers. For example, clients who enroll for certificates must first be configured with end point information that specifies which Policy Server to contact and how to authenticate to it. The Certificate Services System relies upon Group Policy to store and configure this information with the Administrative tool.

File Access Services System [MS-FASOD] — provides a unified view of files and other resources, and includes facilities for centralized data management, file organization, and backup. It enables applications to access and share resources on a network file server, in a secure and managed environment.

The File Access Services System depends on the GP System for the following:

- Configuration of individual protocol capabilities within the File Access Services System. Without the GP System, the File Access Services System cannot be centrally configured and managed.

Print Services System [MS-PRSOD] — supports communication between print clients and **print servers**. The system enables print clients to submit print jobs to print queues that are managed by a print spooler component, which buffers and orders print jobs arriving simultaneously from multiple print clients. The Print Services System uses print drivers associated with the print queues to learn about printer capabilities. The Group Policy: Core Protocol [MS-GPOL] and Group Policy: Deployed Printer Connections Extension [MS-GPDPC] protocol provide support to the Print Services System.

The Print Services System depends on the GP System for the following:

- Propagating policy settings to print clients and print servers through the Group Policy: Core Protocol [MS-GPOL] to control local spooler behavior.
- Restricting print clients from accessing specified print servers.
- Remotely pushing pre-configured print queue connections to print clients, so that print clients have pre-established connections to specified print queues. The Print Services System uses the Group Policy: Deployed Printer Connections Extension [MS-GPDPC] protocol to distribute these pre-configured print queue connections to print clients.

Network Access Protection Protocols System [MS-NAPOD] — manages and enforces compliance with system health requirements for computers in the enterprise. The system uses network infrastructure capabilities and other NAP components to restrict network access to computers that are non-compliant with policy.

The Network Access Protection Protocols depend on the GP System for the following:

- Controlling NAP client behavior through updates to the client registry via the Group Policy: Network Access Protection (NAP) Extension [MS-GPNAP] protocol.
- Automatically updating the local client computer configuration.
- Enabling a client computer to retrieve NAP-specific policy settings from a GP Server.
- Enabling the Administrative tool to retrieve, create, update, and delete NAP-specific policy settings.
- Providing support for domain configuration scenarios with access to directory services and GPOs.

Windows Server Update Services System [MS-WSUSOD] — provides centralized update management in an enterprise computing environment. The system provides automated update discovery, delivery of relevant updates to computers, administrative control over update availability, and update activity monitoring.

The Windows Server Update Services system depends on the GP System for the following:

- The Windows Update Agent uses Group Policy to configure policy settings for Windows Update Services: Client-Server Protocol (WUSP) clients, which includes the specification of an update server, target groups, and detection frequency, as described in [MS-WUSP] section 3.2.1.
- The WSUS administrator uses Group Policy to assign and distribute settings that control the behavior of the WUSP client [MS-WUSP].

Group Policy Extensions — Group Policy is designed to be extended. Microsoft has implemented many extensions that depend on the GP System to implement the specific configuration supported by a given Group Policy extension.

Note that additional extensions to the GP System are possible, beyond those described in this document. Implementers are free to create custom GP Extensions to enhance the functionality of the GP System, as described in [MS-GPOL] section 1.8.

2.3.2 Dependencies on Other Systems

The GP System depends on the following to maintain consistent availability:

- **Connectivity** — the GP System requires physical network connectivity and correctly configured TCP/IP configuration on both the GP Server and the GP Client. There is no specific requirement for the type of physical networking topology.

  The connectivity from the GP Client to the GP Server should be continuous. New and existing policies should be periodically refreshed with updates. The client should be able to tolerate network outages and refresh for policy changes when reconnected to the network.<9>
- **LDAP directory services and file system access** — to provide GP System services to GP Clients, the GP Server must provide LDAP and remote file access services as depicted in the diagram of section 2.1.2.1.
- **Authorization** — the GP System depends on the SPNEGO authentication service [MS-SPNG] to negotiate the specific authentication scheme. The GP System relies on authentication protocols and the SPNEGO service to assist in determining which policies apply to the computer and the user.

- **DC discovery** — the GP Client depends on an IP address of a correctly configured DNS server, to discover and resolve hostnames of GP Servers and connect to them.
- **Policy store** — the GP Client depends on a local store<10> for the storage of specific policy information obtained from the GP Server for the following purposes:
  - To register the extension libraries that will process the settings in the policy files.
  - To persist the policies into user and machine configurations, as this information is not stored in memory.

The GP System depends on the following systems or protocols for the exchange of information between the GP Client and GP Server:

- **Active Directory** — as described in [MS-ADTS]; this is the Windows-based directory service that stores information about objects on a network and makes this information available to users and network administrators. Administrators link GPOs to Active Directory containers such as sites, domains, and OUs, and can also include user and computer objects. This enables policy settings to target specific users and computers throughout an organization.

  The GP System requires Active Directory for storing group policies, so that GP Clients can discover and retrieve them. See [MS-ADOD] for in-depth descriptions of how the directory service is structured and how LDAP operations are conducted.
- **Authentication** — as described by the following authentication protocols:
  - Simple and Protected Generic Security Service Application Program Interface Negotiation Mechanism (SPNEGO) Protocol Extensions, as described in [MS-SPNG] and [MS-AUTHSOD].
  - Kerberos Protocol Extensions, as described in [MS-KILE] and [MS-AUTHSOD].
  - NT LAN Manager Authentication Protocol, as described in [MS-NLMP] and [MS-AUTHSOD].
- **DNS** — for discovering GP Servers.
- **File Access Services System** — as described in [MS-FASOD], for the following:
  - Accessing the GP FS via a remote file access protocol.
  - The distribution of Group Policy.
- **Internet Control Message Protocol (ICMP)** — as described in [RFC792], for use in determining link speed.
- **LDAP** — as described in [RFC2251], for transmitting policy settings and instructions between the GP Client and the GP Server.
- **Netlogon Remote Protocol** — as described in [MS-NRPC], to enable the GP Client and Administrative tool to locate a writeable domain controller ([MS-ADOD] (section 3.1.1)) for the retrieval of GPO data in the Active Directory data store.
- **NetBIOS** — an alternate service for discovering a GP Server, as described in [MS-ADOD] (section 3.1.1).
- **Remote file access protocol** — as described in [MS-FASOD], for transmitting policy settings and instructions between the GP Client and the GP FS.

- **Windows Management Instrumentation Remote Protocol** — as described in [MS-WMI], for Group Policy filtering. During GPO processing, the core GP engine evaluates WMI filters to determine whether a GPO is within scope for computers or users. WMI filtering configurations ensure that policy settings are applied only to specific policy targets, while others are filtered out.

2.3.2.1 Network Connectivity

This system has no additional network connectivity considerations.

2.3.2.2 Underlying Protocols

This system specifies no underlying protocols.

2.3.2.3 Persistent Data Storage Facilities

The GP System requires a persistent storage facility to maintain Abstract Data Model (ADM) elements. Examples of such a facility include file systems and databases. If this requirement is not satisfied, the GP System will not function.

The GP System ADM is based on the conceptual models specified in [MS-GPOL] sections 3.1.1, 3.2.1, and 3.3.1. General information about the GP Server, GP Client, and Administrative tool ADMs for the GP System follows:

Server Abstract Data Model — the GP Server implements AD DS for the storage of managed generic objects known as GPOs, along with the policy information that affects these objects. The GP Server itself has no knowledge of Group Policy protocols and therefore does not introduce any specific ADM elements. Rather, the GP Server maintains state in two conceptual stores: an Active Directory data store and a domain-based GP FS data store that is accessible through a remote file access protocol.

For additional information about the GP Server ADM, see [MS-GPOL] section 3.1.1.

Client Abstract Data Model — the GP Client ADM is described in [MS-GPOL] section 3.2.1.

Administrative Tool Abstract Data Model — the Administrative tool ADM is specified in [MS-GPOL] section 3.3.1.

Note that extending the Administrative tool requires the use of the ADM.

2.4 Assumptions and Preconditions

Preconditions for Group Policy: Core Protocol communications between a GP Client and a GP Server are as follows:

- The GP Server is assumed to be a writeable domain controller.
- The GP Client must be joined to the GP Server domain.
- For user policy mode, the GP Client must be joined to a domain for which the user domain has a bidirectional domain trust.
- All GP Servers in the domain must be configured to require signing of traffic from remote file access operations, for example, as described in [MS-SMB] section 3.2.4.2.4.
- All GP Servers in the domain must be configured to require signing of LDAP traffic, as described in [RFC2251] section 4.2.2.

The following preconditions also apply to the GP Client:

- In order to process a policy that applies to a GP Client, the core GP engine must be able to read the policy data from the directory service so that the policy settings can be applied to the GP Client or the interactive user. It is therefore a requirement that access control list (ACLs) are correctly configured to allow the policy to be read.

2.5 Use Cases

This section describes the basic use cases that explain the main usage of the GP System.

Actors

The following actors support the use cases outlined in this section:

Group Policy Administrator — an individual who is responsible for configuring policy settings that align with organizational and business needs. The primary interests of the GP Administrator are as follows:

- Ensuring that policy settings stored in the GP Server are protected from unauthorized use.
- Targeting policy settings for users and computers at different levels of granularity, which is known as SOM (section 1.1.8).
- Ensuring that management of policy settings can be delegated as described in [MS-ADTS].
- Altering the default processing of policy settings.
- Configuring a large number of computers to execute administrator-specified code at computer start, computer shut-down, user logon, or user logoff, as described in [MS-GPSCR].

GP Server — a domain controller that holds a database of GPOs that GP Clients can retrieve. The primary interests of the GP Server are as follows:

- Enabling a GP Client to retrieve Group Policy information from the domain, based on the group memberships of domain accounts and domain account locations in the Active Directory structure.
- Supporting Administrative tool operations, such as creating, updating, and deleting Group Policy content.

Administrative tool — a tool used to administer policy settings. The primary interests of the Administrative tool are as follows:

- Enabling GP Administrators to create, update, and delete policy settings by writing and reading policy information to and from the logical and file system components of GPOs.

Supporting services — the services that provide a common infrastructure to support GP System operations:

- Remote file services [MS-FASOD]
- LDAP directory services [RFC2251]
- Domain controller discovery ([MS-ADOD] (section 3.1.1))
- WMI services [MS-WMI]

Authentication services — the authentication services specified in [MS-AUTHSOD] provide identity, authentication, and authorization services through NTLM [MS-NLMP] or Kerberos [RFC4120] to secure communications in the GP System. This includes authentication services that support client-to-server communication within the GP System.

2.5.1 Use Case Diagram

The following two GP System use cases are depicted in the diagram that follows:

Applying Group Policy - GP Client

Administering Group Policy - GP Administrator

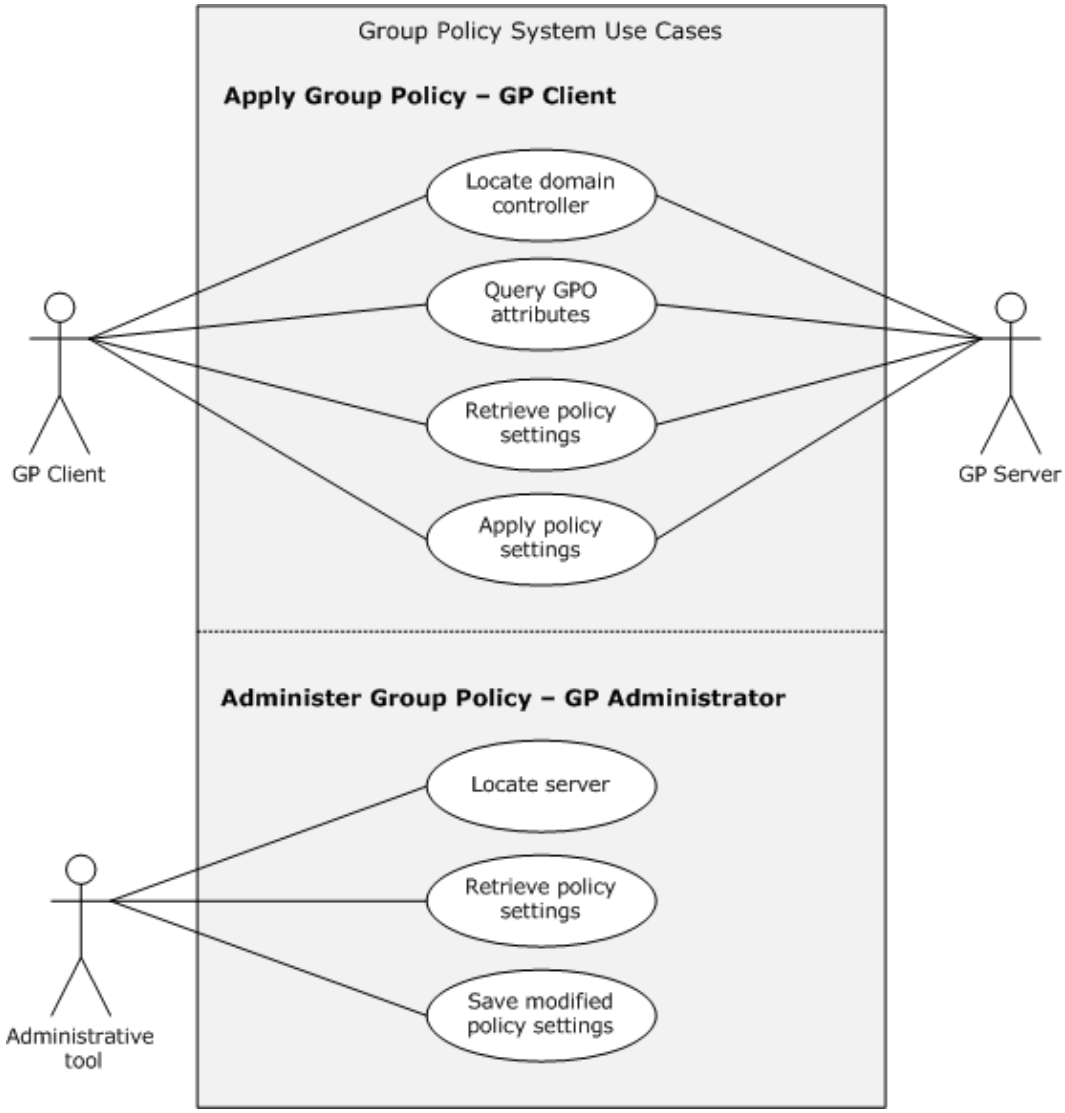

Figure 9: GP System use case diagram

2.5.2 Applying Group Policy — GP Client

Context of use

Group Policy is applied after the GP Client contacts the GP Server and successfully retrieves new or updated content. Based on the SOM, the client retrieves the list of GPOs for policy application, as described in [MS-GPOL] section 3.2.5.1.5.

Goal

The goal of this use case is to retrieve Group Policy information from the GP Server and to apply policy settings on the GP Client.

Actors

GP Client — maintains a policy configuration that is consistent with the policy information stored on the GP Server. This is the primary actor. The primary interests of the GP Client are to:

- Retrieve policy content from the GP Server.
- Ensure that policy settings defined by the GP Administrator are enforced on the GP Client computer.

GP Server — a domain controller that contains a database of GPOs that GP Clients can retrieve. The GP Server responds to requests from the GP Client. The primary interests of the GP Server are to:

- Enable a GP Client to retrieve Group Policy information from the domain, based on the group memberships of domain accounts and domain account locations in Active Directory.
- Support Administrative tool operations, such as creating, updating, and deleting GPOs.

Stakeholders

Users — an individual who uses a Group Policy-enabled computer and whose primary interests are to understand the following:

- How the user experience is influenced by policy settings that affect computers.
- How Group Policy specifically applies to users.

Group Policy Administrator — an individual who is responsible for configuring policy settings that align with organizational and business needs. The primary interests of the GP Administrator are as follows:

- Ensuring that policy settings stored in the GP Server are protected from unauthorized use.
- Targeting policy settings for users and computers at different levels of granularity, which is known as SOM.
- Ensuring that policy setting management can be delegated as described in [MS-ADTS].
- Altering the default processing of policy settings.
- Configuring a large number of computers to execute administrator-specified code at computer start, computer shut-down, user logon, or user logoff, as described in [MS-GPSCR].

Preconditions

- **GP Client** — must be able to access the GP Server.

Main Success Scenario

The main success scenario can be summarized as follows:

1. Triggers: computer startup, user logon, or the periodic timer (sections 2.8.1 and 2.8.2) trigger this use case. When a trigger occurs, the GP Client successfully connects to the GP Server.
2. The GP Client is able to query for applicable policy configuration settings from the GP Server.
3. The GP Client successfully retrieves the policy information based on the results from queries.
4. The GP Client applies the policy settings.

Extensions

- Based on WMI filters, the GP Client decides whether to apply a specific GPO.
- Based on the policy source mode, as described in [MS-GPOL] sections 3.2.1.2 and 3.2.1.3, the GP Client obtains a set of GPOs that apply to itself.

2.5.3 Administering Group Policy — Administrative Tool

Context of use

The GP Administrator initiates a task defined in the goal for this use case.

Goal

The goal of this use case is to create, update, and delete Group Policy content.

Actors

Administrative tool — a tool used by the GP Administrator to manage GPOs. This is the primary actor. The primary interests of the Administrative tool are to:

- Discover the GP Server.
- Ensure read and write access to the GP Server.
- Manage Group Policy.

GP Server — a domain controller implementing Active Directory [MS-ADOD] that contains a database of GPO that GP Administrators can read and write to. The GP Server responds to requests from the GP Administrator. The primary interests of the GP Server are to:

- Support Administrative tool operations, such as creating, retrieving, modifying, and deleting GPOs that apply to groups of domain user and computer accounts in Active Directory.
- Store policy settings and attributes configured by the GP Administrator.

Stakeholders

GP Administrator — an individual who ensures that the GP Server is storing policies that align with business and organizational needs. The primary interests of the GP Administrator are to:

- Ensure that policy settings are stored on the GP Server.
- Create, retrieve, modify, or delete Group Policy content on the GP Server.

Preconditions:

- The Administrative tool must be able to access the GP Server.
- The GP Server must be a read/write domain controller, not a read-only domain controller.

Main Success Scenario

The main success scenario can be summarized as follows:

1. Trigger: the GP Administrator launches the Administrative tool. When a trigger occurs (section 1.1.7.1), the Administrative tool successfully connects to the GP Server.
2. The Administrative tool is able to query for policy information on the GP Server and successfully retrieve the prioritized GPO list based on query results.
3. The Administrative tool displays the prioritized GPO list.
4. The GP Administrator updates, creates, or deletes policy information with the Administrative tool.
5. The Administrative tool successfully writes updated information to the GP Server.

Extensions

- None.

2.6 Versioning, Capability Negotiation, and Extensibility

This section outlines the features of versioning, capability negotiation, and vendor-extensible fields for the GP System.

2.6.1 System Versioning and Capability Negotiation

The GP System is a collection of protocols each with its own system versioning and capability negotiation. The GP System itself does not provide capability negotiation but relies on the member protocols to perform this action.

The GP System relies on the Group Policy: Core Protocol, as implemented in the core GP engine, for the transport of policy information. It provides a versioning capability in an attribute of the Active Directory object class for a GPO, as described in [MS-GPOL] section 2.2.4. The version number is a simple integer that is also written to the gpt.ini file on the GP FS, as described in [MS-GPOL] section 2.2.4. There is currently only one version, and if the GP Client receives anything other than the current version for a GPO, the GPO does not participate in the Group Policy: Core Protocol, as described in [MS-GPOL] section 3.2.5.1.5.

The System Versioning and Capability Negotiation implementation of extension protocols is documented in the respective extension protocol specifications. They are described in section 1.7, Versioning and Capability Negotiation of the respective protocol technical documents.

2.6.2 System Vendor-Extensible Fields

The GP System contains a collection of protocols. The system can incorporate new functionality by adding new extensions to the GP Client or the Administrative tool. Each new extension can also potentially be extended. See [MS-GPOL] section 1.8 for more information about implementing extensions on the GP Client. Extending the Administrative tool requires the use of the ADM specified in [MS-GPOL] section 3.3.1.

The system vendor-extensible fields of each extension protocol are documented in the respective extension protocol specification. These are specified in section 1.8 Vendor-Extensible Fields of the respective technical documents.

2.7 Error Handling

The GP System does not define any error handling requirements beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

Various kinds of errors may occur that affect the system. More precisely, an error condition may affect one or more protocols supported by the system. Such error conditions and the resulting protocol semantics are described in section 2 of the corresponding protocol specifications.

Windows returns the following error codes for the failure scenarios described in this section:

- Connection failures: ERROR_NO_SUCH_DOMAIN.
- Failures related to the operating system: ERROR_OUTOFMEMORY and ERROR_ACCESS_DENIED.
- GP FS access failure: ERROR_FILE_NOT_FOUND and ERROR_ACCESS_DENIED.
- Active Directory or GP FS time-out failures: ERROR_TIMEOUT.
- Client-side extensions indicate errors by returning an error code other than ERROR_SUCCESS or E_PENDING.

2.7.1 Failure Scenarios

This section describes common failure scenarios and specifies the system behavior under such conditions.

2.7.1.1 Connection Failure

A common failure scenario is an unexpected connection breakdown between the GP Server and the GP Client or between the GP Server and the computer hosting the Administrative tool. A disconnection can be caused by the network not being available or by the GP Server becoming unavailable. In both cases, where the network or the GP Server is not available, the effect on the GP Client and the Administrative tool is as follows.

- When the GP Client is unable to reach the GP Server, the policy application fails, and a message is logged in the event log. The GP Client will periodically try to contact the GP Server to refresh its policy settings.<11>
- When the Administrative tool is unable to reach the GP Server, for example, due to network or GP Server unavailability, an error message is displayed to the Group Policy Administrator. It is up to the Group Policy Administrator to retry the task when the issue has been resolved.

2.7.1.2 Internal Failures

2.7.1.2.1 Operating System-Related Failures

It is possible that the GP Client or the Administrative tool may detect an unrecoverable internal state at some point during its operation. For example, this might occur due to the unavailability of some operating system resources. For this kind of failure, the consequences and recovery are similar to those for the connection failure described in section 2.7.1.1. This kind of failure is detected when the

operating system indicates that it could not allocate virtual memory, or was unable to access critical system resources. Recovery from this failure should allow successful policy application.

2.7.1.2.2 Failure in Client-Side Extensions

An internal failure in any CSE does not cause the entire policy application to fail. The consequence of this failure is that the settings corresponding to that protocol extension are not applied to the system. The failure is detected when a CSE indicates an error. At the next scheduled policy application, the GP Client will call the CSE again, in an attempt to recover from the failure. Recovery from the failure allows the successful application of settings corresponding to the CSE. If a CSE for which a policy is configured is missing from the client, the GP Client ignores the policy for that extension and continues with application of policies for other applicable extensions. It is not an error condition for a CSE to be absent from the GP Client.

2.7.1.2.3 Link Speed Determination Failure

If a failure in link speed determination occurs ([MS-GPOL] section 2.2.6), the GP Client assumes link speed to be above the threshold and processes policy settings belonging to all CSEs. At the next scheduled policy application, the GP Client initiates link speed determination again in an attempt to recover from the failure. Recovery from the failure helps prevent application of policies from those CSEs that should not be invoked when link speed is below threshold.

If the link speed cannot be determined, all policies are applied to ensure that critical functionalities are in place.

2.7.1.3 History Repository Errors

The GP Client maintains a history of policy application to optimize client performance and certain cleanup tasks. If the history repository is corrupted or lost, the GP Client proceeds as though the policy is being applied for the first time and re-creates the history repository.

2.7.1.4 GP File Share Access Failure

The GP Client may not be able to access a file on the GP FS via a remote file access protocol for one of the following reasons:

- File replication delays
- File permissions not configured correctly by the Group Policy Administrator

As a consequence of this failure, the GP Client will be unable to apply any policy. At the next scheduled policy application, the GP Client will attempt to apply policy again. Recovery from this failure ensures that the client has the latest set of policies.

2.7.1.5 GP Failures Related to Active Directory Replication

In a single DC domain, there is no impact on Group Policy that is associated with Active Directory replication. However, in multiple-DC domain scenarios, directory replication introduces a time delay that can defer the application of Group Policy in a domain until all data is successfully propagated to all DCs. During this delay period, prior modifications to Group Policy configurations will not be applied to policy targets in replication domains. GP Administrators should take note that this is not an error, although it can appear to be.

However, if AD replication actually fails, Group Policy will continue to function normally in its pre-existing state, but any updates to Group Policy configurations will not be applied until a successful replication occurs and the aforementioned delay period has expired.

2.8 Coherency Requirements

2.8.1 Timers

The GP Client should have the following timer:

Periodic Refresh timer — this timer should be triggered periodically to check for an updated policy for the computer or for each user interactively logged on to the computer. The frequency of this timer is implementation-specific.<12> For more information about GP Client periodic refresh timers, see [MS-GPOL] section 3.2.2.

2.8.2 Non-Timer Events

Events associated with policy application include the following:

- **Computer boot or new connection** — policy application in computer policy mode should be invoked when a client machine boots or connects to a new network.
- **User logon or new connection** — policy application in user policy mode should be invoked when a user logs on or connects to a new network.
- **GPUpdate.exe** — an update event can be set via GPUpdate.exe to supersede the periodic refresh timer functionality and allow policy to be applied at any time.
- **Policy change event** — a local PolicyChange event is triggered at the end of policy application to indicate that a policy has changed. To receive notification of this event, see the **RegisterGPNotification** function described in the Group Policy API reference documentation [MSDN-GroupPolicy].

Policy application can also be invoked at other times, as described in section 2.8.1.

Events associated with the use of the Administrative tool include the following:

- **GPO creation** — Group Policy is created when the GP Administrator uses the Administrative tool to create a GPO. This process triggers a GPO Creation message, as described in [MS-GPOL] section 2.2.8.1.
- **GPO property update**— a Group Policy property update occurs when the GP Administrator uses the policy administration sequence of a GP Extension protocol to change the properties of a GPO. This process triggers a GPO Property Update message, as described in [MS-GPOL] section 2.2.8.3.
- **SOM property update** — an SOM property update occurs when the GP Administrator uses the policy administration sequence of a GP Extension protocol to change the properties of an Active Directory container object in the Group Policy domain that is within SOM. This process triggers an SOM Property Update message, as described in [MS-GPOL] section 2.2.8.4.
- **GPO extension update** — a GP Extension settings update occurs when the GP Administrator changes the settings of an extension in a GPO. This triggers a GPO Extension Update message, as described in [MS-GPOL] section 2.2.8.2. In this message, the GPO container and GPO file system version numbers must be computed as described in [MS-GPOL] section 3.3.4.5.

2.8.3 Initialization and Re-Initialization Procedures

The GP Client should register for computer boot and user logon event notifications in the domain to ensure that during initialization, policy application will occur as a result of these events. If the GP Client computer restarts while it is already up and running, the GP Client should recreate the operational state of the computer and all logged-on users.

2.9 Security

This section documents system-wide security issues that are not otherwise described in the Technical Documents (TDs) of the member protocols listed in section 2.2. This section does not duplicate what is already in these documents unless there is some unique aspect that applies to the system as a whole.

In a distributed environment where information is stored and retrieved from clients to the server, it is essential to protect information exchange from tampering. GP System protocols are not intended to transmit sensitive information, and therefore should not be used to transmit it.

2.9.1 Internal Security

This section describes the internal security of the GP Client. The general guideline for GP System implementers is to ensure that the resources used by the core GP engine and extensions are protected from unauthorized access. In addition, users who do not have the required privileges should be prevented from modifying or tampering with administrative configurations.

The following shows the different components that define the security boundaries of the GP System on the GP Client. Elements that are external to the GP System are described in [MS-GPOL].

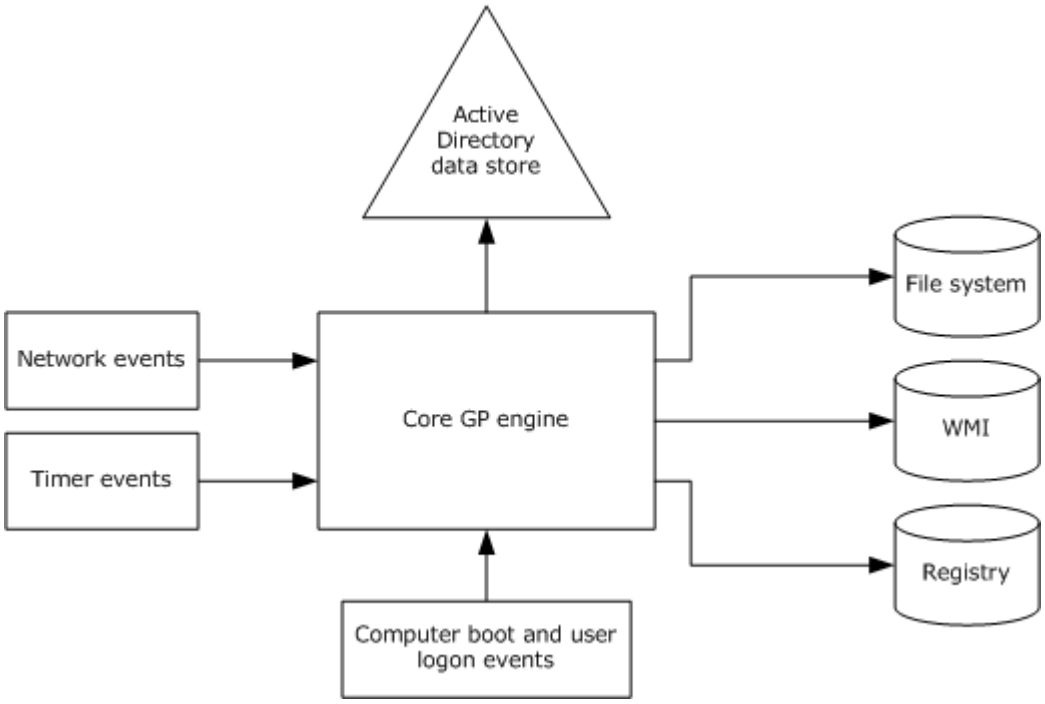

Figure 10: GP System security boundary components

2.9.1.1 Data Store Permissions

The GP System writes policy information to various data stores, such as the GP FS, Active Directory, and the registry, where policy settings are persisted. The GP System ensures that appropriate permissions are set on each resource such that no user can tamper with the data unless the user has permissions to the resource. GP System protocols set user permissions on resources to read only, so they cannot change the data. Group Policy cannot protect against a user with administrative privileges, because that user can take ownership of a resource and change it.

2.9.1.2 Timer and Network Events

The process that applies Group Policy to GP Client computers runs periodically in the background and is triggered by the firing of a timer or a network event, such as a change to the user's network state. Any implementation of GP System protocols should ensure that these event sources are trusted and cannot be spoofed.

2.9.1.3 Computer Boot and Logon Events

The computer boot, computer shutdown, user logon, and user logoff events are used to apply policies to a user or a computer when these events occur. Any implementation of GP System protocols should ensure that the components that generate these events are trusted and cannot be spoofed.

2.9.2 External Security

GP System protocols use the encryption mechanisms provided by the LDAP and remote file access [MS-FASOD] transports to ensure that the data is protected against tampering. The GP System relies on the authentication mechanisms provided by the underlying protocols to establish user and computer identities. These security mechanisms include the following:

- LDAP and remote file access protocol signing, for setting and retrieving policy data
- Kerberos [RFC4120] authentication for application of computer policy, as described in [MS-AUTHSOD] section 3.3.
- SPNEGO authentication for application of user policy, as described in [MS-GPOL] section 5

The GP System does not define any additional external security beyond what is described in the specifications of the protocols supported by the system, as listed in section 2.2.

2.10 Additional Considerations

There are no additional security considerations.

3 Examples

The GP Server allows clients to discover and retrieve policy settings created by domain administrators. Policy settings are directives issued by administrators to control client behaviors. These behaviors are defined by user policy settings and computer policy settings.

This section contains examples that further elaborate the Group Policy concepts outlined in this document, to provide a basis for practical understanding and implementation of the GP Server. Message flow diagrams are included to illustrate the flow of communication as certain events occur.

The examples demonstrate the GP Server system architecture in the context of various scenarios. The functionalities illustrated in these scenarios exemplify some of the purposes of the GP Server:

- Processing Group Policy events.
- Applying policy via the GP Client.
- Populating the Administrative tool with configuration data.
- Authoring new policies.
- Connecting the Administrative tool to a Group Policy server resulting in failure.
- Querying Active Directory for SOM and version information.
- Applying policy via the GP Client resulting in failure to connect to the Group Policy server.

3.1 Example 1: Processing Group Policy Events

This section describes various events that trigger the GP System processing architecture and the resulting sequence of messages that apply Group Policy. This example provides a very high-level view of the sequences that take place in response to specific event occurrences, such as:

- Computer start up.
- User logon to a computer.
- User logoff from a computer.
- Computer shutdown.

This example maps to the use case specified in section 2.5.2, "Applying Group Policy".

Prerequisites

The following prerequisites apply to this example:

- The GP Client must be able to discover and communicate with the GP Server, as described in [MS-GPOL] section 3.2.5.1.1.
- The GP Server must be storing policy and must respond to requests from the GP Client.
- The GP Client must maintain a consistent configuration of policy information retrieved from the GP Server, which includes registry settings, WMI data, and RSoP data.
- The GP Administrator must ensure that the GP Client policy configuration aligns with business requirements.

Initial System State

Prior to the application of Group Policy, the GP System is actively listening for the specific events that will trigger policy application on computers in a domain.

Final System State

The state of the GP System and its components following execution of this example can be described as follows:

- The GP Client retrieved the appropriate policies from the GP System and they were applied on the client.

Sequence of Events

The following figure illustrates the message sequence that occurs in response to events that trigger policy application. The figure also indicates when Group Policy computer startup, computer shutdown, user logon, and user logoff scripts are run.

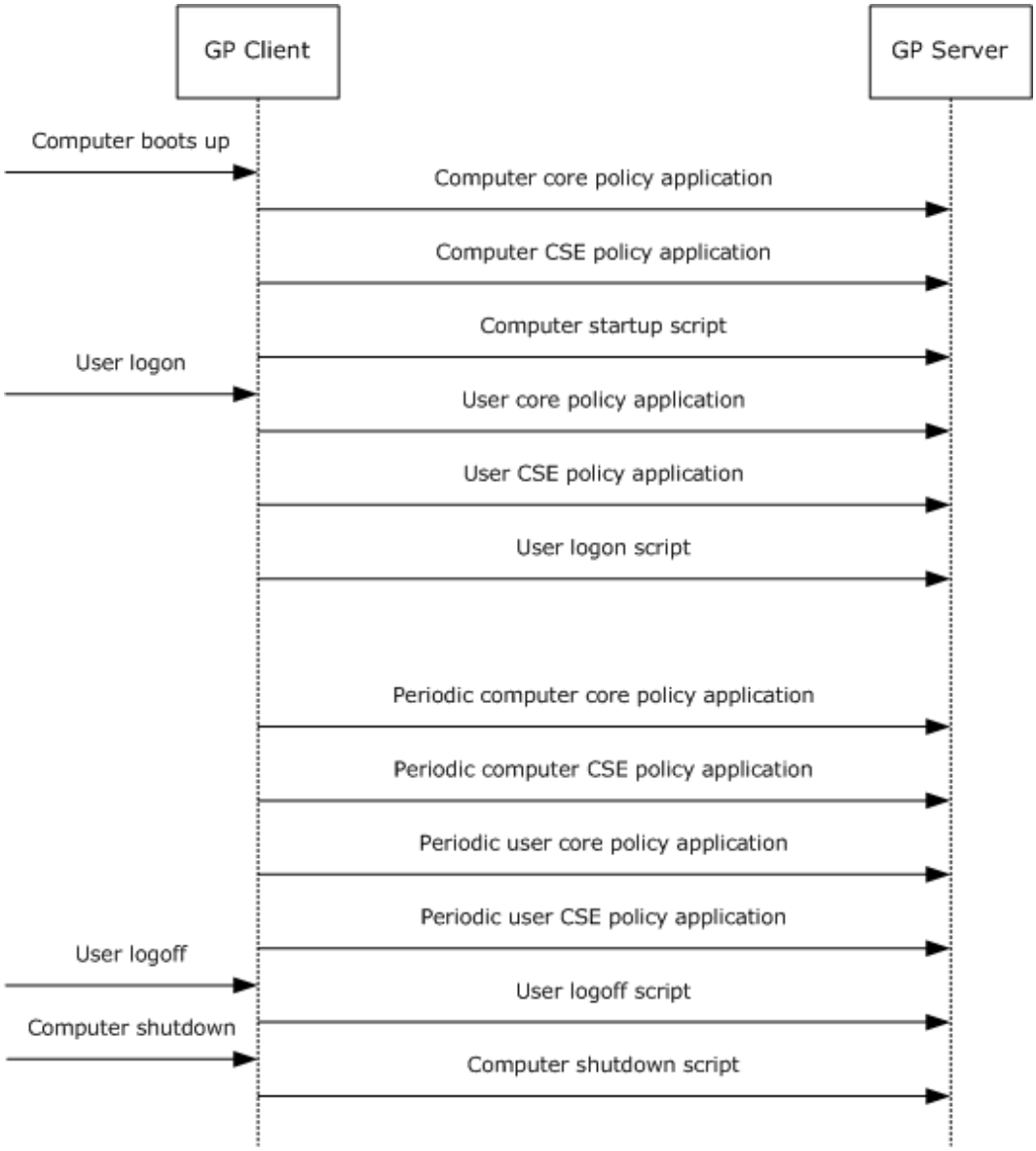

Figure 11: GP System processing internal architecture

The following table provides document references for the messages in the preceding figure.

Group Policy messages and document references

Protocol message	Document name	Section
Computer Core Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	1.3.3, Policy Application
Computer CSE Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	3.2.5.1.10, Extension Protocol Sequences

Protocol message	Document name	Section
Computer Startup Scripts	[MS-GPSCR]: Group Policy Scripts Extension: Protocol Specification	3.2.5, Message Processing Events and Sequencing Rules
User Core Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	1.3.3, Policy Application
User CSE Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	3.2.5.1.10, Extension Protocol Sequences
User Logon Scripts	[MS-GPSCR]: Group Policy Scripts Extension: Protocol Specification	3.2.5, Message Processing Events and Sequencing Rules
Periodic Computer Core Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	1.3.3, Policy Application
Periodic Computer CSE Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	3.2.5.1.10, Extension Protocol Sequences
Periodic User Policy Core Application	[MS-GPOL]: Group Policy: Core Protocol Specification	1.3.3, Policy Application
Periodic User CSE Policy Application	[MS-GPOL]: Group Policy: Core Protocol Specification	3.2.5.1.10, Extension Protocol Sequences
User Logoff Scripts	[MS-GPSCR]: Group Policy Scripts Extension: Protocol Specification	3.2.5, Message Processing Events and Sequencing Rules
Computer Shutdown Scripts	[MS-GPSCR]: Group Policy Scripts Extension: Protocol Specification	3.2.5, Message Processing Events and Sequencing Rules

3.2 Example 2: Applying Policy on the GP Client

The GP Client's interaction with the GP Server in policy application utilizes a pull model, in which the GP Client polls a GP Server to check for new user GPOs.

When the GP Client discovers the GP Server, the GP Client performs two sets of queries to Active Directory on the GP Server using **LDAP** as a transport. The first set of queries determines which GPOs have been assigned.

The second set of queries determines attributes of the relevant policies, discovers the location of the policy files, and determines any exclusionary WMI filtering for GPOs.

The GP Client then checks the link speed and processes any relevant filters to potentially filter down the collective list of extensions.

Lastly, CSEs read the relevant policy settings from the server that are stored in Active Directory and on the GP FS, using LDAP or a remote file access protocol, respectively, and applies them.

This example maps to the use case specified in section 2.5.2, "Applying Group Policy".

Prerequisites

The following prerequisites apply to this example:

- The GP Server must be storing policy information.

- The GP Client must maintain a consistent configuration of policy information retrieved from the GP Server, which includes registry settings, WMI data, and RSoP data.
- The Group Policy Administrator must ensure that the GP Client policy configuration aligns with business requirements.
- The GP Client has discovered the GP Server and connected with Active Directory, as described in [MS-GPOL] section 3.2.5.1.1.
- The GP Client has sent an LDAP **BindRequest** message, as specified in [RFC2251] section 4.2, to the GP Server, and the GP Server has replied with an LDAP **BindResponse** message, as described in [RFC2251] section 4.2.3.
- In this scenario, it is assumed that the GP FS resides on the GP Server.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of this example can be described as follows:

- The GP Client applied the appropriate user and computer policies that were retrieved from the GP data store.

Sequence of events

The following figure illustrates the message sequence that occurs when Group Policy is applied on the GP Client:

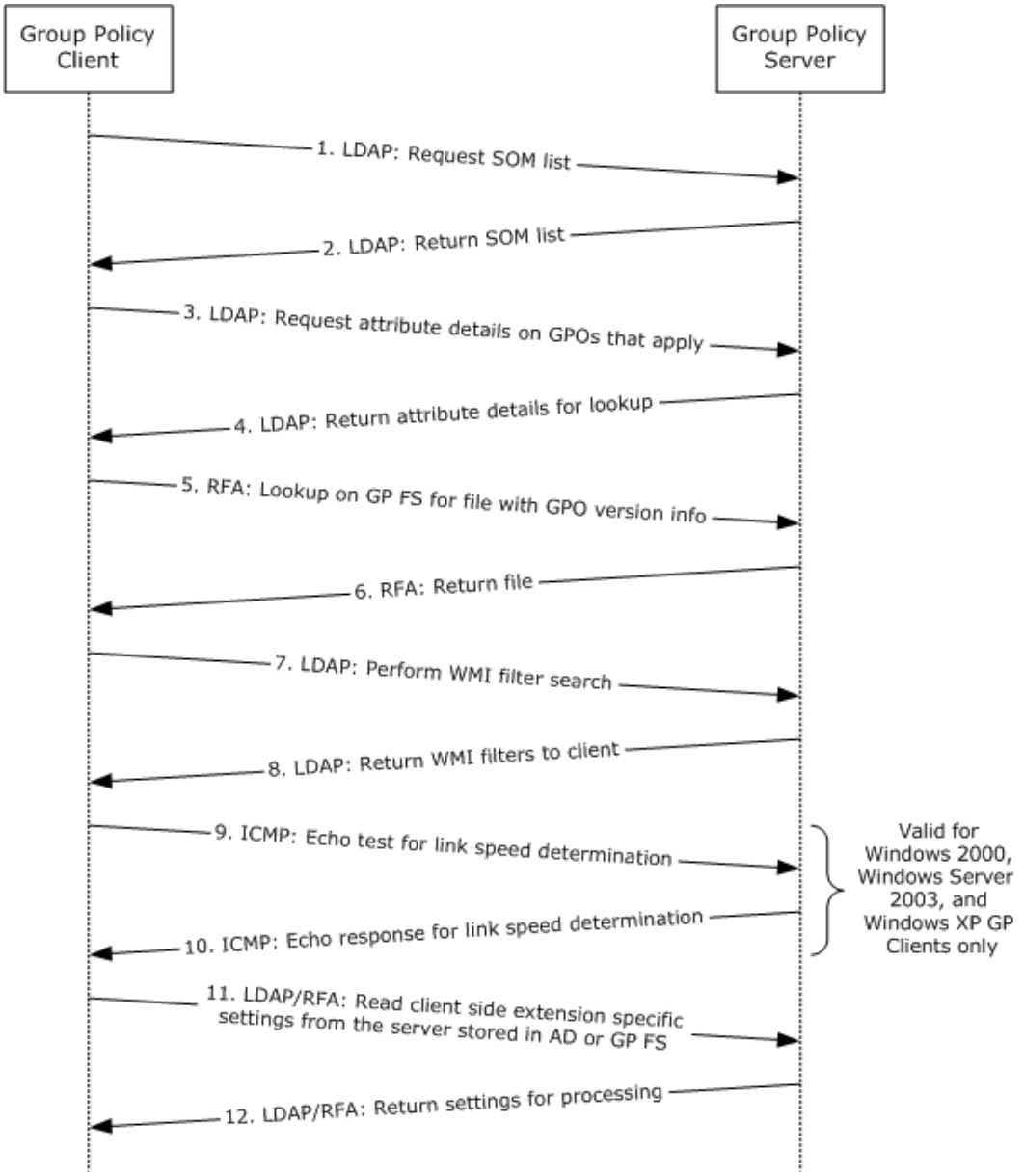

Figure 12: GP Client applying policy

The message sequence for this example is described as follows:

1. The GP Client sends an LDAP request to the GP Server to discover the policies that apply to the user and to the computer. For more information, see [MS-GPOL] sections 2.2.2, 2.2.3, and 3.2.5.1.3.

2. The GP Server sends an LDAP reply to the GP Client that contains a list of policies that apply to the user and to the computer. For more information, see [MS-GPOL] sections 2.2.2, 2.2.3, and 3.2.5.1.3.

3. The GP Client receives the list of policies and then sends an LDAP query to the GP Server to request specific attributes that define further filtering, the location of the policy file, and the precedence order for sequential application of policies and classes of settings. For more information, see [MS-GPOL] sections 2.2.4 and 3.2.5.1.5.

4. Through an LDAP reply, the GP Server returns the list of attributes that the GP Client requested. The GP Client then invokes any extension settings that are defined as part of the returned attributes. For more information, see [MS-GPOL] section 2.2.4 and 3.2.5.1.5.

5. The GP Client sends a remote file access request to the GP FS on the GP Server to read the gpt.ini file that contains version information for the GPO. For more information, see [MS-GPOL] section 2.2.4.

6. The version information from the file is returned to the GP Client in response to the remote file access request. The GP Client parses the file to check the GPO version.

7. The GP Client sends an LDAP request to the GP Server to retrieve any WMI filters that apply to the GPOs in scope for the GP Client. For more information, see [MS-GPOL] sections 2.2.5 and 3.2.5.1.7.

8. The GP Server sends a response back to the client with any relevant WMI filters that apply to the client. For more information, see [MS-GPOL] section 2.2.5.

9. The GP Client may send a separate request to the GP Server to determine the link speed. For more information, see [MS-GPOL] sections 2.2.6 and 3.2.5.1.9.

10. The GP Client receives a response from the GP Server that assists the GP Client in determining link speed. For more information, see [MS-GPOL] section 2.2.6.

11. If a Group Policy update is required, the GP Client sends an LDAP request to the GP Server and a file access request to the GP FS that stores the extension-specific policy settings. For more information, see [MS-GPOL] section 3.2.5.1.

12. The GP Client then retrieves the requested settings and applies them. For more information, see [MS-GPOL] section 3.2.5.1.

3.3 Example 3: Populating the Administrative Tool with Configuration Data

This example demonstrates the processes that occur when the Administrative tool loads and retrieves the appropriate information from the data stores that contain Group Policy data. The Administrative tool is populated with data retrieved from the GP Server.

This example maps to the use case specified in Administering Group Policy (section 2.5.3).

Prerequisites

The following prerequisites apply to this example:

- Policy information stored in the Group Policy data store (GP DS) must align with business and organizational needs.
- The GP Administrator who is running the Administrative tool must have read/write access to Active Directory on the GP Server and to the GP FS.
- The GP Server must be a read/write domain controller (DC).

- The Administrative tool must be able to discover and communicate with the GP Server, as described in [MS-GPOL] section 3.2.5.1.1.

Note that the GP Server (DC) discovery and connection sequence for the GP Client and Administrative tool are identical.

- The computer hosting the Administrative tool must be joined to the domain and the GP Administrator should be logged in with domain credentials of sufficient privilege.
- In this scenario, it is assumed that the GP FS resides on the GP Server.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of this example can be described as follows:

- The Administrative tool retrieved all the existing policies on the GP Server.

Sequence of events

The following figure illustrates the message sequence that occurs when the Administrative tool retrieves GPO data from the GP Server and policy settings from the GP FS.

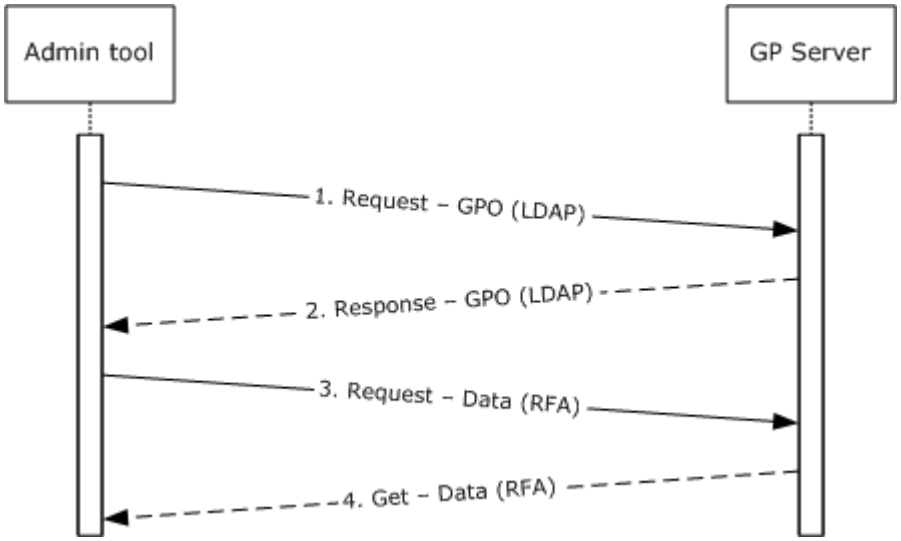

Figure 13: Populating the Administrative tool with data

The message sequence for this example is described as follows:

1. The Administrative tool makes a sequence of LDAP calls to the GP Server to retrieve GPO information via the message types described in [MS-GPOL] sections 2.2.2, 2.2.3, 2.2.4, 2.2.5, and 2.2.7.
2. The GPO information returned in response to the LDAP queries is used to populate the tool.

3. During editing operations, the Administrative tool invokes one or more extension protocols, which communicate with the GP FS via a remote file access (RFA) protocol to return existing policy settings.

4. The returned policy settings information is used to populate the tool.

3.4 Example 4: Authoring a New GPO

This example describes the message flow during new policy authoring. When the GP Administrator chooses to create a new GPO, the GP Server handles the request by provisioning resources in Active Directory for a new GPO and appropriate directories are created on the GP FS. After the new policy is created, the administrator opens the policy and begins setting the policy configuration. As the administrator authors policy settings, the Administrative tool communicates with Active Directory on the GP Server and the GP FS to update these GP data stores with the policy data.

This example maps to the use case specified in Administering Group Policy (section 2.5.3).

Prerequisites

The following prerequisites apply to this example:

- Policy information stored in the GP data store must align with business and organizational needs.
- The Administrative tool must have read/write access to the GP Server.
- The GP Server must be a read/write domain controller.
- The Administrative tool must be able to discover and communicate with the GP Server, as described in [MS-GPOL] section 3.2.5.1.1.
- In this scenario, it is assumed that the GP FS resides on the GP Server.

Note that the GP Server (DC) discovery and connection sequence for the GP Client and Administrative tool are identical.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of this example can be described as follows:

- The GP Server is updated with newly authored Group Policy information.

Sequence of events

The following figure illustrates the message sequence that occurs when the Administrative tool is used to author a new policy.

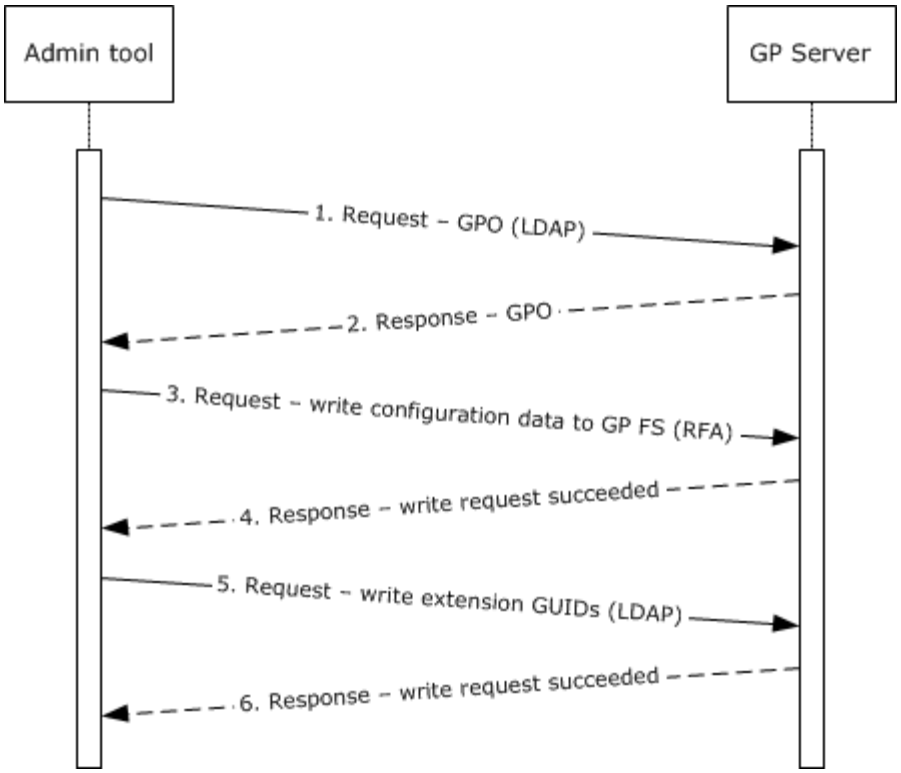

Figure 14: Authoring a new policy

The message sequence for this example is described as follows:

1. When the GP Administrator creates a new GPO, the Administrative tool makes an LDAP call to Active Directory with a request to create the GPO.

2. The GP Server provisions resources in Active Directory to create the new GPO and the appropriate GPO GUID folder and User and Machine subdirectories are created on the GP FS. The GP Server sends a response to the Administrative tool.

3. Extension configuration data is then written to the GP FS via a remote file access protocol, in the folders that were created for the new policy.

4. The GP Server sends a response confirming the success of the write operation.

5. The GP Administrator edits specific policy settings. When a new Administrative tool extension is modified for the first time, an LDAP call is made to Active Directory, and the corresponding Administrative tool extension GUID and the CSE GUID are written to the **gPCMachineExtensionNames** and/or **gPCUserExtensionNames** attributes of the GPO.

6. The GP Server sends a response confirming the success of the write operation.

3.5 Example 5: Administrative Tool Cannot Connect to a GP Server

The examples in this section describe message sequences that occur during the policy administration process that end in failure as a result of a lost connection with the GP Server or a remotely-located GP FS. The following two scenarios are illustrated:

- Failure to contact Active Directory
- Failure to contact the GP FS

The examples in this section map to the use case specified in Administering Group Policy (section 2.5.3).

Prerequisites

The following prerequisites apply to the examples in this section:

- Policy information stored in the GP data store must align with business and organizational needs.
- The GP Server must be a read/write domain controller.
- The Administrative tool must be able to discover the GP Server, as described in [MS-GPOL] section 3.2.5.1.1.

Note that the GP Server (DC) discovery and connection sequence for the GP Client and Administrative tool are identical.

- The Administrative tool must have read/write access to the GP Server.
- For the failure to contact GP FS scenario, it is assumed that the GP FS resides on the GP Server.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of each example in this section can be described as follows:

- The state of the GP System and its components is unchanged.

Sequence of events for Active Directory Connection Failure

The following figure illustrates the message sequence that occurs when the Administrative tool is unable to connect with Active Directory.

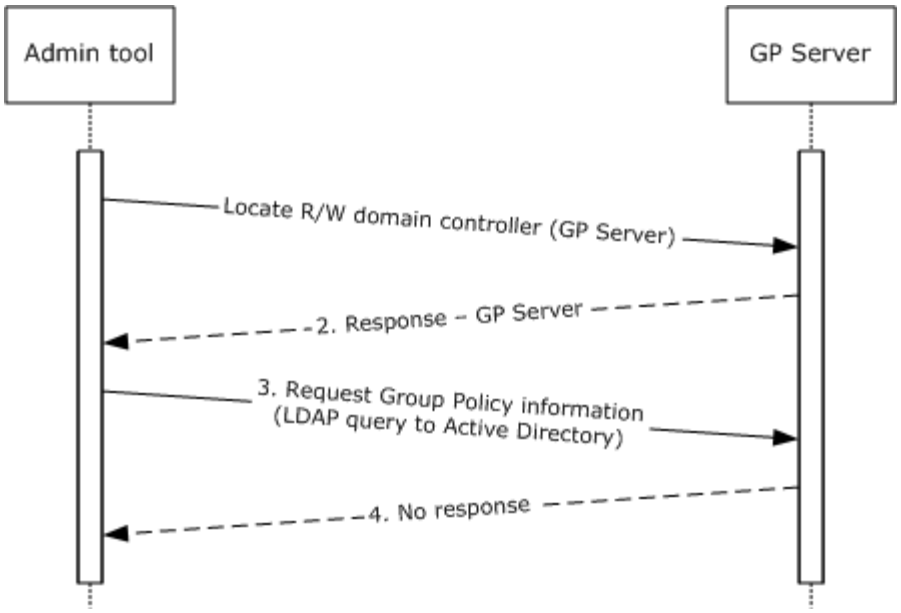

Figure 15: Administrative tool cannot contact Active Directory

The message sequence for this example is described as follows:

1. The Administrative tool attempts to locate the GP Server in the domain by the steps described in [MS-ADOD] (section 3.1.1).
2. The GP Server information for the domain is returned.
3. The Administrative tool sends an LDAP query to Active Directory to retrieve GPO information, as described in [MS-GPOL] sections 2.2.2, 2.2.3, and 2.2.4.
4. The Administrative tool fails to receive a response from the GP Server within a specified time-out interval.

Sequence of events for GP FS Connection Failure

The following figure illustrates the message sequence that occurs when the Administrative tool fails to connect with the GP FS.

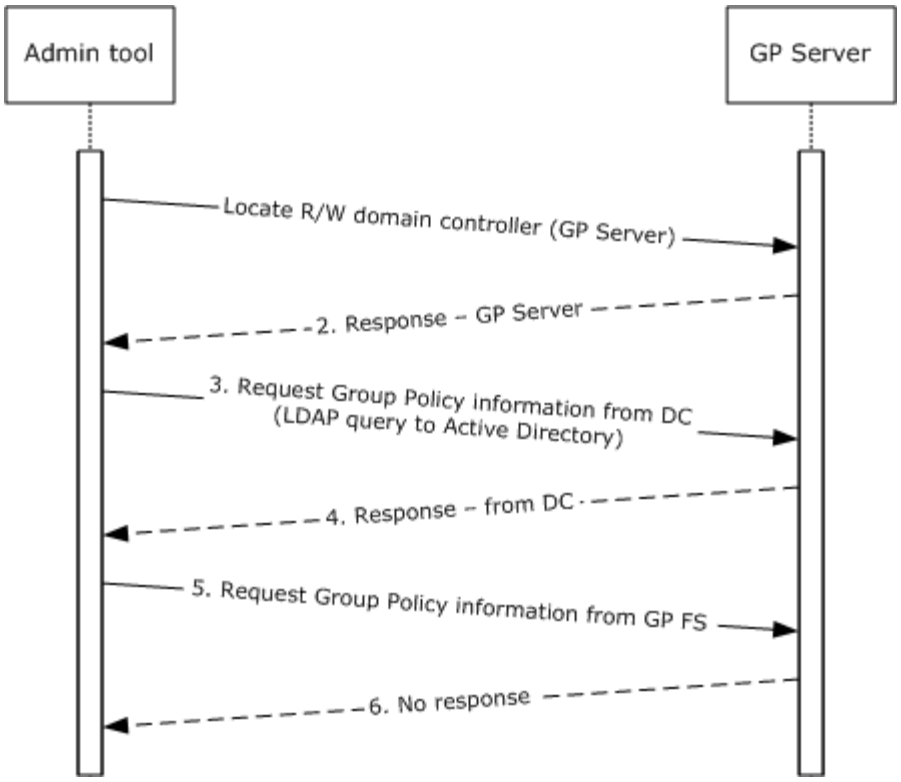

Figure 16: Administrative tool cannot contact the GP FS

The message sequence for this example is described as follows:

1. The Administrative tool attempts to locate the GP Server in the **domain** by following the steps described in [MS-ADOD] (section 3.1.1).
2. The GP Server information for the domain is returned.
3. The Administrative tool sends an LDAP query to Active Directory to request GPO information, as described in [MS-GPOL] sections 2.2.2, 2.2.3, 2.2.4, 2.2.5, and 2.2.7.
4. The Administrative tool receives responses ([MS-GPOL] sections 2.2.2, 2.2.3, 2.2.4, 2.2.5, and 2.2.7) from the GP Server within a specified time-out interval.
5. The Administrative tool requests information from the GP FS on the GP Server, in a manner that is similar to the process described in section 2.1.3.1.7.
6. The Administrative tool does not receive a response from the GP Server within a specified time-out interval.

3.6 Example 6: Querying Active Directory for Scope of Management and Version Information

In this example, a GP Client queries a GP Server for SOM and version information. SOM containers such as domain, site, and OU containers hold user and computer account information and are associated with GPOs. Each GPO is associated with a specific policy target, such as a user or

computer account. Messages exchanged between the GP Client and the GP Server use LDAP as a transport.

This example loosely maps to the use case specified in Applying Group Policy — GP Client (section 2.5.2).

Prerequisites

The following prerequisites apply to this example:

- The GP Client has discovered the GP Server and has connected with Active Directory, as described in [MS-GPOL] section 3.2.5.1.1.
- The GP Server must be storing policy and must respond to LDAP requests from the GP Client.
- The GP Client must maintain a consistent configuration of policy information retrieved from the GP Server, which includes registry settings, WMI data, and RSoP data.
- The GP Administrator must ensure that the GP Client policy configuration aligns with business requirements.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of this example can be described as follows:

- The GP Client successfully retrieved the SOM and version information from the GP Server.

Sequence of Events

The following figure illustrates the message sequence that occurs when the GP Client queries Active Directory for SOM and Version information:

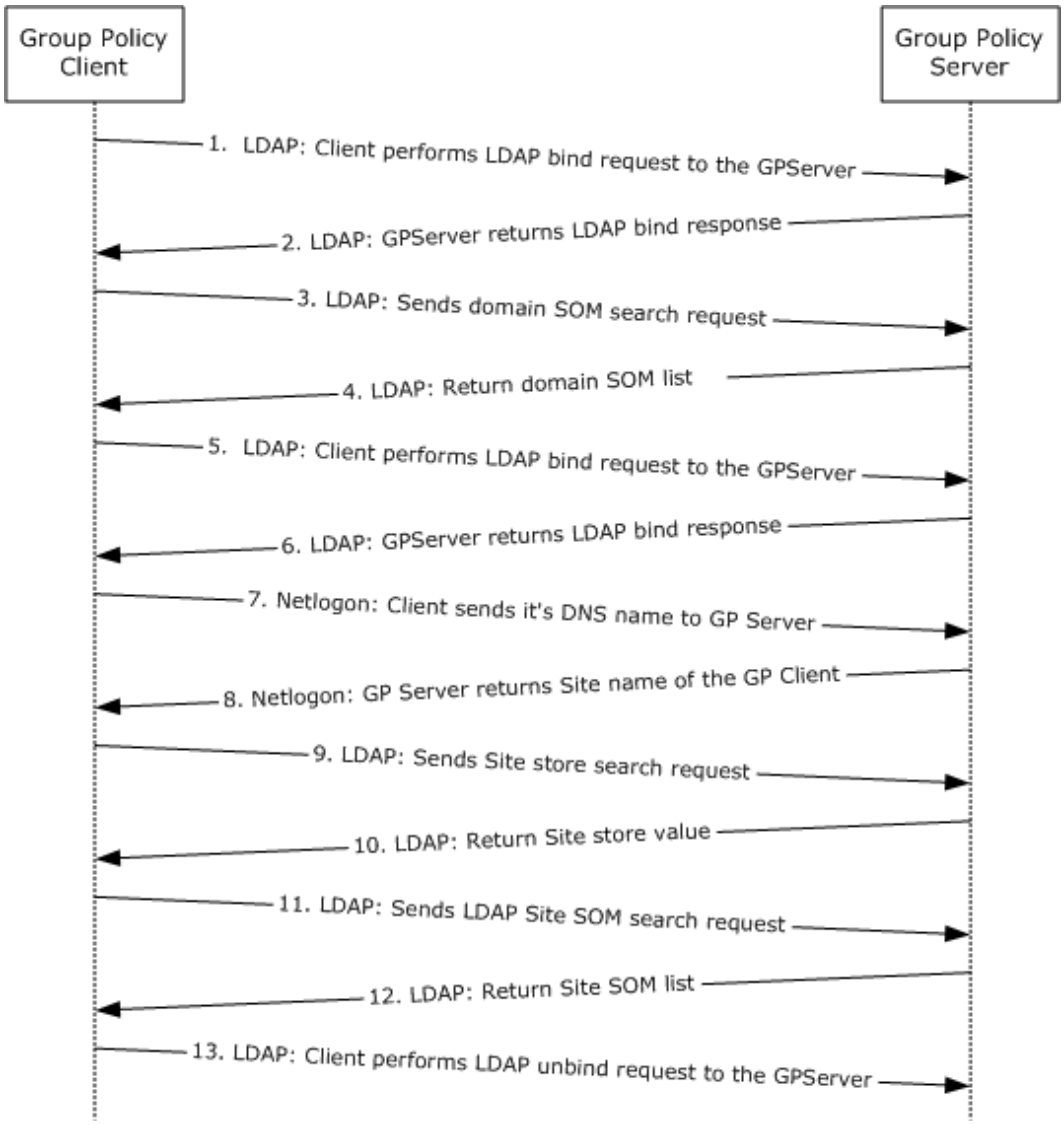

Figure 17: Querying Active Directory for SOM and version information

The message sequence for this example is described as follows:

1. The GP Client sends an LDAP **BindRequest**, as described in [RFC2251] section 4.2, to the GP Server.

2. The GP Server sends an LDAP **BindResponse**, as described in [RFC2251] section 4.2.3, to the GP Client.

3. The GP Client sends an LDAP domain SOM **SearchRequest** to the GP Server, to query for the **gpLink** and **gpOption** attributes for its DN for the **domain naming context (domain NC)**, as described in [MS-GPOL] section 3.2.5.1.3.

4. The GP Server returns the domain SOM list via an LDAP **SearchResponse**, as described in [MS-GPOL] section 3.2.5.1.3.

The GP Client processes the **gpLink** and **gpOption** attributes information for the domain SOM and uses it to search for the list of GPOs for the domain SOM, as described in [MS-GPOL] section 3.2.5.1.5.

5. The GP Client sends an LDAP **BindRequest** to the GP Server.
6. The GP Server sends an LDAP **BindResponse** to the GP Client.
7. The GP Client sends its DNS name to the GP Server via Netlogon.
8. The GP Server returns the site name of the GP Client via Netlogon.
9. The GP Client sends an LDAP **SearchRequest** to the GP Server, querying for the **configurationNamingContext** attribute for the root of the domain, as described in [MS-GPOL] section 3.2.5.1.4.
10. The GP Server returns the site store value via an LDAP **SearchResponse** message.

    The GP Client processes the **configurationNamingContext** attribute information for the root domain and uses it to compute the DN of the site, as described in [MS-GPOL] section 3.2.5.1.4.
11. The GP Client sends an LDAP **SearchRequest** message to the GP Server, to query for the **gpLink** and **gpOption** attributes to obtain the DN for the config NC, as described in [MS-GPOL] section 3.2.5.1.4.
12. The GP Server returns the site SOM list via an LDAP **SearchResponse** message.

    The GP Client processes the **gpLink** and **gpOption** attributes information for the site SOM and uses this information to search for the list of GPOs for the domain SOM, as described in [MS-GPOL] section 3.2.5.1.5.
13. The GP Client sends an LDAP **UnBindRequest**, as described in [RFC2251] section 4.3, to the GP Server.

3.7 Example 7: GP Client Cannot Connect to the GP Server When Applying Policy

The examples in this section describe the message sequences during policy application that end in failure as a result of a lost connection with the GP Server. The following two scenarios are illustrated:

- Failure to contact Active Directory
- Failure to contact the GP FS

This example maps to the use case specified in Applying Group Policy — GP Client (section 2.5.2).

Prerequisites

The following prerequisites apply to the examples in this section:

- The GP Server must be storing policy and must respond to requests from the GP Client.
- The GP Client must maintain a consistent configuration of policy information retrieved from the GP Server, which includes registry settings, WMI data, and RSoP data.
- The GP Administrator must ensure that the GP Client policy configuration aligns with business requirements.

- The GP Client has discovered the GP Server and established a connection with Active Directory, as described in [MS-GPOL] section 3.2.5.1.1.
- The GP Client has sent an LDAP **BindRequest** message, as described in [RFC2251] section 4.2, to the GP Server and the GP Server has replied with an LDAP **BindResponse** message, as described in [RFC2251] section 4.2.3.
- For the failure to contact GP FS scenario, it is assumed that the GP FS resides on the GP Server.

Initial System State

The initial state of the GP System corresponds to the previously specified prerequisites.

Final System State

The state of the GP System and its components following execution of each example in this section can be described as follows:

- The state of the GP System and its components is unchanged.

Sequence of Events for Active Directory Contact Failure

The following figure illustrates the message sequence that occurs when the GP Client fails to contact Active Directory:

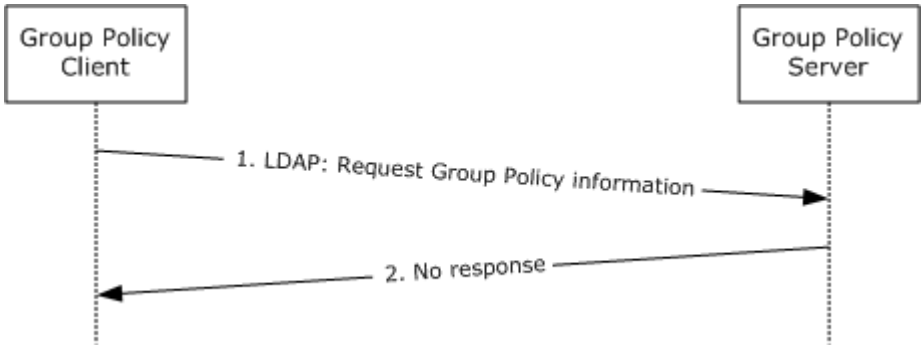

Figure 18: GP Client applying policy cannot contact Active Directory

The message sequence for this example is described as follows:

1. The GP Client sends an **LDAP** search query, as described in [RFC2251] section 4.5.1, to the GP Server to request Group Policy information.
2. The GP Client does not receive a response from the GP Server within the time-out interval.

Sequence of Events for GP File Share Contact Failure

The following figure illustrates the message sequence that occurs when the GP Client fails to contact the GP FS:

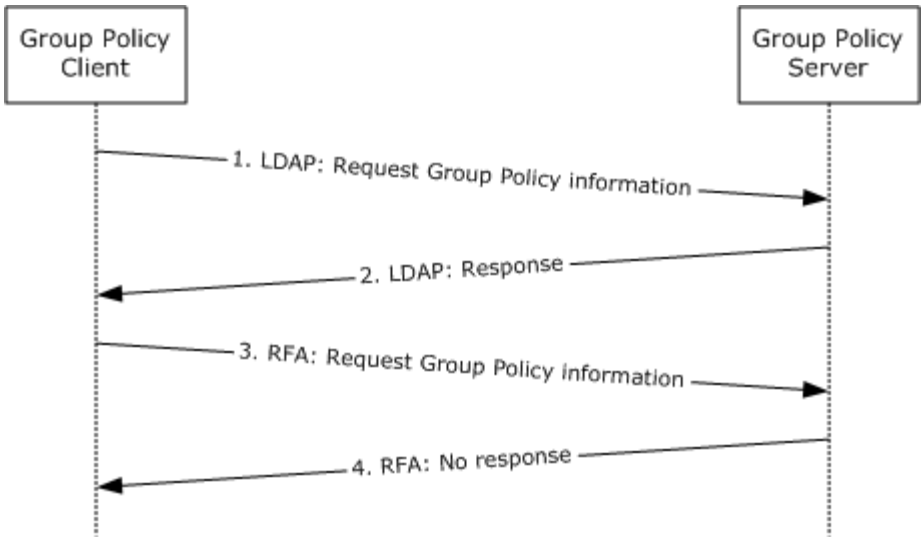

Figure 19: GP Client applying policy cannot contact the GP FS

The message sequence for this example is described as follows:

1. The GP Client sends an LDAP search query, as described in [RFC2251] section 4.5.1, to the GP Server to request Group Policy information.
2. The GP Client receives an LDAP response from the GP Server.
3. The GP Client sends a *File Open* request via a remote file access protocol to the GP Server.
4. The GP Client does not receive a response from the GP Server within a specified time-out interval.

4 Microsoft Implementations

The information in this specification is applicable to the following versions of Windows:

- Windows 2000 operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Server 2003 R2 operating system
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system

Exceptions, if any, are noted in the following section.

4.1 Product Behavior

<1> Section 1.1.3: The Microsoft implementation of the directory service for the GP System is Active Directory Domain Services (AD DS)[MS-ADTS].

<2> Section 1.1.3: The Microsoft implementation of the GP FS repository is a system volume (SYSVOL) share on the GP Server.

<3> Section 1.1.5: In Windows implementations, the GP System uses the SMB or **SMB2** file access protocol [MS-FASOD] for remote file access operations.

<4> Section 2.1.2: The Microsoft implementation of the directory service for the GP System is Active Directory Domain Services (AD DS) [MS-ADTS].

<5> Section 2.1.2.1: For Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2,Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the GP System queries the operating system directly to determine link speed. Windows 2000, Windows XP, and Windows Server 2003 use ICMP to determine the link speed between the GP Client and the domain controller. The following algorithm is used to determine the link speed when ICMP is used.

1. Form an ICMP Echo request with a packet size between 500 and 2,048 bytes.
2. Send the request to the domain controller three times, and compute the round-trip time for each of the echo responses.
3. Divide the packet size by the average response time to estimate the link speed between the GP Client and the domain controller.

<6> Section 2.1.3.1.6: For Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, the Group Policy System queries the operating system directly to determine link speed. Windows 2000, Windows XP, and Windows Server 2003 use ICMP to determine the link speed between the GP Client and the domain controller. Use the following algorithm to determine the link speed when ICMP is used.

1. Form an ICMP Echo request with a packet size between 500 and 2,048 bytes.

2. Send the request to the domain controller three times and compute the round-trip time for each of the echo responses.

3. Divide the packet size by the average response time to estimate the link speed between the GP Client and the domain controller.

<7> Section 2.1.3.1.6: By default, Windows clients (versions Windows 2000, Windows XP, and Windows Server 2003) do not invoke the Software Installation [MS-GPSI] and Folder Redirection [MS-GPFR] extensions if the link speed is less than 500 kilobytes per second. An administrator can use Group Policy to modify the threshold speed and the set of extensions to be skipped.

<8> Section 2.2: The Group Policy: Central Access Policies Extension protocol is introduced in Windows 8 and onwards.

<9> Section 2.3.2: In the Microsoft implementation, the refresh period is every 90 minutes plus or minus a random offset value.

<10> Section 2.3.2: In the Microsoft implementation, the registry is used to store certain Group Policy information.

<11> Section 2.7.1.1: For Windows Vista, Windows Vista SP1, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2, only: When the network is unavailable, the GP Client also listens to network change notifications so that the policy can be refreshed as soon as the network is reachable. When a network change is detected and the GP Server is reachable, the policy application is applied only if the time elapsed is greater than the periodic refresh interval.

<12> Section 2.8.1: Periodic timer expiration for each user interactively logged on to the computer and for the computer itself is: every 90 minutes, by default, plus a random offset between 0 and 30 minutes, by default. Windows Group Policy clients maintain separate timers for the computer and each user interactively logged on to the computer. Time-outs can vary from as low as 1 minute to any number of days. The timer interval is a value determined by the client computer configuration and is typically configured by an administrator.

5 Change Tracking

This section identifies changes that were made to the [MS-GPOD] protocol document between the January 2013 and August 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. 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.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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

- New content added.
- Content updated.
- 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 are always classified with the change type **Editorially updated.**

Some important terms used in the change 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.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
4 Microsoft Implementations	Modified this section to include references to Windows 8.1 operating system and Windows Server 2012 R2 operating system.	Y	Content updated.
4.1 Product Behavior	Modified this section to include references to Windows 8.1 and Windows Server 2012 R2.	Y	Content updated.

6 Index