opentext™

Group Policy Administrator™
User Guide

October 2023

Legal Notice

Contents

7 Reporting on GPOs   117

8 Uninstalling Group Policy Administrator   137

9 Troubleshooting   139

Contents   **9**

About this Book and the Library

The *User Guide* provides conceptual information about NetIQ Group Policy Administrator (GPA). This book defines terminology, guides you through the process of installing GPA and performing many Group Policy tasks.

Intended Audience

This book provides information for individuals responsible for understanding Group Policy concepts, and for individuals designing and implementing a Group Policy solution for their enterprise environment.

This book assumes you are a Group Policy Object (GPO) administrator who is familiar with the following Microsoft Windows management concepts and terms:

- Active Directory (AD)
- Microsoft Management Console (MMC)
- Group Policy Management Console (GPMC)
- GP Edit
- IntelliMirror technologies
- Organizational Units (OUs)
- Group Policy Objects (GPOs)

1 Introduction

Active Directory Group Policy is a powerful technology that lets administrators make changes on thousands of desktops with a single keystroke.

Group Policy technology can help IT organizations centrally manage user, desktop, and server configurations. By deploying Group Policy in your enterprise, you can:

- Enforce secure password and account policies
- Ensure access to network resources
- Secure network and wireless communications
- Comply with government and industry regulations such as SOX, HIPAA, FISMA, VISA CISP and many others

IT organizations want to leverage this technology but know that making changes to live Group Policy Objects (GPOs) can be risky and have unintended and costly consequences.

Using Group Policy can help secure and unify enterprise operations. Group Policy can also help you meet compliance objectives, especially those that require you to document changes that affect network security or access to sensitive files, such as financial or personnel data.

Most IT organizations do not have the luxury of hiring additional staff to comply with these regulations. To safely leverage Group Policy, you need ways to:

- Model changes to GPOs safely without interrupting services
- Thoroughly test GPOs and secure approval from all stakeholders before deploying them
- Deploy tested GPOs into trusted or untrusted Active Directory domains
- Maintain consistent GPOs across business units, regions, or worldwide locations
- Roll back to a last-known good GPO to quickly recover from errors

The following sections provide more information:

- Section 1.1, "What Is Group Policy Administrator?," on page 13
- Section 1.2, "How GPA Works," on page 14

1.1 What Is Group Policy Administrator?

Group Policy Administrator (GPA) is an enterprise-class Group Policy change control solution that provides:

- A secure offline repository to reduce the risk of changing GPOs in live Active Directory
- A robust workflow and delegation model to safely involve all Group Policy stakeholders
- Built-in tools that help you analyze, compare, troubleshoot, and test GPOs
- Comprehensive reporting to help you document regulatory compliance

GPA lets you manage changes to GPOs in a safe offline environment, without risking potentially catastrophic changes that can impact network or service availability.

Using the GPA offline repository (GP Repository), change management workflow, and powerful productivity features gives you many advantages over using other tools:

- Buffers you from making errors in a live Active Directory environment
- Lets you compare and view differences between GPOs
- Performs health checks to ensure GPOs are not corrupted
- Lets you quickly roll back to a last-known good version of a GPO
- Stores backup copies of GPOs including WMI filters and links
- Lets you centrally manage GPOs in untrusted domains
- Lets you migrate GPOs from one domain to another
- Lets you delegate GPO changes to appropriate people while limiting Active Directory permissions

The GP Repository is a safe offline environment where you can test changes to Group Policy before rolling GPOs into the live environment.

GPA has many features designed specifically for large enterprises to help automate the following types of tasks:

- Enforcing consistent use of GPOs across your enterprise
- Managing and maintaining GPOs across trust barriers
- Automatically deploying GPOs during non-peak hours
- Diagnosing problems and differences between GPOs

If your enterprise needs a solution to help efficiently manage GPOs and streamline your auditing and reporting process, GPA can help you.

1.2   How GPA Works

GPA is an enterprise-wide, Group Policy administration solution that helps you take advantage of the powerful features Group Policy offers. GPA provides a mechanism for creating, changing, and testing GPOs away from your Active Directory environment, along with a complete change management workflow. To perform these functions, GPA uses several software components:

- GPA Console
- GP Repository
- GPA Server

1.2.1   Understanding GPA Components

GPA consists of software components you can install in a number of ways to integrate with your Active Directory environment and meet your Group Policy management objectives. Deciding which components to install and where to install them depends on the GPA features you want to

implement and the requirements of your network environment. For example, if you want to restrict the ability to change GPOs in your Active Directory environment to the Export Only service account, you need to install the GPA Server.

Additionally, the GP Repository uses Microsoft SQL Server. If your organization has specific requirements for Microsoft SQL Server, such as installing it only on designated computers that are managed by a separate group of database administrators, you may need to have this group install the GP Repository on one of these computers.

NOTE: This guide uses the terms **production environment** and **test environment** to describe how GPA works. A production environment is a live network environment. A test environment is a separate Active Directory network environment that you use exclusively for testing purposes. A test environment limits access to specific users. The term **environment** is an Active Directory network consisting of one or more domains or a collection of domains grouped in forests. This book uses **environment** to refer to multiple domains or forests and the term **domain** to refer to a single domain.

GPA Console

The GPA Console is an MMC snap-in that enables you to use and administer GPA. You can perform the following tasks using the GPA Console:

- Define a GPO workflow and security model
- Edit GPOs in the GP Repository or Active Directory
- Create comparison, diagnostic, and RSoP analysis reports
- Ensure GPO consistency between multiple domains
- Back up and restore GPOs
- Import and export GPOs between the GP Repository and Active Directory
- Search for specific GPO settings

GP Repository

The GP Repository provides a secure location away from your production Active Directory environment where you can create, change, analyze, and approve GPOs before you deploy them. The GP Repository also performs the following functions:

- Enforces version control
- Maintains a GPO revision history
- Enables you to roll back to a previous version of a GPO
- Stores information about the tasks each GPA user can perform using the GPA Console
- Maintains information about GPA service accounts and the Repository Authorization Code

Event logging records detailed information about the changes made to GPOs in the GP Repository using GPA. The event log includes what changes have been made, who made them, and when they were made. For more information, see Section 3.2.5, "Configuring GPA Event Logging," on page 46 or Section 3.2.6, "Viewing GPA Event Logs," on page 48.

GPA Server

The GPA Server performs the following functions:

- Enables the export of GPOs from the GP Repository to untrusted domains in your Active Directory environment using the Export Only account
- Allows you to restrict the rights to export GPOs to the Export Only service account
- Provides centralized event logging for GPO changes made with the GPA Console
- Sends email notifications of GPO changes made with the GPA Console
- Indexes GPOs in Active Directory and the GP Repository to provide accurate data for GPA Search reports

The event log includes what changes have been made, who made them, and when they were made. For more information, see Section 3.2.5, "Configuring GPA Event Logging," on page 46 or Section 3.2.6, "Viewing GPA Event Logs," on page 48.

1.2.2 Understanding Test and Production Environments

The GP Repository enables you to create, change, and evaluate GPOs without having to deploy them to your production Active Directory environment. This capability allows you to thoroughly test and evaluate GPOs before you implement them, which minimizes the risk of introducing harmful Group Policy errors into your production Active Directory environment.

However, you cannot be fully certain what a GPO will do until you actually deploy it to Active Directory. To insulate your production Active Directory environment as much as possible from any unintended consequences caused by an error or oversight in a GPO, you can first deploy a GPO into a test Active Directory environment.

A typical test Active Directory environment consists of a separate domain or forest with its own set of users and computers. To further separate the test Active Directory environment from production, the test Active Directory environment is typically untrusted by the production Active Directory environment.

Setting up a separate test Active Directory environment enables you to take full advantage of the managed GPA workflow while ensuring the GPOs you deploy to your production Active Directory environment work as intended. How you choose to implement GPA depends on your particular environment and GPO management requirements. For more information about test Active Directory environments, see Section 2.3.2, "Understanding Test Environment Configurations," on page 25.

1.2.3 Understanding the GPA Workflow

GPA provides a managed change control workflow to help you administer Group Policy in your Active Directory environment. The following figure shows a basic GPA workflow using separate production and test Active Directory environments. Having a separate test Active Directory environment is the

most effective way to ensure the security and reliability of your production Active Directory environment. For more information about test Active Directory environments, see Section 2.3.2, "Understanding Test Environment Configurations," on page 25.

The GPA Server is used in environments with untrusted domains or if you want to use features like centralized logging or GPO change notification. To emphasize the basic GPA workflow, which remains the same whether or not you install the GPA Server, the GPA Server is not included in this figure. For more information about the GPA Server, see Section 1.2.1, "Understanding GPA Components," on page 14.
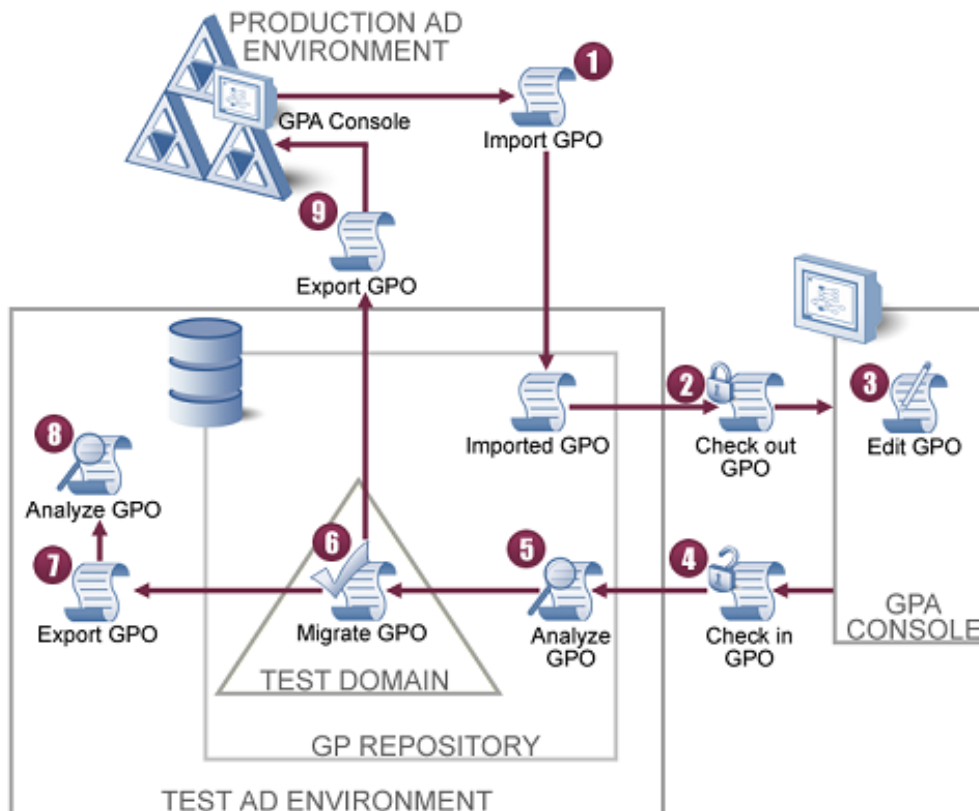
A high-level GPO change management workflow using GPA includes the following steps, which correspond to the numbers in the preceding illustration:

1   Import GPOs from your production Active Directory environment into the GP Repository.

2   Check out a GPO, locking it from changes by other users.

3   Edit the GPO as needed.

4   Check in the updated GPO, unlocking the GPO and updating the version number of the GPO.

5   Analyze the GPO to verify your changes (for example, RSoP analysis), and then approve the GPO.

6   Migrate the approved GPO to a test domain in the GP Repository.

7   Export the approved GPO into the Active Directory test domain.

8   Analyze the GPO to verify your changes (for example, RSoP analysis or diagnostic reports).

9   Export the GPO to Active Directory in the production environment.

1.2.4   Supported GPOs

GPA supports managing GPOs for all editions, service packs, and releases of the following Microsoft Windows operating systems, unless otherwise noted:

- Microsoft Windows Server 2022
- Microsoft Windows Server 2019
- Microsoft Windows Server 2016
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2008
- Microsoft Windows Server 2003
- Microsoft Windows 11
- Microsoft Windows 10
- Microsoft Windows 8.1
- Microsoft Windows 8
- Microsoft Windows 7
- Microsoft Windows Vista
- Microsoft Windows XP

NOTE: If you know that the GPMC supports managing the GPO, you can trust that GPA supports it as well.

2 Installing Group Policy Administrator

To install Group Policy Administrator (GPA), you need a working knowledge of Microsoft Windows Server and Active Directory. Some features of GPA rely on Microsoft SQL Server.

The checklist in this chapter provides a summary of the installation tasks along with references to more detailed information for each step in the process.

- Section 2.1, "Installation Checklist," on page 19
- Section 2.2, "Confirming GPA Installation Requirements," on page 20
- Section 2.3, "Understanding Common GPA Setup Scenarios," on page 25
- Section 2.4, "Creating GPA Service Accounts," on page 29
- Section 2.5, "Installing All Components on One Computer," on page 33
- Section 2.6, "Installing Components on Multiple Computers," on page 33
- Section 2.7, "Upgrading GPA Components," on page 36
- Section 2.8, "Installing or Upgrading a GPA License," on page 37

2.1 Installation Checklist

The following checklist provides the GPA installation tasks and directs you to the detailed steps to complete each task. Install GPA by completing the checklist.

	Steps and Related Information
☐	Review the information about how GPA works. For more information, see Section 1.2, "How GPA Works," on page 14.
☐	Ensure your user account has the necessary permissions to complete the installation and the computers on which you want to install the GPA components meet the minimum hardware and software requirements. For more information, see Section 2.2, "Confirming GPA Installation Requirements," on page 20.
☐	Determine the GPA components you want to install and whether you will manage group policy using an untrusted or a trusted test environment. For more information, see Section 2.3.1, "Determining Which GPA Components to Install," on page 25. For trial installations, install all GPA components on the same computer.
☐	Create the GPA service accounts. For more information, see Section 2.4, "Creating GPA Service Accounts," on page 29. **NOTE:** Although it is not required, you should use a service account to install GPA.
☐	Install GPA. For more information, see one of the following sections depending on how you plan to distribute the GPA components: - Section 2.5, "Installing All Components on One Computer," on page 33. - Section 2.6, "Installing Components on Multiple Computers," on page 33.

	Steps and Related Information
❑	Make any post-installation configuration changes to the GP Repository, GPA Server, or GPA Console. For more information, see: • Section 2.6.1, "Installing the GP Repository," on page 34. • Section 2.6.2, "Installing the GPA Server," on page 35. • Section 2.6.3, "Installing the GPA Console," on page 35.

2.2 Confirming GPA Installation Requirements

The account you use to install GPA must have the following Microsoft SQL Server and Microsoft Windows administrator permissions:

- Database administrator permissions on Microsoft SQL Server
- Domain administrator permissions in the domain where you create the GPA service accounts
- Local administrator permissions on the computers where you install the GPA Server and GPA Console

GPA supports 64-bit platform, including Itanium, ensuring you can run GPA in any Microsoft Windows environment.

Ensure the computers on which you install GPA meet the minimum hardware and software requirements for each GPA component you plan to install. The following sections list these requirements for each GPA component. You can also run the prerequisite checker included in your installation kit to verify that you meet the minimum requirements for each component you plan to install.

To check for GPA requirements on the target computer:

1 Log on to a computer where you want to install GPA components.

2 Extract NetIQGPAInstallationKit.

3 Run `SETUP.EXE` from the GPA installation kit.

4 Select the components you plan to install on that computer, and then click **Check**.

5 *If the prerequisite checker identifies missing software*, install and configure the software before you begin installing GPA. For example, you may need to install Microsoft SQL Server.

6 *If you are installing GPA components on different computers*, run the prerequisite checker on each computer to verify you meet the requirements for a given component.

2.2.1 GP Repository Requirements

For more information about the prerequisite checker, see Section 2.2, "Confirming GPA Installation Requirements," on page 20. The following table lists the hardware and software requirements for the GP Repository computer.

Element	Requirements
Disk space	1 GB (able to expand to 5 GB)
Database	One of the following versions of Microsoft SQL Server: ◆ Microsoft SQL Server 2016 Standard or Enterprise Edition (United States - English version, Case Insensitive) ◆ Microsoft SQL Server 2019 Standard or Enterprise Edition (United States - English version, Case Insensitive) ◆ Microsoft SQL Server 2022 Standard or Enterprise Edition (United States - English version, Case Insensitive)
Other software	Other prerequisites are: ◆ .NET Framework 4.8 ◆ Microsoft Windows Installer 3.1 or 4.0 ◆ Microsoft Visual C++ 2022 Redistributable Package (x86) ◆ Microsoft ODBC Driver 13 for SQL Server

Configuring Microsoft SQL Server

GPA requires the following Microsoft SQL Server configurations:

◆ Default collation order for US English, Case Insensitive SQL Server installation. For more information about supported collation orders, contact Technical Support.
◆ Mixed-mode authentication for managing GPOs in untrusted domains
◆ Windows authentication for trusted domains

GPA supports the following Microsoft SQL Server configurations:

◆ Default or named instances
◆ Microsoft SQL Server clusters

Specifying the Repository Authorization Code

The Repository Authorization Code is a unique identifier for every GP Repository that GPA Consoles must use to communicate with the GP Repository. You specify the Repository Authorization Code when you install the GP Repository. Although you have the option to accept a default value, defining your own Repository Authorization Code ensures a higher level of security for your GPA installation. Follow best practices for creating strong passwords, such as using a combination of uppercase and lowercase letters, numbers, and special characters.

NOTE: Record the Repository Authorization Code you define for later use. You must provide the Repository Authorization Code whenever you install a GPA Console to enable communication between the GP Repository and the GPA Console.

If you are setting up an environment with more than one GP Repository, you can use the same Repository Authorization Code for each one. Using the same code greatly simplifies your administration of GPA Consoles and ensures that each GPA Console can communicate with any GP Repository.

Setting the GPA Repository Management Group

GPA requires you to specify a user group during the GP Repository installation. You can specify an existing group or accept the default group `GPA_REPOSITORY_MANAGEMENT`, which GPA creates if you do not specify an existing group. Any users you add to the `GPA_REPOSITORY_MANAGEMENT` group or the group you specify have full permissions to perform all GPA-related tasks and manage all levels of GPA Security. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

2.2.2 GPA Server Requirements

For more information about the prerequisite checker, see Section 2.2, "Confirming GPA Installation Requirements," on page 20. The following table lists the hardware and software requirements for the GPA Server computer.

Element	Requirements
CPU	1 GHz (x86 processor) or 1.4 GHz (x64 processor)
RAM	4 GB
Disk space	10 GB
Operating system	One of the following operating systems: ◆ Microsoft Windows Server 2022 ◆ Microsoft Windows Server 2019 ◆ Microsoft Windows Server 2016
Group Policy Management Tools	Dependant upon operating system. Windows Server 2016, 2019, 2022– No action required.
Other software	Other prerequisites are: ◆ .NET Framework 4.8 ◆ Microsoft Windows Installer 3.1 or 4.0 ◆ Microsoft Core XML Services 6.0 Service Pack 1 ◆ Microsoft Visual C++ 2022 Redistributable Package (x86)

2.2.3 GPA Console Requirements

For more information about the prerequisite checker, see Section 2.2, "Confirming GPA Installation Requirements," on page 20. The following table lists the hardware and software requirements for the GPA Console computer.

Element	Requirements
CPU	Pentium 4, 1 GHz
RAM	2 GB
Drive space	1 GB
Operating system	One of the following operating systems: ◆ Microsoft Windows Server 2022 (64-bit) ◆ Microsoft Windows Server 2019 (64-bit) ◆ Microsoft Windows Server 2016 (64-bit) ◆ Microsoft Windows 11 (64-bit) ◆ Microsoft Windows 10 (64-bit)
Group Policy Management Tools	Depending on the version of Windows, install Group Policy Management Console (GPMC) or install Remote Server Administration Tools (RSAT), and then enable Group Policy Management Tools: ◆ Microsoft Windows 10 - Install RSAT for Microsoft Windows 10 and then enable Group Policy Management Tools. For more information, see "Enabling Group Policy Management Tools" on page 24. ◆ Microsoft Windows 11 - Install RSAT for Microsoft Windows 11 and then enable Group Policy Management Tools. For more information, see "Enabling Group Policy Management Tools" on page 24. You can download these tools from the Microsoft Download Center at www.microsoft.com/downloads/en/default.aspx. **NOTE:** Download the GPMC from the Microsoft website, install it, and then proceed to install the GPA console on Windows 10 and 11 machines.
Web browser	Microsoft Edge
Languages (Reports)	All reports are generated in American English. The following reports can also be generated in German: ◆ GPO Settings Report ◆ RSoP Analysis Report ◆ RSoP Analysis Comparison Report ◆ Group Policy Results Diagnostics Report ◆ Group Policy Comparison or Differential Report

Element	Requirements
Other software	Other prerequisites are: .NET Framework 4.8 - Microsoft Windows Installer 3.1 or 4.0 - Microsoft Core XML Services 6.0 Service Pack 1 - Microsoft Visual C++ 2022 Redistributable Package (x 86 and x64 for 64-bit computers) - PowerShell 2.0 or above

Enabling Group Policy Management Tools

If you are installing the GPA Console on a computer running Microsoft Windows Vista or later, install Remote Server Administration Tools (RSAT), and then enable Group Policy Management Tools.

To install RSAT and enable GPMC:

1 Download and install RSAT for the version of Microsoft Windows installed on the GPA Console computer.

2 Start the Programs application in Control Panel.

3 Under Programs and Features, select Turn Windows features on or off.

4 In the Windows Features window, expand Remote Server Administration Tools > Feature Administration Tools, select Group Policy Management Tools, and then click OK.

Creating the GPA Console Nodes Visibility Group

GPA can create a global group whose members have full access to manage node visibilty on the GPA Console. To create this account, GPA must use a domain administration account or an account with the following permissions:

- On the CN=Users AD container, add the Create Group Objects permission to this object and all descendant objects.

- On the CN=System AD container, add the following permissions to this object and all descendant objects:
    - Create Container Objects
    - Modify Permissions
    - Modify Owner
    - Read All Properties
    - Write All Properties

You can name this group yourself, select an existing group name, or accept the default group name of GPA-CONSOLE-NODES-VISIBILITY-MANAGEMENT. You must also provide a domain administration account user name and password. The owner of the domain administration account will have the power to add users or groups to the GPA-CONSOLE-NODES-VISIBILITY-MANAGEMENT group. For more information about managing node visibility, see Section 3.3.5, "Managing Node Visibility on the GPA Console," on page 51.

2.3 Understanding Common GPA Setup Scenarios

A minimum GPA installation requires a GP Repository, a GPA Server, and at least one GPA Console. You must also install a GPA Console in each untrusted domain if you want to manage GPOs in untrusted domains.

Your GPA installation depends on your environment and your GPO change management and security requirements. For more information about GPA components and how they work in relation to your Active Directory environment, see Section 1.2, "How GPA Works," on page 14.

2.3.1 Determining Which GPA Components to Install

You can install GPA components on one computer or on several computers. The primary factors affecting your decision are:

- ◆ Your network environment
- ◆ Whether you are managing GPOs in both trusted and untrusted domains

For more information about how GPA works with your network environment, see Section 1.2, "How GPA Works," on page 14.

The simplest approach is to install all GPA components on one computer, the typical method if you are installing a trial installation. For a production installation, however, installing all GPA components on one computer has limitations and may not adapt well to your actual environment. For production installations, consider installing the GPA components on separate computers.

For example, the GP Repository uses Microsoft SQL Server. Your company may install SQL Server on dedicated server-class computers and manage these computers with designated SQL Server database administrators. In this case, you may need a SQL Server database administrator to install the GP Repository on one of these computers. You would then use a GPA Console from another computer on the network to connect to the GP Repository.

You must install the GPA Server in a trusted domain of the domain where you install the GP Repository. If you maintain multiple GP Repositories, each GP Repository requires a separate installation of the GPA Server.

If you are managing GPOs in untrusted domains, you must install at least one GPA Console in each untrusted domain. Depending on the number and location of GPA users, you may need to install several GPA Consoles in multiple domains.

2.3.2 Understanding Test Environment Configurations

The GPA architecture provides you with tremendous flexibility in configuration options. The following sections describe common installation scenarios to help you determine the configuration that best suits your environment and GPO management and security requirements.

Configuring Untrusted Test Environments

An untrusted test environment configuration includes your production environment and an untrusted test environment. The test environment, untrusted by the production environment, allows you to work with GPOs before you deploy them to the production environment.

Verifying GPO changes in an untrusted test environment configuration minimizes the number of issues you may face in your production Active Directory environment. For example, users in the test environment cannot accidentally implement GPO changes made in the test environment to GPOs in the production environment using native GPO tools such as GPMC or GPEdit. These tools cannot work with GPOs in untrusted environments.

Although GPA can work with GPOs in untrusted Active Directory environments, setting up an untrusted test Active Directory environment also reduces the risk of making unauthorized changes to GPOs using GPA. A user account must have GPO Creator Owner permissions to modify GPOs. A user account in an untrusted test Active Directory environment is very unlikely to know user account credentials in the production Active Directory environment with these permissions. Test environments and users are purposely isolated from production environments to ensure that no changes made in the test environment can be made in the production environment. For more information about production and test environments, see Section 2.3.2, "Understanding Test Environment Configurations," on page 25.

A typical GPA untrusted test environment configuration includes the GP Repository, a GPA Server, and at least one GPA Console installed in the untrusted test environment. Install another GPA Console in the production environment.

Use the GPA Console in the production environment to import GPOs into the GP Repository. Use the GPA Console in the untrusted test environment to check in, check out, and edit GPOs in the test environment.

NOTE

- In an untrusted test environment configuration, you must use a GPA Console in the production environment to import GPOs into the GP Repository. The GPA Console cannot import required information about GPOs, such as any links to Active Directory objects or security filters, from a GPO in an untrusted environment.

- In an untrusted test environment configuration, you must use a GPA Console in the untrusted test environment where you installed the GP Repository to check out and edit GPOs. The GPA Console uses native Microsoft interfaces to edit GPOs, and these interfaces do not allow the editing of GPOs from untrusted domains.

Use the GPA Server to export GPOs from the GP Repository to the untrusted production environment. This method is the most reliable and secure way to perform a GPO export into an untrusted environment. You can export GPOs from the GP Repository using a GPA Console in either the production or test environment. For more information about using the GPA Server to export GPOs, see Section 2.3, "Understanding Common GPA Setup Scenarios," on page 25.

The following figure shows a basic GPA workflow in an untrusted test environment configuration, including the GPA components used in each step. Domain B represents an untrusted domain in the test environment. Domain A represents an untrusted domain in the production environment.
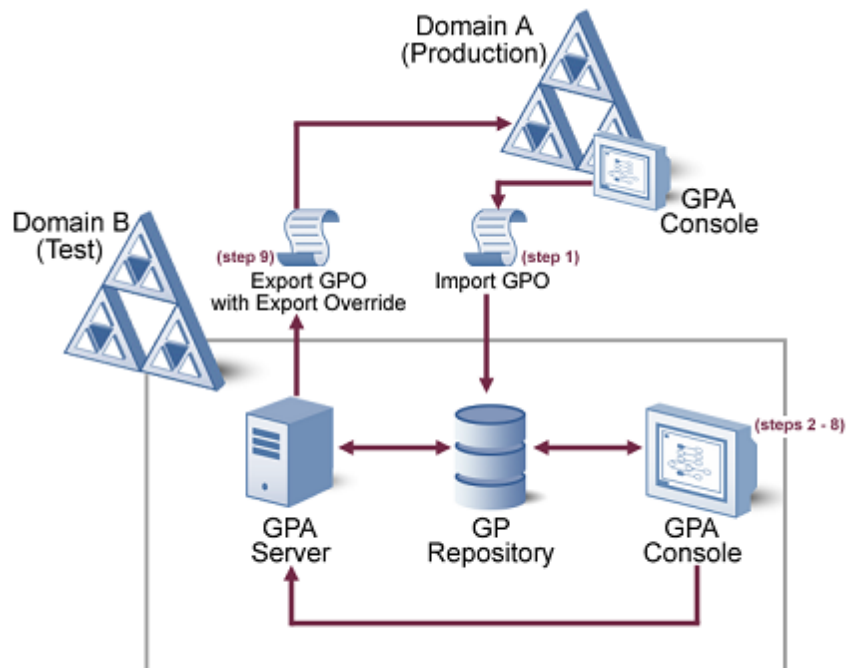
A high-level GPO change management workflow using GPA in an untrusted test configuration includes the following steps, which correspond to the numbers in the preceding illustration:

1 Import GPOs from your production Active Directory environment into the GP Repository using a GPA Console in production domain A.

2 Check out a GPO, locking it from changes by other users, using a GPA Console in test domain B.

3 Edit the GPO as needed.

4 Check in the updated GPO, unlocking the GPO and updating the version number of the GPO.

5 Analyze the GPO to verify your changes (for example, RSoP analysis), and then approve the GPO.

6 Migrate the approved GPO to test domain B in the GP Repository.

7 Export the approved GPO into test domain B in Active Directory.

8 Analyze the GPO to verify your changes (for example, RSoP analysis or diagnostic reports).

9 Export the GPO to the Active Directory production domain A using the GPA Server and the Export Only service account for domain A.

Configuring Trusted Test Environments

A simple GPA configuration includes your production environment and a test environment that is trusted by the production environment. This configuration includes a GP Repository and a GPA Console installed in the test environment. This configuration is simpler than an untrusted test environment for the following reasons:

- You do not need a GPA Console in the production environment to import GPOs into the GP Repository
- You can perform all GPA operations from a single GPA Console
- You do not need to use the GPA Server and an Export Only account to export GPOs from the GP Repository into the production environment.

This configuration requires fewer GPA components and allows you to perform all operations, including importing GPOs, from a single GPA Console. However, since the production and test domains are trusted, there is a much higher risk that GPO changes you make in your test environment could also be made in your production environment accidentally. Users in the test environment have credentials that are also trusted by the production environment, so it is much easier for a test environment user to make unintentional changes to GPOs in the production environment.

The following figure shows a basic GPA workflow in a trusted test environment configuration, including the GPA components used in each step. Domain B represents a trusted domain in the test environment. Domain A represents a trusted domain in the production environment.
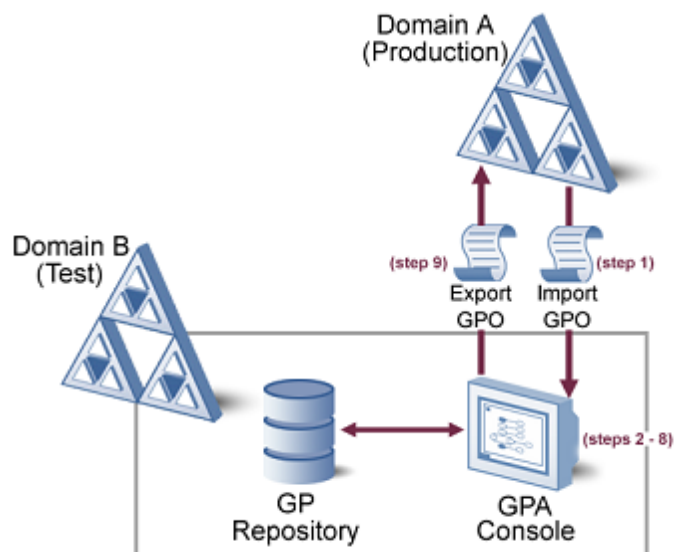
A high-level GPO change management workflow using GPA in a trusted test configuration includes the following steps, which correspond to the numbers in the preceding illustration:

1 Import GPOs from your production Active Directory environment into the GP Repository using the GPA Console in production domain A.

2 Check out a GPO, locking it from changes by other users, using a GPA Console in test domain B.

3 Edit the GPO as needed.

4 Check in the updated GPO, unlocking the GPO and updating the version number of the GPO.

5 Analyze the GPO to verify your changes (for example, RSoP analysis), and then approve the GPO.

6 Migrate the approved GPO to test domain B in the GP Repository.

7 Export the approved GPO into test domain B in Active Directory.

8 Analyze the GPO to verify your changes (for example, RSoP analysis or diagnostic reports).

9 Export the GPO to the Active Directory production domain A using the GPA Console in test domain B.

NOTE: You can install the GPA Server in the test environment and use an Export Only account to export GPOs to your production environment. This modified configuration allows only the Export Only account to make these GPO changes, and enforces a higher degree of security and control over changes to GPOs in the trusted test environment.

2.4 Creating GPA Service Accounts

GPA uses the following service accounts. Creating these accounts in advance allows you to complete the installation process without interruption.

NOTE: Although it is not required, you should use a service account to install GPA.

GPA Security

Used to access the GP Repository and to publish GPA Server information in AD.

Export Only account

Used for exporting GPOs.

Untrusted Access account

Used for generating reports and any repository update operation in an untrusted domain (migration, sync, retrieving data for mapping entries).

2.4.1 Creating the GPA Security Account

GPA requires the GPA Security account, a service account you specify during the GPA Server installation, to operate.

The GPA Server uses the GPA Security account to access the GP Repository. The GPA Security account also has special permissions on the GP Repository that the GPA Server needs to export GPOs. The GPA Security account also improves change control and auditing by being able to uniquely identify the tasks the GPA Security account performs.

GPA uses the GPA Security account to publish GPA Server information as a Service Connection Point (SCP) in AD during server start up. To do this, the GPA Security account requires specific permissions in each trusted domain where a GPA Console is installed.

Create the GPA Security account before installing GPA and save the user name and password to use when you install the GPA Server. The GPA Security account needs permission to export GPOs from the GP Repository and to write GPOs to Active Directory.

To create the GPA Security account:

1 Create a user account named GPA Security Account.

2 Add the GPA Security account to the Domain Admins group. *If you do not want to grant the GPA Security account domain administrator permissions*, add the account to the global Group Policy Creator Owners group.

3 Add the following permissions on the CN=System container for the GPA Security account in each trusted domain where a GPA Console is installed:

```
Read Permissions
Modify Permissions
Read all properties
Write all properties
Delete subtree
Create Container objects
Delete Container objects
Create serviceConnectionPoint objects
Delete serviceConnectionPoint objects
```

4 On the computer where you plan to install the GPA Server, add the account to the local Administrators group.

While installing GPA, the setup program configures the GPA Server permission to start and run the GPA Server service by setting the Startup type property to `automatic` and the Log on as this account property to the GPA Security account.

The setup program also creates a SQL Server login account for the GPA Security account and adds it to the following database roles for the GPA database:

- Public role
- netiq_gpr_server role

2.4.2 Creating the Export Only Account

GPA can use an optional service account, the Export Only account, to export GPOs from the GP Repository to Active Directory. Exporting GPOs requires elevated Active Directory permissions that you might not want a GPA user to have.

To overcome this limitation, you can configure GPA to use the GPA Server and an Export Only service account in the domain where you want to export GPOs. If you do not configure GPA to use an Export Only account, GPA uses the credentials of the user logged onto the GPA Console to export GPOs.

You can create the Export Only account before or after installing GPA. You can create an Export Only account for each domain to which you will export GPOs or create a single account across all trusted domains.

If you are managing multiple untrusted domains, create an Export Only account for each untrusted domain.

To create an Export Only account:

1 Log on to the domain into which you need to export GPOs with an account that has domain administrator permissions.

2 Create a user account with a name that describes its function, such as GPO Export Only.

3 Add the Export Only account to the Domain Admins group. ***Otherwise, if you do not want to grant the Export Only account domain administrator permissions***, complete the following steps:

NOTE

 • If you do not add the Export Only account to the Domain Admins group, every time you create a GPO in the GP Repository you must modify the GPO to grant the Export Only account all permissions except **Apply Group Policy** and **All Extended Rights**.

 • If you want to use one Export Only account across multiple trusted domains, the Export Only account must have domain administrator permissions in each domain.

3a Add the Export Only account to the global Group Policy Creator Owners group.

3b In the domain where you want to export GPOs, grant the Export Only account FullEdit permission to all GPOs in Active Directory. Grant these permissions before you import GPOs into the GP Repository. To grant these permissions, run the following PowerShell script.

Import-Module GroupPolicy

$params = @{

   All   = $true

   TargetName   = "AccountName"

   TargetType   = 'User'

   PermissionLevel = 'GpoEditDeleteModifySecurity'

   Replace   = $true

}

Set-GPPermission @params

Replace "AccountName" with Export Only Account name.

For more information about GPMC scripts, see the Microsoft documentation at the following Web site: Microsoft Documentation.

NOTE: If you do not set these permissions before importing GPOs, manually set the permissions for each GPO in the GP Repository. For more information, see Knowledge Base article NETIQKB28252 at www.netiq.com/NETIQKB28252.

3c In the domain where you want to export GPOs, grant the Export Only account permissions to link GPOs to SOM containers (OUs, sites, and the domain). You can use GPMC to delegate permissions to link Group Policy Objects.

- Navigate to the Delegation tab of the container.

- In the results pane, select "Link GPOs" in the Permission drop down-list box and click Add.

For more information, see Microsoft Documentation.

4 Configure the GPA Console to use the Export Only account for each domain where you want to export GPOs. If you have multiple untrusted domains, ensure you configure a GPA Console in each untrusted domain to use a different Export Only account. For more information, see Section 3.3.1, "Configuring GPA to Use the Export Only and Untrusted Access Accounts," on page 49.

2.4.3   Creating the Untrusted Access Account

GPA requires an Untrusted Access account for each untrusted domain with GPOs you want to manage. This service account is used to generate reports and perform any repository update operation in an untrusted domain, including, but not limited to, the following operations:

- Migration
- Synchronization
- Retrieving data for mapping entries

You can create the Untrusted Access account before or after installing GPA. Create an Untrusted Access account for each untrusted domain where you want to manage GPOs.

To create an Untrusted Access account:

1 Log on to the untrusted domain with an account that has domain administrator permissions.

2 Create a user account with a name that describes its function, such as, GPO Untrusted Access.

3 Add the Untrusted Access account to the Domain Users group.

4 Configure the account to have the following permissions to the following container, applied to this object and all its children: `%Domain%->FullArmor->FAZAM GP REPOSITORY SERVERS->%SQL SERVER%->SYSTEM->POLICIES`:

- Read
- Write
- List Contents
- Read All Properties
- Write All Properties
- Delete
- Delete Sub-tree
- Modify Permissions
- Read Permissions
- Modify Owner
- All Validated Writes

- Create groupPolicyContainer
- Delete groupPolicyContainer

5 Configure the GPA Console to use an Untrusted Access account for each untrusted domain. For more information, see Section 3.3.1, "Configuring GPA to Use the Export Only and Untrusted Access Accounts," on page 49.

2.5 Installing All Components on One Computer

If you do not need to install GPA components on multiple computers, follow these instructions to install all GPA components on a single computer. This configuration is common for trial installations, but may not work well for certain production installations. For more information about installing GPA components on one or multiple computers, see Section 2.3.1, "Determining Which GPA Components to Install," on page 25.

To install all components on one computer:

1 Log on to the computer where you want to install GPA with an account that has administrative rights in the domain as well as database administrator rights for Microsoft SQL Server. For more information about the requirements, see Section 2.2, "Confirming GPA Installation Requirements," on page 20.

2 Close all open applications.

3 Run the setup program from the GPA installation kit.

4 Click **Start Installation** on the left menu.

5 Follow the instructions in the setup program until you finish installing all GPA components.

6 To complete the installation, click **Finish**.

2.6 Installing Components on Multiple Computers

A complete installation of GPA requires one GP Repository, one GPA Server, and at least one GPA Console. Install an additional GPA Console in each untrusted domain where you want to manage GPOs. To complete the installation successfully on multiple computers, install these components in the following order:

1 GP Repository

2 GPA Server

3 GPA Console

If you want to install the GP Repository on a remote computer, follow the steps for installing the GPA Server, and select both the GP Repository and the GPA Server components from the setup program.

2.6.1 Installing the GP Repository

The GP Repository uses Microsoft SQL Server. If your organization has specific policies determining where you can install Microsoft SQL Server and who can administer it, these policies may affect where you install the GP Repository. For more information about installation requirements, see Section 2.3, "Understanding Common GPA Setup Scenarios," on page 25. For more information about where to install the GP Repository, see Section 1.2, "How GPA Works," on page 14.

During the setup process, the setup program prompts you for a Repository Authorization Code. The Repository Authorization Code is a unique identifier you create when you install the GP Repository. Each GPA Console must use the Repository Authorization Code to gain access to the GP Repository.

NOTE

- ◆ Record the Repository Authorization Code you define for later use. You must provide the Repository Authorization Code whenever you install a GPA Console to enable communication between the GP Repository and the GPA Console. For more information about the Repository Authorization Code, see "Specifying the Repository Authorization Code" on page 21.
- ◆ When you install the GP Repository, you can specify either the default instance or a named instance for the Microsoft SQL Server.

The following procedure guides you through the process of installing the GP Repository on the local computer. Optionally, you can also install the GP Repository on a remote computer.

To install the GP Repository:

1 Log on to the computer where you want to install the GP Repository with an account that has administrative rights in the domain as well as database administrator rights for Microsoft SQL Server. For more information about the requirements, see Section 2.2.1, "GP Repository Requirements," on page 21.

NOTE: If you want to install the GP Repository on a remote computer, follow the steps for installing the GPA Server, and select both the GP Repository and the GPA Server components from the setup program.

---

2 Close all open applications.

3 Run the setup program from the GPA installation kit.

4 Click Start Installation on the left menu.

5 Follow the instructions in the setup program until you finish installing the GP Repository.

6 To complete the installation, click **Finish**.

2.6.2   Installing the GPA Server

You must install the GPA Server in the same domain or a domain trusted by the domain where you installed the GP Repository and to export GPOs to AD. The GPA Server cannot operate across untrusted domains. For more information about where to install the GPA Server, see Section 1.2, "How GPA Works," on page 14.

Consider the following requirements:

- You must configure the GPA Server with a GPA Security account. The GPA Server uses this account to access the GP Repository. For more information about the GPA Security account, see Section 2.4.1, "Creating the GPA Security Account," on page 29.
- You can use only one GPA Server for each GP Repository. If you have more than one GP Repository, you must install a GPA Server for each GP Repository.

For more information about installation requirements, see Section 2.3, "Understanding Common GPA Setup Scenarios," on page 25.

To install the GPA Server:

1 Log on to the computer where you want to install the GPA Server with an account that has administrative rights in the domain as well as database administrator rights for Microsoft SQL Server. For more information about the requirements, see Section 2.2.2, "GPA Server Requirements," on page 22.

2 Close all open applications.

3 Run the setup program from the GPA installation kit.

4 Click **Start Installation** on the left menu.

5 Follow the instructions in the setup program until you finish installing the GPA Server.

6 To complete the installation, click **Finish**.

2.6.3   Installing the GPA Console

You can install the GPA Console on multiple computers to distribute Group Policy management tasks among several GPA users. If you want to manage GPOs in untrusted domains, install the GPA Console in each untrusted domain. For more information about where to install the GPA Console, see Section 1.2, "How GPA Works," on page 14.

Consider the following requirements:

- Install the GPA Console on a computer in either the same domain or a domain trusted by the domain where you installed the GP Repository.
- Configure the GPA Console with the Repository Authorization Code of the GP Repository you want the GPA Console to access.

For more information about installation requirements and installing on 64-bit platforms, see Section 2.3, "Understanding Common GPA Setup Scenarios," on page 25.

To install the GPA Console:

1 Log on to the computer where you want to install the GPA Console with an account that has local administrator rights on the computer where you want to install the GPA Console. You must also know the Repository Authorization Code for the GP Repository with which the GPA Console is going to communicate. For more information about the requirements, see Section 2.2.3, "GPA Console Requirements," on page 23.

2 Close all open applications.

3 Double-click the `setup.exe` file from the GPA installation kit.

4 Follow the instructions in the setup program until you finish installing the GPA Console.

5 To complete the installation, click **Finish**.

2.7 Upgrading GPA Components

You can upgrade from GPA version 6.8.

TIP: If you want to upgrade from GPA version 6.7, you must first upgrade to version 6.8.

Upgrade GPA components in the following order:

1. GP Repository
2. GPA Server
3. GPA Console

If you installed all GPA components on the same computer, the setup program upgrades the GPA components in the correct order.

NOTE

 Ensure you check in or undo the check out for all GPOs before you upgrade GPA.

 Ensure you back up the GPA database on the GP Repository before upgrading to the latest version. If you cancel the upgrade process, the setup program cannot roll back changes to the database and you will need to restore the database.

If you installed GPA components on separate computers, ensure you upgrade the GP Repository first. However, you can run the setup program on the GPA Server first and upgrade the GPA database while upgrading the GPA Server. If you are upgrading the GPA Server, the setup program automatically upgrades the GPA database on the GP Repository.

The process for upgrading to the latest version of GPA is the same as for installing GPA. For more information, see Section 2.6, "Installing Components on Multiple Computers," on page 33 or Section 2.5, "Installing All Components on One Computer," on page 33.

NOTE

- ◆ Ensure you upgrade all components to avoid compatibility issues between the latest version and previous versions of GPA. Do not attempt to export any GPOs until you have finished upgrading all GPA consoles.

- ◆ If you are upgrading the GPA database on a computer with other GPA components installed, the setup program does not remove the registry entries or binary files associated with previous versions of the GP Repository.

- ◆ You cannot create a new Security group or change the Security group you previously created. If you rename or delete the Security group you previously created, the upgrade process will create the Security group again and the setup program disables the **Browse** button on the User Group for GP Repository window.

2.8 Installing or Upgrading a GPA License

If you plan to use GPA past the 30-day trial period, install your license. To obtain a license, contact your GPA sales associate. Install your license on each GPA Console in your environment.

To install a license:

1 Log on to a GPA Console computer with an account that has local administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Explorer**.

4 In the left pane, expand **Forest**.

5 *If your domain is not listed in the left pane*, show domains by completing the following steps:

   5a Select **Forest**.

   5b From the Action menu, select **Show Domains**.

   5c Select the domains you want to show, and then click **OK**.

6 Select a domain in the left pane.

7 From the Action menu, select **Show License**.

8 Click **Install New License**.

9 Browse to the license file you want to install, and then click **Open**.

10 Review the license agreement. *If you accept the terms of the agreement*, click **Accept**.

11 Click **Close**.

3 Configuring Group Policy Administrator

After you install the product, you can customize configuration settings for the GP Repository, the GPA Server, and the GPA Console for your specific network and security needs.

3.1 Configuring the GP Repository

The installation process properly configures the GP Repository. You might need to change settings on the GP Repository after installation, such as the Repository Authorization Code or the GPA Security account. For example, you might need to change the Repository Authorization Code if it is no longer secure.

Additionally, you can customize the following GPO options:

- GPO Check In and Check Out
- GPO Backup
- Link Order
- GPO Naming
- GPO Paste
- GPO Migration
- GP Editor Link Security
- GP Extensions

3.1.1 Configuring GPO Options

To customize GPOs in the GP Repository:

1 Log on to a GPA Console computer with an account that has Customize Deployment Options permissions in GPA and database administrator permissions for Microsoft SQL Server.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** and select the GP Repository you want to configure.

4 Click **Action** > **Properties**.

5 Click **Customize Options**.

   5a *If you want to change GPO Check In and Check Out options,* see "GPO Check In and Check Out Options" on page 40.

   5b *If you want to change GPO backup options,* see "GPO Backup Options" on page 40.

5c *If you want to retain link order during operations,* select **Retain Existing AD Link Order upon Export and for RSoP Reports**.

5d *If you want to prevent duplicate GPO names,* select **Do not allow GPOs with same Name**.

5e *If you want to change GPO Paste options,* see Section 3.1.2, "GPO Paste Options," on page 41.

5f *If you want to change GPO migration and synchronization options,* see Section 3.1.3, "GPO Migration and Synchronization Options," on page 41.

5g *If you want to specify where GP Editors can link GPOs,* see Section 3.1.4, "Enabling GP Editor Link Security," on page 42.

5h *If you want to enable GPA to use third-party extensions on a central store,* see Section 3.1.5, "Enabling GP Extensions," on page 43.

GPO Check In and Check Out Options

You can configure how you check in and check out GPOs from the GP Repository. For example, you can configure the GP Repository to display a Comments dialog in the GPA Console when you check in or check out a GPO. Displaying the Comments dialog is useful to maintain detailed notes about changes to GPOs, such as the purpose of the change or who requested the change.

To configure the check in and check out options:

1 Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.

2 *If you want to enable check out options*, under **Check In/Out Options**, select the options you want to enable, and then click **OK**.

3 *If you want to disable check out options*, under **Check In/Out Options**, clear the options you want to disable, and then click **OK**.

4 Click **OK**.

GPO Backup Options

You can configure whether GPA creates backup copies of GPOs when you export GPOs from the GP Repository. Creating a backup copy of a GPO in Active Directory before you export the same GPO from the GP Repository enables you to recover the Active Directory version of the GPO. Recovering the Active Directory version of the GPO is useful if you decide not to implement the GP Repository version of the GPO in Active Directory. You can also configure the GP Repository to recover the Active Directory version of a GPO if an export from the GP Repository fails.

To configure GPO backup options:

1 Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.

2 *If you want to enable backup options*, under **Backup Options**, select the options you want to enable, and then click **OK**.

3 *If you want to disable backup options*, under **Backup Options**, clear the options you want to disable, and then click **OK**.

4 Click **OK**.

GPO Link Order Options

You can set an option to retain AD link order during exports and for RSoP reports. Select **Retain Existing AD Link Order upon export and for RSoP Reports** in the Link Order section.

GPO Naming

You can prevent users from creating different GPOs with duplicate names. Select **Do not allow GPOs with same Name** in the GPO Naming section.

3.1.2 GPO Paste Options

By default, copying and pasting a GPO in the GP Repository includes all GPO properties. You can change the GPO copy and paste options to exclude the following properties:

- GPO settings
- GPO name
- GPO security filters
- GPO WMI filters
- GPO links

To customize GPO copy and paste options:

1. Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.
2. Click the **Paste** tab.
3. *If you want to enable paste options*, select the options you want to enable.
4. *If you want to disable paste options*, clear the options you want to disable.
5. *If you want to enforce the default settings*, select **Enforce Default Paste Settings**.

   If you do not select **Enforce Default Paste Settings**, GPA displays the GPO Paste Options window to allow GPA administrators to customize the paste options when pasting GPOs.
6. Click **OK** until you close the window.

3.1.3 GPO Migration and Synchronization Options

By default, GPA does not maintain migration logs. If you enable migration logs, GPA saves every GPO migration in a log file on the computer where you initiated the migration. GPA saves the `MigrateReport.log` file in the following location:

`C:\Program Files (x86)\NetIQ\Group Policy Administrator\Log Files`

Migration logs record how GPA applied the migration map to a GPO during the migration. The migration mapping information is useful for diagnosing problems, such as a GPO that has incorrect settings after a migration.

To enable migration logs:

1. Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.
2. Click the **GPO Migration** tab.

3 Select **Enable Migration Log**.

4 Click **OK**, and then click **OK** again.

By default, users who migrate GPOs must have Migrate GPO permissions in both the source and target domains. If you want users to be able to migrate GPOs when they have Migrate GPO permissions only on the target domain, select **Enable Unidirectional Migration** on the **GPO Migration** tab.

To enable unidirectional GPO migration:

1 Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.

2 Click the **GPO Migration** tab.

3 Select **Enable Unidirectional Migration**.

4 Click **OK** until you close the window.

By default, a GPO migration or synchronization includes all GPO properties. You can change the migration and synchronization options to exclude the following GPO properties:

- ◆ Delegation rights
- ◆ Active Directory links
- ◆ WMI filters

Excluding GPO properties is useful for GPO migrations when certain properties do not apply to the domain where you are migrating the GPO.

You can also select whether to update GPO names during migrations.

To customize GPO migration and synchronization options:

1 Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.

2 *If you want to exclude certain GPO properties during migration and synchronization*, under **GPO Properties to Exclude During a Migration/Synchronization**, select the options you want to exclude, and then click **OK**.

3 *If you want to include certain GPO properties during migration and synchronization*, under **GPO Properties to Exclude During a Migration/Synchronization**, clear the options you want to include, and then click **OK**.

4 *If you changed the name of the GPO and want the name change reflected in the target domain*, select **Update GPO Name**. This option applies only to GPOs you migrate, not synchronize.

5 Click **OK**.

3.1.4 Enabling GP Editor Link Security

GPA allows the GP administrator to specify where GP Editors can link GPOs at a domain, OU, and site level, providing granular management of GP security. GPA disables this feature by default, which allows GP Editors to link GPOs to any domain, OU, or site. When you enable this feature, GP administrators can select which targets GP Editors can link GPOs to.

To enable GP Editor Link Security:

1 Expand GP Repository and select the repository to configure.

2 Click **Action** > **Properties**.

3 Click **Customize Options**, and then click the **GP Editor Link Security** tab.

4 Select **Enable GP Editor Link Security**, and then click **OK**.

For more information about configuring GP Editor Link Security, see Section 5.2.8, "Configuring GP Editor Link Security," on page 73.

3.1.5 Enabling GP Extensions

If you are using a third-party application to manage GP extensions, GPA allows you to store and edit these GPOs in the GP Repository when you select an option. This option is disabled by default. After you enable the option, when you edit a GPO in the GP Repository, GPA handles the PolicyPak extensions from a central store and a local store.

To enable editing GP extensions from third-party applications:

1 Follow the steps for Section 3.1.1, "Configuring GPO Options," on page 39.

2 Click the **GP Extensions** tab.

3 Select **Enable third-party GP extensions in the GP Repository**.

4 Click **OK** until you close the window.

3.2 Configuring the GPA Server

The installation process configures most settings on the GPA Server and prepares it for initial use. However, you might need to change these settings after installation.

3.2.1 Associating the GPA Server with a Different GP Repository

You associate the GPA Server with a GP Repository during installation. You may need to change the GP Repository association you created during the GPA Server installation. For example, if you have more than one GP Repository, you may want to associate the GPA Server with a different GP Repository.

To associate the GPA Server with a different GP Repository:

1 Log on to the computer where you installed the GPA Server with an account that has domain administrator permissions.

2 Start the **GPA Server Configuration** utility in the Group Policy Administrator program folder.

3 Under **GPA Repository**, select the Microsoft SQL Server where the new GP Repository is installed in the **SQL Database** list.

4 Ensure the GP Repository name in the **Database name** field is correct, and then click **OK**.

3.2.2 Changing or Updating the GPA Security Account

You configure the GPA Server to use a GPA Security account during installation. You may need to change the GPA Security account or update the account password.

To change the GPA Security account, configure the new account information on the GP Repository and the GPA Server. Configure the GPA Security account on the GP Repository before you configure the GPA Server. For more information about configuring the GPA Security account on the GP Repository, see Section 3.2.3, "Changing the GPA Security Account and Repository Authorization Code," on page 44.

To change the GPA Security account or update the account password:

1 Log on to the computer where you installed the GPA Server with an account that has domain administrator permissions.

2 Start the GPA Server Configuration utility in the Group Policy Administrator program folder.

3 Under GPA Security Account, click Change.

4 *If you want to change the GPA Security account*, perform the following steps:

    4a Type or browse to the new user account in the User field.

    4b Type the new password for the user account in the Password field, and then click OK.

5 *If you want to change the password for the GPA Security account*, type the new password for the user account in the Password field, and then click OK.

IMPORTANT

You can execute the GPAServerConfig command to change the GPA Security account or update the account password.

3.2.3 Changing the GPA Security Account and Repository Authorization Code

You can change either the Repository Authorization Code or the GPA Security account, using the NqGPARepConfig command line tool.

If you change the Repository Authorization Code, you must also update every GPA Console configuration to incorporate the same change. For more information about changing the authorization code on a GPA Console, see Section 3.3, "Configuring the GPA Console," on page 48.

If you change the GPA Security account, you must also make the same change on the GPA Server. For more information about changing the GPA Security account on the GPA Server, see Section 3.2, "Configuring the GPA Server," on page 43.

Event logging records detailed information about the changes made to the Repository Authorization Code. The event log includes what changes have been made, who made them, and when they were made. For more information see, Section 3.2.5, "Configuring GPA Event Logging," on page 46 or Section 3.2.6, "Viewing GPA Event Logs," on page 48.

To change the Repository Authorization Code or GPA Security account using the `NqGPARepConfig` tool:

1 Log on to the computer where you installed the GP Repository with an account that has domain administrator permissions and database administrator permissions for Microsoft SQL Server.

2 Open a command prompt window.

3 Navigate to the `C:\Program Files (x86)\NetIQ\Group Policy Administrator\Tools` folder.

4 Use the `NqGPARepConfig` tool to change the Repository Authorization Code or GPA Security account. For more information about using the tool, see Section A.7.2, "Change Repository Authorization Code or GPA Security Account," on page 162, or the command-line Help, which is available by typing `NqGPARepConfig /?` at the command prompt.

3.2.4   Setting Up GPO Change Email Notifications

GPA can notify you using email when a GPA user makes changes to GPOs. You can use these email notifications to support your GPO workflow. GPA can notify single users as well as distribution groups.

To configure GPA to send email notifications:

1 Log on to the computer where you installed the GPA Server with an account that has domain administrator permissions.

2 Start the **GPA Server Configuration** utility in the Group Policy Administrator program folder.

3 Under **Notifications**, select **Enable Notifications**.

4 In the **SMTP Server** field, type the name of the SMTP server that will send email notifications.

5 In the **Mail From** field, type the sender's email address. You must use a valid SMTP email address format, but the address you specify does not need to be a working email account.

6 In the Notifications section, click **Test**.

7 In the **Target SMTP Address** field, type the email address where you want to send the test email, and then click **Test**.

8 Click **OK**.

9 Confirm receipt of the test email message.

10 In the left pane, expand **GP Repository**, and select the Microsoft SQL Server.

11 On the Action menu, click **Configure Notification**.

12 Click **Add**.

13 Click **Manage**.

14 In the **Recipient Name** field, type the name of the person who will receive email notifications.

15 In the **Recipient Email** field, type the email address for the person who will receive notifications.

16 Click **Add**.

17 Click **Close**.

18 In the **Recipients** list, select the name of the person who will receive notifications.

19 In the **Objects Tree**, select the domain, category, or GPO for which you want to send notifications.

20 In the **Operations** list, select the operations for which you want to send notifications, and then click **OK**.

21 GPA displays a summary of configured notifications. Ensure the correct name appears in the **Notification Recipient** column for each operation, and then click **Close**.

3.2.5 Configuring GPA Event Logging

You can configure GPA to record information about changes made to GPOs in the GP Repository, on the GPA server, or from the GPA Repository Configuration tool (`NqGPARepConfig.exe`). GPA records information for the following GPO events in event logs, which you can view later:

Event ID	Event Type	Task Category	Source	Action
22101	GPA Server Service	GPA Server	GPA Server	Start GPA Server Service Stop GPA Server Service
22201	GPA Indexing process	GPA Indexes	GPA Indexes	Start Indexing Stop Indexing
22301	GPO Operation	Repository GPO Approval	GP Repository	Approve GPO
22302	GPO Operation	Repository GPO Approval	GP Repository	Unapprove GPO
22303	GPO Operation	Repository GPO Approval	GP Repository	Send for Approval
22304	GPO Operation	Repository GPO Approval	GP Repository	Reject GPO
22401	GPO Operation	Repository GPO Change Management	GP Repository	Create GPO
22402	GPO Operation	Repository GPO Change Management	GP Repository	Check Out GPO
22403	GPO Operation	Repository GPO Change Management	GP Repository	Check In GPO
22404	GPO Operation	Repository GPO Change Management	GP Repository	Undo Check Out GPO

Event ID	Event Type	Task Category	Source	Action
22405	GPO Operation	Repository GPO Change Management	GP Repository	Rollback GPO
22406	GPO Operation	Repository GPO Change Management	GP Repository	Delete GPO
22407	GPO Operation	Repository GPO Change Management	GP Repository	Rename GPO
22408	GPO Operation	Repository GPO Change Management	GP Repository	Link Order of GPO Changes
22409	GPO Operation	Repository GPO Change Management	GP Repository	Block Inheritance Changes
22501	GPO Operation	Repository GPO Synchronization	GP Repository	Import GPO
22502	GPO Operation	Repository GPO Synchronization	GP Repository	Export GPO
22503	GPO Operation	Repository GPO Synchronization	GP Repository	Migrate GPO
22504	GPO Operation	Repository GPO Synchronization	GP Repository	Copy GPO link
22601	GPO Operation	Repository GPO Delegation	GP Repository	GPO Masked or Locked
22701	GPA Repository Configuration Tool	GPA Repository Configuration Tool	GPA Repository Configuration Tool	Add Security Account Change Repository Authorization Code Disconnect GPA Server and GP Repository Database Associations

GPA records specific GPO event details in a Windows application event log, which you can view from the Microsoft Windows Event Viewer application. Use the Event Viewer to see log details, such as the GPO name, version, category, and domain. In addition, the event log contains details, such as the type of change, time made, comments made, the location of the GPA Console and GP Repository where changes occurred, and the user account making the change.

To configure GPA event logging:

1 Log on to the computer where you installed the GPA Server with an account that has domain administrator permissions.

2 Start the **GPA Server Configuration** utility in the Group Policy Administrator program folder.

3 Under **Centralized Event Logging**, select **Enable Event Logging**, and then click **OK**.

For more information on event logging, see "Section 3.2.6, "Viewing GPA Event Logs," on page 48."

3.2.6 Viewing GPA Event Logs

The Microsoft Windows operating system records important system occurrences in event log files. Through GPA, you can configure and view logs for GPO-specific events occurring in the GP Repository, on the GPA Server, or using the GPA Repository Configuration tool (`NqGPARepConfig.exe`). For more information, see "Section 3.2.5, "Configuring GPA Event Logging," on page 46."

To view GPA event log information:

1 Start the Microsoft Windows Event Viewer application in the Administrative Tools program folder.

2 View the Application log in the Microsoft Windows Event Viewer.

3 Use filters in Event Viewer to more quickly find and view GPA event logs. For example, set a filter to view only GPA specific events occurring in the GP Repository.

4 Use the arrow keys in the Event Viewer to scroll through all the stored GPO event logs.

3.3 Configuring the GPA Console

After installing GPA, you may need to make specific changes to the configuration of the GPA Console, such as:

- ◆ Configuring the GPA Console to use Export Only and Untrusted Access accounts
- ◆ Changing the Repository Authorization Code the GPA Console uses
- ◆ Connecting to a GPA Server
- ◆ Configuring Domain Indexing
- ◆ Hiding or showing GPA Console nodes

3.3.1 Configuring GPA to Use the Export Only and Untrusted Access Accounts

After you create the Export Only and Untrusted Access accounts and have installed or upgraded GPA, configure GPA to use the accounts. Configure GPA to use these accounts for each untrusted domain. You do not need to configure Untrusted Access accounts for trusted domains. You can configure Export Only accounts for trusted domains, but this is not required.

For more information about creating an Export Only account, see Section 2.4.2, "Creating the Export Only Account," on page 30. For more information about creating the Untrusted Access account, see Section 2.4.3, "Creating the Untrusted Access Account," on page 32.

To configure GPA to use the Export Only and Untrusted Access accounts:

1. Log on to the GPA Console computer with an account that has GPA Security Manager permissions for the GP Repository domain that you want to use the Export Only or Untrusted Access account. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

2. Start the **GPA Console** in the Group Policy Administrator program folder.

3. In left pane, expand **GP Repository** and select the untrusted domain you want to configure to use the Export Only or Untrusted Access account.

4. On the Action menu, click **Properties**.

5. On the Accounts tab, select the account you want to use and then type the user name and password.

6. Click **OK**.

7. Repeat the previous steps for each domain that you want to use an Export Only or an Untrusted Access account.

IMPORTANT

You can execute the `GPAServerConfig` command to change the Export Only and Untrusted Access accounts or update the account password.

3.3.2 Changing the Repository Authorization Code

Every GPA Console requires a Repository Authorization Code to connect with a GP Repository. The Repository Authorization Code is a unique identifier for each GP Repository that is set on the GP Repository. You can configure each GPA Console with one Repository Authorization Code, and each GPA Console can display any GP Repository with the same code.

Changing the Repository Authorization Code on the GPA Console does not change the Repository Authorization Code for the GP Repository. For more information about changing the Repository Authorization Code on the GP Repository, see Section 3.2.3, "Changing the GPA Security Account and Repository Authorization Code," on page 44.

NOTE: Each GPA user who starts the GPA Console for the first time must provide the Repository Authorization Code for the GP Repository with which the GPA Console is communicating. Each GPA user needs to provide the Repository Authorization Code only once.

To change the Repository Authorization Code the GPA Console uses:

1 Log on to the GPA Console computer with an account that has GPA Security Manager permissions.

2 Start the **GPA Console** in the Group Policy Administrator program folder.

3 In the left pane, click **GP Repository**.

4 On the Action menu, click **Properties**.

5 Click the Repository Authorization Code tab.

6 Type the Repository Authorization Code you want to use in the **Authorization Code** field, and then click **OK**.

3.3.3 Connecting to a GPA Server

You must connect to a GPA Server for some activities, such as searching for GPOs.

To connect to a GPA Server:

1 In the left pane of the GPA Console, expand GP Explorer, and right-click the forest of the domain you want the console to use.

2 Click **Pick GPA Server**.

3 Follow the instructions on the window to connect to a GPA Server.

4 *If you want to change the GPA Server,* type a name in the **Server Name** field or click **Connect to DB**.

5 *If you want to verify that the console can reach the server,* click **Validate**.

6 Click **OK**.

3.3.4 Configuring Domain Indexing

You can configure default indexing settings for the domains you want GPA to include when you search for GPOs. These settings can be changed later when you are searching for GPOs or creating reports.

To configure domain indexing:

1 In the left pane of the GPA Console, expand **GP Analysis** and click **Search**.

2 Click **Action** > **Configure Domain Index**.

3 Select or clear the check boxes for each AD domain or GP Repository you want to include or exclude for indexing.

4 *If you want to change the preferred domain controller,* click on the domain controller in the **Preferred DC** column and type the name of the DC you want GPA to use to collect GPO data for the AD domain on that row.

5 *If you want to add unmanaged domains,* click **Get Unmanaged Domains** and select additional trusted AD domains to include in your searches.

6 Click **OK**.

3.3.5 Managing Node Visibility on the GPA Console

GPA lets you choose which users or groups can see which nodes on the GPA Console. To show or hide nodes you must be a member of the `GPA-CONSOLE-VISIBILITY-MANAGEMENT` group, which is created at the time of installation if the group does not already exist in the Active Directory. For more information about the `GPA-CONSOLE-VISIBILITY-MANAGEMENT` group, see "Creating the GPA Console Nodes Visibility Group" on page 24.

To manage node visibility:

1 In the left pane of the GPA Console, right-click Group Policy Administrator and select **Show/Hide nodes**.

2 If it is not already selected, select Configure GPA Console Nodes Visibility.

3 Use the Users/Groups list box to add or remove objects whose viewing abilities you want to configure.

4 Select an object from the Users/Groups list box and select or clear the appropriate check boxes in the Console Nodes list box. Selecting the check box will make the node visible to the targeted user or group; clearing the check box will hide the node from the targeted user or group.

5 Click **OK**.

4 Configuring Security and Permissions

Group Policy Administrator (GPA) provides a comprehensive security model to ensure the safety and reliability of your Active Directory environment when you are using GPA to manage Group Policy. This security model, implemented in the GPA Repository, enables you to enforce a secure workflow for creating, modifying, testing, approving, and deploying GPOs to your production Active Directory environment. GPA enforces security over GPO changes in a number of ways:

- GPA defines the specific tasks or roles a user can perform, as well as the domains, categories, and GPOs each GPA user can work with.
- Using the Export Only account limits the user accounts that need Active Directory permissions to modify GPOs.
- Using the Untrusted Access account limits the user accounts that need access to untrusted domains to run reports or perform GPA operations such as migrations, synchronizations, or retrieving mapping data.
- Each GPA Console uses a Repository Authorization Code to connect to the GP Repository.
- The GPA Security account limits the ability to change the Repository Authorization Code.
- Using GPA allows you to control where a GPO Editor can link a GPO to specific AD containers.

The following sections explain how GPA enforces security over GPO change management.

- Section 4.1, "Understanding the GPA User Security Model," on page 53
- Section 4.2, "Defining GP Repository Security Permissions and Scope," on page 55
- Section 4.3, "Setting Required Permissions for GP Repository Tasks," on page 56
- Section 4.4, "Increasing File Security of a GPO after Checking It Out," on page 59

4.1 Understanding the GPA User Security Model

GPA enables you to define who can use GPA, what tasks each user can perform, and what parts of your Active Directory environment users can modify, down to the GPO level. To accomplish this level of control, GPA uses a security model with two levels:

- Authentication
- Access control

4.1.1 Authenticating Users

Authentication establishes the first layer of security in the GPA user security model. Authentication confirms the identity of any user trying to connect to a GP Repository with a GPA Console. You must explicitly identify users to whom you want to provide access to the GP Repository. The GP Repository supports both Microsoft Windows and SQL Server authentication so users in both environments can connect and work with the GP Repository.

Microsoft Windows Authentication

Microsoft Windows authentication enables GPA users to take advantage of a single logon. With single logon, users who are already logged on to the domain need not supply their user name and password again when starting the GPA Console or connecting to the GP Repository from the GPA Console. In addition, you can use Active Directory user and group accounts instead of creating and managing additional SQL Server accounts.

SQL Server (Mixed Mode) Authentication

When you are managing GPOs in untrusted domains, you must connect to the GP Repository using SQL Server authentication. Since the GP Repository cannot validate the Microsoft Windows credentials of a GPA Console user in an untrusted domain, GPA must rely upon SQL Server credentials you have configured in the GP Repository. GPA uses these SQL Server credentials to establish a connection between a GPA Console in an untrusted domain and the GP Repository.

Your organization may configure its SQL Servers to use only Microsoft Windows authentication. You need to configure the SQL Server instance for the GP Repository to allow mixed mode authentication. Mixed mode authentication enables both Microsoft Windows and SQL Server authentication.

All GPA users, whether they connect to the GP Repository with Microsoft Windows or SQL Server authentication, must have Microsoft Windows credentials recognized as valid by the GP Repository. The GP Repository uses the Microsoft Windows credentials of the user in the untrusted domain to assign specific access control to the user in GPA. You use the Remote User Login wizard to add Microsoft Windows credentials for users from untrusted domains to the GP Repository. For more information about adding users from untrusted domains, see "Setting Up Untrusted Domains" on page 66.

4.1.2 Granting Access Control

The second layer in the GPA Security model is granting authenticated users specific permissions to perform the various tasks available in GPA. Authorizing users or groups to perform tasks in GPA is **access control**. Access control enables you to precisely define users who are able to perform particular tasks. Additionally, you can define access control for individual objects in the GP Repository, including domains, categories, and individual GPOs.

You define access control for users and groups by granting permissions for GPA tasks for every object in the GP Repository. By defining a specific set of permissions for GPA users, you can limit users to tasks appropriate for their job roles. For example, assign permissions to modify GPOs to users whose job is to define and maintain GPOs. Assign GPO approval permissions to users whose job is to approve changes to GPOs.

4.2 Defining GP Repository Security Permissions and Scope

Other than members of the GPA_REPOSITORY_MANAGEMENT group created during GPA installation, by default, GPA users do not have permissions to perform any GPA tasks. You need to specifically grant permissions to each GPA user. You must also specify the objects in the GP Repository to which the user permissions apply, whether domains, categories, or GPOs.

You define GPA user security permissions and scope in GPA by adding GPA users to Active Directory groups and assigning specific GPA roles to those groups or by defining individual permissions for a GPA user for each object in the GP Repository.

4.2.1 Understanding Tasks and the GP Repository Structure

The GP Repository has a hierarchical structure similar to the structure of Active Directory. Starting at the top, the GP Repository has the following levels:

- GP Repository
- Domain
- Category (similar to an OU in Active Directory)
- GPO

Users can perform numerous tasks in the GP Repository. Refer to "GP Repository Task Specifics" on page 275, for a complete list of tasks and permission levels in the GP Repository.

4.2.2 Understanding GPR Security Management

Under the server node in the GP Repository there is a GPR Security Management node. When you expand this node, there are three sub-nodes, ActiveViews, Roles, and Assignments.

You use these three sub-nodes to (1) create an ActiveView and define the permissions scope (where permissions get applied), (2) define access roles, using built-in or custom roles, and (3) select users and assign them to the roles on the ActiveView that you have defined.

4.2.3 Determining Security Inheritance Permissions

The GP Repository implements security inheritance. The following sections detail the rules that determine how security inheritance works.

Security Permission Levels

By default, any permission set at a higher level is propagated to lower levels. For example, assigning GPO approval permissions to a specific user for a category in the GP Repository grants approval permissions to that user for all GPOs in the category.

When you add a user or a group to a higher level in the GP Repository, the same user or group is automatically propagated to existing lower levels and to any newly created lower levels. For example, if you add a user account with certain security permissions to a GP Repository domain, the user account has the same security permissions for any categories and GPOs in the domain.

Local Permission Overrides

Permissions set on a lower level override permissions inherited from a higher level. If you explicitly set permissions on an object in the GP Repository, these security permissions override any inherited permissions.

4.2.4 Configuring Security Attributes at Multiple Levels

Certain GP Repository tasks require configuration of security permissions at several levels in the GP Repository hierarchy. Some tasks in the GP Repository require a user to have a combination of permissions at several levels in the GP Repository hierarchy. For example, to import a GPO into a particular category in the GP Repository, a GPA user must have **Import GPO from AD** permissions at the domain level and **Create GPO** permissions at the category level.

4.3 Setting Required Permissions for GP Repository Tasks

You must set permissions for each user and for each task you want a user to perform. Refer to "GP Repository Requirements" on page 280 to learn the levels in the GP Repository hierarchy and where you need to set permissions for a user to perform a particular task.

The GPA Console has a specialized node called GPR Security Management that allows you to manage users and their permissions, mainly through the creation and maintenance of roles.

4.3.1 Understanding GPA Roles

You can assign several roles to GPA users. Each role corresponds to one of the job functions a user performs in GPA. Each role defines the security permissions required to perform the tasks in the GP Repository appropriate to the GPA-related job function.

GPA provides the following roles:

GPO Importer

> Has permission to import GPOs from Active Directory into the GP Repository and synchronize ADMX files from the central store.

GPO Exporter

> Has permission to export GPOs from the GP Repository to Active Directory and export ADMX files to the central store.

GPO Editor

> Has permission to send for approval and modify GPOs in the GP Repository and add ADMX files.

GPO Approver

> Has permission to approve, reject, or unapprove GPOs for export from the GP Repository to Active Directory. This role also has permissions to approve or unapprove ADMX files for export from the GP Repository to the central store.

GPO Synchronizer

> Has permission to modify GPOs in the GP Repository to synchronize controlled GPOs with master GPOs.

GPR Security Filtering

Has permissions to set users and groups to mask or lock the GPOs in the GP repository.

Assigning a role to a GPA user configures the security settings of the role for the GPA user. The advantage of assigning roles to GPA users is that you do not have to configure security settings individually for each task a particular GPA-related job function requires. For example, assigning a GPA user to the GPO Editor role configures all the permissions that the user needs to edit GPOs in the GP Repository.

Understanding Role Scope

GPA roles are specific to a GP Repository domain. For example, if you want a GPA user to be a GPO Editor for more than one GP Repository domain, you must assign the user to the GPO Editor role in each domain.

GPO roles are also specific to categories and GPOs in a domain. When you assign a role to a GPA user, you must also define the categories and GPOs to which the role applies.

Understanding Roles and Workflow

To assure a secure and controlled GPA workflow, assign different roles to different users. Assigning different roles to different users prevents any one user from having broad permissions and enforces a system of checks and balances. For example, if you assign the GPO Editor, GPO Approver, and GPO Exporter roles to separate people, you prevent one person from being able to both modify and implement GPOs in your Active Directory environment. You can also ensure that you properly test and verify any GPO changes before you implement them.

Before you can create and assign roles, you need to first create an ActiveView to define the scope where the permissions are applied, which can include categories, domains, and GPOs.

Creating an ActiveView

The ActiveViews node is located in the GPR Security Management container under the domain node of the GP Repository. From this node you create new active views for the GP Repository, so you can define permission roles and assign those roles to users or groups.

To create a new ActiveView:

1 Log on to a GPA Console computer as a member of the GPA_REPOSITORY_MANAGEMENT group.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then expand the domain in which you want to assign a user or group to a role.

4 Expand **GPR Security Management**.

5 Right-click **ActiveViews**, and select **New ActiveView**.

6 In the New ActiveView dialog box, type a name and description for the active view.

7 Do one or more of the following, as required:

- ◆ Click **Add Category**, expand the server tree, select a domain or category node, and click **OK**.
- ◆ Click **Add GPO**, expand the server tree, select a GPO, and click **OK**.

> **IMPORTANT:** If you want to exclude a particular scope, then select **Rule** as **Exclude**. If you want to include a particular scope, then select **Rule** as **Include** and choose the required **Inheritance** (**Objects in Nested Categories** or **Only GPOs in Category**).

- ◆ Add additional categories, domains, and GPOs.

8 Click **OK** in the New ActiveView dialog box.

Customizing Roles for Users or Groups

GPA comes with a set of built-in roles that should address common GPO tasks. If you need to extend these roles, or create new ones, you can use the GPR Security Management feature to customize security settings that more precisely define the permissions that apply to a user or group.

To create a new role:

1 Log on to a GPA Console computer as a member of the GPA_REPOSITORY_MANAGEMENT group.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then expand the domain in which you want to assign a user or group to a role.

4 Expand **GPR Security Management** and then expand **Roles**.

5 Right-click **Custom Roles** and select **New Role**.

6 Use the New Role window to name the role and assign the permissions that the role should grant.

Assigning Roles to Users

You assign roles to GPA users using the GP Repository's GPR Security Management node, which assigns a role to one or more users or groups and also defines the domains, categories, and GPOs in a GP Repository to which the role applies.

To assign roles to users or groups using the GPR Security Management node:

1 Log on to a GPA Console computer as a member of the GPA_REPOSITORY_MANAGEMENT group.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then expand the domain in which you want to assign a user or group to a role.

4 Expand **GPR Security Management**, and right-click **Assignments**.

5 Select **New Assignment**.

6 Use the New Assignment window to assign roles to users or groups on ActiveView.

Assigned to an ActiveView

To assigned to an ActiveView:

1 Log on to a GPA Console computer as a member of the GPA_REPOSITORY_MANAGEMENT group.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP_Repository**, and then log into database.

4 Expand **GPR Security Management**, expand activeview and select the required ActiveView want to view the associated assignment, a user or group to a role.

Also, when the Assignment container is expanded and required Assignment is selected, in the right pane of the assignment you can view the linked ActiveView.

View Current User's Assigned Roles

To view the roles assigned to the current user:

1 Start the **GPA Console** in the Group Policy Administrator program group.

2 In the left pane, expand **GP Repository**, and then right-click the repository node.

3 Select **Show Your Assigned Roles**.

4.4 Increasing File Security of a GPO after Checking It Out

You can further restrict what others can do with GPOs that you have checked out:

- You can prevent others—except for members of the GPA Repository Management group—from undoing your check out.
- You can prevent others—except for members of the GPA Repository Management group—from running a GPO Settings report while the GPO is checked out to you.

To add this extra security see "GPO Check In and Check Out Options" on page 40 and select **Enable file security after checkout**.

5 Working with GPOs in the GP Repository

You can use the GP Repository to plan and evaluate your GPOs before implementing them in your production environment. Using the GP Repository enables you to perform a number of tasks that assist you in managing Group Policy in your Active Directory environment:

- Implement a process for creating, testing, and deploying GPOs to minimize risks for your production environment
- Track the history of changes made to GPOs and restore prior versions
- Avoid changing GPOs directly in your production environment
- Take advantage of features such as version control and reporting
- Delegate GPO administration capabilities and control the tasks each Group Policy Administrator (GPA) user can perform

GPA also allows you to directly edit GPOs in your production environment. For more information about directly working with GPOs in Active Directory, see Chapter 6, "Working with GPOs in Active Directory," on page 103.

- Section 5.1, "Workflow for Managing GPOs with the GP Repository," on page 62
- Section 5.2, "Setting Up the GP Repository," on page 63
- Section 5.3, "Creating GPOs," on page 73
- Section 5.4, "Importing GPOs," on page 77
- Section 5.5, "Modifying GPOs," on page 80
- Section 5.6, "Merging GPOs," on page 84
- Section 5.7, "Managing GPO Versions," on page 86
- Section 5.8, "Exporting GPOs," on page 88
- Section 5.9, "Synchronizing GPOs," on page 92
- Section 5.10, "Migrating GPOs," on page 94
- Section 5.11, "Managing Administrative Template Files," on page 97
- Section 5.12, "Setting GPO Security Filters," on page 100

5.1 Workflow for Managing GPOs with the GP Repository

The following checklists summarize the activities you need to perform when setting up and maintaining the GP Repository and working with GPOs.

☑	**Setting Up the GP Repository**
❑	Connect to a GP Repository. For more information, see Section 5.2.1, "Connecting to a GP Repository," on page 64.
❑	Add domains to the GP Repository. For more information, see Section 5.2.2, "Adding Domains to the GP Repository," on page 65.
❑	Determine how you are going to organize GPOs in the GP Repository. For more information, see Section 5.2.3, "Understanding GPO Categories," on page 68.
❑	Create categories and subcategories in the GP Repository. For more information, see Section 5.2.5, "Creating Custom Categories and Subcategories," on page 70.
❑	Add GP Repository users and define user security. For more information, see Section 5.2.7, "Adding Users and Defining User Security," on page 71 and Section 5.2.8, "Configuring GP Editor Link Security," on page 73.

☑	**Creating and Modifying GPOs in the GP Repository**
❑	Create a new GPO in the GP Repository production domain or import a GPO from Active Directory. For more information, see the appropriate section: ♦ Section 5.3.1, "Creating a GPO Directly in the GP Repository," on page 73. ♦ Section 5.4.1, "Importing an Active Directory GPO," on page 77.
❑	Check out the GPO. For more information, see Section 5.5.1, "Checking Out a GPO," on page 80.
❑	Edit policy settings, add or edit preferences, or edit GPO properties. For more information, see the appropriate section: ♦ Section 5.5.2, "Editing Group Policy Settings, Preferences, and Properties," on page 81. ♦ "Setting Preferences" on page 82. ♦ "Editing GPO Properties" on page 82.
❑	Check the GPO into the GP Repository and add comments. For more information, see Section 5.5.4, "Checking in a GPO," on page 82.
❑	Approve the GPO for export to your Active Directory test environment. For more information, see Section 5.8.3, "Managing GPOs for Export," on page 89.
❑	Migrate the GPO to the GP Repository test domain. For more information, see Section 5.10.3, "Migrating a GPO Between GP Repository Domains," on page 95.

☑	**Testing and Evaluating GPOs in Your Test Environment**
☐	Export the approved GPO to your Active Directory test environment. For more information, see Section 5.8.5, "Exporting GPOs to AD Domains," on page 90.
☐	Test and fix any problems. For more information, see Chapter 7, "Reporting on GPOs," on page 117.

☑	**Deploying GPOs to Your Production Environment**
☐	Export the GPO to the production domain. For more information, see Section 5.8.5, "Exporting GPOs to AD Domains," on page 90.
☐	Test the GPO in the production domain. For more information, see Chapter 7, "Reporting on GPOs," on page 117.

☑	**Ongoing Management, Troubleshooting, and Disaster Recovery**
☐	Consider using the Offline Mirror wizard to create a copy of Active Directory GPOs for use in the GP Repository. For more information, see Section 5.4.2, "Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)," on page 78.
☐	If you need to synchronize GPO link order from the GP Repository to match the GPO link order in AD, select the Sync Link Order option in the Offline Mirror wizard and choose Match AD Link Order.
☐	Consider backing up the GPA database on a regular basis to minimize the risk of data loss. If the GPA database becomes corrupted, you can restore the GPA database from a backup. For more information about backing up or restoring a SQL Server database, see the Microsoft documentation.

5.2 Setting Up the GP Repository

You must perform several tasks to prepare the GP Repository for use. These tasks include:

- ◆ Connecting a GPA Console to a GP Repository
- ◆ Adding domains to the GP Repository
- ◆ Creating categories and sub-categories in the domains
- ◆ Adding GPA users and defining user security

The following sections provide instructions to complete each of these tasks.

5.2.1 Connecting to a GP Repository

Before you can begin working with GPOs in the GP Repository, you must establish a connection between the GPA Console you are using and the GP Repository. You can connect to the GP Repository using either Microsoft Windows or SQL Server credentials. Using Microsoft Windows credentials gives you the advantage of a single logon. You can connect to the GP Repository using the same credentials you used to log on to the GPA Console computer.

If your Microsoft SQL Server does not accept your Microsoft Windows credentials as valid or if you are connecting to a GP Repository from a GPA Console in an untrusted domain, you have the option to use SQL Server credentials to connect to the GP Repository. For more information about untrusted domains, see "Setting Up Untrusted Domains" on page 66.

Users who connect to the GP Repository using Microsoft Windows credentials only need to provide their Microsoft Windows credentials the first time they connect to the GP Repository. The GPA Console continues to use the same Microsoft Windows credentials each time a user connects to the GP Repository.

Users who connect to the GP Repository using SQL Server credentials must provide their SQL Server password once when they start the GPA Console and connect to the GP Repository. The GPA Console will not prompt for a user's SQL Server password again when connecting to the GP Repository unless the user stops and starts the GPA Console.

NOTE: If you are connecting to a GP Repository for the first time, you need to first configure the GPA Console with the Repository Authorization Code for the GP Repository. For more information about configuring the Repository Authorization Code on the GPA Console, see Section 3.3.2, "Changing the Repository Authorization Code," on page 49.

To connect to a GP Repository:

1 Log on to a GPA Console computer with an account that has domain administrator privileges.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**.

4 On the Action menu, click **New**, and then click **Connect to Database**.

5 *If you have installed the GPA Console on the same computer as the Microsoft SQL Server*, type `local` in the **SQL Server** list.

6 *If you have not installed the GPA Console on the same computer as the Microsoft SQL Server*, select the computer where you have installed the Microsoft SQL Server.

NOTE: If the **Server** list does not display the name of the Microsoft SQL Server, verify that the Microsoft SQL Server is accessible over the network from the GPA Console computer you are using. If the server is accessible, type the server name in the **Server** list. If DNS does not provide name resolution, type the IP address of the Microsoft SQL Server in the **Server** list.

---

7 Select whether you want to use Microsoft Windows or SQL Server authentication to access the GP Repository.

8 *If you specify SQL Server credentials*, provide valid credentials for the GP Repository.

9 Click **OK**.

To disconnect a particular GP Repository, select the GP Repository. On the Action menu, click **Disconnect**.

5.2.2 Adding Domains to the GP Repository

The first step after connecting to a GP Repository is to set up domains within the GP Repository that correspond to the domains in Active Directory where you are managing GPOs. These domains provide an Active Directory context for the GPOs you import into the GP Repository. The GP Repository domains enable GPA to maintain information about each GPO in Active Directory, including:

- Security settings
- Active Directory links to OUs, domains, and sites
- Network paths
- Active Directory domain where the GPO is located

GPA uses this information to export GPOs from the GP Repository to Active Directory. If you are working with multiple domains, GPA uses this information to migrate GPOs between domains in the GP Repository. For more information about the interaction between Active Directory and the GP Repository, see Section 1.2, "How GPA Works," on page 14.

Configuring a Domain Controller

From the GPA Console, you can select the Domain Controller (DC) the software uses for GP Explorer or GP Repository operations. When using domains with multiple domain controllers (DCs), you can specify for the console to use only the Primary Domain Controller (PDC), any available DC, or the currently specified DC to perform a GPO operation from the GP Repository.

To configure a domain controller:

1 Log on to the GPA Console computer with any domain user account.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then select the GP Repository.

4 Select the domain.

5 Click **Action** > **Properties**.

6 On the **General** tab of the domain properties window, select one of the following options:

Primary Domain Controller (PDC) Operations Master: GPA Console uses a specific PDC, and if that DC later becomes unavailable on the network, GPA notifies you that the domain controller is inaccessible and asks you to change the DC and refresh the GP Repository for any future operations.

Any Available Domain Controller: GPA Console uses any available DC, and if the DC in use becomes unavailable on the network, GPA finds another DC and continues to work without notifying you or sending any error messages.

This Domain Controller: GPA Console uses a specific domain controller, and if the DC becomes unavailable on the network, the GPA Console notifies you the specified DC is not accessible and asks you to confirm that the DC is online and try again or select another available DC.

> **NOTE:** It is recommended to use the Domain Controller located in the local site or the nearest site.

7 Click **OK**.

8 Refresh the database connection in the GP Repository to see the changes to the selected DC.

When the GPA Console does not detect any available DCs, the software lets you work in the domain with limited scope.

Indexing Domains

You can include trusted and untrusted domains when GPA indexes GPOs, and you can view indexing statistics for each domain. GPA updates the indexes for the GP Repository any time you make changes to Repository GPOs. When you include trusted and untrusted domains in the indexes, you also set the schedule for GPA to update domain indexes.

To include a domain for indexing and view indexing statistics:

1 In the left pane of the GPA Console, expand **GP Repository**, and then select the GP Repository.

2 Select the domain.

3 Click **Action** > **Properties**.

4 On the **Indexing Properties** tab, select the option to include the domain for indexing.

5 *If you are managing multiple domains,* repeat Step 2 through Step 4 for each domain to index.

6 Review the indexing statistics for that domain. If the status is in progress, click **Refresh** to see if the indexing completes while you have the window open.

7 *If you want to delete current indexes and completely rebuild them,* click **Rebuild Indexes**

Adding Trusted Domains

You can add trusted Active Directory domains to the GP Repository from a GPA Console computer in any domain trusted by the domain you want to add.

To add a trusted domain to the GP Repository:

1 Log on to a GPA Console computer with an account that has domain administrator permissions on the domain you want to add.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then select the GP Repository.

4 On the Action menu, select **New > Domain**.

5 Specify or browse to the domain you want to add, and then click **OK**.

Setting Up Untrusted Domains

Adding an untrusted domain to the GP Repository requires you to complete several related steps:

- ◆ Add the untrusted domain to the GP Repository
- ◆ Create a SQL Server login to the GP Repository that GPA Consoles in the untrusted domain use to communicate with the GP Repository

◆ Specify Microsoft Windows credentials from the untrusted domain to which you assign specific permissions in GPA.

NOTE

◆ If the untrusted domain is on a different DNS server from the domain on which the GP Repository resides, you need to add users from this untrusted domain using a GPA Console computer belonging to the untrusted domain.

◆ By setting up untrusted access credentials, users can perform operations such as generating reports on untrusted domains.

◆ If there are untrusted domains with different DNS servers, you need to configure DNS forwarding from each of these domains to other untrusted domains and these domains should be able to resolve their names and IP addresses.

For more information about permissions and security in GPA, see Section 4.1, "Understanding the GPA User Security Model," on page 53. For more information about the interaction between Active Directory and the GP Repository, see Section 1.2, "How GPA Works," on page 14.

Adding Untrusted Domains to the GP Repository

You can add untrusted domains to the GP Repository from any GPA Console computer.

To add an untrusted domain to the GP Repository:

1 Log on to a GPA Console computer with an account that has domain administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then select the GP Repository.

4 On the Action menu, select **New > Domain**.

5 Select **Untrusted Domain**.

6 Specify the DNS or NetBIOS name of the untrusted domain you want to add, and then click **OK**.

7 Provide the user name and credentials of an account that has domain administrator permissions in the untrusted domain, and then click **OK**.

Creating a SQL Server Logon Account to the GP Repository

To manage GPOs in the untrusted domain you added, you must configure the GP Repository with a SQL Server logon account that GPA Consoles in the untrusted domain use to connect to the GP Repository. GPA Consoles in the untrusted domain should connect using SQL Server authentication since the domain where you installed the GP Repository does not trust Microsoft Windows credentials from the domain where you installed the GPA Console.

You must also specify a Microsoft Windows user account in the untrusted domain to which you will assign specific GPA permissions. GPA users log on to GPA Console computers in the untrusted domain to perform GPA operations using this Microsoft Windows account. You can specify multiple Microsoft Windows accounts to support your GPA security model.

Use the Remote User Login wizard to create the SQL Server logon account as well as specify the Microsoft Windows credentials from the untrusted domain to which you will assign specific GPA permissions.

To set up a SQL Server logon account and remote user for an untrusted domain using the Remote User Login wizard:

1 Log on to a GPA Console computer in a domain trusted by the domain where you installed the GP Repository with an account that has domain administrator permissions and database administrator permissions and is a member of the securityadmin role for Microsoft SQL Server.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then select the GP Repository.

4 On the Action menu, select **Add Untrusted Domain User**, and then click **Next**.

5 *If there are no SQL Server logon accounts listed, or if you want to create a new SQL Server logon account*, click **Add** and follow Steps **a - c**. Otherwise, click **Next**.

   5a Type the name of the SQL Server logon account you want to create in the **User Login** field.

   5b Type the password for the new SQL Server logon account in the **Password** field.

   5c Confirm the password in the **Re-Enter Password** field, and then click **OK**.

To specify Windows Credentials from an Untrusted Domain

1 Add the untrusted domain and create the SQL Server logon for the domain

2 Type the user name of an account from the untrusted domain you are adding to the GP Repository in the **User Name** field. Specify an account to which you will assign specific GPA permissions. For more information about permissions and security in GPA, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

3 Specify the NetBIOS name or IP address of the domain controller for the untrusted domain in the **Domain Controller** field.

4 Under **Connect As**, type the user name of a domain administrator account in the untrusted domain in the **User** field.

5 Under **Connect As**, type the password for the domain administrator account in the **Password** field, and then click **Next**. The Remote User Login wizard uses these credentials to verify the account to which you will assign GPA permissions.

6 Click **Finish**.

You can use the Remote User Login wizard as many times as necessary to set up the GPA users you need in the untrusted domain. You must define security permissions for each remote user you add to the GP Repository. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

5.2.3 Understanding GPO Categories

Categories allow you to group and organize GPOs in the GP Repository. For example, you can group GPOs related to security in a Security Settings category, while you can group those related to desktop management in a Desktop Settings category. You can also create categories in the GP Repository that

correspond to OUs in Active Directory. Determine how you want to organize and manage GPOs in the GP Repository before you create your categories. You need to create at least one category before you can create or import GPOs into the GP Repository.

Categories in the GP Repository are not equivalent to OUs in Active Directory. You can create categories that correspond to OUs to help organize GPOs in the GP Repository the same way they are linked in Active Directory, but categories have no effect on GPOs. For example, placing a GPO in a category is not the same as linking a GPO to an OU.

Categories allow you to specify security permissions, such as creating, editing, deleting, importing, and editing, for GPOs within a particular category. For more information about permissions and security in GPA, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

Group Policy Administrator defines categories as one of these types:

- System Categories
- User-Defined Categories

5.2.4 Understanding System Categories

The GP Repository has several predefined system categories:

All

> This category contains a list of all GPOs in a GP Repository domain. The **All** category provides a quick way to find any GPO by GPO name. You can only view reports on GPOs in this category. You cannot modify or delete GPOs in this category.

Backup

> This category contains a backup of Active Directory GPOs. The GP Repository export feature allows users to export GPOs you create in the GP Repository to an Active Directory domain. During this export process, if a GPO with the same GUID already exists in the Active Directory domain, the GP Repository backs up the live domain GPO in the Backup category. This node does not appear until the GP Repository creates the first backup GPO. For more information, see Section 5.8.2, "Backing Up GPOs Prior to Export," on page 89.

GPOs Pending Approval

> This category contains a list of all GPOs in a GP Repository domain that are waiting for approval for export to Active Directory.

GPOs Pending Export

> This category contains a list of all GPOs in a GP Repository domain that you have approved for export to Active Directory.

5.2.5 Creating Custom Categories and Subcategories

You can add your own categories to the GP Repository in addition to those categories available in GPA. You can also create subcategories within a category or another subcategory.

IMPORTANT: When naming a category, avoid ending the name with a backslash (\). A category with a name that ends with a backslash will have the following effects:

- Operations that depend on indexing, such as merging or searching in GP analysis mode, will not function properly.
- The **Preview Export** command will not run for any GPOs in the category.
- Attempting to export a GPO from this category will fail with a "System.Runtime.InteropServices.COMException".
- The **Browse for Category** window only shows the categories listed above the misnamed category and does not show the misnamed category itself, nor the categories that are listed after it. For example, if there are five categories named A, B, C\, D, and E, only categories A and B will be listed.

To create a category or subcategory:

1 Log on to a GPA Console computer with an account that has permissions to create categories in the GP Repository.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**.

4 *If you want to create a new category*, select the domain where you want to create the category.

5 *If you want to create a new subcategory*, select the category under which you want to create the subcategory.

6 On the Action menu, click **New > Category**.

7 Specify the name for the category, and click **OK**. A category name should be unique within a domain.

NOTE: You cannot create categories called All or Backup. These are special categories reserved by GPA.

5.2.6 Determining Approval Status

You can determine the approval status of a GPO in several ways. When you select a category in the left pane of the GPA Console, the result pane provides a summary of every GPO in the category. The **Approval Status** column displays the approval status of each GPO.

A GPO can have the following approval states:

- Waiting for Approval
- Not Approved
- Approved
- Exported

You can also display the **GPOs Pending Approval** category and the **GPOs Pending Export** category in the GP Repository. The **GPOs Pending Approval** category displays all GPOs that are waiting for approval for export to Active Directory. The **GPOs Pending Export** category displays all GPOs in the GP Repository that have approval for export to Active Directory. Specify which of these category options display by selecting GP Repository and choosing **Properties** from the Action menu.

To configure GPA to display the GPOs Pending Export and GPOs Pending Approval categories:

1 Log on to a GPA Console computer with an account that has Customize Deployment Options permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, click **GP Repository**.

4 On the Action menu, click **Properties**.

5 Click the GPO Category Options tab.

6 Select the check box next to the category you want to display, and then click **OK**.

The left pane displays the GPOs Pending Export and GPOs Pending Approval categories.

5.2.7   Adding Users and Defining User Security

The final step in preparing the GP Repository for use is adding GPA users and defining the permissions each user has to perform GPO-related tasks in the GP Repository. To add new users to the GP Repository, you need to be a member of the `GPA_REPOSITORY_MANAGEMENT` group. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

NOTE

- If you add any users to the `GPA_REPOSITORY_MANAGEMENT` group, those users must log off and log on again for the new security permissions to take effect.

- You may have specified your own group account instead of the `GPA_REPOSITORY_MANAGEMENT` group during the GP Repository installation.

Adding Users

When adding new GPA users, you can specify whether the new user will use Microsoft Windows credentials or SQL Server credentials to connect to the GP Repository. Using Microsoft Windows credentials gives you the advantage of a single sign-on. You connect to the GP Repository using the same credentials you used to log on to the GPA Console computer.

If your Microsoft SQL Server does not accept your Microsoft Windows credentials as valid or if you are connecting to a GP Repository from a GPA Console in an untrusted domain, you have the option to use SQL Server credentials to connect to the GP Repository. If you add a new user with SQL Server credentials, this creates a new SQL Server logon account on the Microsoft SQL Server. For more information about untrusted domains, see "Setting Up Untrusted Domains" on page 66.

NOTE: A non-SQL administrator user must have the **securityadmin** role in SQL Server and the **db_owner** database role in the GPO_Repository database to be able to add users to the GP Repository and to remove users from the GP Repository.

To add a new user to the GP Repository:

1 Log on to a GPA Console computer as a member of the GPA_REPOSITORY_MANAGEMENT group and a member of the security admin role for Microsoft SQL Server.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** and select the GP Repository to which you want to add a user.

4 On the Action menu, click **Add Repository User**. GPA displays a window that contains the list of current GP Repository users.

NOTE: You can also use this window to remove GP Repository user or group accounts. Do not delete the default accounts, such as the Administrator account or the account you use to install the GP Repository.

5 *If you want to add a Microsoft Windows user account to the GP Repository server*, click **Microsoft Windows User** and then click **Add**.

    5a Select from a list of user accounts granted public access to the GPA Repository, or type the user account or group you want to add. You can add the user or a domain group from the current domain or from the list of trusted domains. You can add **Domain**, **Local**, **Global**, or **Universal** groups.

    5b Click **OK**.

NOTE: To perform this action, you must have permission to create new SQL Server logon accounts on the GPA Repository SQL Server.

6 *If you want to create a new SQL Server user account*, click **SQL User** and then click **Add**.

    6a Specify the user name and password for a new SQL Server user account.

    6b Click **OK**.

7 Click **Close**.

Defining User Security

By default, GPA users do not have permissions to perform any GPA tasks. You need to specifically grant permissions to each GPA user. Additionally, you need to define the scope of the permissions you grant. That is, you need to specify what levels in the GP Repository the user permissions apply to, whether domains, categories, or GPOs.

You define GPA user security permissions and scope in GPA by assigning GPA users to specific GPA roles or by defining individual permissions for a GPA user for each object in the GP Repository. For more information about assigning roles and defining individual security permissions for GPA users, see Section 4.2, "Defining GP Repository Security Permissions and Scope," on page 55.

You can also set security filtering to mask and lock GPOs from certain users and groups. When you set this level of security, the GPA Console no longer allows all users to see and edit all GPOs.

5.2.8 Configuring GP Editor Link Security

GPA allows the GP administrator to specify where GP Editors can link GPOs at a domain, OU, and site level, providing granular management of GP security. GPA disables this feature by default, which allows GP Editors to link GPOs to any domain, OU, or site. When you enable this feature, GP Admins can select which targets GP Editors can link GPOs to.

To enable GP Editor Link Security:

1 Expand GP Repository and select the repository to configure.

2 On the Action menu, click **Properties**.

3 Click **Customize Options**, and then click the **GP Editor Link Security** tab.

4 Select **Enable GP Editor Link Security**, and then click **OK**.

When you enable this feature, only GPA security administrators will be allowed to add or modify GPO links in the Repository by default. Use the GP Editor Link Security window to grant GPO link permissions to GP Editors on specific domains, OUs, and sites.

To grant GPO link permissions to GP Editors:

1 Expand GP Repository and select the domain to configure.

2 On the Action menu, select GP Editor Link Security.

3 Select the containers and users to grant appropriate permissions for your environment.

4 Have users test their ability to link GPOs to appropriate containers, and adjust permission settings when needed.

5.3 Creating GPOs

You can create GPOs in the GP Repository in the following ways:

- Create a new GPO directly in the GP Repository
- Import a GPO in an AD domain into the GP Repository
- Import all GPOs linked to any AD container in an AD domain into the GP Repository (also called creating an Offline Mirror)
- Import a GPO from backup
- Copy and paste an existing GP Repository GPO

5.3.1 Creating a GPO Directly in the GP Repository

You can create GPOs in any category or sub-category in the GP Repository.

NOTE: If you use the Export Only account to export GPOs and you did not add the Export Only account to the Domain Admins group, every time you create a GPO in the GP Repository you must modify the GPO to grant the Export Only account all permissions except **Apply Group Policy** and **All Extended Rights**.

To create a GPO:

1 Log on to a GPA Console computer with an account that has permissions to create GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** and select the category where you want to create the GPO.

4 On the Action menu, click **New > GPO**.

5 Specify the name of the new GPO, and then click **OK**.

The new GPO has the following attributes:

◆ It has a new Globally Unique ID (GUID)

◆ None of the policy settings are defined

◆ The GPO revision number is set to 0 for both Computer and User

◆ The GP Repository version number of the GPO is set to 1

◆ The GPO is not linked to any Active Directory domain, OU, or site

◆ The security filters on the GPO are the same default accounts and permissions set for GPOs you create in Active Directory

To view a report on the GPO settings of the newly created GPO, click the Settings tab in the right pane. To view a report on the status of security and integrity checks of the GPO, click the Health Check tab in the right pane.

5.3.2   Copying and Pasting a GPO from a Category or GPO Link

You can create a new GPO from an existing GP Repository GPO by using the copy and paste feature. The GP Repository offers several ways to copy and paste a GPO based on the location where you initiate the paste operation:

◆ Copy and paste a GPO from a category.

◆ Copy and paste a GPO link.

When you perform a copy and paste operation on a category, GPA pastes the settings in the latest version of the copied GPO into a newly-created GPO. The new GPO has the same name, GPO settings, security filters, and links to the same Active Directory objects as the original GPO. However, it has a different GUID and its GP Repository version number is always set to 1, irrespective of the version number of the copied GPO.

The same GPO can exist under more than one category in the GP Repository. The GPO appears in multiple categories because it is linked to those locations. Linking GPOs is useful when organizations have developed exceptionally large GPOs containing functionality that needs to be classified under more than one category.

Copying a GPO link is similar to copying a GPO. However, when you paste a GPO link, only one copy of the GPO exists in the GP Repository.

To copy and paste a GPO from a category or GPO link:

1 Log on to a GPA Console computer with an account that has permissions to create GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to copy.

4 On the Action menu, click **Copy**.

5 Select the category under which you would like to create a copy of this GPO.

6 On the Action menu, click **Paste as New GPO** or **All Tasks > Paste GPO Link**.

7 Click **OK**.

GPA creates the new GPO with the same name under the selected category. Even though the two GPOs have the same name, they have different GUIDs and version numbers. To change the name of the new GPO you need to first check it out of the GP Repository. For more information, see Section 5.5, "Modifying GPOs," on page 80.

5.3.3   Managing GPO Link Order

When you import GPOs from Active Directory into the GP Repository, you do not import the link order. If you import a linked GPO to a GP Repository that already contains a set of GPOs linked to the same site, domain, or OU, GPA lists the imported GPO at the bottom of the offline link order list. GPA does not change the existing link order of GPOs in the GP Repository. For more information about importing a GPO into the GP Repository, see Section 5.4.1, "Importing an Active Directory GPO," on page 77.

GPA lets you update GPO link order from several places:

- When you import all GPOs into the GP Repository
- When modifying links in GPOs already in the GP Repository
- When exporting GPOs to Active Directory
- When modifying links in Active Directory GPOs

When you import all GPOs from Active Directory into a fresh installation of the GP Repository using the Offline Mirror wizard, GPA imports GPOs linked to the domain, site, or OU in Active Directory. If you choose the Sync Link Order option, the wizard synchronizes GPO link order in the GP Repository. You can tell the wizard to use AD or the existing GP Repository link order as the basis for ordering links. For more information about importing all GPOs from Active Directory and synchronizing link order using the Offline Mirror wizard, see Section 5.4.2, "Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)," on page 78.

When you export GPOs to Active Directory, the resulting link order depends on whether you configure GPA to retain the existing link order in Active Directory or overwrite it with the link order you defined in the GP Repository.

If you configure Group Policy Administrator to retain the Active Directory link order by enabling the **Retain Existing AD Link Order upon Export and for RSoP Reports** option on the GP Repository Custom Options property window, and then export a GPO, Group Policy Administrator applies the link order specified in Active Directory to the GPO. If no link order is specified in Active Directory, Group Policy Administrator creates a link for the GPO and moves it to the bottom of the list (lowest precedence).

By default, Group Policy Administrator is not configured to retain the existing Active Directory link order. If you use the default setting, and then export a GPO, Group Policy Administrator overwrites the link order in Active Directory with the link order in the GP Repository according to the scenarios illustrated in the following table.

Export Scenario	Description
Same GPO to Active Directory	When you export a GPO from the GP Repository to Active Directory, if Active Directory already contains the same GPO, the GPO link order you have defined in the GP Repository overrides the GPO link order in Active Directory.
New GPO to Active Directory	When you export a new GPO from the GP Repository to Active Directory, Active Directory lists the exported GPO at the top of the link order.
	You may have a scenario where one set of GPOs in Active Directory are linked to a site, domain, or OU and a different set of GPOs in the GP Repository are linked to the same site, domain, or OU. When you export a GPO, Active Directory lists the exported GPO at the top of the link order even if there are GPOs in the GP Repository that are at a higher link order than the exported GPO.
	For example, suppose there are three GPOs in the GP Repository (A, B, and C) that are linked to an OU (Z) and three GPOs (D, E, and F) in Active Directory that are linked to the same OU (Z). If you export GPO C from the GP Repository to Active Directory, Active Directory lists GPO C at the top of the link order.
GPO with lower link order to Active Directory	You may have a scenario where one or more GPOs in both the GP Repository and Active Directory are linked to the same site, domain, or OU. If you export any GPO other than the first GPO in the link order list, Active Directory lists the exported GPO above all GPOs in Active Directory that do not exist in the GP Repository.
	For example, suppose there are three GPOs in the GP Repository (A, B, and C) that share the same link order as three GPOs (D, E, and F) in Active Directory. When you export GPO B from the GP Repository to Active Directory, Active Directory lists GPO B before GPO D. If you export GPO C to Active Directory, Active Directory lists GPO C after GPO B and before GPO D.
GPO with higher link order to Active Directory	You may have a scenario where one or more GPOs in both the GP Repository and Active Directory are linked to the same site, domain, or OU. If you export any GPO that has a higher link order than GPOs that exist in both the GP Repository and in Active Directory, Active Directory lists the exported GPO above these GPOs in the link order.
	For example, suppose there are three GPOs (A, B, and C) in the GP Repository that share the same link order as three GPOs (D, E, and F) in Active Directory. When you export GPO B from the GP Repository to Active Directory, Active Directory lists GPO B at the top of the Active Directory link order. If you then export GPO A from the GP Repository to Active Directory, Active Directory lists GPO A above GPO B in the link order.

For more information about exporting a GPO from the GP Repository, see .

5.3.4   Modifying GPO Link Order Using the GP Repository

GPA allows you to use the GP Repository to define the link order of GPOs for sites, domains, and OUs.

TIP: You do not need to check out the GPO before editing its link order.

Also, to ensure that the GP Repository link order is included with GPOs upon export and in RSoP reports, do not configure GPA to retain the existing Active Directory link order. For more information, see Section 5.3.3, "Managing GPO Link Order," on page 75.

To modify the GPO link order:

1 Log on to a GPA Console computer with an account that has permissions to modify GPO settings.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to modify.

4 On the Action menu in the Group Policy Administrator Console, click **Properties**.

5 On the AD Links tab select the site, domain, or OU for which you want to modify the link order.

6 Click Edit.

7 In the Link Options window, select the GPO and click **Up** or **Down** to change the order of the GPO in the link order list.

8 Click OK.

5.4   Importing GPOs

If you have already implemented Group Policy in Active Directory (AD), you may import a GPO from your domains into the GP Repository. If you have multiple GPOs to import into the GP Repository, use the Offline Mirror wizard. From the wizard, you can also synchronize the GPO link order between AD and the GP Repository.

From the Offline Mirror wizard, you can indicate that you want to import GPOs and also specify how the wizard creates new versions of GPOs in the GP Repository. You can specify for the wizard to skip any unlinked GPOs and skip any OUs that do not contain links, and you can indicate if you want the wizard to mimic the AD category structure in the GP Repository or import all GPOs into a specified single category. For more information on importing all GPOs from within the Offline Mirror wizard, see Section 5.4.2, "Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)," on page 78.

5.4.1   Importing an Active Directory GPO

Once you have GPOs in AD, GPA lets you import each GPO into the GP Repository. Ensure the account you plan to use to export GPOs has sufficient rights before you import the GPOs. For more information, see Section 2.4, "Creating GPA Service Accounts," on page 29.

To import a GPO from an Active Directory domain:

1 Log on to a GPA Console computer with an account that has permissions to import GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** and select the category where you want to import the GPO.

4 On the Action menu, click **All Tasks > Import GPO from AD**.

5 Browse for the GPO to import, and then click **OK**.

NOTE

- If you attempt to import a GPO that already exists in the GP Repository, you receive a message that the GPO already exists in the GP Repository. You have the option to import the GPO and create a new version of the GPO in the GP Repository. Creating a new version increments the GPO version number by 1.

- You can only import GPOs from the Active Directory domain corresponding to the domain you have selected in the GP Repository.

- When you import a GPO from Active Directory, you do not import the block inheritance settings for the OU associated with the GPO.

5.4.2   Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)

The Offline Mirror wizard lets you import all GPOs that are linked to any AD containers in an AD domain into the GP Repository and synchronize their link order. This is also called creating an **offline mirror**. The Offline Mirror wizard streamlines the process of importing all GPOs or GPOs linked to specific AD containers.

The Offline Mirror wizard lets you:

- Select a GP Repository where you have added the domains from which you want to import GPOs

- Select specific AD Containers for importing GPOs to the GP Repository

- Specify the GP Repository category structure for importing the GPOs linked from the selected AD containers

- Select whether to overwrite or replace the security permissions of the category with the corresponding AD OU.

- Save your offline mirror settings as a template file for later use

- Import an offline mirror template file to reuse previously saved settings

- Select whether to sync GPO link order between AD and the GP Repository

- View a summary of your offline mirror wizard selections

- View the progress of your offline mirror operation

When you run the Offline Mirror wizard, GPA creates a category folder in the GP Repository with the name of the domain from which you are importing GPOs. GPA also creates category folders for each OU within the domain in which to place GPOs. You can configure the wizard to also overwrite or replace the security permissions of the category with the corresponding AD OU. Then, GPA imports all the GPOs into the corresponding category folders in the GP Repository, creating a mirror of the AD hierarchy. Alternatively, you can import all GPOs under a single category folder that you specify in the Offline Mirror wizard.

The Offline Mirror wizard also lets you choose whether to import GPOs that do not have links to any objects in Active Directory. When unlinked GPOs are imported to the GP Repository, they are placed in a category called Domain Name - Unlinked GPOs.

The Offline Mirror wizard places GPOs that are linked to a site in a category called *DomainName -* *Site*. Using the wizard, you can import new or updated GPOs from existing domains as well.

NOTE: You must add the domains in the GP Repository corresponding to the Active Directory domains before you run the Offline Mirror wizard.

As a part of the offline mirror process, the Offline Mirror wizard determines whether the GPO link order is not synchronized between AD and the GP Repository. From the wizard, you can choose to have the offline mirror process synchronize the link order based on AD or on the GP Repository. You can also choose to only perform the sync link order process. For information on synchronizing the link order between the GP Repository and AD, see Section 5.9, "Synchronizing GPOs," on page 92.

To import GPOs into the GP Repository using the Offline Mirror wizard:

1 Log on to a GPA Console computer with an account that has the following roles and permissions:
   - GPO Importer role
   - GPO Editor role
   - Create Category
   - Paste GPO Category Link

2 If the source domain does not exist, add a domain in the GP Repository that corresponds to the Active Directory domain you want to mirror. When adding a new domain name, specify the fully qualified domain name, such as `mydomain.company.corp`.

3 Select the domain you created in the GP Repository and click **Run Offline Mirror** on the **Action** menu.

4 Select the target repository, scope, import options, and link order options in the Offline Mirror wizard. If you choose, you can set the offline mirror options by importing an offline mirror template you created previously.

5 If you plan to use the current Offline Mirror wizard settings for another GPO import, save a template of the settings from the Summary window.

6 View the status of the offline mirror in progress or access a log of the progress from the Status window. Click **Finish** when the offline mirror process completes.

NOTE: Depending on the number of GPOs and the complexity of your domain structure, importing all your GPOs can take some time. You can use the Offline Mirror command-line tool, `NetIQ GPA Offline Mirror Wizard.exe` (located in the `\Bin` folder under the product installation path), to run during off-peak hours using a Microsoft Windows scheduled task. For more information about the Offline Mirror command-line tool, see Section A.8.11, "Offline Mirror," on page 179.

TIP: If the Offline Mirror wizard fails to import all of the child OUs of a top-level OU in a domain, inspect the name of the top-level OU. If the name ends with a backslash (\), the wizard will only import the first child OU of that misnamed OU, and then it will skip all of its remaining child OUs and move on to the next top-level OU.

5.4.3 Importing a GPO from Backup

GPA enables you to import GPOs from the folder containing backed up GPOs. If you previously backed up GPOs from Active Directory, you can see the list of backed up GPOs.

To import a GPO from backup:

1 Log on to a GPA Console computer with an account that has permissions to create GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the category to which you want to import the backed up GPO.

4 On the Action menu, click **Import GPO from Backup**.

5 On the Import GPO from Backup window, click **Browse** to select the folder that contains the backed up GPOs.

6 Click **OK**.

7 Select the GPO you want to import from the list of backed up GPOs, and then click **OK**.

5.5   Modifying GPOs

To change the settings of a GPO within GPA, you must first check out the GPO from the GP Repository. When you have modified the GPO, you check the GPO back into the GP Repository. The check out and check in process prevents many users from editing the same GPO at the same time.

NOTE: To see the effect of the GPO changes in Active Directory, you need to export the GPO after approving it. For more information about approving a GPO, see Section 5.8.3, "Managing GPOs for Export," on page 89. For more information about exporting a GPO, see Section 5.8.5, "Exporting GPOs to AD Domains," on page 90.

5.5.1 Checking Out a GPO

When you check out a GPO, no other user can modify it. You cannot make modifications to a GPO in GPA until you check out the GPO.

To check out a GPO from the GP Repository:

1 Log on to a GPA Console computer with an account that has permissions to check in and check out GPOs and modify GPO settings.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to check out.

4 On the Action menu, click **Check Out**. Group Policy Administrator launches the Group Policy Management Editor so you can modify the GPO policy or preference settings.

NOTE: When you check out an approved GPO, its approval status remains set to Approved. But when you check the GPO back in, the GPA console changes the GPO's approval status to Not Approved. You must approve the GPO again to change its approval status back to Approved. For more information about approving a GPO, see Section 5.8.3, "Managing GPOs for Export," on page 89.

To learn how to further restrict what others can do with GPOs that you have checked out, see Section 4.4, "Increasing File Security of a GPO after Checking It Out," on page 59.

5.5.2   Editing Group Policy Settings, Preferences, and Properties

After you check out the GPO, you can edit the policy settings, preferences and properties of a GPO. When you check out a GPO, GPA opens the Group Policy Management Editor, which you can use to add or modify any Group Policy setting. By editing these settings in the GP Repository, GPA only makes changes to the GPO in the GP Repository. Your edits do not change the Active Directory instance of the GPO until you export the GPO from the GP Repository to Active Directory.

To edit GPO settings:

1 Log on to a GPA Console computer with an account that has permissions to check in and check out GPOs and modify GPO settings. To add or modify preferences, log on to a GPA Console computer that supports preference management.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to check out.

4 *If the GPO is not already checked out,* on the Action menu, click **Check Out**. Group Policy Administrator launches the Group Policy Management Editor.

5 *If the GPO is already checked out*, on the Action menu, click **Edit GPO**.

6 In the left pane of the Group Policy Management Editor, expand the GPO to the level of the Group Policy setting you want to modify or to the Group Policy Preference extension for the preference you want to add or edit.

7 Complete one of the following steps, based on what you want to configure or add:

   ◆ *To edit policy settings,* click the specific Group Policy setting you want to modify in the right pane. Then, click **Properties** on the Action menu. For more information about modifying policy settings, see your Microsoft Windows documentation.

   ◆ *To add a preference,* click **New** on the Action menu, then select the Group Policy Preference type. For more information about GPO preferences, see Setting Preferences.

   ◆ *To edit a preference,* click the preference in the right pane, then click **Properties** on the Action menu. For more information about GPO preferences, see Setting Preferences.

   ◆ *To edit GPO properties,* click **Properties** on the Action menu. For more information about GPO properties, see Setting Properties.

8 Configure or modify the settings as needed, click **Apply**, and then click **OK**.

9 Close the Group Policy Management Editor.

10 To save your changes, select the GPO and then, on the Action menu, click **Check In**.

Setting Preferences

After you check out the GPO, you can add or edit Group Policy Preferences (preferences).

Unlike policy extensions, preference extensions typically do not include preferences. You can add one or more preferences to a preference extension. While adding a preference, you can edit the preference settings. You can also change the settings for a preference you added previously.

Editing GPO Properties

In addition to editing the policy and preference settings within a GPO, you can also modify the following properties of a GPO:

- Disabling either Computer Configuration Settings or User Configuration Settings
- Editing, adding, and removing Active Directory links
- Viewing and Editing GPO comments
- Setting a GPO as the master GPO
- Adding WMI filters
- Setting security filters
- Setting the block inheritance option

> **NOTE:** You can use the `NqGPASyncLinkOrder.exe` file to synchronize the block inheritance settings in the GP Repository to match the block inheritance settings in Active Directory during the upgrade process. For more information about this tool, see Section A.10.24, "Synchronize GPO Link Order," on page 225.

5.5.3 Undoing a Check Out

When you undo a check out, GPA reverts the GPO and the GPO settings remain as they were before you checked out the GPO.

To undo a GPO check out:

1 Log on to a GPA Console computer with an account that has permissions to modify GPO settings.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to revert.

4 On the Action menu, click **Undo Check Out**.

5 To confirm you want to undo the check out, click **Yes**.

5.5.4 Checking in a GPO

When you have finished making changes to a GPO, check the GPO back into the GP Repository. When you check in a GPO, GPA saves the changes you made to the GPO in the GP Repository, and not in Active Directory. GPA also releases the GPO to make it available to other GPA users.

NOTE: You cannot save modifications to a GPO in GPA until you check the GPO back in. A check in operation creates a new version of the GPO in the GP Repository.

To check in a GPO:

1 Log on to a GPA Console computer with an account that has permissions to check in and check out GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to check in.

4 On the Action menu, click **Check In**.

5.5.5 Modifying a GPO with Copy and Paste

You can modify an existing GPO by pasting settings, Active Directory links, WMI filters, and security information from a different GPO. When you perform a copy and paste operation on a GPO, you paste the contents of the source GPO into another GPO. Modifying a GPO by pasting contents from another GPO is useful if you want to overwrite the contents of one GPO with another.

The contents you paste from one GPO into another depends on the paste options you configure on the GP Repository. For more information about configuring paste options, see Section 3.1.2, "GPO Paste Options," on page 41.

If you select the option to paste the GPO name, GPA renames the target GPO with the source GPO name. The GUID of the target GPO does not change.

To modify a GPO using copy and paste:

1 Log on to a GPA Console computer with an account that has permissions to modify GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO you want to copy.

4 On the Action menu, click **Copy**.

5 Select the GPO you want to update with the settings from the copied GPO.

6 On the Action menu, click **Paste GPO**. GPA displays the GPO Paste Options window.

7 *If you want to use the default paste options*, click **OK**.

8 *If you do not want to use the default settings*, perform the following steps:

    8a Clear **Use Default Settings**.

    8b Select the specific GPO settings you want to copy.

    8c *If you want to save the paste options for future use*, click **Save**.

9 Click **OK**.

NOTE: If you configure the GP Repository to enforce the default settings, you cannot clear the **Use Default Settings** check box. For more information, see Section 3.1.2, "GPO Paste Options," on page 41.

5.6 Merging GPOs

Maintaining a large number of GPOs can be very tedious and inefficient. Merging them can lead to reduced complexity, making maintenance and troubleshooting simpler and less time-consuming.

GPOs with the following traits make excellent candidates for merging:

- The same settings or policies configured in multiple GPOs.
- Small GPOs that have only one or two settings.

5.6.1 Status of the Merged Object

You have the following options when merging the GPOs:

- You can create a new GPO that contains the merged data from the source GPOs.
- You can overwrite the data of either source GPO with the merged data from both of the source GPOs.
- You can overwrite the data of a non-source GPO with the merged data from the source GPOs.

5.6.2 Status of the Other Attributes

You can only merge the settings from the source GPOs into the merged GPO. The other GPO attributes are handled as follows:

GPO Attribute	Merged GPO is new	Merged GPO created from other GPO
Links	Empty.	Links from the original GPO will be retained.
Delegations	Default delegation and security information.	Delegation and security information from the original GPO will be retained.
WMI Filter	Empty	WMI information from the original GPO will be retained.
GPR Security	Default GPR Security information	Original GPR Security information from the original GPO will be retained.

5.6.3 Considerations

Other things to consider before merging GPOs:

- You can only merge two GPOs at a time. The first GPO that you select to be merged will be considered the *primary GPO*; the second GPO that you select will be considered the *secondary GPO*. During the merge, if there is a settings conflict between the source GPOs, the primary GPO's settings will be used; however, you can override this default behavior and manually select which setting should be merged.

◆ Both GPOs to be merged must come from the GP Repository and must come from the same domain.

◆ You cannot merge GPOs in an untrusted domain.

◆ Third-party settings, such as extension snap-ins, cannot be merged.

◆ To merge the Advanced Audit Policy settings, both source GPOs must have been created in the same language as that used by the computer performing the merge.

◆ The Comment attribute of Administrative Template files are not saved from either source GPO.

5.6.4   Merging Two GPOs

You must have the following permissions before you can merge GPOs:

◆ The **GPO Editor** role on the domain or category that contains the source GPOs.

◆ If you want to preview the GPO before creating it, you must have the **Delete GPO** permission on the category where the target GPO will be created or overwritten.

To merge two GPOs:

1 Log on to a GPA Console computer with an account that has permissions to modify GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** and right-click the domain or category that contains the GPO that you intend to merge.

4 Select **Merge GPOs** and use the browse dialog to select the source GPOs.

5 Click **OK**.

6 *If you want to create the merged GPO in a different container*, do the following:

   6a Select **Create a new GPO in this container** and then browse to the container and select it.

   6b Type a name for the merged GPO in the **New GPO name** field.

7 *If you want to replace a GPO with the merged GPO*, select **Overwrite this GPO** and then browse to the GPO you want to replace and select it.

8 *If you want to pick which settings to merge*, do the following:

   8a In the Customize Settings pane, choose **Select Settings Manually to Merge**.

   This will open a granular configuration window that displays the two GPOs side by side with the different categories of settings. Potential conflicts appear in red text. Under the Select Settings Manually to Merge option, you can change which GPO has the Higher Priority. The highest priority GPO will have all the settings selected.

   8b Select the settings granularity that you require in the different categories from the two GPOs to configure the settings output of the merged GPO.

9 *If you want to merge all settings*, select **Merge All Settings**.

TIP: If you want the merged GPO to inherit the settings from the secondary GPO by default, you will have the option to select **Set "<GPO Name>" GPO to higher priority during merging**.

---

10 *If you want to delete one or both of the source GPOs after merging the settings*, select **Delete Source GPO**.

11 ***If you want to see a preview of the merged output before executing the merge***, click **Preview** when you finish choosing your settings for the merged GPO.

Click **Commit** to create the merged GPO after previewing the pending configuration.

12 ***If you want to bypass the preview***, click **Merge** instead of **Preview**.

5.7 Managing GPO Versions

The GP Repository maintains a history of changes you make to each GPO in the GP Repository. The history is a list of every version of the GPO in the GP Repository. The GPO history includes the following information for each version:

- GP Repository action performed on the GPO
- User account that performed the action
- Date and time when the action occurred

The following table displays the list of GP Repository actions and their effect on the GPO version numbers.

GP Repository Action	Effect on GPO Version Number
Create a new GPO	Creates a new GPO in the GP Repository with the version number set to `1`.
Import a GPO	If the GPO does not exist in the GP Repository, GPA creates a new GPO in the GP Repository with the version number set to `1`. If the GPO already exists in the GP Repository, GPA increments the Repository GPO version number by `1`.
Check out a GPO	Creates a temporary GPO with a version number incriminated by `1`. For example, if you check out GPO version 4, GPA creates a temporary GPO with version number 5.
Check in a GPO	Updates the GP Repository with the contents of the temporary GPO created at check out. GPA updates the version number of the GPO in the GP Repository with the same version number as the temporary GPO.
Undo check out	Deletes the temporary GPO created during check out. The version number remains the same as it was before the check out.
Rollback	Creates a new version number, but rolls back the contents of the GPO to a previous version. For example, if the current version of a GPO in the GP Repository is 5, and you roll back the GPO to version 3, GPA creates a new GPO with version number 6 that has the same settings as version 3. The GP Repository retains all previous versions.
Migrate	Creates a new version on the target GPO after migrating as new GPO or over an existing GPO.
Send for Approval	Moves the GPO to the GPOs Pending Approval folder. Sending a GPO for approval does not change the version number.

GP Repository Action	Effect on GPO Version Number
Reject	Moves the GPO out of the GPOs Pending Approval folder. Rejecting a GPO does not change the version number.
Approve	Approves the latest version of the GPO for export and moves it to the GPOs Pending Export folder. Approving a GPO does not change the version number.
Unapprove	Removes approved status of the GPO. Unapproving a GPO does not change the version number
Export to AD	Exports latest version of the GP Repository GPO to Active Directory. Exporting does not change the GP Repository GPO version number. If the exported GPO does not exist in Active Directory, its revision number is set to: `1 (Computer)` and `1 (User)`. If the GPO already exists in Active Directory, its revision numbers are incremented by `1`. For example, if a live GPO has revision numbers `24 (Computer)` and `42 (User)`, the export of a GPO from the GP Repository changes the revision numbers to `25 (Computer)` and `43 (User)`.
Import from backup	Creates a new GPO in the GP Repository with the version number set to `1`. Checks in the GPO to the GP Repository.

5.7.1 Working with GPO Versions

You can alter the version history, compare versions, and roll back GPA to a previous version, all by completing the following steps:

1 Log on to a GPA Console computer with an account that has permissions to modify GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level and select the GPO for which you want to change version information.

4 On the Action menu, click **View History**. Complete one of the following steps, depending on what you want to do:

- To view and change the version history, see Section 5.7.2, "Viewing Version History," on page 87.
- To compare two versions of GPOs, see Section 5.7.3, "Comparing Versions," on page 88.
- To roll back to a previous version of a GPO, see Section 5.7.4, "Rolling Back to a Previous Version," on page 88.

5.7.2 Viewing Version History

You can access version information for a GPO with the GPO History window. From this window, you can perform the following tasks:

- View a report of the settings for every version
- View a report of the differences between two versions

- Roll back the GPO to a previous version
- See the details of each GP Repository operation performed on the GPO
- View a report of the policy settings in each version of the GPO

To view the version history of a GPO:

1 Follow the steps for Section 5.7.1, "Working with GPO Versions," on page 87.

2 ***To display the details of a GP Repository operation for a particular version***, click the version you want, and then click **Details**.

3 ***To display a report of the policy settings for a particular version***, click the version you want, and then click **Report**.

5.7.3 Comparing Versions

You can generate a comparison or difference report to determine changes between any two versions of a GPO. For more information about comparison and difference reports, see Section 7.4, "Comparing and Differentiating GPOs," on page 121.

To generate a comparison report of two different versions of the GPO:

1 Follow the steps for Section 5.7.1, "Working with GPO Versions," on page 87.

2 Select the two GPO versions that you want to compare. To select two versions of the same GPO, click on one GPO version, press and hold the **Ctrl** key, then click the other GPO version.

3 ***If you want to create a comparison report***, click **Compare**.

4 ***If you want to create a Difference report***, click **Differentiate**.

5.7.4 Rolling Back to a Previous Version

GPA enables you to revert to a previous version of a GPO. Reverting to a previous version is useful if you need to restore the policy settings from an earlier version of the GPO.

To roll back a GPO to an earlier version:

1 Follow the steps for Section 5.7.1, "Working with GPO Versions," on page 87.

2 Select the GPO version to which you would like to roll back.

3 Click **Rollback**, and then click **Yes** to confirm the operation.

4 Click **OK**.

5.8 Exporting GPOs

To see the effect of the GPO changes in Active Directory, you need to export the GPO from the GP Repository. Before exporting a GPO into Active Directory, you can back up the Active Directory version of the GPO. GPOs are available for export only after they have been approved. Ensure that the GP Repository has been synchronized with the latest Active Directory structure before exporting GPOs.

5.8.1 Previewing GPO Export

You can preview whether a GPO is ready to export without causing errors. When you preview a GPO export, GPA performs a health check to detect common errors and offer potential solutions to detected errors, such as checking permissions in Active Directory or the GP Repository or checking whether specific features and services are enabled or running.

When you preview a GPO export, you see a report that details each item checked and the status of the check. You can also begin the export operation or schedule it for later from this window.

5.8.2 Backing Up GPOs Prior to Export

You have the option to back up Active Directory versions of GPOs before you export the GP Repository version of the GPO into Active Directory. If you configure the GP Repository to back up Active Directory GPOs prior to export, the GP Repository creates a backup copy of any Active Directory GPOs that it overwrites during an export and stores the backup copy in a system category called **Backup**. GPA creates this category the first time a GPO overwrite occurs during export. For more information about configuring the GP Repository to back up Active Directory GPOs, see "GPO Backup Options" on page 40.

To restore a GPO from the Backup category into Active Directory, you must approve the GPO and export it to Active Directory in the same manner as other GPOs in the GP Repository. You can only approve and export GPOs in the Backup category. You cannot check out or roll back GPOs in the Backup category. GPA maintains only the latest version of live GPOs in the Backup category.

5.8.3 Managing GPOs for Export

By default, GPOs you create or modify in the GP Repository are unapproved and you need to approve these GPOs before you can export them to Active Directory. GPA users with GPO editing permissions can send GPOs for approval. GPA users with approval permissions can approve or reject GPOs for export. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

GPA users with approval permissions can also return approved GPOs to an unapproved status. Setting a GPO to a status of Unapproved prevents the GPO from being exported to Active Directory.

To manage a GPO export to an Active Directory domain:

1 Log on to a GPA Console computer with an account that has permissions to approve or unapprove GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the category level to the level of the GPO you want to approve or unapprove.

4 Select the GPO you want to approve or unapprove.

5 On the Action menu, choose one of the following options:

   ◆ *To send a GPO for approval,* click **Send for Approval**.

   ◆ *To reject a GPO version,* click **Reject GPO**.

- ◆ *To approve a GPO for export,* click **Approve Version**.
- ◆ *To unapprove a GPO for export,* click **Unapprove Version**.

6 To confirm the approval or unapproval, click **Yes**.

5.8.4 Synchronizing GPOs with AD Before Export

You can link GPOs in the GP Repository to Active Directory container objects such as sites, domains, and OUs. However, if you change the name or location of one of these Active Directory container objects or delete any, the export process no longer links the exported GPO from the GP Repository to Active Directory because the corresponding container object in the GP Repository retains the previous name. To correctly export a linked GPO from the GP Repository, you need to first synchronize the GP Repository with Active Directory.

To synchronize the GP Repository with Active Directory:

1 Log on to the GPA Console computer with an account that has GPA Security or GPO Editor permissions.

2 Start the **GPA Console** in the Group Policy Administrator program folder.

3 In the left pane, click **Group Policy Administrator** and select the domain.

4 On the Action menu, click Properties.

5 Click Sync with AD on the GPO Link Scope tab of the domain Properties dialog box.

NOTE: Clicking Sync with AD clears an OU if the OU is deleted from the Active Directory. This action removes the OU link from the GPOs present in the Repository.

6 Click OK.

5.8.5 Exporting GPOs to AD Domains

Exporting GPOs is the process of moving GPOs from the GP Repository into your Active Directory environment. When you export a GPO into Active Directory, a copy remains in the GP Repository. You can export GPOs into Active Directory in both trusted and untrusted domains.

To export to an untrusted domain, you need to configure the GPA Console to use an Export Only account for that domain. You also need to ensure the domain controller name of the untrusted domain is in the DNS format. For example, if the domain controller name is `corp001` and the untrusted domain name is `mycompany.com`, the DNS name should be `corp001.mycompany.com`. You also have the option to use an Export Only account for trusted domains. For more information about configuring domains to use an Export Only account, see Section 2.4.2, "Creating the Export Only Account," on page 30.

NOTE: If you did not add the Export Only account to the Domain Admins group and if you are exporting a GPO you created in the GP Repository, ensure you modify the GPO to grant the Export Only account all permissions except **Apply Group Policy** and **All Extended Rights**.

To export a GPO from the GP Repository into Active Directory:

1 Log on to the GPA Console computer with an account that has permissions to export GPOs and is a member of the local Administrators group.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the level of the GPO you want to export, and then select the GPO.

4 On the Action menu, click **Export to AD**.

5 Click **Yes**.

6 *If the GPO already exists in Active Directory*, click **Yes** to overwrite it.

7 Type a comment about the export, and then click **OK**.

On successful completion of the export operation, GPA creates the latest version of the GP Repository GPO in Active Directory. If the GP Repository GPO has links to Active Directory objects, the exported GPO has the same links to Active Directory objects, such as domains, OUs, and sites. The link order is exported, too, unless you configure GPA to use the Active Directory link order instead of the link order you configure in the GP Repository. The exported GPO also maintains the same security filters as those of the GPO in the GP Repository. The export increments the Active Directory revision number of the GPO by 1, or sets the version number to 1 if the GPO was not present in Active Directory previously. Exporting does not change the GP Repository GPO version number.

5.8.6 Scheduling GPOs for Export

The Scheduled GPO Export wizard enables you to identify GPOs you want to export to Active Directory and schedule a Microsoft Windows task to perform the export at a specified time. This wizard is useful when you need to export a large number of GPOs at one time or export GPOs after normal business hours. For example, after running a GPO synchronization, you may determine that you have several GPOs you need to export to ensure the consistency of the GPO throughout your Active Directory environment. You can use the wizard to schedule these GPOs for export at a particular time. For more information about GPO synchronization, see Section 5.9, "Synchronizing GPOs," on page 92.

To schedule GPOs for export using the Scheduled GPO Export wizard:

1 Log on to a GPA Console computer with an account that has permissions to export GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**, and then select the GP Repository.

4 On the Action menu, select **Schedule GPOs for Export**.

5 Follow the instructions until you have finished scheduling GPOs for export.

NOTE: The Scheduled GPO Export wizard gives you the option to create an export batch file without creating a Microsoft Windows task to run the batch file. You can then run the batch file yourself at another time or return to the wizard and schedule a Microsoft Windows task to run the batch file when you are ready.

For more information about running the Export batch file, see "The Export Batch File" on page 205.

5.9 Synchronizing GPOs

GPA enables you to match multiple copies of a GPO to a single GPO known as a master GPO. A master GPO is one you select to use as a controlling source for other GPOs. The GPOs you select to match the master GPO are controlled GPOs. The process of matching controlled GPOs to a master GPO is called GPO synchronization.

5.9.1 Understanding GPO Synchronization

You can use GPO synchronization to ensure the consistency of GPOs across your Active Directory environment. Depending upon how you choose to implement Group Policy, you may need to deploy copies of a particular GPO to multiple computers, sites, domains, and OUs. Over time, these GPO copies may be modified and become inconsistent with the original GPO. By defining the original GPO as a master GPO and then making each copy a controlled GPO, you can use GPO synchronization to ensure every controlled GPO remains consistent with the master GPO.

GPA provides two other tools you can use in conjunction with GPO synchronization to help you manage GPOs:

- ◆ Enterprise Consistency Check
- ◆ Scheduled GPO Export wizard

The Enterprise Consistency Check identifies any difference between master and controlled GPOs so you can establish which GPOs need to be synchronized. Once you have identified and synchronized GPOs, you can use the Scheduled GPO Export wizard to export the GPOs to Active Directory automatically at a specific time.

A typical workflow for synchronizing GPOs includes the following steps:

1 Perform an Enterprise Consistency Check to identify controlled GPOs that are no longer synchronized to their master GPOs.

2 Perform a GPO synchronization to synchronize the controlled and master GPOs.

3 Run the Scheduled GPO Export wizard to export the synchronized GPOs to your production Active Directory environment.

For more information about the Enterprise Consistency Check report, see Section 7.5, "Analyzing Multi-Domain GPOs against a Master GPO," on page 124. For more information about the Scheduled GPO Export wizard, see Section 5.8.6, "Scheduling GPOs for Export," on page 91.

5.9.2 Performing a GPO Synchronization

The following procedure describes how to specify master and controlled GPOs and perform a GPO synchronization to match the controlled GPOs to the master GPO. This process synchronizes controlled GPOs to master GPOs in the GP Repository only. To ensure your Active Directory

environment remains synchronized, you must export the GPOs you synchronized in the GP Repository to Active Directory. For more information about exporting GPOs, see Section 5.8, "Exporting GPOs," on page 88.

To synchronize GPOs:

1 Log on to the GPA Console computer with an account that has GPO synchronization permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository**.

4 Expand the appropriate domain hierarchy to the GPO you want to identify as a master GPO, and then select the GPO.

5 On the Action menu, click **Properties**.

6 Click the GPO Sync Options tab.

7 Select the **Make this GPO a master GPO** check box, and then click **OK**.

8 Click the Synchronization tab in the GPO result view.

9 To select controlled GPOs for this master GPO, click **Add**.

10 *If you want to select GPOs from the GP Repository*, accept the default selection, and then click **OK**.

11 *If you want to select GPOs from an Enterprise Consistency Check report XML file*, select **ECC Wizard XML file**, and then browse to the location of the file.

12 *If you want to determine whether the controlled GPOs are in sync with the master GPO*, select the controlled GPOs you want to check, and then click **Run Sync Report**. GPA generates an Enterprise Consistency Check report on the master GPO and the selected controlled GPOs in the GP Repository. For more information about the Enterprise Consistency Check report, see Section 7.5, "Analyzing Multi-Domain GPOs against a Master GPO," on page 124.

NOTE: The GPA console disables the Run Sync Report button on the Synchronization tab for GPOs in untrusted domains in the GP Repository. However, you can generate these reports from a GPA console installed on a computer in the untrusted domain.

---

13 *If you want to update the domain map for a controlled GPO and ensure the Active Directory links for these controlled GPOs are synchronized with the master GPO*, select the GPO and click **Update Domain Map**. You do not need to perform this step for a controlled GPO if the **Mapped** column indicates `Yes`. For more information about updating domain maps, see Step 8 on page 96 of Section 5.10.3, "Migrating a GPO Between GP Repository Domains," on page 95.

14 *If you want to synchronize a controlled GPO with the master GPO*, select the controlled GPO and click **Synchronize**. You do not need to perform this step if the **In Sync** column indicates `Yes`.

NOTE

- If any controlled GPOs are in untrusted domains, you must provide credentials for each untrusted domain that have domain administrator permissions.

- If you want to synchronize the forest root default domain policy with the default domain controller policies, including the default domain policies in the child domains, you need to first create a custom policy in each domain and then synchronize these policies because these GPOs have the same GUID.

5.9.3 Synchronizing GPO Link Order Using the Offline Mirror Wizard

Use the Offline Mirror wizard to synchronize the GPO link order of repository GPOs with either the relative GPO link order in AD or the GP Repository.

NOTE: The Synchronize GPO Link Order tool, `NqGPASyncLinkOrder.exe`, also synchronizes link order from the command-line. GPA installs this tool in the `\Tools` folder under the product installation path. For more information, see Section A.10.24, "Synchronize GPO Link Order," on page 225.

To synchronize the GPO link order between Active Directory and the GP Repository using the wizard:

1 Run the Offline Mirror wizard on the domain where you have GPOs for which you want to synchronize the link order. For more information on using the Offline Mirror wizard, see Section 5.4.2, "Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)," on page 78.

2 Select the scope, including the domain and OUs. Select the option to import GPOs if you plan to update the repository with GPOs from on the specified domain.

3 On the Link Order Options window of the Offline Mirror wizard, select **Sync Link Order** and specify whether you want the wizard to synchronize GPO link orders based on the GP Repository or AD.

4 Complete the rest of the wizard windows, then view the status of the sync link order process from the Status window.

5.10 Migrating GPOs

GPA allows you to migrate a GPO in the GP Repository from one domain to another. For example, you can migrate a GPO from a production domain in the GP Repository to a test domain in the GP Repository. You can then modify and evaluate the GPO in the test domain before you implement the GPO in your production Active Directory environment. You migrate GPOs between GP Repository domains with the GPO Migration wizard. You can also map the source domains from which you want to migrate GPOs before you start the GPO Migration wizard. For more information about mapping source domain information, see Section 5.10.2, "Mapping Source Domain Information," on page 95.

One of the key challenges when migrating GPOs from one domain to another is that some information in the GPO is specific to the domain to which the GPO is linked. When you transfer the GPO to a new domain, it may not be possible to use the same settings. Settings that are specific to a domain include references to Universal Naming Convention (UNC) paths, GPO links to a specific container, and security principals such as users, groups, and computers.

You can use a single GP Repository to manage GPOs from multiple domains. Most organizations with Active Directory implementations maintain separate test and production domains for their regular operations. They carry out changes to GPOs in their test environments and then move the tested GPOs into the production domains. For more information about test and production domains, see Section 1.2.2, "Understanding Test and Production Environments," on page 16.

5.10.1 Understanding Migration Tables

A migration table allows you to map references to GPO settings from the source GPO to new values in the destination GPO. A migration table consists of one or more mapping entries. Each mapping entry consists of a type, source reference, and destination reference.

Migration tables store the mapping information in XML format and have the file name extension `.migtable`. You can create migration tables using the Migration Table Editor (MTE) that is available with GPMC. The MTE is a convenient tool for viewing and editing migration tables if you are not familiar with XML.

5.10.2 Mapping Source Domain Information

You can map source domain information to the target domain using the information available in migration tables. The migration table allows you to map all default settings from the source domain to the target domain.

After you map the default settings, you do not need to again map this information during migration of GPOs between domains. For more information about migrating GPOs between domains, see Section 5.10.3, "Migrating a GPO Between GP Repository Domains," on page 95.

To map source domain information:

1 Log on to the GPA Console computer with an account that has GPA Security or GPO Editor permissions.

2 Start the **GPA Console** in the Group Policy Administrator program folder.

3 In the left pane, expand **GP Repository** to the domain you want to map, and then select the domain.

4 On the Action menu, click Properties.

5 *If you want to add mapping information from another migration table*, on the Map tab, click **Load**.

    5a Browse to and select the migration table you want to use.

    5b Click **Open**.

6 *If you want to save the changes to the migration table locally*, on the Map tab, click **Save**.

    6a Browse to the folder where you want to save the migration table.

    6b Click **Save**.

7 Click **Apply**, and then click **OK**.

5.10.3 Migrating a GPO Between GP Repository Domains

The GP Repository supports the following migration scenarios:

- Migrating GPOs between trusted domains in the same or different forests
- Migrating GPOs between untrusted domains in the same or different forests

GPA stores GPO mapping information in the GP Repository. You only need to define the mapping information the first time you migrate a GPO. During subsequent migrations, you only need to update the mapping information as required.

To migrate a GPO from one GP Repository domain to another with the GPO Migration wizard:

1 Log on to the GPA Console computer with an account that has permissions to migrate GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 Expand **GP Repository** to the level of the GPO you want to migrate.

4 Select the source GPO you want to migrate.

5 On the Action menu, click **Migrate GPO**.

6 Select whether you want to create a new GPO or update an existing GPO, and then click **Next**.

7 Specify the target domain and category information. If you are updating an existing GPO, you must also specify the GPO to replace.

NOTE: If you are migrating a GPO to an untrusted domain, you need to provide user credentials in the untrusted domain that have the permission to create GPOs.

8 *If you want to modify an existing domain map or add a new domain map,* click **Edit Domain Map**, modify the map for each tab as appropriate, and then click **OK**:

   ◆ *To modify a mapping*, double-click any entry you want to modify, and then follow the instructions on the window**.**

   ◆ *To delete a mapping*, select the entry and click **Clear**.

   ◆ *To reset all mappings to the default*, click **Reset**.

   NOTE: If you are modifying the AD links mapping, ensure both the source and target domains are the same.

9 Click **Next**.

10 Review the summary information to confirm you have not left any portions of the GPO unmapped.

   NOTE

   ◆ Be sure you have not left any security account unmapped. If a source GPO contains a security account you have not mapped to a corresponding target account, the migration process ignores that particular security account and proceeds with the rest of the migration. Important security information could be lost in the process, which could cause problems when exporting the GPO to Active Directory.

   ◆ Any Active Directory object that does not have corresponding mapping information in the target domain is not linked in the target domain.

11 Click **Finish**.

GPOs you have migrated to a production domain in the GP Repository you can then export directly to the corresponding Active Directory production domain.

5.11   Managing Administrative Template Files

Administrative template (ADM) files are unicode-formatted text files that Group Policy uses to describe the location of registry-based policy settings in the Windows Vista and Windows 7 registry. ADMX files are XML-based administrative template files, which were introduced with Microsoft Windows Vista Service Pack 1 and are used instead of ADM files.

ADM files are no longer supported. On a computer running Microsoft Windows Vista Service Pack 1 or later, GPA ignores the following default ADM files and only uses the corresponding ADMX files:

- `system.adm`
- `inetres.adm`
- `conf.adm`
- `wmplayer.adm`
- `wuau.adm`

You need to apply any changes you make to custom ADM files to the corresponding ADMX files. GPA neither displays nor applies new policy settings and modifications you make on ADM files if there is a corresponding ADMX file. For more information about ADM and ADMX files, see the Microsoft Windows documentation.

5.11.1   Understanding ADMX Files

ADMX files are XML-based administrative template files, which were introduced with Microsoft Windows Vista Service Pack 1 and used instead of ADM files. ADMX files are language-neutral and support multilingual display of policy settings. The file structure comprises a language-neutral (.admx) file and a language-specific (.adml) resource file. Multilingual support allows administrators in different countries to work with the same ADMX files and see the descriptions of the Group Policy settings in the local language. You can only manage ADMX file-based Group Policy settings on computers running Microsoft Windows Vista Service Pack 1 or later. You can create and edit ADMX files using any XML-compatible editor.

Microsoft Windows manages ADMX files from the central store that is a central location in the domain. Before you install the GPA Console on a computer running Microsoft Windows Vista Service Pack 1 or later, manually create a central store on the domain controller. The central store enables you to read ADMX files from a single domain-level location on the domain controller. You cannot manage ADMX files if you do not create a central store. For more information about creating the central store, see the Microsoft Windows documentation.

The ADMX and ADML files are available in the default local policy definition folder, `%systemroot%\PolicyDefinitions`. When you install the GPA Console on a computer running Microsoft Windows Vista Service Pack 1 or later, the installation process replaces the default local policy definition folder with a local folder, `\installDir\Local GPOs\domain name\`, and redirects all future ADMX files to the local folder. GPA uses this local folder to temporarily store and work with ADMX files.

 ◆ If you did not configure the central store before installing the GPA Console, you can retrieve the default ADMX and ADML files from the `%systemroot%\Windows\PolicyDefinitions_old path` folder.

 ◆ After installing the GPA Console, synchronize the ADMX and ADML files from the central store. For more information about synchronizing ADMX and ADML files from the central store, see Section 5.11.3, "Synchronizing or Exporting ADMX Files from the Central Store," on page 99. After synchronizing the ADMX and ADML files, run the GPO Settings report on any GPO in the GP Repository to ensure tools using the local ADMX and ADML files can correctly display the administrative templates policies. For more information about running the GPO Settings Report, see Section 7.2, "Viewing GPO Setting Information," on page 118.

5.11.2 Working with ADMX Files in the GP Repository

While administering domain-based GPOs, you can use ADMX files available in the GP Repository. You can also share custom ADMX files with other Group Policy administrators in a domain by adding these files to the GP Repository.

In the GPA Console, you can perform the following tasks:

 ◆ Add ADMX files
 ◆ Remove ADMX files
 ◆ Approve ADMX files
 ◆ Unapprove ADMX files
 ◆ Synchronize ADMX files from the central store
 ◆ Export ADMX files to the central store

NOTE: You can perform these tasks only on computers running Microsoft Windows Vista Service Pack 1 or later.

To add or remove ADMX files:

 1 Log on to the GPA Console computer with an account that has Create GPO permissions (for adding an ADMX file) or Delete GPO permissions (for removing an ADMX file).

 2 Start the **GPA Console** in the Group Policy Administrator program group.

 3 In the left pane, expand **GP Repository**, and then select the GP Repository.

 4 Select the domain.

 5 On the Action Menu, click **Properties**.

 6 On the domain Properties dialog box, click the Repository ADMX tab.

 7 Complete one of the following steps:

 ◆ Click **Add**, and Browse to and select the `.admx` file you want to add, and then click **OK**.

 ◆ Select the ADMX file you want to remove, and then click **Remove**.

You can add ADMX files from any folder to any domain in the GP Repository. All corresponding `.adml` files should be in the culture subfolders before you add `.admx` files into the domain in the GP Repository. The culture subfolders are available under the folder where `.admx` files are available. When you add ADMX files into the domain in the GP Repository, GPA automatically scans the culture subfolders for the language-specific `.adml` files.

NOTE: You can add ADMX files without adding the corresponding ADML files. However, GPA does not allow you to edit the settings available in ADMX files until the corresponding ADML files are available in the culture subfolders.

You can select and remove ADMX files from any domain in the GP Repository. Removing `.admx` files from the GP Repository also removes corresponding `.adml` files from the GP Repository culture subfolder.

NOTE: If you delete any domain and its corresponding GPOs, you need to manually delete the related ADMX files from the `\installDir\Local GPOs\domain name\` folder on all computers running Microsoft Windows Vista Service Pack 1 or later and the GPA Console.

5.11.3 Synchronizing or Exporting ADMX Files from the Central Store

When you install or upgrade GPA, the GP Repository does not contain any ADMX files. To add the default ADMX files provided by Microsoft Windows Vista Service Pack 1 or later, you need to synchronize these files from the central store to the GP Repository. You can also synchronize new or modified ADMX files from the central store to the GP Repository. You need to ensure the central store is available at all times. For more information about the central store, see the Microsoft Windows documentation.

When you synchronize ADMX files from the central store to the GP Repository, the GP Repository contains the central store version of ADMX files. If you create a custom ADMX file with the same name as the central store version of the ADMX file, and then add this file from any folder on your local computer to the GP Repository, GPA overwrites the central store version of the ADMX file in the GP Repository.

Users with Export GPO permissions can export ADMX files into the central store. If the same ADMX file exists in the central store, you have the option to either overwrite the existing file or skip exporting the specific ADMX file and export the remaining ADMX files.

  To synchronize or export ADMX files:

  1 Log on to the GPA Console computer with an account that has Import GPO from AD permissions (for synchronizing ADMX files) or Export GPO to AD and Modify Export Status permissions (for exporting ADMX files).

  2 Start the **GPA Console** in the Group Policy Administrator program group.

  3 In the left pane, expand **GP Repository**, and then select the GP Repository.

  4 Select the domain.

  5 On the Action Menu, click **Properties**.

  6 On the domain Properties dialog box, click the Repository ADMX tab.

7 Perform one of the following tasks:

- ◆ Click **Sync from Central Store** to synchronize ADMX files.
- ◆ Click **Export** to export all approved ADMX files.

NOTE: Changes you make to ADMX files you exported to the central store will be available in the GPO Settings report after you close and re-open the GPA Console.

---

8 Click **OK**.

The `AdmxSyncCentralStoreReport.log` file provides a report that lists the ADMX files GPA has and has not synchronized from the central store. The `AdmxSyncCentralStoreReport.log` file is available on the local computer in the `\installDir\Log files` folder.

The `ADMXExport_SummaryReport.log` file provides the list of ADMX files that GPA did not export. It is available on the local computer in the `\installDir\Log file` folder.

5.12   Setting GPO Security Filters

You can set security filtering on users and groups to mask or lock the GPOs in the GP repository. When you set this level of security, the GPA Console no longer allows the users or groups to see or edit the targeted GPOs.

You must have the "GPO Security Filtering" privilege to mask or lock a GPO.

You cannot mask or lock GPOs from your own user account or GPOs from the GPA Repository Management Group Users group.

5.12.1   Masking or Locking GPOs

You can set two types of security filters from the GPA Console:

- ◆ You can *mask* the GPO to hide it from the selected user or group.
- ◆ You can *lock* the GPO to prevent the selected user or group from editing it.

TIP: You can filter a single GPO or you can filter all or some of the GPOs contained in a domain or category.

To filter a GPO:

1 Log on to a GPA Console computer with an account with the `GPO Security Filtering` role to filter GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the domain, category, or GPO level, depending on your intent, and right-click the object to which you want to apply the filter.

4 Select **GPO Security Filtering**.

5 Browse to and select the GPOs to be masked or locked.

6 Browse to and select the users or groups to be prevented from viewing or editing the GPOs selected in step 5.

7 Set the **Filter** to the appropriate type and click **Add**.

8 Click **OK**.

5.12.2 Unmasking or Unlocking GPOs

To remove a security filter from a GPO:

1 Log on to a GPA Console computer with an account with the `GPO Security Filtering` role to filter GPOs.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Repository** to the domain, category, or GPO level, depending on your intent, and right-click the filtered object.

4 Select **GPO Security Filtering**.

5 Locate the filtered object in the **GPOs Security Filter** table and select its check box.

TIP: If you want to unfilter all of the GPOs displayed in the table, select the check box at the top of the column.

---

6 Click **Remove** and then click **OK**.

6 Working with GPOs in Active Directory

When using Group Policy Administrator (GPA) to work with GPOs, you can use the GP Repository to effectively plan and evaluate your Group Policy before implementing it in your production environment. The GP Repository also provides change management features. However, GPA also allows you to directly edit GPOs in Active Directory with the GP Explorer.

6.1 Workflow for Managing GPOs in Active Directory

When you work with GPOs in Active Directory, you need to be very familiar with Group Policy concepts and how Group Policy affects objects across your production environment. When working in Active Directory instead of the GP Repository, you cannot evaluate GPOs before they are in effect. Therefore, you can disrupt activity and productivity with an incorrectly defined GPO. To help avoid costly mistakes, use a test environment to verify the effects of a change before you deploy that change in your production environment.

NOTE: If you need to modify your production environment, change your OUs and GPOs during off-hours when your organization has reduced network use and you can verify the effects of the GPOs you have implemented.

The following checklist outlines the general process and workflow for working with GPOs in Active Directory.

☑	**Installing and Configuring Your Environment**
☐	If you are upgrading from a previous version of GPA, consider the important differences in this version. For more information, see the Release Notes.
☐	Install GPA. For more information, see Chapter 2, "Installing Group Policy Administrator," on page 19.
☐	Delegate permissions in your test and production environment to define who can modify GPOs and OUs.
☐	Configure global settings, such as default copy options and showing or hiding domains.

☑	**Creating and Modifying GPOs in Your Test Environment**
☐	Create a new GPO in the test domain. For more information, see Section 5.3.1, "Creating a GPO Directly in the GP Repository," on page 73.
☐	Configure policies and preferences. For more information, see the following sections: ◆ Section 5.5.2, "Editing Group Policy Settings, Preferences, and Properties," on page 81. ◆ "Setting Preferences" on page 82

☑	**Testing and Evaluating GPOs in Your Test Environment**
☐	Use RSoP what-if reports to check changes. For more information, see Chapter 7, "Reporting on GPOs," on page 117.
☐	Edit the test environment to fix any problems. For more information, see Section 5.5.2, "Editing Group Policy Settings, Preferences, and Properties," on page 81.
☐	Use RSoP after fixing any problems (as a double-check). For more information, see Chapter 7, "Reporting on GPOs," on page 117.

☑	**Deploying GPOs to Your Production Environment**
☐	Back up live GPOs for disaster recovery. For more information, see Section 6.14, "Backing Up and Restoring GPOs," on page 113.
☐	Migrate within the GP Repository to the production domain. For more information, see Section 5.10.3, "Migrating a GPO Between GP Repository Domains," on page 95.
☐	Test the production environment. For more information, see Chapter 7, "Reporting on GPOs," on page 117.

☑	**Troubleshooting, Disaster Recovery, and Ongoing Management**
❑	Use RSoP to double-check domain controller settings. For more information, see Chapter 7, "Reporting on GPOs," on page 117.
❑	Run diagnostics on client computers.
❑	Generate reports to verify settings and find GPOs. For more information, see Chapter 7, "Reporting on GPOs," on page 117.
❑	Back up live GPOs for disaster recovery. For more information, see Section 6.14, "Backing Up and Restoring GPOs," on page 113.
❑	Use scripts to make common tasks easier. For more information, see Appendix A, "Automating GPA Operations with .NET," on page 145.

6.2 Understanding GP Explorer

GPA offers cross-forest management, giving you the ability to control enterprise management capabilities from a single GPA Console. If the required trust relationships are set up properly, you can select a domain belonging to a different forest and administer policies on that domain.

NOTE: For cross-forest management to perform successfully, ensure the proper trusts are in place between forests.

GP Explorer provides a Group Policy-centric interface to Active Directory. It provides a hierarchical view of policies associated with OUs, domains, and sites. GP Explorer also provides the administrator with basic and advanced GPO management features, such as GPO reporting, backups and restores, search, and delegation.

Each GPA Console connects to a domain controller and its associated domain. When you expand GP Explorer, GPA displays the domain to which you are connected.

6.2.1 Domain Node

From the domain node, you can search for and link to GPOs, back up, restore, and replicate GPOs. In addition, GPA allows you to toggle the inheritance option for all GPOs within the domain and view and change domain node properties.

You define GPOs at the domain level, unless you define a local policy. When you expand a domain node you can see the OU hierarchy and the associated GPOs. Use the Action menu to work with GPOs in the domain node.

6.2.2 Organizational Unit Node

From an OU node, you can link a Group Policy to an OU. When you expand an OU node, you can see the GPOs that are associated with the OU. In addition, GPA allows you to toggle the inheritance option for all GPOs within the domain. Use the Action menu to work with GPOs in the organizational unit node.

6.2.3   GPO Node

From a GPO node, you can use the Action menu to edit, back up, import, delete, rename, and specify properties for GPOs.

6.2.4   Unlinked GPOs Node

The Unlinked GPOs node lists the GPOs in a domain that have no links to any container in the same domain. This information can be very useful when you need to find and resolve orphaned GPOs.

In any domain, GPOs can exist that were created for test purposes but not subsequently deleted. Also, links to a GPO can be deleted, but not the entire GPO itself. Though such orphaned GPOs exist in the domain, they are not seen on any GPO property page for an OU-level or domain-level node. In an ideal production environment, the Active Directory containers do not have any orphaned GPOs.

You can use the Unlinked GPOs container to clean up the Active Directory environment by identifying orphaned GPOs. Administrators can then delete them or link them to an OU, domain, or site.

6.2.5   Sites GPO Node

The Sites GPO node under the GP Explorer node displays the available list of sites. GPA displays any GPOs linked to these sites. To view the report on the site GPOs, click the site node in the left pane, and then select the Summary of GPOs tab in the right pane.

6.3   Connecting to, Viewing, or Hiding a Domain or Forest

The first time you start GPA, you must connect to a domain in a forest to manage. By default, GPA connects to the PDC. You can use the GPA Console to connect to a different domain or domain controller. The GPA Console displays the connected domain in square brackets next to each snap-in name.

When you connect to a forest, GPA shows all the domains in that forest. You can show and hide domains for a particular forest.

To connect to or view a domain or forest:

1 Log on to a GPA Console computer.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, click **GP Explorer**, then navigate to the forest that contains the domain you want to connect to or view.

4 Perform one of the following actions:

   ◆ ***If you are connecting to a domain***, click **Action** > **Add Forest** and click **OK**. The GPA Console connects to the domain and adds a forest node for that domain.

   ◆ ***If you want to show or hide domains,*** select the domains you want to show or clear the domains you want to hide, click **Action** > **Show Domains**, then click **OK**.

6.3.1 Changing the Domain Controller

From the GPA console, you can select the Domain Controller (DC) the software uses for GP Explorer operations. When using domains with multiple DCs, you can specify for the console to use only the Primary Domain Controller (PDC), any available DC, or the currently specified DC to perform a GPO operation from the GP Explorer.

To configure a domain controller:

1 Log on to the GPA Console computer with any domain user account.

2 Start the GPA Console in the Group Policy Administrator program group.

3 In the left pane, expand GP Explorer, and navigate through the forest to the domain you want the console to use for all GPA operations.

> **NOTE:** For improved performance, consider selecting a domain controller that is close on the network to the GPA Console. You can also consider choosing a domain controller that you want to initiate replication or that you want to use as a temporary back up if you are taking your preferred domain controller offline for service.

4 Select the domain.

5 Click **Action** > **Change Domain Controller**.

6 In the **Change Domain Controller** window, select the preferred domain option.

  ◆ Choose Primary domain controller (PDC Emulator) for GPA to use this domain controller for operations. When you configure the GPA Console to use a specific PDC, and that DC later becomes unavailable on the network, GPA notifies you that the domain controller is inaccessible and asks you to change the DC.

  ◆ Choose Any available domain controller for GPA to choose an available DC for operations. If you specify that the GPA Console use any available DC, and if the DC in use becomes unavailable on the network, GPA searches for another available DC. If another DC is available, GPA continues to work without notifying you or sending any error messages. If another DC is not available, GPA notifies you that the domain controller is inaccessible.

  ◆ Choose This domain controller for GPA to contact the selected DC for operations. If a you indicate for the console to use a specific domain controller, and the DC becomes unavailable on the network, the GPA Console notifies you that the specified DC is not accessible and asks you to select another available DC.

7 Click **OK**.

When the GPA Console does not detect any available DCs, the software does not allow you to work with that domain at the GP Explorer level.

6.3.2 Connecting to a GPA Server

You must connect to a GPA Server for some activities, such as searching for GPOs. For more information, see Section 3.3.3, "Connecting to a GPA Server," on page 50.

6.4   Creating or Linking a GPO

When you create a GPO, GPA automatically links it to the domain or OU in which you are working.

Although GPA links new GPOs automatically to whatever Active Directory container you selected (domain or OU) when you created the GPO, you can link the GPO to other Active Directory containers.

To create or link a GPO:

1   Log on to a GPA Console computer.

2   Start the **GPA Console** in the Group Policy Administrator program group.

3   In the left pane, expand **GP Explorer** to the domain or OU where you want to create or link to a GPO.

4   Perform one of the following actions:

◆   *To create a new GPO,* on the Action menu, click **Create and Link New GPO**. Type a name for the GPO, and then click **OK**.

◆   *To link to an existing AD container,* on the Action menu, click **Link an Existing GPO**. Select the GPO to which you want to link to the domain or OU, and then click **OK**.

6.5   Editing Policies and Preferences

You can configure policy settings and preferences for a GPO in Active Directory with the GP Explorer in the GPA Console.

You can add, delete, or modify policies in the Software Settings folder. However, you cannot add or delete policies in the Windows Settings folder. You can either define the policies or choose not to define the policies in the Windows Settings folder.

You can add, delete, or modify preferences.

GPA provides reports in which you see what policy settings and preferences have been configured with a GPO and the values assigned to each. GPA automatically generates the report when you select the Settings tab of any GPO.

To configure the policy settings or preferences for a GPO:

1   Log on to a GPA Console computer with an account that has permissions to modify GPO settings in Active Directory.

2   Start the **GPA Console** in the Group Policy Administrator program group.

3   In the left pane, expand **GP Explorer** to the GPO you want to configure

4   Click the GPO you want to configure.

5   In the right pane, click the Settings tab. GPA displays the GPO Settings Report in the right pane.

6 *If you are configuring GPO settings, or if you want to review the GPO settings prior to modifying them*, save the GPO Settings Report and then print it:

    6a Click the GPO in the left pane.

    6b On the Action menu, click **Save Report**.

    6c View and print the report you just saved.

7 On the Action menu, click **GPO Editor**.

8 In the left pane, expand the folders to the policy or preference you want to configure and click the policy or preference in the right pane.

9 Perform one of the following steps, depending on what you are configuring:

    ◆ *To edit a policy setting,* on the Action menu, click **Properties**.

    ◆ *To add edit a preference,* on the Action menu, click **New**, and then select the appropriate type.

    ◆ *To change a preference setting,* on the Action menu, click **Properties**.

10 Edit the settings, and then click **OK**. For more information, see the Microsoft Windows documentation.

6.6   Adjusting GPO Filters

By default, a new GPO applies to every user. Therefore, a new GPO has just one filter named `Authenticated Users`. You must filter each policy to apply to users, groups, and computers you want.

NOTE: By default, every GPO is assigned to the Authenticated Users group.

To adjust filtering and manage the assignment of users, groups, and computers to a GPO:

1 Log on to a GPA Console computer.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Explorer** to the GPO.

4 Click the Filters tab in the right pane.

5 Review the effective security filters displayed in the right pane.

6 Select the GPO in the left pane.

7 On the Action menu, click **Properties**.

8 *If you need to add a user, group, or computer to the list of filters*, perform the following steps on the Delegation tab:

    8a Click **Add**.

    8b Select the appropriate domain in the **From this location** field. The default location is the parent domain, even if you are connected to a child domain. Make sure you are selecting an object from the appropriate domain.

8c Select the user, group, or computer to add, and then click **OK**. You can repeat this step multiple times to add multiple objects to the filter.

8d Select the **Receive Settings [Read, Apply]** permissions for the selected group or user, and then click **OK**.

9 When you have finished adjusting the filters to the GPO, click **Apply**, and then click **OK**.

6.7 Working with WMI Filters

Windows Management Instrumentation (WMI) filters let you dynamically detect the scope of Group Policy objects (GPOs), based on the attributes of the targeted computer.

For example, to ensure that each GPO associated with a group can only be applied to devices running the correct version of Windows, assign WMI filters to the GPO. Although you can create a separate membership group for each GPO, you would then have to manage the memberships of the different groups. Instead, use only a single membership group, and let WMI filters automatically ensure the correct GPO is applied to each device.

To create a WMI filter:

1 Expand the target domain, and locate the WMI Filters node in the domain's tree.

2 Right-click the WMI Filters node, and select **New WMI Filter**.

3 Provide a name and description for the filter.

4 Add at least one query. To learn about creating queries, see WMI Queries.

5 Click **Save**.

To import a WMI filter:

1 Expand the target domain and locate the WMI Filters node in the domain's tree.

2 Right-click the WMI Filters node and select **Import**.

After creating or importing a WMI filter, you can perform the following actions against it:

- Edit
- Export
- Copy
- Delete
- Rename
- Backup
- View its properties
- Link to GPOs
- Delegate permissions to users or groups

To perform an action against a WMI filter, right-click it and select the desired action.

To export a WMI filter:

1 Expand the target domain, and locate the WMI Filters node in the domain's tree.

2 Right-click a filter, and select **Export**.

3 Locate the directory where you want to save the filter, and click **Save**.

To link a WMI filter to a GPO:

1 Expand the target domain, and locate the WMI Filters node in the domain's tree.

2 Expand the WMI Filters node, and click the desired filter.

3 In the General tab of the filter configuration, right-click in the GPO pane and select **Add**.

4 Select a GPO from the Group Policy objects, and click **OK**.

To delegate a WMI filter to a user or group:

1 Expand the target domain, and locate the WMI Filters node in the domain's tree.

2 Expand the WMI Filters node, and click the desired filter.

3 Select the **Delegation** tab in the filter configuration, and click the **Add** button at the bottom of the page.

4 Enter, check, and select the object name, and click **OK**.

6.8   Setting Indexing Properties

GPA indexes GPOs in Active Directory periodically according to a schedule you set.

To set indexing properties for a domain:

1 In the left pane of the GPA Console, expand **GP Explorer** to the domain where you want to set indexing properties.

2 Right-click the domain and click **Properties**.

3 In the Domain Properties window, click **Indexing Settings** tab, then select the option to include the domain for indexing and set the schedule you want in **Index Schedule**.

4 *If you want to delete current indexes and completely rebuild them,* click **Rebuild Indexes**.

5 *If you want to update current indexes outside the configured schedule,* click **Update Indexes**.

6 *If indexing was in progress when you opened the window,* click **Refresh** to update the status area.

7 Click **OK**.

6.9   Copying GPOs

Use this menu option to link the GPO to an Active Directory container.

To copy GPOs:

1 Select the GPO from which you are going to copy the policy settings.

2 On the Action menu, click **Copy**.

3 Select the **All GPOs** node into which you are going to paste.

4 On the Action menu, click **Paste**.

6.10   Importing GPO Settings

Unlike **Copy**, this action completely overwrites policy objects in the second GPO, including software and IP securities. The **Import** feature uses a backup folder as the source of the policy objects, rather than using the GPA tree. In other words, the GPO from which you are copying information must be backed up before you import the information to another GPO.

To import GPO settings:

1 Select the GPO into which you are going to import.

2 On the Action menu, click **Import**.

3 Follow the Import Group Policy Object wizard until you have completed importing the GPO settings.

6.11   Updating GPOs Remotely

GPA enables you to remotely refresh the group policy settings of all computers in an OU from a single GPA console running Windows Server 2012/2012 R2 or Windows 8/8.1.

To initiate the update you must have the following permissions:

- Generate resultant set of policy (logging)
- Generate resultant set of policy (planning)

The following groups have the permissions necessary to initiate a group policy update:

- SYSTEM
- Domain admins
- Enterprise admins
- Administrators

To start a remote group policy update:

1 Log on to a GPA Console computer with an account that has permissions to starts remote GPO updates.

2 Start the GPA Console in the Group Policy Administrator program group.

3 In the left pane, expand GP Explorer and right-click the OU or GPO that you want to receive the update.

4 Select **Group Policy Update**.

6.12   Deleting GPOs

You can permanently delete a GPO, which includes deleting all its settings and links.

NOTE: If you want to temporarily remove the GPO from the set of applied policies, you can remove the GPO links. This will have the same effect as deleting the GPO, without destroying the GPO.

To permanently delete a GPO from an OU:

1 Select the GPO you want to remove.

2 On the Action menu, click Delete.

NOTE: *If you only want to remove the GPO link*, select the Remove the Link from the list radio button.

---

3 *If you want to permanently delete the GPO and related links*, select the Remove the Link and delete the GPO permanently radio button.

4 Click OK.

6.13 Understanding the GPO Settings Report

The GPO Settings Report is a robust and extremely useful tool within GPA.

- ◆ You can use the GPO Settings Report to maintain hard copies of the settings for each GPO in your organization.

- ◆ If you need to compare two GPOs, you can compare two GPO Settings Reports rather than switch between two GPOs.

- ◆ You can save the GPO Settings Report as an HTML file for future reference by clicking **Save Report** on the Action menu.

To view the GPO Settings Report, select the GPO in the left pane, and then click the Settings tab in the right pane. GPA displays the report in the right pane.

NOTE: If you make changes to ADMX files in Active Directory on the central store, close and re-open the GPA Console before viewing the GPO Settings Report.

6.14 Backing Up and Restoring GPOs

It is recommended that you back up your GPOs. Consider the following:

- ◆ IT departments can spend many weeks, and sometimes months, creating GPOs and assigning them to Active Directory objects.

- ◆ Your organization may require and create a potentially large number of policies.

- ◆ The Group Policy environment can become complicated quickly.

- ◆ Unless the entire Active Directory can be restored (if it has been backed up), Microsoft Windows has no way to recover the GPO information or the time that was spent creating one or more GPOs.

- ◆ If a GPO is lost through deletion or corruption, it could take several hours to recreate the exact settings.

Consider the following scenarios:

- ◆ Two or more administrators (or delegated users) can simultaneously change the settings of a policy, which can effectively overwrite the changes made by one of the administrators.

◆ Changes to GPO settings and links could have far-reaching consequences if a GPO is corrupted or accidentally deleted.

◆ Many users may be affected by the loss of a GPO, effectively denying them access to their business-critical applications.

6.14.1 Backing Up Single or Multiple Objects

GPA provides you with GPO backup capabilities for one or many objects and provides the ability to restore those objects. To protect your organization from wasted hours of recreating policies, It is recommended that you use these features to back up your policy objects.

For example, once you have determined that GPOs in a test domain work properly, you can use the backup and restore functions to migrate the GPOs to your production domain. This feature is also handy when you need to restore a single GPO that may have become corrupted.

6.14.2 Storing/Backing Up Policy Templates

GPA provides a mechanism to store policy settings offline through templates. With a large number of policy settings, organizations might choose to implement only a subset of the policy settings. The group of policy settings at the domain level could be different from those at the OU level. However, your organization might want to standardize on these groupings across the entire enterprise.

Therefore, you can use the backup function to create company-specific policy templates. You can make any grouping of policy settings constitute a template. Domain policy templates, OU policy templates, desktop policy templates, and server policy templates are typical examples. You can use these templates to standardize policy implementation and share policy information.

6.14.3 Backing Up GPOs

The backup function allows you to back up the following information:

◆ GPO settings

◆ Security information

The same information is available when you restore GPOs.

To create a backup copy of your Group Policy Objects:

1 Create a folder for this backup procedure. Give it a name and location that is easy to remember.

You use this folder later to store the information that GPA needs to back up your policy settings. You can create as many folders for as many backups as you like. For example, you could have a folder named `Development Backup` and another called `Finance Backup`. In this example, a folder called `GPO Backups` is created in a shared archive directory.

2 Select the GPO you want to backup.

3 On the Action menu, click **Back Up**.

4 Specify the location of the backup folder and any appropriate comments.

5 Click **Back Up**.

6 Click **OK** on the information window.

7 Click **OK** on the BackUp GPO window.

6.14.4 Restoring GPOs

You can restore the objects of a backed-up policy to your current domain (single-target restoration) or to the parent domain *and* one or more child domains (multi-target restoration). For example, you are on a domain named `Headquarters.com`. If you restore a GPO that you previously backed up, you perform a single-target restoration. You decide to restore the GPO but also want the `Finance.Headquarters.com` domain, a child of the `Headquarters.com` domain, to have the same GPO. If you restore that GPO to both domains, you perform a multi-target restoration.

NOTE: You can restore GPOs to domains that do not have GPA installed.

The following illustration depicts a multi-target GPO restore.
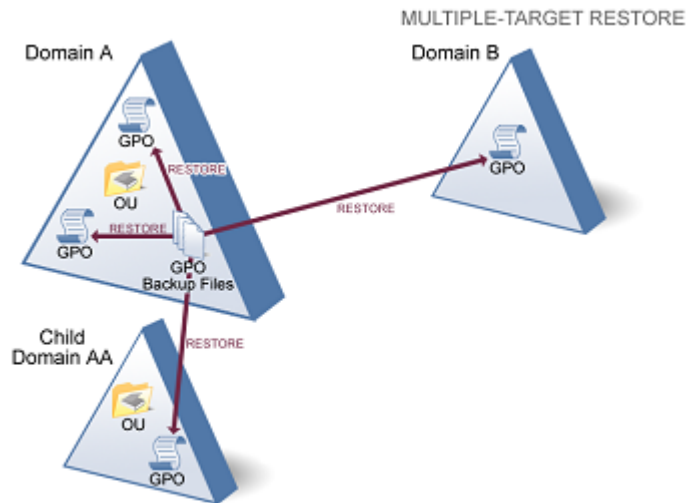
The following illustration depicts a single-target GPO restore.

SINGLE-TARGET RESTORE
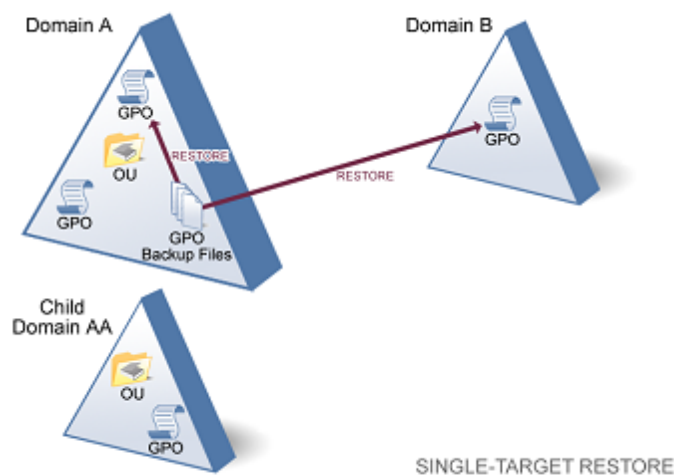
To restore saved GPOs from the same domain:

1 Log on to a GPA Console computer.
2 Start the **GPA Console** in the Group Policy Administrator program group.
3 In the left pane, expand **GP Explorer** to the domain where you want to restore GPOs.
4 In the left pane, click **All GPOs**.
5 On the Action menu, click **Restore**.
6 Complete the Restore Group Policy Object wizard.

7 Reporting on GPOs

Group Policy Administrator (GPA) offers extensive reporting for GPOs in the GP Repository and in Active Directory, including reports that provide the following information:

- GPO Settings Report: provides complete information about policy and preference settings for GPOs stored in the GP Repository or in Active Directory
- GPO Health Check Report: determines data accuracy of GPOs in GP Explorer and the GP Repository and identifies issues
- Group Policy Comparison or Differential Report: compares two GPOs in the same GP Repository, in different GP Repositories, or in the GP Repository and in Active Directory
- Enterprise Consistency Check Report: compares GPOs from multiple domains against a master GPO
- Point in Time Analysis and Activity Report: identifies when and who performed a particular action, such as who approved and exported a GPO
- RSoP Analysis Report: predicts the final result of all the policy and preference settings that apply to a particular user logged on to a particular computer
- RSoP Analysis Comparison Report: compares two RSoP Analysis reports to help you determine the source of differences in settings for different users on different computers
- Group Policy Results Diagnostics Report (Actual RSoP): shows the actual, rather than predicted, policies and preferences that are taking effect for a specific user on a specific computer, and the history of GPOs that have been applied
- GPO Search Report: shows the result of searches for GPOs in the GP Repository and in Active Directory according to the criteria specified and allows you to perform actions, such as starting the GPO Editor or performing workflow operations on repository GPOs

7.1 Built-In Search Reports

GPA provides a set of useful search filters that you can use to quickly generate the following reports:

Available from GP Repository and GP Explorer

- Disabled GPOs Report. Generates a list of GPOs with disabled user and computer configurations.
- Duplicate Named GPOs Report. Generates a list of GPOs with the same name of at least one other GPO.
- Empty GPOs Report. Generates a list of GPOs that have no settings.
- GPOs with no security filtering Report. Generates a list of GPOs that have no security group filter assigned to them.

Available from GP Repository only

- Out of sync GPOs Report. Generates a list of GPOs with modification times that are different from their Active Directory versions. This filter is not available in untrusted domains.
- Unlinked GPOs Report. Generates a list of GPOs that have no links.

The built-in search filters display on the left pane of the GPA Console along with the other OUs and categories. Before generating a report, make sure the appropriate domain is indexed. To generate a report, click the appropriate node and the report displays in the right pane.

7.2 Viewing GPO Setting Information

The GPO Settings Report is an important and useful tool. The GPO Settings Report gives you these abilities:

- Maintains copies of the configured settings for each GPO in your organization as an HTML file.
- Fully documents the GPO as part of a GPO change and release process. This helps you assure compliance with internal policies and regulatory requirements.

7.2.1 Viewing the GPO Settings Report

You can view the GPO Settings Report for a GPO in the GP Repository or in Active Directory.

NOTE

- GPA may take a few moments to generate the GPO Settings Report for the first time.
- If you installed the GPA Console on a computer that supports managing ADMX files, before you generate the GPO Settings report for the first time for a GPO in the GP Repository, synchronize the ADMX and ADML files from the central store. If you do not synchronize the files, the GPO Settings report cannot display the administrative template policies. For more information about synchronizing the files, see Section 5.11.3, "Synchronizing or Exporting ADMX Files from the Central Store," on page 99.

- If you make changes to ADMX files, close and re-open the GPA Console before viewing the GPO Settings Report.

- When using multiple domain controllers (DCs), if you generate a settings report, the GPO Console creates a container in the primary domain controller (PDC) to create the GPO Settings Report. The other DCs on the domain are not contacted when this report is generated.

To view the GPO Settings Report:

1 In the left pane, expand **GP Repository** or **GP Explorer**.

2 Expand the appropriate domain hierarchy to the GPO for which you want to generate the report.

3 In the right pane, click the Settings tab.

7.2.2 GPO Settings Report Layout

The GPO Settings Report contains the same sections as GPMC reports. The GPO Settings Report is divided into the following main sections:

Details

This section contains information about the GPO itself, such as the GPO owner, its GUID, the GP Repository version, GPO source, and its status. The Details section also contains information about the change history of the GPO, including when it was created, when it was last modified, and how many revisions have been made to the settings contained in the GPO. This field is only available for GPOs in the GP Repository.

Links

This section contains the following information:

- The name of the Active Directory objects (containers) to which the GPO is linked
- The GPO-specific properties of the container (Enforced and Link Status)
- The path to the Active Directory containers to which the GPO is linked

Security Filtering

This section contains a list of the users, groups, and computers to which the GPO applies.

WMI Filtering

This section contains the name and description of the WMI filter associated with the GPO.

Delegation

This section contains the following summary information:

- A list of security groups and users who have security permissions for the GPO
- The security permissions set on each user or group, such as Read, Edit settings, Delete, and Modify security

GPO Settings

This section provides several subsections that summarize the GPO settings. The subsections vary depending on if the GPA Console is installed on a computer supporting ADMX files or Group Policy Preferences.

7.3   Analyzing GPO Security and Structural Health

The GPO Health Check Report provides information about whether the security filtering, GPO data, security policies, and GPO structural integrity in a GPO are accurate or corrupted. This report also helps you to identify failures in the File Replication Service (FRS) on domain controllers by comparing the GPO template with the GPO container.

7.3.1   Viewing the GPO Health Check Report

You can view the GPO Health Check Report for a GPO in the GP Repository or in Active Directory.

To view the GPO Health Check Report:

1   In the left pane, expand **GP Repository** or **GP Explorer**.

2   Expand the appropriate domain hierarchy to the GPO for which you want to generate the report.

3   In the right pane, click the Health Check tab.

7.3.2   GPO Health Check Report Layout

The Health Check Report is divided into the following main sections:

General

This section contains information about the GPO, such as the GPO owner, its GUID, and its status. The General section also contains information specific to the GPO in the GP Repository, such as the change history of the GPO, including when it was created, when it was last modified, and how many revisions have been made to the settings contained in the GPO.

GPO Health Check Results

This section contains the following information:

- Status on whether the GPO display name is correct or wrong and also whether the GPO revisions are consistent

- Status on the integrity of the GPO file system and the GPO Active Directory data

- Status on whether the GPO is applied to users or computers and also whether the inheritance flag is set on the GPO security descriptor

- Status on the integrity checks for file security, registry security, and services security

You can use this information to help identify corrupt GPOs in your Active Directory environment.

7.4 Comparing and Differentiating GPOs

Determining the similarities and differences between GPOs is often necessary as part of the GPO change and release process. GPA allows you to compare:

Two versions of the same GPO within the GP Repository

Useful for determining what changed from a prior version of the same GPO. For more information, see "Reporting on Two Versions of the Same GPO in the GP Repository" on page 122.

Two distinct GPOs from the same or different GP Repositories

Useful for determining the similarities or differences between two GPOs in the same or two different GP Repositories. You can use either the GP Repository node or the GP Analysis node to make this comparison. For more information, see "Reporting on Two Different GPOs in the GP Repository" on page 122.

A GPO in the GP Repository with the version of the same GPO in Active Directory

Useful for determining if changes have been made in Active Directory to the approved version of a GPO. You can use either the GP Repository node or the GP Analysis node to make this comparison. For more information, see "Reporting on a GPO in the GP Repository and the Same GPO in Active Directory" on page 124.

A GPO in the GP Repository and a different GPO in Active Directory

Useful for determining the similarities or differences between a GPO in Active Directory and a GPO in the GP Repository. Use the GP Analysis node to make this comparison. For more information, see "Reporting on a GPO in the GP Repository and a Different GPO in Active Directory" on page 123.

Indicators in the Status column report the following information:

Indicator	Meaning
S	Content of GPO setting is the same in both versions
D	Content of GPO setting is different between versions
Z	Content of GPO setting is similar in both versions
L	Only GPO on the left of the report contains information
R	Only GPO on the right of the report contains information

The comparison reports contain the same sections and have a layout similar to the GPO Settings Report. For more information, see Section 7.2, "Viewing GPO Setting Information," on page 118. The steps to produce the Group Policy Comparison Report and the Group Policy Differential Report vary depending on the type of comparison you are making.

7.4.1 Understanding Comparison and Differential Reports

The Group Policy Comparison Report and the Group Policy Differential Report may seem similar. Despite their similarities, they have important differences:

- **Group Policy Comparison Reports** display *all* of the settings in the GPOs being compared and shows which are the same and which are different.
- **Group Policy Differential Reports** display *only the settings that are different* between the GPOs.

Reporting on Two Versions of the Same GPO in the GP Repository

In many cases, you may find it necessary to look at reports on two versions of the same GPO in the GP Repository.

To report on two versions of a GPO within the GP Repository:

1 In the left pane, expand **GP Repository**.

2 Select the domain containing the GPO.

3 Select the appropriate GPO.

4 On the Action menu, click **View History**.

5 Select one of the versions to be compared and then press **Ctrl** while selecting the second version.

6 Click **Compare** for a Group Policy Comparison Report or click **Differentiate** for a Group Policy Differential Report.

Reporting on Two Different GPOs in the GP Repository

Using the GP Repository node or GP Analysis node, you can compare the differences or similarities between two different GPOs from the same GP Repository or two different GP Repositories.

From the GP Repository

You can use the GP Repository node to run the report.

To report on two different GPOs from the same or different GP Repositories using the GP Repository node:

1 In the left pane, expand **GP Repository**.

2 Select the domain containing the GPOs you want to compare.

3 On the Action menu, click **Compare GPOs**.

4 In the left pane, select the first GPO to compare.

5 In the right pane, select the second GPO to compare, either from the same GP Repository or from a different GP Repository.

6 Click **Compare** for a Group Policy Comparison Report or click **Differentiate** for a Group Policy Differential Report.

With the GP Analysis Node

You can use the GP Analysis node to run the report.

To report on two different GPOs from the same or different GP Repositories using the GP Analysis node:

1. In the left pane, expand **GP Analysis** and click **GPO Comparison**.
2. On the Action menu, click **GPO Comparison Wizard**.
3. Read the introduction text, and then click **Next**.
4. *If you want to run a differential report*, select **Difference Report**.
5. Under **Select First GPO**, click **Browse**.
6. Select **Repository**, and then click **OK**.
7. Specify the GP Repository containing the first GPO to compare in the **SQL Server** field and the credentials to connect to the GP Repository, and then click **OK**.
8. Expand the domain and category, and then select the first GPO to compare.
9. Click **OK**.
10. Under **Select Second GPO**, click **Browse**.
11. Select **Repository**, and then click **OK**.
12. Specify the GP Repository containing the second GPO to compare in the **SQL Server** field and the credentials to connect to the GP Repository, and then click **OK**.
13. Expand the domain and category, and then select the second GPO to compare.
14. Click **OK**.
15. Click **Next**, and then click **Finish**.

Reporting on a GPO in the GP Repository and a Different GPO in Active Directory

This report is useful for determining the similarities or differences between a GPO in Active Directory and a GPO in the GP Repository. You can run this report on the same GPO, or on a different GPO.

To report on a GPO in Active Directory and a GPO in the GP Repository:

1. In the left pane, expand **GP Analysis** and click **GPO Comparison**.
2. On the Action menu, click **GPO Comparison Wizard**.
3. Read the introduction text, and then click **Next**.
4. *If you want to run a differential report*, select **Difference Report**.
5. Under **Select First GPO**, click **Browse**.
6. Select **Repository**.
7. Specify the GP Repository containing the GPO to compare in the **SQL Server** field and the credentials to connect to the GP Repository, and then click **OK**.
8. Expand the domain and category, and then click the GPO in the GP Repository to compare.
9. Click **OK**.
10. Under **Select Second GPO**, click **Browse**.

11 Select **Active Directory of a specific domain**.

12 Click **Browse** to select a domain, and then click **OK**.

13 Click **OK**.

14 Select the GPO in Active Directory to compare, and then click **OK**.

15 Click **Next**, and then click **Finish**.

Reporting on a GPO in the GP Repository and the Same GPO in Active Directory

This report is useful for determining whether the Active Directory version of a GPO still matches the GP Repository version.

NOTE

- You can also make this comparison using the GP Analysis node. For more information, see "Reporting on a GPO in the GP Repository and a Different GPO in Active Directory" on page 123.

- The GPA console disables the Compare AD version and Differentiate AD version options in the Context, Action, and toolbar menus for GPOs in untrusted domains in the GP Repository. However, you can generate these reports from a GPA console installed on a computer joined to the same untrusted domain.

To report on a GPO with versions in the GP Repository and in Active Directory:

1 In the left pane, expand **GP Repository**.

2 Expand the domain containing the GPO.

3 Expand the category containing the GPO, and then click the GPO.

4 On the Action menu, click **Compare AD version** or **Differentiate AD version**.

7.5　Analyzing Multi-Domain GPOs against a Master GPO

Maintaining GPO consistency in your enterprise is key in keeping your business functioning effectively. The Enterprise Consistency Check report lets you compare GPOs from multiple domains against a master GPO. This report is useful in determining whether a GPO deployed throughout your Active Directory is still consistent with the original GPO. For example, you may be enforcing the same password policy throughout the enterprise with the deployment of a password policy GPO to every domain in your Active Directory. Over time, these copies of the original GPO may no longer match the original policy. The Enterprise Consistency Check report identifies any GPOs that are no longer consistent with the original GPO. For more information about generating the Enterprise Consistency Check report, see Section 7.5.1, "Running the Enterprise Consistency Check Report with the Wizard," on page 125.

Used in conjunction with GPO synchronization, the Enterprise Consistency Check report helps you maintain consistency with your GPOs in Active Directory. By comparing the master GPOs in Active Directory or the GP Repository with the controlled GPOs in the GP Repository, you can quickly identify any controlled GPOs that are no longer synchronized with their master GPOs. For more information about GPO synchronization, see Section 5.9, "Synchronizing GPOs," on page 92.

7.5.1 Running the Enterprise Consistency Check Report with the Wizard

You can run the Enterprise Consistency Check wizard to generate the Enterprise Consistency Check report if you have only a few GPOs to compare. If you have many GPOs to compare, consider running the report with the `NqGpoCompare.exe` command-line utility. For more information, see Section A.4.3, "Generating the Enterprise Consistency Check Report Using Scripts," on page 156.

In addition to generating the report, the Enterprise Consistency Check wizard creates or updates an `.xml` report configuration file that both the wizard and the `NqGpoCompare.exe` command-line utility require to create the Enterprise Consistency Check report.

To create the Enterprise Consistency Check report using the Enterprise Consistency Check wizard:

1 Log on to a GPA Console computer with an account that has domain administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, click **GP Analysis**.

4 In the right pane, click **Run Enterprise Consistency Check**.

> **NOTE:** The GPA console disables the Run Enterprise Consistency Check option for GPOs in untrusted domains in the GP Repository. However, you can generate these reports from a GPA console installed on a computer in the untrusted domain.

5 Read the introduction text, and then click **Next**.

6 Choose whether to compare GPOs from the GP Repository or Active Directory or load an existing configuration file. If this is the first time you are running the wizard, choose one of the first two options.

> **NOTE:** If you compare GPOs from your Active Directory, you can only select GPOs from trusted domains. If you compare GPOs from the GP Repository, you can select GPOs from both trusted and untrusted domains.

7 Click **Next**.

8 *If you chose to compare GPOs in the GP Repository*, specify the credentials to connect to the GP Repository, and then click **OK**.

9 Click **Browse** to select a master domain in the **Master Domain (DNS)** field. The domain you select should contain the master GPOs you want to use as the basis for comparison.

10 Click **Add** to select the domain or domains that contain the GPOs you want to compare to the master GPOs, and then click **Next**.

11 Click **Add** to open the Repository GPO Browser window.

12 Select the GPOs you want to use as master GPOs, and then click **OK**. By default, the wizard compares the master GPO to any GPOs with the same name as the master GPO in all comparison domains.

13 *If you want to compare a master GPO with only a subset of the comparison domains*, perform the following steps:

    13a Select a master GPO, and then click **Edit**.

    13b Clear the check box next to any domains you want to exclude from the comparison, and then click **OK**.

14 *If you cannot compare a master GPO to another GPO using the GPO name*, perform the following steps:

    14a Select a master GPO, and then click **Edit**.

    14b Select the domain containing the comparison GPO you cannot compare by name, and then click **Edit**.

    14c Expand **GP Repository** to the location of the GPO you want to compare to the master GPO.

    14d Click the comparison GPO, and then click **OK**. The configuration file now uses the GUID of the comparison GPO instead of the name.

15 *If you want to compare the GP Repository and Active Directory versions of each master GPO*, select this option at the bottom of the window. This comparison helps to prevent errors when synchronizing any comparison GPOs to master GPOs.

16 When you have added and mapped all the GPOs you want to compare, click **Next**.

17 Specify or browse to the location and file name of the report configuration file you want to create or update. You need to specify the file extension as `.xml`.

18 Select the **Run a report at the completion of this wizard** check box.

19 Specify or browse to the location and file name of the report file you want to create. You must specify the file extension as `.htm` or `.html`.

20 Click **Next**.

21 Review the Summary window to ensure you have properly specified the master GPO and comparison GPOs, and then click **Finish**.

22 If you want to display the Enterprise Consistency Check, navigate to the folder where you stored the HTML file and open the file in a web browser.

7.6   Analyzing GPO Activity

Auditing user actions is among the most important aspects of a sound security implementation. The Point in Time Analysis and Activity Report helps you identify when a particular action was performed and who performed it. You can use this report to track changes related to GPO approval and GPO export. This report also allows you to identify the changes between two versions of the same GPO.

7.6.1   Running the Activity Report

The GPA Console allows you to generate a GPO activity tracking report as an HTML file. When you are connected to a GP Repository, you can run the Point in Time Analysis and Activity Report using various criteria:

- Domain
- All users or a specific user

- ◆ Type of activity
- ◆ Various date ranges

To run an Activity Report from the GPA Console, select a GP Repository, and on the **Action** menu, click **Activity Report**.

7.6.2 Running the Activity Report Command-line Tool

GPA provides a command-line tool, `ActivityReport.exe,` to generate Point in Time Analysis and Activity Reports as HTML files. Using or scheduling this utility to run periodically lets you quickly create audit-tracking records of GPO changes. For more information, see Section A.4.4, "Generating the Point in Time Analysis and Activity Report Using Scripts," on page 158.

7.6.3 Activity Report Layout

The Point in Time Analysis and Activity Report is divided into the following main sections:

Criteria

This section contains information related to the criteria you selected to generate the report.

GPO Export Summary

This section contains the following information:

- ◆ The name of the GPO
- ◆ The revision history of the GPO
- ◆ The date on which the revision was done
- ◆ The name of the user who exported the GPO
- ◆ Comments added by the user
- ◆ Description of the changes made to the GPO

This information helps you in auditing the changes made to a GPO.

7.7 Viewing Planned User Policies (RSoP)

When many different GPOs are assigned to a user or a computer, you might have difficulty predicting what policy or preference settings ultimately apply.

To determine what settings apply, run an RSoP Analysis report. An RSoP Analysis report predicts the final result of all the settings that apply to a particular user logged on to a particular computer.

Active Directory assigns GPOs to users and computers. Some of the settings in each GPO may conflict with one another. For example, one setting may enable **Remove Desktop icons** while another setting may disable the same item. GPA uses a complex algorithm to arrive at the RSoP for a particular user on a particular computer. This algorithm uses the SDOU (Site, Domain, Organizational Unit) hierarchy to evaluate policy.

NOTE: Link order also affects an RSoP. If you configure GPA to retain the existing Active Directory link order, the RSoP Analysis report predicts the RSoP based on the link order in Active Directory, not in the GP Repository. You can change this option on the Customize Options window in the GP Repository database properties.

The following figure illustrates two sites that belong to one domain as well as a number of associated GPOs.
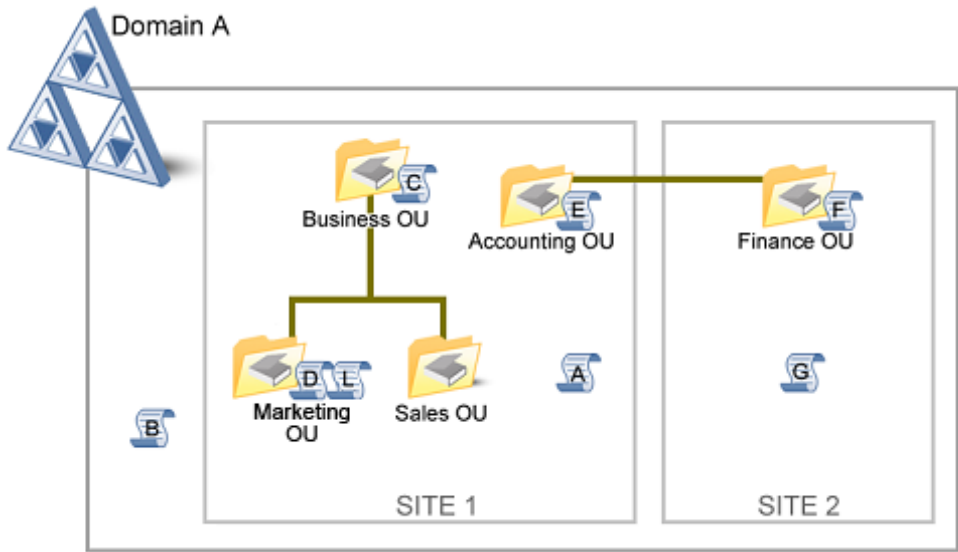
The policies or preferences are defined as follows:

  ◆ A and G are site-level.

  ◆ B is domain-level.

  ◆ C, D, E, F, and L are OU-level.

The following table shows the results of two different RSoP analyses using the SDOU hierarchy in the previous diagram.

OU in which user resides	OU in which computer resides	RSoP: S + D + (OUs)
Marketing	Marketing	A + B + (C + D + L)
Finance	Finance	G + B + (E + F)

The RSoP column of the table shows how the policies or preferences apply, in the order of the SDOU hierarchy. Each level of the hierarchy adds to the next, including GPOs from nested OUs. If any of the GPOs has either a *Block Inheritance* or *Enforced* setting, the algorithm processes additional rules to arrive at the RSoP. For more information, see the Microsoft documentation.

7.7.1 What-If Scenarios in RSoP Analysis Reports

You can run a simple RSoP Analysis report or use what-if criteria when running the report to simulate a certain scenario.

Simple RSoP Analysis Scenario

When you want to find the effective RSoP from domain and OU policy or preference settings, select a user and a computer from either the same or different domains.

RSoP with What-if Criteria Example Scenario

What-if criteria allows you to deploy a GPO hypothetically rather than to actually implement a new or modified policy or preference without knowing it might affect. For example, if you are planning to deploy a GPO that defines a new corporate email policy, the ability to run a what-if scenario helps you ensure the deployment of one GPO concerning email does not have a detrimental impact on all your users.

To determine the result of deploying a GPO under different conditions, use what-if criteria in the RSoP Analysis report. The what-if criteria helps you simulate what would actually occur in Active Directory. For example, you can simulate the following what-if scenarios:

- What if the user or computer is moved to a different OU?
- What if the user or computer is removed from a security group?
- What if the user or computer is added to a different security group?
- What if the computer is moved to another site?
- What if the GP Repository version of the GPO is exported to Active Directory?
- What if the GP Repository version of the GPO is exported and its link order is modified?
- What if you perform the analysis ignoring the existing loop back mode?
- What if you perform loop back analysis in Replace mode?
- What if you perform loop back analysis in Merge mode?

In each case, GPA calculates and reports the resultant effect of policies and preferences on the computer and user combination without actually making changes in Active Directory.

NOTE

- By default, the RSoP Analysis report does not analyze local and site-level policies and preferences. You can analyze site-level policies and preferences by including the *What if this computer is moved to another site?* scenario.
- If you are creating an RSoP Analysis report based on the *What if the GP Repository version of the GPO is exported to Active Directory?* scenario, only those links that belong to the domain for the user and computer you select are analyzed for block inheritance and link order. The report does not analyze link order and block inheritance settings of other domains. Also, the user and computer you select should be in the same domain to yield accurate results.

7.7.2   Running an RSoP Analysis Report

When you run an RSoP Analysis report, you select the user and computer for which you want to perform the analysis. You have the option to select a user and a computer from different domains and also simulate WMI filter results. You also have the option to incorporate what-if scenarios such as selecting or ignoring site-level policies and preferences for a computer when processing the RSoP Analysis report.

NOTE: When you run an RSoP Analysis report using the *What if the GP Repository version of the GPO is exported to Active Directory?* scenario on a GPO in the GP Repository, GPA always simulates the WMI filter results as `True` for this GPO.

To run an RSoP Analysis report:

1 Log on to a GPA Console computer with an account that is a member of the Domain Administrators or Enterprise Administrators group. The account also needs read access to all GPOs and SDOU hierarchies that are included in the RSoP analysis.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Analysis**, and then click **RSoP Analysis**.

4 On the Action menu, select **RSoP Wizard**.

5 Follow the instructions on the wizard until you have finished the RSoP analysis and created an RSoP analysis report.

7.8   Comparing or Differentiating RSoP Analysis Reports

GPA enables you to run a comparison or difference report for two RSoP analysis reports. This is similar to running a comparison or difference report for two GPOs, except you are comparing the results of two RSoP analysis reports rather than the settings in two different GPOs.

Comparing two RSoP analysis reports is very useful in determining the source of differences in settings for different users on different computers. For example, if two different users logging on to the same computer appear to have different settings applied, you can easily determine what these differences are by running an RSoP analysis for each user on the computer in question and then comparing the two RSoP analysis reports.

A similar scenario might occur when the same user logs on to two different computers. By running an RSoP analysis for the same user on each computer and then comparing the two RSoP analysis reports, you can identify what differences there are in the settings.

7.8.1   Running an RSoP Analysis Comparison Report

You create an RSoP Analysis Comparison report by running the RSoP Comparison wizard.

NOTE: If you have RSoP Analysis reports from a previous version of GPA, ensure you regenerate the reports before comparing them. If you select any RSoP Analysis report from a previous version of GPA without regenerating the report, GPA displays an error.

To run an RSoP Analysis Comparison report:

1 Log on to a GPA Console computer with an account that has domain administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Analysis**, and then click **RSoP Report Comparison**.

4 On the Action menu, click **RSoP Comparison Wizard**.

5 Follow the instructions in the wizard until you have finished the RSoP comparison and created an RSoP comparison report.

7.9 Analyzing GPO Infrastructure Status

The GPO Infrastructure Status report provides information about whether the domains and GPOs in a domain are synchronizing consistently. This is a very useful report to run before performing any other tasks involving a GPO or domain.

7.9.1 Generating the GPO Infrastructure Status Report

To generate the GPO Infrastructure Status report:

1 Log on to a GPA Console computer with an account that has domain user and Active Directory read access permissions.

2 In the GP Explorer, select the domain or GPO for which you want to generate a report.

3 Click the Infrastructure Status tab on the right pane.

4 *If you want to change the baseline domain controller that will be used as the reference domain controller for generating the report*, click Change DC.

5 Click Get Status on the right pane.

7.9.2 Checking the Results of the GPO Infrastructure Status Report

To validate the synchronization consistency of the GPOs, GPA compares the Group Policy information stored in Active Directory, and it separately compares the Group Policy information that is stored in SYSVOL on all the domain controllers in the selected domain.

GPA performs the following comparisons:

- Report AD Creation Date Details
- Report AD Modification Date Details
- Report AD ACL Details
- Report AD Version Details
- Report AD GPO Count Details
- Report Sysvol Creation Date Details
- Report Sysvol Modification Date Details
- Report Sysvol ACL Details
- Report Sysvol Version Details
- Report Sysvol GPO Count Details
- Report Sysvol Hash Details

During the report generation process, if GPA cannot contact a GPO, or if a GPO is found to not be consistent, the GPO in question is added to the Domain controller(s) NOT in Sync list under the Status Details section of the Infrastructure status tab.

7.10 Viewing the GPO Security Filter Settings for a GPO

You can generate a report that lists all of the security filters—masks and locks—that have been set on a GPO.

To generate and save a GPO Security Filtering report:

1 Log on to a GPA Console computer with an account that has permissions to filter GPOs.

2 Start the GPA Console in the Group Policy Administrator program group.

3 In the left pane, expand GP Repository to the domain, category, or GPO level, depending on your intent, and right-click the object to which you want to apply the filter.

4 Select Save GPO Security Filtering report.

5 Browse to a save location and click OK.

7.11 Viewing Group Policy and Preference Settings for a User

GPA provides you with diagnostics that list the set of policies and preferences for a specific user on a specific computer.

◆ Diagnostics show the policies and preferences that are actually taking effect, as opposed to RSoP Analysis, which predicts what policies or preferences would take effect.

◆ Diagnostics show the history of GPOs that have been applied. Network and Help Desk administrators can use this functionality to understand why a particular setting is not working correctly.

◆ Diagnostics are run on individual computers, not on the GPA Console computer. They are particularly beneficial to the Help Desk in supporting policy management, because they can quickly track down the settings assigned to a user.

7.11.1 Understanding the Difference Between Predicted and Actual RSoP

GPA acquires planned RSoP information with the RSoP Analysis node, which predicts the effective policy or preference that would be applied to a user when that user logs on to a specific computer.

GPA retrieves actual RSoP information with the Diagnostics node, which connects to the WMI data store on a remote computer and gets the information about the actual settings for the affected user and computer.

The difference between these GPO lists and their corresponding resultant settings provides valuable information for troubleshooting issues.

7.11.2 Running the Group Policy Results Diagnostics Report

The Group Policy Results Diagnostics Report is available in the GP Analysis node.

To run a Group Policy Results Diagnostics Report:

1 Log on to a GPA Console computer with an account that has domain administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Analysis**, and then click **Diagnostics**.

4 On the Action menu, click **GP Diagnostics Wizard**.

5 Read the introduction text, and then click **Next**.

6 Browse and select the computer you want to analyze, and then click **Next**.

NOTE: The computer you select should support the RSoP logging mode. For more information, see the Microsoft documentation.

---

7 Select the user from the list, and then click **Next**.

8 Review the Summary window, and then click **Finish**.

7.12 Renaming Reports

You can rename reports you create using GPA. Renaming reports can be useful to help clearly distinguish one report from another. For example, when you run an RSoP Analysis report, GPA creates a new report under **RSoP Analysis** following the naming convention of *User A* at *Machine X*, such as `Administrator at BOS-QAC-DC-01`. If you perform another RSoP analysis with the same user and computer, GPA gives the same name to the second RSoP Analysis report. Having the same name can make it difficult to distinguish one RSoP Analysis report from another. To distinguish between the RSoP Analysis reports, you can rename one or both reports.

To rename a GPA report:

1 Log on to a GPA Console computer with an account that has domain administrator permissions.

2 Start the **GPA Console** in the Group Policy Administrator program group.

3 In the left pane, expand **GP Analysis** and any sub folders containing the report you want to rename.

4 Select the report you want to rename.

5 On the Action menu, click **Rename**.

6 Enter the new report name.

7.13　Searching for GPOs

Your group policy environment can grow extremely large over time, even for a smaller business. GPA features a comprehensive and interactive Search GPO function that can help you locate GPO data in Active Directory (AD) and in your GP Repository.

The Search GPO function lets you perform an advanced search on multiple criteria. You can search for GPOs based on various criteria, such as user or computer policy settings, GPO information, security group details, GUID information, and other criteria in AD trusted domains. GPA can also search multiple domains at the same time and recent versions of GPOs in the GP Repository. GPA lists this data in the Search Results section of a GPO Search report.

NOTE: GPA determines and populates the available search criteria in the Search window based on the operating system and version on your computer.

When you change the domain for your search, GPA removes any domain-specific search criteria from the search criteria list. Click **Cancel** to keep the current search criteria or **OK** to continue.

When GPA displays the search results, you can do the following:

- Act on GPOs by starting the GPO Editor or performing workflow operations on AD GPOs and GP Repository GPOs.
- Save the GPO Search Results, Settings Report, and Health Check Report in HTML format.
- Rerun the search with the current search criteria.
- Edit the current search criteria.
- Make a copy of the search criteria or rename the search results record you saved. (GPA copies the Report node to another node and adds the prefix "Copy of" to the original report name.)
- Print the search results.
- Delete the current search results.

To search for GPOs:

1 Log on to a GPA Console computer with a domain user account with Local Administrator permissions.
2 Start the GPA Console in the Group Policy Administrator program group.
3 Make sure the GPA Console is connected to a GPA Server. For more information about connecting to a GPA Server, see Section 3.3.3, "Connecting to a GPA Server," on page 50.
4 In the left pane, expand **GP Analysis**, and then select **Search**.
5 Click **Action** > **Search GPO**.
6 Use the **Search Options** to select AD domains and GP Repositories to include in your search.
7 Use the Search Criteria to specify the type of search and what criteria to use.

   When you use the Search Criteria, you:
   - Select the type of search
   - Add specific search criteria using list boxes
8 After specifying the domain and search criteria, click **OK**.

GPA saves the search results and stores each report by date.

9 *If you do not have an existing SQL Server connection in the GP Repository*, enter the appropriate information in the connection dialog.

To use existing reports to quickly generate the same report or a similar report:

1 In the left pane, expand **GP Analysis**, expand **Search**, and then select the report you want to use.

2 On the Action menu, select **Rerun Search** to generate the same report with updated information based on changes made in AD. GPA uses the same criteria so you can see what has changed since you last ran the report.

3 On the Action menu, select **Edit** to display the Search window with the same search criteria as the selected report.

4 Add or remove criteria to fit your needs.

5 Click **OK**.

8 Uninstalling Group Policy Administrator

Complete the following tasks to uninstall Group Policy Administrator:

1. Close all GPA console instances.
2. It is recommended that you run the GPA installer `setup.exe` and select Uninstall for each of the individual components, GP Repository, GPA Server or GPA Console installed on one or more of the machines.

   NOTE: Ensure User Account Control (UAC) is set to low if you must use CLI or Control Panel to uninstall GPA.

3. Delete the GPO_REPOSITORY database from SQL Server manually after uninstallation.

9 Troubleshooting

The following sections provides solutions to the problems you might encounter when configuring or using Group Policy Administrator:

9.1 GPA Security Account Is Not Given Any Permissions after a Distributed Installation

Issue: The GPA security account is not given any permissions after a distributed installation when the GPA server is installed.

Workaround: Add the GPA security account as a member of the 'GPA Repository Management' group and restart the 'GPA Server' service.

9.2 GPA Closes Unexpectedly If You Click the Manage GPO Changes Offline Link

Issue: If, without closing the GPA console, an administrator grants a user access to the GP Repository node after no access to any nodes was previously granted, the MMC will stop unexpectedly if the user clicks the **Manage GPO Changes Offline** link.

Workaround: Close the GPA console and re-open it immediately after granting access to the GP repository node.

9.3 Cannot Choose Target Object When Migrating a GPO

Issue: You cannot choose a target object when migrating a GPO if the target object is specified in the FQDN\user format.

Workaround: Use the NetBIOS domain\user format to specify the target object.

9.4 Category May Have Incorrect FAGPR Path

Issue: When a category has "=" in its name, the FAGPR path of that category and its child categories and GPOs will be incorrect with the parent category's name missing from the path.

Workaround: Remove the "=" from the name of the category.

9.5 Web Installer Fails to Install Some Files

Issue: The web installer will not install the msxml6_x86.msi file if the folder to which the installation files are extracted is not empty.

Workaround #1: Click **Retry** three times.

Workaround #2: Abort the installation and delete all of the files from the PreRequisites folder before restarting the web installer.

9.6 GPAPolicyDefinitions Folder Receives Wrong Permissions If You Install GPA to a Non-system Drive

Issue: If you install GPA to any drive other then the C: drive, the Authenticated Users group will not be included in the GPAPolicyDefinitions folder.

Workaround: Install GPA on the C: drive.

9.7 Analysis Node Fails to Save SQL Name in a Distributed Environment

Issue: In a distributed environment, analysis node operations such as the RSoP Wizard, ECC Report, and the GPO comparison tool fail to save the SQL name, defaulting to the console server name.

Workaround: Manually enter the correct SQL Server name hosting the GPA Repository.

9.8 Attempting to Assign an IP Security Policy Results in an Error

Issue: When attempting to assign an IP Security Policy, the following error results: `The following error occurred when saving IP security data: No such interface supported.` This error can occur from either the GP Repository or the GP Explorer.

Workaround: Edit the GPO in Group Policy Management Console and assign an IP Security policy.

9.9 Faulty Schedule Export Wizard

Issue: The Schedule Export wizard will not let you pick an export account from a non-local domain.

Workaround: Use an export account from a local domain to complete the Schedule Export wizard and then change the account to a non-local domain account in the task from the Task Scheduler.

9.10 Not All Settings Display on the GPA Console Merge GPO Window in a Distributed Environment

Issue: Only the Registry Based settings display on the GPA console's Merge GPO window in a distributed environment; however, all of the settings display in the Merge GPO window in a consolidated environment.

Workaround:

1 Start the Group Policy Management Console (GPMC) on the GPA server using the GPA security account.

2 Browse to each source GPO and generate a settings report.

> **TIP:** If a source GPO is not present in AD—such as a newly created GPO, for instance—then export the GPO to AD and then generate a settings report using GPMC.

3 Log in to the console machine and re-index the repository.

9.11 The GPA Console Slows Down and Becomes Unresponsive after Multiple GPO Merges

Issue: Multiple GPO merge operations on large GPOs increases the GPA Console's memory utilization but the GPA Console does not release the increased memory after the operation completes. Over time, this results in sluggish performance.

Workaround: If you observe sluggish behavior, close and re-open the GPA console.

9.12 GPO Merge Fails Due to Name Conflict

Issue: If the source GPOs have a script with the same name, GPA will append a random 38-character GUID to the name of the script from the secondary GPO. If the script has a long name, this can cause the merge to fail because the length of the directory path to the script cannot exceed 255 characters.

Workaround: Rename one of the conflicting scripts.

9.13 The Wired and Wireless Policy Does Not Appear in the GP Repository Settings Report

Issue: If you generate a GP repository settings report after creating a repository GPO with the wired and wireless policy on a Windows 8 or Windows Server 2012 machine, the wired and wireless policy will not appear in the report.

Workaround: Run the `nltest /dsgetdc:<domain name> /force` command to find the domain controller to which the GPA editor will write the wireless/wired configuration. When the target domain controller is identified, change the domain controller in the GPA Repository domain to the target domain controller and generate the settings report.

9.14 The GP Analysis Node Fails to Appear After Installing GPA

Issue: After installing GPA, the GPA Analysis node fails to appear in the GPA Console tree on the newly installed server.

Workaround

To register the `faGPAnalysis.dll` and the `faGPDiagnostics.dll`:

1 Run the command line application in administrator mode.

2 Change the directiory to *<GPA Install Directory>*\Bin.

3 Enter `regsvr32 faGPAnalysis.dll`.

4 Enter `regsvr32 faGPDiagnostics.dll`.

9.15 GPA Does Not Save The Repository Authorization Code If the User Has No Profile

Issue: If a user with no profile logs in to the GPA Console, GPA displays an error when the user tries to save the GPA repository autorization code.

Workaround: Create a local profile, which is needed to encrypt or decrypt the autorization code, before logging in to the GPA Console. .

9.16 Cannot Create or Display GPO Links from GPA Console Running on Windows Server 2016 or Windows 10

Issue: Under certain circumstances you might not be able to link an existing GPO to an organizational unit from a Windows Server 2016 or a Windows 10 environment, or you might not be able to display the GPO's list of links.

Workaround #1: Use an earlier version of a Windows operating system such as Windows Server 2012 R2 or Windows 8 to create the link or display the GPO's list of links.

Workaround #2:

1 On a Windows Server 2012 R2 machine, go to C:\\Windows\SysWOW64 and copy the `gpedit.dll` file.

2 Paste the file into the *<GPA installation path>*\Bin folder on the Windows 10 or Windows Server 2016 machine.

3 Go back to the C:\\Windows\SysWOW64 folder and locate the appropriate language folder. For instance, if you are running an English language environment, locate the en-US folder.

4 Copy the `gpedit.dll.mui` file from the langauge folder.

5 Go to the *<GPA installation path>*\Bin folder on the Windows 10 or Windows Server 2016 machine and create a folder and give it the appropriate language name. For example, en-US.

6 Paste the `gpedit.dll.mui` file into the language folder.

NOTE: This change will not impact native and other external applications. Only the GPA Console will use the local copy of gpedit.dll while launching the appropriate user interfaces.

A Automating GPA Operations with .NET

To make your policy administration easier, you can schedule Group Policy Administrator (GPA) to perform backups or imports regularly using Microsoft Scheduled Tasks by creating scripts or .NET applications as documented in this section.

GPA has packaged all the functionality associated with policy management and provided it as a COM `.DLL` file, enabling it to be scripted. GPA also supports PowerShell and .NET language applications to automate group policy administration. This feature abstracts the complex ADSI concepts related to Group Policy, which saves you time and effort in coding. The Microsoft Windows Script Host enables you to run scripts directly on any computer running the GPA Console by either:

- Clicking a script file on the Microsoft Windows desktop
- Typing the name of a script file at the command prompt

You can also create .NET applications and schedule them to run automatically. The biggest benefit of this GPA feature for administrators is the simplicity and ease of programming. You can use any text editor, such as Notepad, to create the Visual Basic scripts with minimal programming effort. This section provides information for writing Visual Basic scripts and C# methods.

NOTE

- Because the width of some commands is too long for the width of this page, many of the command lines run over to a second or third line. When you create a script or method, only include line breaks where the extra space appears between paragraphs in this document.
- If you are running GPA on a 64-bit platform, you need to run scripts and .NET applications using a 32-bit command prompt window. On a 64-bit computer, you can access the 32-bit command prompt window from the `%WINDIR%\SysWOW64` folder.

A.1   Backups Script

One of the most important tasks in any software environment is to perform backups. The following information shows you how to write a script or an application to do that task.

To write a Visual Basic script for performing backups:

1 In Notepad, type the following script:

```
dim x
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** BackupPolicy(GPO LDAP path, Backup folder path,
REM ***** Domain, DC, comment, options)
x.BackupPolicy "LDAP://DomainController/
CN={GUID},CN=Policies,CN=System,
DC=Domain","BackupFolder","DomainController","Domain","Comment",
Reserved
```

2 To obtain the LDAP path of the GPO ("LDAP://... DomainName"), use the ADSI Edit tool. If the ADSI Edit tool is not available, substitute the following information for the variables:

DomainController

> Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

GUID

> Type the GUID number that corresponds to the GPO you want to back up.

Domain

> Type the name of the domain to which the GPO belongs.

BackupFolder

> Type the path and the name of the folder to which you are going to back up the data.

Comment

> Type a comment.

Reserved

> Specify 0.

3 When you are finished, your Visual Basic script should look similar to the following example:

```
REM **** BACKUP ******************************************
dim x
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** BackupPolicy(GPO LDAP path, Backup folder path,
REM ***** Domain, DC, comment, options)
x.BackupPolicy "LDAP://chld-dc-01.chld.prnt.net/CN={2FC79BD5-2288-4C37-9BAF-
49E6F8D21A17},CN=Policies,CN=System,DC=chld,DC=prnt,DC=net", "C:\GPO
Backups","chld-dc-01.chld.prnt.net","chld.prnt.net", "Backup from Scripts",0
set x = nothing
Wscript.Echo "Backup Completed!"
REM ****************************************************
```

4 To back up multiple GPOs, copy the `BackupPolicy` line as many times as necessary, modifying only the *GUID* for each GPO.

5 Save the file with any suitable name, and an extension of `.vbs` for Visual Basic scripts, to the location of your choice (for example, `backupGPOs.vbs`).

To write a C# application for performing backups:

1 In Microsoft Visual Studio, create a new project, then select **Visual C#** > **Console Application**.

2 In **Solution Explorer**, select **References**, then right-click on **Reference** and select **Add Reference**.

3 Select the Browse tab, navigate to *GPA install location*/`Tools`, and add the `Interop.FULLARMORGPRSCRIPTOBJLib.dll` assembly.

4 Repeat Step 2, then select the Browse tab, navigate to *GPA install location*/`Bin`, and add the `Interop.fazam2003expobj.dll` assembly.

5 Add the following libraries at the beginning of each application

```
using Interop.FULLARMORGPRSCRIPTOBJLib;
using Interop.fazam2003expobj;
```

NOTE: Repeat steps 1-5 for each C# application you write for GPA operations.

6 Create the following C# method:

```
public static void BackupAD()
        {
            string sGPOPath = "LDAP://DomainController/
CN={GUID},CN=Policies,CN=System,DC=Domain";
            string sPath = "BackupFolder";
            string sDomainController = "DomainController";
            string sDomain = "Domain";
            string sComment = "Comment";
            long sReserved = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.BackupPolicy(sGPOPath, sPath, sDomain,
sDomainController, sComment, sReserved);
            Console.WriteLine("GPO backed up successfully");
            Console.ReadKey();
        }
```

7 To back up multiple GPOs, copy the `sGPOPath` line as many times as necessary, modifying only the *GUID* for each GPO.

NOTE: Add the `[STAThread]` attribute before the main method, which indicates that the COM threading model for the application is single-threaded apartment.

8 Your C# method should look similar to the following example:

```csharp
public static void BackupAD()
        {
            string sGPOPath = "LDAP://MYDOMAIN.LAB/CN={04835D24-7FAC-4B7B-B677-
419E598593B0},CN=Policies,CN=System,DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/";
            string sDomainController = "MYDOMAINCONTROLLER.MYDOMAIN.LAB";
            string sDomain = "MYDOMAIN.LAB";
            string sComment = "Backup from Scripts";
            long sReserved = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
           oEXPRroot.BackupPolicy(sGPOPath, sPath, sDomain, sDomainController,
sComment, sReserved);
            Console.WriteLine("GPO backed up successfully");
            Console.ReadKey();
        }
```

9 Compile the solution, then execute the executable build.

A.2 Imports Script

After you have used GPA to back up a GPO, you have several ways to return the GPO to the Active Directory environment:

- ◆ Use the Restore feature, in which you can restore GPO settings, Active Directory links, and security settings individually or all together.
- ◆ Use the Import feature, in which you can import only the GPO settings of a backed up GPO into an existing GPO.

The key difference between these two methods is that importing a GPO requires that another GPO already exist, whereas restoring a GPO does not require that another GPO be present in the Active Directory environment.

To write a Visual Basic script or method for importing backed up policies:

1 In Notepad, create the following script:

```
dim x , a
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** RestorePolicy(GPO LDAP path, Backup folder path,
REM ***** Domain, DestDC, options)
x.RestorePolicy "LDAP://DomainController/
CN={ImportGUID},CN=Policies,CN=System,DC=Domain","BackupFolder\{BackupGUID}","
Domain","DomainController", Options
set x = nothing
```

or

Create the following C# method:

```
public static void Import()
        {
              string sGPOPath = "LDAP://DomainController/
CN={ImportGUID},CN=Policies,CN=System,DC=Domain";
              string sPath = "BackupFolder/{BackupGUID}";
              string sDomainController = "DomainController";
              string sDomain = "Domain";
              long sReserved = 0;
              IfaExplorerRoot oEXPRroot = new PolicyManager();
              oEXPRroot.RestorePolicy(sGPOPath, sPath, sDomain,
sDomainController, sReserved);
              Console.WriteLine("GPO imported successfully");
              Console.ReadKey();
        }
```

2 Modify the variables as follows:

DomainController

> Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

ImportGUID

> Type the GUID number that corresponds to the GPO into which you want to import the backed up GPO.

Domain

> Type the name of the domain to which the GPO belongs.

BackupFolder

> Type the path and the name of the backup folder from which you are going to import the GPO.

BackupGUID

> Type the GUID number that corresponds to the GPO you want to import.

Options

> Specify 0.

3 When you are finished, your script should look similar to the following example:

```
REM ****** Import (use RestorePolicy) ***********************
dim x , a
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** RestorePolicy(GPO LDAP path, Backup folder path,
REM ***** Domain, DestDC, options)
x.RestorePolicy "LDAP://chld-dc-01.chld.prnt.net/CN={FC53832A-7AB2-4100-A971-
C3AEB48A645D},CN=Policies,CN=System,DC=CHLD,DC=PRNT,DC=net", "C:\GPO
Backups\{79ED2956-3F25-43B4-AD8C-513CB850731A}", "chld.prnt.net","chld-dc-
01.chld.prnt.net",0
set x = nothing
Wscript.Echo "Import Completed"
REM ******************************************************
```

or

Your method should look similar to the following example:

```
public static void Import()
        {
            string sGPOPath = "LDAP://MYDOMAIN.LAB/CN={04835D24-7FAC-4B7B-B677-
419E598593B0},CN=Policies,CN=System,DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/{AA968F38-B89D-4688-B766-4B7A4D5CD5C2}/
";
            string sDomainController = "MYDOMAINCONTROLLER.MYDOMAIN.LAB";
            string sDomain = "MYDOMAIN.LAB";
            long sReserved = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.RestorePolicy(sGPOPath, sPath, sDomain,
sDomainController, sReserved);
            Console.WriteLine("GPO imported successfully");
            Console.ReadKey();
        }
```

4 To import multiple GPOs, copy the `x.RestorePolicy` line, for Visual Basic scripts, or the `sGPOPath` line, for C# methods, as many times as necessary, modifying only the *ImportGUID* and *BackupGUID*.

5 Save the file with any suitable name, and an extension of `.vbs` for Visual Basic scripts, to the location of your choice (for example, `importGPOs.vbs`), or compile the solution, then execute the executable build.

A.3   Restore Script

Use the Restore feature to restore GPO settings, Active Directory links, and security settings individually or all together.

Restoring a GPO does not require that another GPO be present in the Active Directory environment.

To write a Visual Basic script or method for restoring backed up policies:

1 In Notepad, create the following Visual Basic script:

```
dim x, a
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** RestorePolicy(NULL, Backup folder path, Domain Name, DC Name, option)
x.RestorePolicy NULL, "BackupFolder\{BackupGUID}",
"Domain","DomainController", Option
set x = nothing
```

or

Create the following C# method:

```
public static void Restore()
        {
            string sPath = "BackupFolder/{BackupGUID}";
            string sDomainController = "DomainController";
            string sDomain = "Domain";
            long sReserved = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.RestorePolicy(null, sPath, sDomain, sDomainController,
sReserved);
            Console.WriteLine("GPO restored successfully");
            Console.ReadKey();
        }
```

2 Modify the variables as follows:

BackupFolder

Type the path and the name of the backup folder from which you are going to restore the GPO.

BackupGUID

Type the GUID number that corresponds to the GPO you want to restore.

Domain

Type the name of the domain to which you want to restore the GPO.

DomainController

Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

Reserved

Specify 0.

3 When you are finished, your script should look similar to the following example:

```
REM **** Restore ********************************************
dim x , a
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** RestorePolicy(NULL, Backup folder path, Domain Name, DC Name, option)
x.RestorePolicy NULL, "E:\GPO Backups\Backup
Scripts\Backups\{33A49A2C-6A2A-45D1-B7CE-A8AEC3A3C325}",
"child-domain.com", "develchild.child-domain.com", 0
set x = nothing
Wscript.Echo "Restore Completed"
REM **********************************************************
```

or

Your C# method should look similar to the following example:

```
public static void Restore()
        {
            string sPath = "C://Folder/{AA968F38-B89D-4688-B766-4B7A4D5CD5C2}/
";
            string sDomainController = "MYDOMAINCONTROLLER.MYDOMAIN.LAB";
            string sDomain = "MYDOMAIN.LAB";
            long sReserved = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.RestorePolicy(null, sPath, sDomain, sDomainController,
sReserved);
            Console.WriteLine("GPO restored successfully");
            Console.ReadKey();
        }
```

4 To restore multiple GPOs, copy the `x.RestorePolicy` line, for Visual Basic scripts, or the `sPath` line, for C# methods, as many times as necessary, modifying only the *BackupGUID* section.

5 Save the file with any suitable name, and an extension of `.vbs` for Visual Basic scripts, to the location of your choice (for example, `importGPOs.vbs`), or compile the solution, then execute the executable build.

A.4   Report Scripts

With the right script, you can use GPA to generate GPO reports.

A.4.1   Generating the GPO Settings Report Using Scripts or Methods

To write a Visual Basic script or method to generate the GPO Settings report:

1 In Notepad, create the following script:

```
dim x
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** GenerateReport(GPO LDAP path, output file path, options)
x.GenerateReport("LDAP://DomainController/CN={GUID},CN=Policies,
CN=System,DC=DistinguishedDomainName","OutputFile",Options)
set x = nothing
```

or

Create the following C# method:

```
public static void GenerateGPOReportAD()
        {
            string sGPOPath = "LDAP://DomainController/
CN={GUID},CN=Policies,CN=System,DC=DistinguishedDomainName";
            string sPath = "OutputFile";
            long sOptions = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.GenerateReport(sGPOPath, sPath, sOptions);
            Console.WriteLine("AD GPO report generated successfully");
            Console.ReadKey();
        }
```

2 Modify the variables. To obtain the LDAP path of the GPO ("LDAP... DomainName"), use the ADSI Edit tool. If the ADSI Edit tool is not available, substitute the following information:

DomainController

Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

GUID

Type the GUID number that corresponds to the GPO you want to report on.

DistinguishedDomainName

Type the distinguished name format of the domain to which the GPO belongs.

OutputFile

Type the path and the name of the file for the output report.

Options

Type 0 to specify the GPO Settings report.

3 When you are finished, your script should look similar to the following example:

```
REM ********************************************************
dim x
set x = CreateObject("GPExplorer.PolicyManager.1")
REM ***** GenerateReport(GPO LDAP path, output file path, options)
x.GenerateReport "LDAP://devel-child.child-domain.com/CN={31B2F340-016D-11D2-
945F-00C04FB984F9}, CN=Policies,CN=System,DC=child-domain,DC=com","E:\GPO
Backups\Backup Scripts\Reports\Report.htm",0
set x = nothing
Wscript.Echo "Report Generation Completed"
REM ********************************************************
```

or

Your C# method should look similar to the following example:

```
public static void GenerateGPOReportAD()
        {
            string sGPOPath = "LDAP://MYDOMAIN.LAB/CN={04835D24-7FAC-4B7B-B677-
419E598593B0},CN=Policies,CN=System,DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/ADGPOReport.htm";
            long sOptions = 0;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.GenerateReport(sGPOPath, sPath, sOptions);
            Console.WriteLine("AD GPO Report generated successfully");
            Console.ReadKey();
        }
```

4 To generate reports of multiple GPOs, copy the `x.GenerateReport` line, for Visual Basic scripts, and the `sGPOPath` line, for C# methods, as many times as necessary, modifying ONLY the *GUID* for each GPO to include in the report.

5 Save the file with any suitable name, and an extension of `.vbs` for Visual Basic scripts, to the location of your choice (for example, `ReportGPOs.vbs`), or compile the solution, then execute the executable build.

A.4.2   Generating the Health Check Report Using Scripts or Methods

To write a Visual Basic script or method to generate the Health Check report:

1 In Notepad, create the following script:

```
dim x
set x = CreateObject("GPExplorer.PolicyManager.1")
x.ConnectTo ("ConnectionString")
REM ***** GenerateReport(GPO LDAP path, output file path, options)
x.GenerateReport("LDAP://DomainController/CN={GUID},CN=Policies,
CN=System,DC=DistinguishedDomainName","OutputFile",Options)
set x = nothing
```

or

Create the following C# method:

```
public static void GenerateHealthCheckReport()
        {
            string sGPOPath = "LDAP://DomainController/
CN={GUID},CN=Policies,CN=System,DC=DistinguishedDomainName";
            string sPath = "OutputFile";
            long sOptions = 1;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.ConnectTo("ConnectionString");
oEXPRroot.GenerateReport(sGPOPath, sPath, sOptions);
            Console.WriteLine("Health Check report generated successfully");
            Console.ReadKey();
        }
```

2 ***If you are using a script,*** modify the `x.ConnectTo` line.

or

If you are using a method, modify the `oEXPRroot.ConnectTo` line.

For more information about obtaining the *ConnectionString*, see Section 5.2.1, "Connecting to a GP Repository," on page 64.

3 Modify the variables. To obtain the LDAP path of the GPO ("LDAP... DomainName"), use the ADSI Edit tool. If the ADSI Edit tool is not available, substitute the following information:

DomainController

Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

GUID

Type the GUID number that corresponds to the GPO you want to report on.

DistinguishedDomainName

Type the distinguished name format of the domain to which the GPO belongs.

OutputFile

Type the path and the name of the file for the output report.

Options

Type `1` to specify the Health Check report.

4 When you are finished, your script should look similar to the following example:

```
REM ********************************************************
On Error Resume Next
set x = CreateObject("GPExplorer.PolicyManager.1")
x.ConnectTo ("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL
Server Instance
Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
REM ***** GenerateReport(GPO LDAP path, output file path, options)
x.GenerateReport "LDAP://devel-child.child-domain.com/CN={FA586CE5-
9C2C-47BE-86AD-1618E183537D},CN=Policies,CN=System,DC=child-
domain,DC=com","E:\GPO backups\Backup Scripts\Reports\HCReport.htm",1
set x = nothing
Wscript.Echo "Report Generation Completed"
REM ********************************************************
```

or

Your C# method should look similar to the following example:

```
public static void GenerateHealthCheckReport()
        {
            string sGPOPath = "LDAP://MYDOMAIN.LAB/CN={04835D24-7FAC-
4B7B-B677-419E598593B0},CN=Policies,CN=System,DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/HealthCheckReport.htm";
            long sOptions = 1;
            IfaExplorerRoot oEXPRroot = new PolicyManager();
            oEXPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use
Procedure for Prepare=1;Auto Translate=True;Packet
Size=4096;Workstation ID=GPA_SERVER;Use Encryption for Data=False;Tag
with column collation when possible=False");
oEXPRroot.GenerateReport(sGPOPath, sPath, sOptions);
            Console.WriteLine("Health Check Report generated
successfully");
            Console.ReadKey();
        }
```

5 To generate reports of multiple GPOs, copy the `x.GenerateReport` line, for Visual Basic scripts, or the `sGPOPath` line, for C# methods, as many times as necessary, modifying ONLY the *GUID* for each GPO to include in the report.

6 Save the file with any suitable name, and an extension of `.vbs` for Visual Basic scripts, to the location of your choice (for example, `HCReport.vbs`), or compile the solution, then execute the executable build.

A.4.3 Generating the Enterprise Consistency Check Report Using Scripts

The Enterprise Consistency Check report can take time to run. You can run the run the report at a later time by using the `NqGpoCompare.exe` command-line utility with the Microsoft Windows Task Scheduler.

To run the report, create an `.xml` report configuration file with the Enterprise Consistency Check wizard, and then run the report at a time that is convenient for you by using the `NqGpoCompare.exe` command-line utility. You can run the report using either of the following methods:

- Run `NqGpoCompare.exe`, located by default in the `C:\Program Files\NetIQ\Group Policy Administrator\Tools` folder.
- Create a batch file specifying the `NqGpoCompare.exe` file and command-line parameters and schedule the batch file to run at a later time using the Microsoft Windows Task Scheduler.

For more information about using the Microsoft Windows Task Scheduler, see the Microsoft Windows documentation. For more information about creating the `.xml` report configuration file, see Section 7.5.1, "Running the Enterprise Consistency Check Report with the Wizard," on page 125.

Required Permissions

To run the `NqGpoCompare` utility, you must be logged on as a member of the `GPA_REPOSITORY_MANAGEMENT` group. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

Syntax

```
NqGpoCompare.exe {[/Overwrite] [/SQLConnStr:"<SQLConnectionString>" |
/RepServer:<SQLServerName> /RepDB:<RepositoryDBName>
[/SQLUser:<SQLLoginName>] [/SQLPwd:*|<SQLUserPassword>]]
<DomainList.xml> <ReportPath.htm>} | [/?|/Help]
```

To display help for the command, type:

```
NqGpoCompare /Help
```

Options

The following list defines the input parameter options available for the `NqGpoCompare` utility.

Overwrite

Overwrites an existing HTML file or terminates the report if the file already exists.

SQLConnStr

Full SQL Server connection string to the GP Repository, in double quotes. Used instead of the other SQL Server parameters.

RepServer

> Name of the GP Repository Microsoft SQL Server. Not needed if provided in the report configuration `.xml` file and will take precedence over the report configuration file if specified.

RepDB

> Name of the GP Repository database on the Microsoft SQL Server. Not needed if provided in the report configuration `.xml` file and will take precedence over the report configuration file if specified.

SQLUser

> If present, SQL Server authentication is used with this SQL Server logon name.

SQLPwd

> Password for the SQL Server logon account specified with `/SQLUser`. If `/SQLUser` is specified but not `/SQLPwd`, or the specified password is `"*"`, then `NqGpoCompare.exe` prompts for the password on the command line.

DomainList.xml

> Path and filename of the `.xml` report configuration file that specifies the master and comparison GPOs. Use double quotes if the path or filename contains spaces.

ReportPath.htm

> Path and filename of the `.htm` report file. Use double quotes if the path or filename contains spaces.

Help

> Displays the usage statement.

If you want to supply a parameter value that contains spaces, such as the file name `GPO Compare October 2010.html`, place quotation marks around the value. For example, type `"GPO Compare October 2010.html"`.

Example 1

To create an Enterprise Consistency Check report from an `.xml` file, overwriting the report file if it exists, connect to the GP Repository using a SQL Server connection string:

```
NqGpoCompare /overwrite /SQLConnStr:"DRIVER={ODBC Driver 13 for SQL
Server};SERVER="<SQL Server Instance
Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;"
"c:\GPOComparisonReports\October_08.xml"
"c:\GPOComparisonReports\GPO Compare September 2010.html"
```

Example 2

To create a batch file to submit to Microsoft Windows Task Scheduler to create an Enterprise Consistency Check report from an `.xml` file, connect to the GP Repository using SQL Server authentication:

1 Create a text file using a tool such as Notepad.

2 Type the following command in the file:

```
NqGpoCompare /RepServer:mysqlserver /RepDB:GPO_REPOSITORY
/SQLUser:myusername /SQLPwd:mypassword!
"c:\GPOComparisonReports\20100920_30.xml"
"c:\GPOComparisonReports\GPO Compare October 2010.html"
```

3 Save the file.

4 Schedule the batch file to run using Microsoft Windows Task Scheduler. For more information, see the Microsoft Windows documentation.

GPA installs the utility in the following folder by default:

```
\installDir\Tools
```

A.4.4 Generating the Point in Time Analysis and Activity Report Using Scripts

The Point in Time Analysis and Activity Report identifies when a particular action was performed on a GPO and who performed it. Use this report to track changes related to GPO approval and export or to identify the changes between two versions of the same GPO.

To run the Point in Time Analysis and Activity Report, you must be logged in as a member of the `GPA_REPOSITORY_MANAGEMENT` group and have local administrator permissions. For more information about GPA security, see Section 4.1, "Understanding the GPA User Security Model," on page 53.

Using or scheduling this utility to run periodically lets you quickly create audit-tracking records of GPO changes. For more information, see Section A.4.3, "Generating the Enterprise Consistency Check Report Using Scripts," on page 156.

GPA installs the utility in the following folder by default:

```
\installDir\Bin
```

To display help for the command, type:

```
ActivityReport /?
```

Syntax

```
ActivityReport [/file:OutputFileName] [/type:[Export|Approve]]
[/start:StartDate] [/end:EndDate] [/domain:DomainName] [/user:UserName]
[/SQLConnstr:"SQLConnString"]
```

Options

The following list defines the input parameter options available for the `ActivityReport` utility:

/file:OutputFileName

Specifies the path and name of the HTML file in which to store the report results. By default, the utility stores the file in the current folder in the file named `ActivityReport.html`.

/type:[Export|Approve]

Specifies whether to include information about all exported GPOs or to include information only about GPOs approved for export. The default value reports on all exported GPOs.

/start:Start

Specifies the start date for the report in `mm/dd/yyyy` format. See the examples following the list.

/end:End

Specifies the end date for the report in `mm/dd/yyyy` format. See the examples following the list.

/domain:DomainName

Specifies the name of one domain to audit for GPA activity. By default, the utility collects activity about all domains. Specify the domain by DNS name.

/user:UserName

Specifies the name of one user to audit for GPA activity. By default, the utility collects activity about all users. Specify the user by `Domain\User`.

/SQLconnstr:"SQLConnString"

Specifies the SQL Server connection string to use when connecting to the GP Repository.

If you want to supply a parameter value that contains spaces, such as the file name `September Activity Report.html`, place quotation marks around the value. For example, to specify the sample file name, type `"September Activity Report.html"`.

Example 1

To create Point in Time Analysis and Activity Report file named `20100920_30.html` that lists information about all GPOs exported between 9/20/2010 and 9/30/2010:

```
ActivityReport /file:c:\GPA_Reports\20100920_30.html /start:9/20/10
/end:9/30/10
```

Example 2

To create Point in Time Analysis and Activity Report for all GPOs approved for the TestDom1 domain by user Admin1GPA in a month:

```
ActivityReport /file:c:\GPA_Reports\Admin1GPA_Approved092010.html
/type:Approve /start:9/01/10 /end:9/30/10 /domain:TestDom1
/user:Admin1GPA
```

Example 3

To create Point in Time Analysis and Activity Report for all GPOs approved for the TestDom1 domain by user Admin1GPA in a month and use Microsoft Windows Authentication in the SQL Server connection string:

```
ActivityReport /file:c:\GPA_Reports\Admin1GPA_Approved092010.html
/type:Approve /start:9/01/10 /end:9/30/10 /domain:TestDom1
/user:Admin1GPA /SQLConnStr:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=%SERVERNAME%;Use Procedure for
Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=%LOCALCOMPNAME%;Use Encryption for Data=False;Tag with column
collation when possible=False"
```

Example 4

To create Point in Time Analysis and Activity Report for all GPOs approved for the TestDom1 domain by user Admin1GPA in a month using SQL Server Authentication in the SQL Server connection string:

```
ActivityReport /file:c:\GPA_Reports\Admin1GPA_Approved092010.html
/type:Approve /start:9/01/10 /end:9/30/10 /domain:TestDom1
/user:Admin1GPA /details:Difference /overwrite:No
/SQLConnStr:"Provider=SQLOLEDB.1;Password="[Your Password]";User
ID=sa;Initial Catalog=GPO_REPOSITORY;Data Source=%SERVERNAME%;Use
Procedure for Prepare=1;Auto Translate=True;Packet
Size=4096;Workstation ID=%LOCALCOMPNAME%;Use Encryption for
Data=False;Tag with column collation when possible=False"
```

A.5   Scheduling Scripts

Now that you have written multiple scripts for several different tasks, you need to schedule them to run on a set schedule.

To schedule the scripts:

1 Click **Start > Control Panel**.

2 Click **System and Maintenance > Administrative Tools**.

3 Double-click **Task Scheduler**.

4 Double-click the **Add Scheduled Task** icon. Windows displays the Scheduled Task wizard.

5 Click **Next** at the window.

6 Click **Browse** to locate your `.vbs` file, then select it and click **Open**.

7 Select the name of the script you created earlier and click **Open**.

8 Follow the instructions in the wizard to choose the time to run the script and supply any other necessary information.

9 When you click **Finish**, your task is scheduled. For more information about scheduling, see your Microsoft Windows documentation.

A.6   GP Repository Scripting Object Model

Most of the GP Repository operations that are performed from the user interface can be automated by using the GP Repository scripting objects and their methods. The following list identifies the primary scripting objects:

Root   **Creation of this object is a prerequisite for any GP Repository script operation as this exposes the key method to connect to the GP Repository. Other methods include domain creation, enumeration and creation of other GP Repository objects.**

Domain   Offers functionality to create categories, enumerate categories and delete domains.

Category   Offers functionality to create GPO, Enumerate GPOs in category, Import GPO from Active Directory, Delete Category.

GPO   Offers functionality to report, Copy, Paste, Migrate, Query GPO status and modify GPO status.

A.7   Root Node Operations

The following sections provide the scriptable operations you can perform on the root node object and include C# method syntax and samples where applicable.

A.7.1   Root Object Creation

Initialize the GP Repository script operations.

Syntax (Visual Basic Script)

```
Wscript.CreateObject("faGPRRoot.faGPRRoot")
```

Sample Code (Visual Basic Script)

The following code allows you to initialize the GP Repository script operations.

```
Dim oGPRroot
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
```

Syntax (C# Method)

```
IfaGPRRoot oGPRroot = new faGPRRoot();
```

Sample Code (C# Method)

The following code allows you to initialize the GP Repository script operations.

```
public static void CreateRootObject()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            Console.WriteLine("Root object created: " +
oGPRroot.GetType().ToString());
            Console.ReadKey();
        }
```

A.7.2   Change Repository Authorization Code or GPA Security Account

Use the `NqGPARepConfig.exe` tool to change the Repository Authorization Code or the GPA Security Account. For more information, see Section 3.2.2, "Changing or Updating the GPA Security Account," on page 44.

Syntax

```
NqGpaRepConfig.exe [/Repository:<server>] [/DB:<database name>]
[/User:<login_name> /Pass:<password> | *]
[/RepAuthCode:<Rep_code>]
[/AddSecurityAccount:<domain\account_name>] [/LSecurityAccounts]
[/Disconnect] [/? | /Help]
```

Options

The following table describes the command-line parameters and variables.

Variable Name	Replace With
/Repository:*ServerName*	Name of the Microsoft SQL Server computer. The default value is the local computer.
/DB:*DatabaseName*	Name of the GP Repository database. The default value of the GP Repository database name is GPO_REPOSITORY.
/User:*LoginName*	The SQL Server account name that can perform the operation.
/Pass:*Password*	The SQL Server account password for the account provided in the /User option.
/RepAuthCode:*Rep_code*	Sets the Repository Authorization Code.
/AddSecurityAccount: *DomainAccountName*	Adds the specified account as a GPA Security account.
/LSecurityAccounts	Lists all accounts specified as GPA Security accounts.
/Disconnect	Removes all connections between the GP Repository and the GPA Server.

Example 1

Specify a new Repository Authorization Code:

```
NqGpaRepConfig.exe /RepAuthCode:New Authorization Code
```

Example 2

Add a new GPA Security Account:

```
NqGpaRepConfig.exe /AddSecurityAccount:domain\account
```

A.7.3 Manage GPA Access Accounts

Use the `GPAServerConfig.exe` tool to modify the Export Only, Untrusted Access, and GPA Security accounts, including updating the account passwords. For more information, see Section 3.2.2, "Changing or Updating the GPA Security Account," on page 44.

IMPORTANT: Execute the `GPAServerConfig.exe` tool as Administrator.

Syntax

```
GPAServerConfig.exe [/D:DomainFQDN] [/SA:ServiceAccount]
[/SP:ServiceAccountPassword] [/EA:ExportOnlyAccount]
[/EP:ExportOnlyAccountPassword] [/UA:UntrustedAccount]
[/UP:UntrustedAccountPassword] [/? | /Help]
```

Options

The following table describes the command-line parameters and variables.

Variable Name	Replace With
/D	FQDN name of the Domain.
/SA	GPA Security Account Name.
/SP	Password of the GPA Security Account.
/EA	Export Only Account Name.
/EP	Password of the Export Only Account.
/UA	Untrusted Access Account Name.
/UP	Password of the Untrusted Access Account.

Example 1

Modify a GPA Service account:

```
GPAServerConfig /D:NetIQLab.com /SA:NetIQUser /SP:NetIQPassword
```

Example 2

Set or modify the Export Only Account for a domain:

```
GPAServerConfig /D:NetIQLab.com /EA:NetIQUser /EP:NetIQPassword
```

Example 3

Set or modify an Untrusted Access Account:

```
GPAServerConfig /D:NetIQLab.com /UA:NetIQUser /UP:NetIQPassword
```

A.7.4   Connect to GP Repository

Every GP Repository script must connect to a GP Repository to obtain data.

Syntax (Visual Basic Script)

```
Objectname.ConnectTo("ConnectionString")
```

Sample Code (Visual Basic Script)

The following code allows you to connect to the GP Repository.

```
Dim oGPRroot
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
```

Syntax (C# Method)

```
Objectname.ConnectTo("ConnectionString")
```

Sample Code (C# Method)

The following code allows you to connect to the GP Repository.

```
public static void ConnectToGPRepository()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
          oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL
Server Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;");
            Console.WriteLine("Connected to the Repository");
            Console.ReadKey();
        }
```

Obtaining the Connection String Value

The connection string is the parameter required to connect to the database.

To obtain this connection string:

1 Launch the GPA Console from the computer where you are going to execute the script or application.

2 After connecting to the GP Repository, select the **GP Repository** node.

3 On the Action menu, click **Properties**. The connection string is displayed in the Properties window.

4 Copy the connection string and paste it into your script or method file.

NOTE: The connection to the GP Repository is also based on security permissions for the user account in whose context the script or application is executed. Hence, if that user does not have permission to connect to the database, the connection command returns an error.

If you connect to the database with SQL Server authentication by providing a SQL Server user name and password, then the Connect String window displays the password as "<Password>". You need to replace this variable with the actual password.

A.7.5 Create Domains

Add a new domain to GP Repository. This operation requires special permissions in GPR Security to create Container Objects.

Syntax (Visual Basic Script)

```
RootObject.CreateDomain("DomainName")
```

or

```
Set ObjectVariable = RootObject.CreateDomain("DomainName")
```

Sample Code (Visual Basic Script)

The following code creates a new domain in the GP Repository.

```
Dim oGPRroot
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
oGPRroot.CreateDomain("NetIQlabs.com")
```

Syntax (C# Method)

```
RootObject.CreateDomain("DomainName")
```

Sample Code (C# Method)

The following code allows you to create a new domain in the GP Repository.

```
public static void CreateDomain()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            oGPRroot.CreateDomain("NetIQlabs.com");
            Console.WriteLine("Domain created");
            Console.ReadKey();
        }
```

A.7.6   Enumerate Domains

Enumerate the list of domains under the root node.

Syntax (Visual Basic Script)

```
For Each Domain in RootObject
     [. . . perform operations . . .]
Next
```

Sample Code (Visual Basic Script)

The following code prints all existing domain names in the GP Repository.

```
Dim oGPRroot, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
For Each oDomain in oGPRroot
     Wscript.Echo oDomain.Name
Next
```

Syntax (C# Method)

```
foreach (Domain in RootObject)
{
[. . . perform operations . . .]
}
```

Sample Code (C# Method)

The following code allows you to display the GP Repository domain names.

```
public static void EnumerateDomains()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            foreach (IfaGPRDomain gprDomain in oGPRroot)
            {
                Console.WriteLine(gprDomain.Name);
            }
            Console.ReadKey();
        }
```

A.7.7   Get Object

Initiate instance of an object, such as a GPO, category, or domain, in the GP Repository. The object should exist in the GP Repository.

Syntax (Visual Basic Script)

```
Set ObjectVariable = RootObject.GetObject("RepositoryObjectPath")
```

Sample Code (Visual Basic Script)

The following code allows you to initiate an instance of a category.

```
Dim oGPRroot, oCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
wscript.echo oCategory.Name
```

Syntax (C# Method)

```
<Data type> ObjectVariable = RootObject.GetObject("Repository Path")
```

Sample Code (C# Method)

The following code allows you to initiate an instance of a category and a GPO.

```
public static void GetObject()
        {
        IfaGPRRoot oGPRroot = new faGPRRoot();
        oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
        IfaGPRCategory oCategory = oGPRroot.GetObject("FAGPR://
CN=Desktop,DC=NetIQLabs,DC=com");
        IfaGPRGpo oGPO = oGPRroot.GetObject("FAGPR://CN={C104C9C7-9355-4FEC-
8824-22D7BF4797A9}, CN=Desktop,DC=NetIQLabs,DC=com");
        Console.WriteLine("Category name obtained: " + oCategory.Name + " GPO
name obtained: " + oGPO.Name);
        Console.ReadKey();
        }
```

Repository Object Path

The *Repository object path* is the location of a node for a GPO, category, or domain in the GP Repository. The format for the Repository object path is similar to an LDAP path with the following exceptions:

- Use `faGPR://` for the GP Repository path instead of `LDAP://`.
- Category names are preceded by `CN=` and each element of the domain name is preceded by `DC=`.

The simplest method for viewing the Repository object path is to right-click the node for the appropriate GPO, category, or domain, and then click **Properties**. Use the **Path** property.

A.8   Domain Operations

The following sections provide the scriptable operations that can be carried out on the domain object.

A.8.1   Create Offline Policy Container Hierarchy

Run the `NQCreateOfflinePolicyContainerHierarchy.exe` file to create a temporary copy of the settings information of all GPOs in the GP Repository. GPA automatically creates the offline policy container hierarchy when you add domains to the GP Repository. This only works for domains that have a trust relationship with the repository member domain.

To run the `NQCreateOfflinePolicyContainerHierarchy.exe` file, you should have Domain Admin permissions in the domain for which you want to create the offline policy container hierarchy.

The `NQCreateOfflinePolicyContainerHierarchy.exe` file displays a status report in the command prompt window as it runs. After execution, the `NQCreateOfflinePolicyContainerHierarchy.exe` file creates a log that lists the domains it successfully recreated and those domains it failed to recreate. The log displays the "`ATTENTION REQUIRED`" text next to the domain name of any domain the tool failed to recreate.

Syntax

```
NQCreateOfflinePolicyContainerHierarchy /D:Domain_DNS_Name
/S:Repository_Server /DB:DatabaseName
```

Options

The following table describes the command-line parameters and variables.

Variable Name	Replace With
/D:*Domain_DNS_Name*	DNS name of the evaluation domain, such as abc.xyz. If you specify the domain name, then GPA creates the offline policy container hierarchy for only that domain. If you do not specify the domain name, then GPA creates an offline policy container hierarchy for each domain in the GP Repository (optional).
/S:*Repository_Server*	Name of the Microsoft SQL Server where you have installed the GP Repository. The default value, period (.), indicates the local Microsoft SQL Server.
/DB:*DatabaseName*	Name of the GP Repository database. The default value of the GP Repository database name is GPO_REPOSITORY. If the database name is different, specify the correct database name (optional, if you specify the domain name).
/?	Command-line Help for the tool.

Sample Code

```
NQCreateOfflinePolicyContainerHierarchy /D:ABC.com /S:ABCSQLServer /
DB:ABCDatabaseName
```

A.8.2   Create Category

Create a new category.

Syntax (Visual Basic Script)

```
DomainObject.CreateCategory "CategoryName"
```

Sample Code (Visual Basic Script)

The following code creates a domain-level category in the GP Repository.

```
Dim oGPRroot, oCategory, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomain = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
oDomain.CreateCategory "Software Policies"
```

Syntax (C# Method)

DomainObject.CreateCategory("*CategoryName*")

Sample Code (C# Method)

The following code creates a domain-level category in the GP Repository.

```
public static void CreateCategory()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain oDomain = oGPRroot.GetObject("FAGPR://
DC=MYDOMAIN,DC=LAB");
            oDomain.CreateCategory("NewCategory");
            Console.WriteLine("Category created");
            Console.ReadKey();
        }
```

A.8.3 Delete Domain

Delete a domain from GP Repository. This operation would delete all GPOs under various categories and subcategories in the domain. This operation requires all GPOs in the domain to be checked in.

Syntax (Visual Basic Script)

DomainObject.Delete

Sample Code (Visual Basic Script)

The following code deletes a domain from the GP Repository.

```
Dim oGPRroot, oCategory, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomain = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
oDomain.Delete
```

Syntax (C# Method)

DomainObject.Delete()

Sample Code (C# Method)

The following code deletes a domain from the GP Repository.

```
public static void DeleteDomain()
        {
            string sDomainSource = "FAGPR://DC=MYTARGETDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomain = oGPRroot.GetObject(sDomainSource);
            oDomain.Delete();
            Console.WriteLine("Domain deleted");
            Console.ReadKey();
        }
```

A.8.4   Enumerate AD Links

Enumerates AD Links defined in the domain map for the specified domain.

Syntax (Visual Basic Script)

StringArrayOfLinks = DomainObject.EnumerateADLinks()

Sample Code (Visual Basic Script)

The following code allows you to enumerate the AD links defined in the domain map.

```
Dim oGPRroot, oDomainSource, oDomainTarget, aLinks, strPath
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
aLinks = oDomainSource.EnumerateADLinks()
For each strPath in aLinks
     wscript.echo strPath
Next
```

Syntax (C# Method)

ObjectArrayOfLinks = DomainObject.EnumerateADLinks()

Sample Code (C# Method)

The following code allows you to enumerate the AD links defined in the domain map.

```
public static void EnumerateADLinks()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomain = oGPRroot.GetObject(sDomainSource);
            object[] obj = oDomain.EnumerateADLinks();
            foreach (object ob in obj)
            {
                Console.WriteLine(ob.ToString());
            }
            Console.ReadKey();
        }
```

A.8.5   Enumerate Categories

Enumerate the categories under a node for a domain.

Syntax (Visual Basic Script)

```
For Each "Category" in "Domain"
     [. . . perform operations . . .]
Next
```

Sample Code (Visual Basic Script)

The following code prints all domain-level category names for all existing domains in the GP Repository.

```
Dim oGPRroot, oCategory, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
For Each oDomain in oGPRroot
     Wscript.Echo oDomain.Name
     For Each oCategory in oDomain
          Wscript.Echo
oCategory.Name
     Next
Next
```

Syntax (C# Method)

```
foreach (Category in Domain)
{
[. . . perform operations . . .]
}
```

Sample Code (C# Method)

The following code prints all domain-level category names for all existing domains in the GP Repository.

```
public static void EnumerateCategories()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomain = oGPRroot.GetObject(sDomainSource);
            object[] obj = oDomain.EnumerateADLinks();
            foreach (IfaGPRDomain gprDomain in oGPRroot)
            {
                Console.WriteLine(gprDomain.Name);
                foreach (IfaGPRCategory gprCategory in gprDomain)
                {
                    Console.WriteLine(gprCategory.Name);
                }
            }
            Console.ReadKey();
        }
```

A.8.6   Enumerate GPO Map

Enumerates the GPOs in the domain map.

Syntax (Visual Basic Script)

```
ArrayOfGPOCNStrings =
TargetDomainObject.EnumerateGPOMap(SourceDomainObject)
```

Sample Code (Visual Basic Script)

The following code enumerates the GPOs in the domain map.

```
Dim oGPRroot, oDomainSource, oDomainTarget, aGPOs, strGPO
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
aGPOs = oDomainTarget.EnumerateGPOMap(oDomainSource)
For each strGPO in aGPOs
     wscript.echo
Next
```

Syntax (C# Method)

```
ObjectArrayOfGPOCN =
TargetDomainObject.EnumerateGPOMap(SourceDomainObject)
```

Sample Code (C# Method)

The following code enumerates the GPOs in the domain map.

```
public static void EnumerateGPOMap()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            string sDomainTarget = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomain = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 otarget = oGPRroot.GetObject(sDomainTarget);
            object[] obj = otarget.EnumerateGPOMap(oDomain);
            foreach (object gprDomain in obj)
            {
                Console.WriteLine(gprDomain.ToString());
            }
            Console.ReadKey();
        }
```

A.8.7   Enumerate Users

Enumerates the accounts in the domain map.

Syntax (Visual Basic Script)

arrayOfUserStrings = *Domain*.EnumerateUsers()

Sample Code (Visual Basic Script)

The following code allows you to enumerate the accounts in the domain map.

```
Dim oGPRroot, oDomain
Dim aUsers
Dim strUser
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomain = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=coml")
aUsers = oDomain.EnumerateUsers()
For each strUser in aUsers
     wscript.echo strUser
Next
```

Syntax (C# Method)

ObjectArrayOfUsers = *Domain*.EnumerateUsers()

Sample Code (C# Method)

The following code allows you to enumerate the accounts in the domain map.

```
public static void EnumerateUsers()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomain = oGPRroot.GetObject(sDomainSource);
            object[] obj = oDomain.EnumerateUsers();
            foreach (object ob in obj)
            {
                Console.WriteLine(ob.ToString());
            }
            Console.ReadKey();
        }
```

A.8.8   Get Mapped AD Link

Retrieves AD Link mapping information from the domain map.

Syntax (Visual Basic Script)

```
StrADLinkPath = DomainObject.GetMappedADLink("DomainObject", "DCName")
```

Sample Code (Visual Basic Script)

The following code allows you to enumerate the AD links of the OU in the domain map with the corresponding AD Link mapping information of the OU in the specified domain.

```
Dim oGPRroot, oDomainSource, oDomainTarget, aLinks, strPath, strMappedPath
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
aLinks = oDomainSource.EnumerateADLinks()
For each strPath in aLinks
     strMappedPath =
    oDomainTarget.GetMappedADLink
    (oDomainSource,strPath)
     wscript.echo strPath & " --> " &
strMappedPath
Next
```

Syntax (C# Method)

```
StrADLinkPath = DomainObject.GetMappedADLink("DomainObject", "DCName")
```

Sample Code (C# Method)

The following code allows you to enumerate the AD links of the OU in the domain map with the corresponding AD Link mapping information of the OU in the specified domain.

```
public static void GetADLinkMap()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainsource = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomaintarget = oGPRroot.GetObject("FAGPR://
DC=MYTARGETDOMAIN,DC=LAB");
            object[] aLinks = oDomainsource.EnumerateADLinks();
            foreach (object ob in aLinks)
            {
                if (oDomaintarget.GetMappedADLink(oDomainsource, ob.ToString()) !=
null)
                {
                    Console.WriteLine(ob.ToString() + " --- >" +
oDomaintarget.GetMappedADLink(oDomainsource, ob.ToString()));
                }
                else
                {
                    Console.WriteLine(ob.ToString() + " --- >  AD link is not mapped
in the Target Domain");
                }
            }
            Console.ReadKey();
        }
```

A.8.9   Get Mapped GPO

Retrieves GPO mapping information from the domain map.

Syntax (Visual Basic Script)

```
GPOCNString =
TargetDomainObject.GetMappedGPO(SourceDomainObject,
"SourceGPO_CN_NAME")
```

Sample Code (Visual Basic Script)

The following code retrieves GPO mapping information from the domain map.

```
Dim oGPRroot, oDomainSource, oDomainTarget, aGPOs, strGPO,
strMappedGpo
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
aGPOs = oDomainTarget.EnumerateGPOMap(oDomainSource)
For each strGPO in aGPOs
     strMappedGpo =
oDomainTarget.GetMappedGPO
(oDomainSource,strGPO)
     wscript.echo strGPO & " --> " &
strMappedGpo
Next
```

Syntax (C# Method)

```
GPOCNString = TargetDomainObject.GetMappedGPO(SourceDomainObject,
"SourceGPO_CN_NAME")
```

Sample Code (C# Method)

The following code retrieves GPO mapping information from the domain map.

```
public static void GetMappedGPO()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainsource = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomaintarget = oGPRroot.GetObject("FAGPR://
DC=MYTARGETDOMAIN,DC=LAB");
            object[] aGPOs = oDomaintarget.EnumerateGPOMap(oDomainsource);
            foreach (object ob in aGPOs)
            {
                if (oDomaintarget.GetMappedGPO(oDomainsource, ob.ToString()) !=
null)
                {
                    Console.WriteLine(ob.ToString() + " --- >" +
oDomaintarget.GetMappedGPO(oDomainsource, ob.ToString()));
                }
                else
                {
                    Console.WriteLine(ob.ToString() + " --- >  GPO is not mapped in
the Target Domain");
                }
            }
            Console.ReadKey();
        }
```

A.8.10 Get Mapped User

Returns the domain mapping for a specified account and domain.

Syntax (Visual Basic Script)

```
MappedUserString =
TargetDomainObject.GetMappedUser(SourceDomainObject,
"Source_UserName")
```

Sample Code (Visual Basic Script)

The following code allows you to enumerate the accounts in the domain map with the corresponding mapping account for the specified domain.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
aUsers = oDomainSource.EnumerateUsers()
For each strUser in aUsers
    strMappedUser =
    oDomainTarget.GetMappedUser
(oDomainSource,strUser)
    wscript.echo strUser & " --> " &
strMappedUser
Next
```

Syntax (C# Method)

```
MappedUserString = TargetDomainObject.GetMappedUser(SourceDomainObject,
"Source_UserName")
```

Sample Code (C# Method)

The following code allows you to enumerate the accounts in the domain map with the corresponding mapping account for the specified domain.

```
public static void GetMappedUser()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainsource = oGPRroot.GetObject("FAGPR://
DC=MYDOMAIN,DC=LAB");
            IfaGPRDomain2 oDomaintarget = oGPRroot.GetObject("FAGPR://
DC=MYTARGETDOMAIN,DC=LAB");
            object[] aUsers = oDomainsource.EnumerateUsers();
            foreach (object ob in aUsers)
            {
                Console.WriteLine(ob.ToString() + " --- >" +
oDomaintarget.GetMappedUser(oDomainsource, ob.ToString()));
            }
            Console.ReadKey();
        }
```

A.8.11   Offline Mirror

The Offline Mirror process imports GPOs from an Active Directory domain into the GP Repository and synchronizes link order based on AD or the GP Repository, also called creating an **offline mirror**. The command-line interface uses a template you save using the Offline Mirror wizard that defines the options on AD containers to process. This only works for domains that have a trust relationship with the repository member domain.

You can use the Offline Mirror command-line tool, located in the `\Bin` folder under the product installation path, to configure the update process to run during off-peak hours using a Microsoft Windows scheduled task.

NOTE: If you are running GPA on a 64-bit platform, you need to run the Offline Mirror tool using a 32-bit command prompt window. On a 64-bit computer, you can access the 32-bit command prompt window from the `%WINDIR%\SysWOW64` folder.

You can also create an offline mirror from the GPA Console using the Offline Mirror wizard by selecting a GP Repository domain and clicking **Run Offline Mirror** from the **Action** menu. For more information, see Section 5.4.2, "Importing All GPOs Linked to Any AD Container in an AD Domain (Creating an Offline Mirror)," on page 78.

Syntax

```
C:\Program Files\NetIQ\Group Policy Administrator\Bin\"NetIQ GPA Offline
Mirror.exe"
```

```
[/F:"C:\MyNewTemplate.xml]"
```

Options

The following table describes the command-line parameters and variables.

Variable name	Replace with

`/`
`F:Offine_Mirror_Template.xm`
`l`
	Specifies the path to the .xml offline mirror template file, which loads and runs the offline mirror import and sync link order processes from the command-line window. Use quotation marks if the path or file name includes spaces.

`/? or /h`	Displays command-line help for this tool.

NOTE: When you do not specify any parameters, the tool opens and runs the Offline Mirror wizard and closes the command prompt window.

Sample Code

```
"NetIQ GPA Offline Mirror Wizard.exe"/F:"%USERPROFILE%\My
Documents\Offline Mirror Input.xml"
```

Offline Mirror Status Report

The Offline Mirror tool displays a status report in the command prompt window as it runs, similar to the following example. All of the information in the command line window is also written to a log found in the %appdata% path of the user running the tool. The import process may take some time for domains with a large number of OUs and GPOs.

```
NetIQ GPA Offline Mirror
(c) 2011 NetIQ Corporation.
File Path:C:\Users\Administrator.GPDOM300\Desktop\OMW Sample Import.xml
Offline mirror process started
Import in progress...
Percentage Completed:0
Import started  for container:: LDAP://HOUDVGP106V.GPDOM300.lab/DC=GPDOM300,DC=lab
Importing GPO: Default Domain Policy
GPO Default Domain Policy exists in the GP Repository.
The version of GPO Default Domain Policy in AD is newer than the version in the GP
Repository.
GPO imported successfully: Default Domain Policy
Import Completed for container:: LDAP://HOUDVGP106V.GPDOM300.lab/
DC=GPDOM300,DC=lab
Percentage Completed:100
Import operation completed for SyncID: 62a2066e-c3ca-43cd-af35-16ae7edb20f3
Completed successfully without any errors and total AD objects synced: 1
```

A.8.12   Read Domain Name

Read the **Name** property of a domain.

Syntax (Visual Basic Script)

DomainObject.Name

Sample Code (Visual Basic Script)

The following code allows you to print all domain names in the GP Repository.

```
Dim oGPRroot, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
For Each oDomain in oGPRroot
     Wscript.Echo oDomain.Name
Next
```

Syntax (C# Method)

```
DomainObject.Name
```

Sample Code (C# Method)

The following code allows you to print all domain names in the GP Repository.

```
public static void ReadDomainName()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            foreach (IfaGPRDomain gprDomain in oGPRroot)
            {
                Console.WriteLine(gprDomain.Name);
            }
            Console.ReadKey();
        }
```

A.8.13   Set AD Link Map

Sets mapping information in the domain map for the specified AD Link.

Syntax (Visual Basic Script)

```
TargetDomainObject.SetADLinkMap SourceDomainObject, "Source Domain OU
Path",  "Target Domain OU Path"
```

Sample Code (Visual Basic Script)

The following code allows you to map an AD Link from an OU in the source domain, `Org1.com`, to a target AD Link in an OU with the same name and path in `Test.Org1.com`.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=Org1,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=Org1,DC=com")
oDomainTarget.SetADLinkMap oDomainSource,
"LDAP://OU=Houston,DC=Org1,DC=com",
"LDAP://OU=Houston,OU=USA,DC=Test,DC=Org1,DC=com"
```

Syntax (C# Method)

```
TargetDomainObject.SetADLinkMap(SourceDomainObject, "Source Domain OU
Path", "Target Domain OU Path")
```

Sample Code (C# Method)

The following code allows you to map an AD Link from an OU in the source domain, `MyDomain.Lab`, to a target AD Link in an OU with the same name and path in `MyTargetDomain.Lab`.

```
public static void SetADLinkMap()
        {
            string sDomainSource = "FAGPR://DC=Org1,DC=Com";
            string sDomainTarget = "FAGPR://DC=Test,DC=Org1,DC=com";
            IfaGPRRoot oGPRroot = new faGPRRoot();
           oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL
Server Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;");
            IfaGPRDomain2 oDomainSrc = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomainTgt = oGPRroot.GetObject(sDomainTarget);
            oDomainTgt.SetADLinkMap(oDomainSrc, "LDAP://
OU=Bolivia,DDC=Org1,DC=Com", "LDAP://OU=USA,DC=Test,DC=Org1,DC=com");
            Console.WriteLine("AD Link mapped successfully");
            Console.ReadKey();
        }
```

A.8.14   Set Default AD Link Map

Populates the AD Links in the target domain's map from source domain with default information by mapping each source AD Link to an OU with the same name/path in the target domain.

Syntax (Visual Basic Script)

```
TargetDomainObject.SetDefaultADLinkMap SourceDomainObject
```

Sample Code (Visual Basic Script)

The following code allows you to map each AD Link from an OU in the source domain, `Org1.com`, to the corresponding target AD Link in the OU with the same name and path in `Test.Org1.com`.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=Org1,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=Org1,DC=com")
oDomainTarget.SetDefaultADLinkMap oDomainSource
```

Syntax (C# Method)

TargetDomainObject.SetDefaultADLinkMap(*SourceDomainObject*)

Sample Code (C# Method)

The following code allows you to map each AD Link from an OU in the source domain, `MyDomain.Lab`, to the corresponding target AD Link in the OU with the same name and path in `MyTargetDomain.Lab`.

```
public static void SetDefaultADLinkMap()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            string sDomainTarget = "FAGPR://DC=MYTARGETDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainSrc = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomainTgt = oGPRroot.GetObject(sDomainTarget);
            oDomainTgt.SetDefaultADLinkMap(oDomainSrc);
            Console.WriteLine("Default AD Link Map was mapped successfully");
            Console.ReadKey();
        }
```

A.8.15   Set Default User Map

Updates the target domain map for the source domain (the map to target domain from source domain). For each user in the source domain's map, this operation adds a map entry from the source account to the target account with the same account name (if any).

Syntax (Visual Basic Script)

TargetDomainObject.SetDefaultUserMap *SourceDomainObject*

Sample Code (Visual Basic Script)

The following code allows you to map each user account from the source domain, `Org1.com`, to the corresponding target account with the same name in `Test.Org1.com`.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=Org1,DC=com")
Set oDomainTarget = oGPRroot.GetObject("FAGPR://DC=Test,DC=Org1,DC=com")
oDomainTarget.SetDefaultUserMap oDomainSource
```

Syntax (C# Method)

TargetDomainObject.SetDefaultUserMap(*SourceDomainObject*)

Sample Code (C# Method)

The following code allows you to map each user account from the source domain, `MyDomain.Lab`, to the corresponding target account with the same name and path in `MyTargetDomain.Lab`.

```
public static void SetDefaultUserMap()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            string sDomainTarget = "FAGPR://DC=MYTARGETDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainSrc = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomainTgt = oGPRroot.GetObject(sDomainTarget);
            oDomainTgt.SetDefaultUserMap(oDomainSrc);
            Console.WriteLine("Default User Map successful");
            Console.ReadKey();
        }
```

A.8.16   Set Domain Controller

Set domain controller context. This DC would be the one used for subsequent Active Directory Operations, such as exporting a GPO. It is not a mandatory command. If not specified, the system selects any available domain controller. You must run the Set Domain Controller script as a local administrator.

Syntax (Visual Basic Script)

RootObject.SelectDomainDC "*DomainName*", "*DCName*"

Sample Code (Visual Basic Script)

The following code sets a domain controller context.

```
Dim oGPRroot, oCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
oGPRroot.SelectDomainDC "RootDev2.Com","root2-ad-01.rootdev2.com"
```

Syntax (C# Method)

```
RootObject.SelectDomainDC("DomainName", "DCName")
```

Sample Code (C# Method)

The following code sets a domain controller context.

```
public static void SetDomainController()
        {
            string sDomainSource = "MYDOMAIN.LAB";
            string sDomainController = "MYDOMAINCONTROLLER.MYDOMAIN.LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            oGPRroot.SelectDomainDC(sDomainSource, sDomainController);
            Console.WriteLine("Domain Controller was set correctly");
            Console.ReadKey();
        }
```

A.8.17   Set GPO Map

Sets mapping information for GPOs in the domain map. When you migrate GPOs from one domain to another domain, this method allows you map the source GPOs in one domain to the target GPOs in another domain.

Syntax (Visual Basic Script)

```
TargetDomainObject.SetGPOMap SourceDomainObject, "Source_GPO_CN",
"Target_GPO_CN"
```

Sample Code (Visual Basic Script)

The following code sets mapping information for GPOs in the domain map.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = WscriDRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
oDomainTarget.SetGPOMap oDomainSource, "{6E936ED3-00C8-4FE7-95A1-
803874AB7EA0}", "{8435AE6D-DED3-470C-B57C-66BB80B7DA8B}"
```

Syntax (C# Method)

```
TargetDomainObject.SetGPOMap(SourceDomainObject, "Source_GPO_CN",
"Target_GPO_CN")
```

Sample Code (C# Method)

The following code sets mapping information for GPOs in the domain map.

```
public static void SetGPOMap()
       {
           string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
           string sDomainTarget = "FAGPR://DC=MYTARGETDOMAIN,DC=LAB";
           IfaGPRRoot oGPRroot = new faGPRRoot();
           oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
           IfaGPRDomain2 oDomainSrc = oGPRroot.GetObject(sDomainSource);
           IfaGPRDomain2 oDomainTgt = oGPRroot.GetObject(sDomainTarget);
           oDomainTgt.SetGPOMap(oDomainSrc, "{C104C9C7-9355-4FEC-8824-
22D7BF4797A9}", "{A71A3C86-53FD-43B3-AAB1-DC163CBC3EC9}");
           Console.WriteLine("GPO was mapped successfully");
           Console.ReadKey();
       }
```

A.8.18   Set User Map

Adds an entry to the domain map.

Syntax (Visual Basic Script)

```
TargetDomainObject.SetUserMap sourceDomainObject, "Source_Username",
"Target_Username"
```

Sample Code (Visual Basic Script)

The following code allows you to map a user account from the source domain to a target account in the target domain.

```
Dim oGPRroot, oDomainSource, oDomainTarget
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oDomainSource = oGPRroot.GetObject("FAGPR://DC=NetIQLabs,DC=com")
Set oDomainTarget =
oGPRroot.GetObject("FAGPR://DC=Test,DC=NetIQLabs,DC=com")
oDomainTarget.SetUserMap oDomainSource ,"JSmith" ,"SmithJ"
```

Syntax (C# Method)

```
TargetDomainObject.SetUserMap(sourceDomainObject, "Source_Username",
"Target_Username")
```

Sample Code (C# Method)

The following code allows you to map a user account from the source domain to a target account in the target domain.

```
public static void SetUserMap()
        {
            string sDomainSource = "FAGPR://DC=MYDOMAIN,DC=LAB";
            string sDomainTarget = "FAGPR://DC=MYTARGETDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRDomain2 oDomainSrc = oGPRroot.GetObject(sDomainSource);
            IfaGPRDomain2 oDomainTgt = oGPRroot.GetObject(sDomainTarget);
            oDomainTgt.SetUserMap(oDomainSrc, "MYDOMAIN-LAB\JSmith",
"MYTARGETDOMAIN-LAB\SmithJ");
            Console.WriteLine("User account mapped successfully");
            Console.ReadKey();
        }
```

A.8.19   Merge GPOs

Merges the settings from two GPOs into a new GPO in the same domain.

Syntax (Visual Basic Script)

DomainObject.MergeGpos *SourceGPOs*, *TargetFAGPRPath*, *"TargetGPOName"*,
DeleteSourceGPOsFlag

Sample Code (Visual Basic Script)

The following sample merges two GPOs and creates a new GPO in the specified category without deleting the source GPOs.

```
Dim arrSourceGpos, targetGPRPath, targetGPOName, oGPRroot, oDomain
arrSourceGpos = Array("FAGPR://CN={9FCE1105-3661-404A-BB6D-
0EAA8049BC93},CN=MYCAT,DC=MYDOMAIN,DC=LAB","FAGPR://CN={57DCB21E-30D0-4229-97B4-
69F3B30E01BB},CN=MYCAT,DC=MYDOMAIN,DC=LAB")
targetGPRPath = "FAGPR://CN=MYCAT, DC=MYDOMAIN,DC=LAB"
targetGPOName = "MergedGPO"
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=SQLSERVER;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=GPACONSOLE;Use Encryption for
Data=False;Tag with column collation when possible=False")
Set oDomain = oGPRroot.GetObject("FAGPR://DC=MYDOMAIN,DC=LAB")
Wscript.Echo "Ready to Merge GPOs"
oDomain.MergeGpos arrSourceGpos, targetGPRPath, targetGPOName, false
Wscript.Echo "Merge Successful"
```

Syntax (C# Method)

DomainObject.MergeGpos(*SourceGPOs*, *TargetFAGPRPath*, *TargetGPOName*,
DeleteSourceGPOsFlag)

Sample Code (C# Method)

The following sample merges two GPOs and creates a new GPO in the specified category.

```
public static void MergeGPOs ()
{
try
{
IfaGPRRoot GprRoot = new faGPRRoot();
GprRoot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=SQLSERVER;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=GPACONSOLE;Use Encryption for
Data=False;Tag with column collation when possible=False");
IfaGPRDomain3 GprDomain = GprRoot.GetObject("FAGPR://DC=MYDOMAIN,DC=LAB");
      List<string> sourceGpos = new List<string>();
sourceGpos.Add(("FAGPR://CN={9FCE1105-3661-404A-BB6D-
0EAA8049BC93},CN=MYCAT,DC=MYDOMAIN,DC=LAB");
sourceGpos.Add("FAGPR://CN={57DCB21E-30D0-4229-97B4-
69F3B30E01BB},CN=MYCAT,DC=MYDOMAIN,DC=LAB");
string targetGPRPath = "FAGPR://CN=MYCAT, DC=MYDOMAIN,DC=LAB";
      string targetGPOName = "MergedGPO";
List<string> sourceGposToDelete = new List<string>();
      Console.WriteLine("Ready to Merge GPOs");
      GprDomain.MergeGpos(sourceGpos, targetGPRPath, targetGPOName, false);
Console.WriteLine("Merge Successful");
}
catch (Exception ex)
{
string msg = ex.Message;
      Console.WriteLine(msg);
      }
Console.ReadLine()
}
```

A.9   Category Operations

The following sections provide the scriptable operations that can be carried out on the category object.

A.9.1   Create GPO

Create a GPO under a category.

Syntax (Visual Basic Script)

CategoryObject.CreateGPO "*GPOName*"

Sample Code (Visual Basic Script)

The following code allows you to create a GPO in a category in the GP Repository.

```
Dim oGPRroot, oCategory, sCatergory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
oCategory.CreateGPO "Software Policy"
```

Syntax (C# Method)

CategoryObject.CreateGPO("*GPOName*")

Sample Code (C# Method)

The following code allows you to create a GPO in a category in the GP Repository.

```
public static void CreateGPO()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject("FAGPR://CN=MyCategory,
DC=MYDOMAIN,DC=LAB");
            oCategory.CreateGPO("Software Policy");
            Console.WriteLine("GPO was created correctly");
            Console.ReadKey();
        }
```

A.9.2   Delete Category

Deletes a category. This operation would delete all GPOs and subcategories. To use this command, all GPOs in the category must be checked in.

Syntax (Visual Basic Script)

CategoryObject.Delete

Sample Code (Visual Basic Script)

The following code deletes a category from the GP Repository.

```
Dim oGPRroot, oCategory, oGPO
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory =
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
oCategory.Delete
```

Syntax (C# Method)

CategoryObject.Delete()

Sample Code (C# Method)

The following code deletes a category from the GP Repository.

```
public static void DeleteCategory()
        {
             IfaGPRRoot oGPRroot = new faGPRRoot();
             oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
             IfaGPRCategory oCategory = oGPRroot.GetObject("FAGPR://CN=NewCategory,
DC=MYDOMAIN,DC=LAB");
             oCategory.Delete();
             Console.WriteLine("Category deleted");
             Console.ReadKey();
        }
```

A.9.3   Enumerate GPOs

Enumerate the list of GPOs under a category. Prior to the enumeration loop, you need to specify the enumeration type as GPO.

Syntax (Visual Basic Script)

```
CategoryObject.EnumType = "GPO"
For Each GPOObject in CategoryObject
     [. . . perform operations . . .]
Next
```

Sample Code (Visual Basic Script)

The following code allows you to enumerate all GPOs inside a category.

```
Dim oGPRroot, oCategory, oGPO, sCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB"
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
Wscript.Echo oCategory.Name
For Each oGPO in oCategory
     Wscript.Echo oGPO.Name
Next
```

Syntax (C# Method)

```
CategoryObject = oGPRroot.GetObject(CategoryPath)
foreach (GPOObject in CategoryObject)
{
[. . . perform operations . . .]
}
```

Sample Code (C# Method)

The following code allows you to enumerate all GPOs inside a category.

```
public static void EnumerateGPOs()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            foreach (IfaGPRGpo gprGpo in oCategory)
            {
                Console.WriteLine(gprGpo.Name);
            }
            Console.ReadKey();
        }
```

A.9.4   Enumerate Subcategories

Enumerate the subcategories in a category. This task is similar to enumerating the list of GPOs under
a category. However, the extra step prior to the enumeration loop is for you to specify the
enumeration type to be CATEGORY. If you do not, the enumeration would return the list of GPOs.

Syntax (Visual Basic Script)

```
CategoryObject.EnumType = "CATEGORY"
For Each SubCategoryObject in CategoryObject
     [. . . perform operations . . .]
Next
```

Sample Code (Visual Basic Script)

The following code allows you to enumerate all child categories inside a parent category.

```
Dim oGPRroot, oCategory, oSubCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory =
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
oCategory.Enumtype = "CATEGORY"
Wscript.Echo oCategory.Name
For Each oSubCategory in oCategory
     Wscript.Echo oSubCategory.Name
Next
```

Syntax (C# Method)

```
CategoryObject.EnumType = "CATEGORY"
foreach (SubCategoryObject in CategoryObject)
{
[. . . perform operations . . .]
}
```

Sample Code (C# Method)

The following code allows you to enumerate all child categories inside a parent category.

```
public static void EnumerateSubCategories()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "Category";
            foreach (IfaGPRCategory gprCategory in oCategory)
            {
                Console.WriteLine(gprCategory.Name);
            }
            Console.ReadKey();
        }
```

A.9.5   Import GPO from Active Directory

Import an existing Active Directory GPO into a category.

Syntax (Visual Basic Script)

```
CategoryObject.ImportGPO "GPOLDAPPath", OverwriteFlag
```

or

```
Set GPOObject = CategoryObject.ImportGPO("GPOLDAPPath",
OverwriteFlag)
```

or

```
CategoryObject.ImportGPO ADSIGPOObject, OverwriteFlag
```

Sample Code 1 (Visual Basic Script)

The following code imports a GPO from Active Directory into the GP Repository.

```
Dim oGPRroot, oDomain, oCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory =
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
oCategory.ImportGPO "LDAP://maboslpt03.NetIQLabs.com/
CN={D162D0C0-6B7C-4F77-9846-F6EEF520FAD3},
CN=Policies,CN=System,DC=NetIQLabs,DC=com",True
```

Sample Code 2 (Visual Basic Script)

The following code imports a GPO from Active Directory into the GP Repository and associates the newly imported GPO to an object.

```
Dim  oGPRroot,  oDomain,  oCategory, oGPO
Set  oGPRroot  =  Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set  oCategory  =  oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
set oGPO = OCategory.ImportGPO ("LDAP://maboslpt03.NetIQLabs.com/CN={D162D0C0-
6B7C-4F77-9846-F6EEF520FAD3},CN=Policies,CN=System,DC=NetIQLabs,DC=com",True)
wscript.echo "The '" & oGPO.Name & "' GPO has been imported successfully."
```

Sample Code 3 (Visual Basic Script)

The following code imports a GPO from Active Directory into the GP Repository by passing the Active Directory GPO object as a parameter.

```
Dim oGPRroot, oDomain, oCategory
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory =
oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
Set oGpo = GetObject("LDAP://maboslpt03.NetIQLabs.com/
CN={D162D0C0-6B7C-4F77-9846-F6EEF520FAD3},
CN=Policies,CN=System,DC=NetIQLabs,DC=com")
oCategory.ImportGPO oGpo,True
```

Syntax (C# Method)

```
CategoryObject.ImportGPO("GPOLDAPPath", OverwriteFlag)
```

or

```
GPOObject = CategoryObject.ImportGPO("GPOLDAPPath", OverwriteFlag)
```

or

```
CategoryObject.ImportGPO(ADSIGPOObject, OverwriteFlag)
```

Sample Code (C# Method)

The following code imports a GPO from Active Directory into the GP Repository.

```
public static void ImportGPOFromAD()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sGPOLDAP = "LDAP://MYDOMAIN.LAB/CN={000344FD-1494-45A4-BF39-
5022C4B4741A},CN=Policies,CN=System,DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.ImportGPO(sGPOLDAP, true);
            Console.WriteLine("GPO imported");
            Console.ReadKey();
        }
```

Understanding LDAP Path and Overwrite Flag

This operation includes parameters that are defined as follows:

GPOLDAPPath

Specifies the LDAP path of the GPO that needs to be imported from a live Active Directory domain. To obtain the LDAP path of the GPO (`LDAP://...`), use the ADSI Edit tool. If the ADSI Edit tool is not available, substitute the following information in the LDAP path:

`"LDAP://DomainController/CN={GUID},CN=Policies,CN=System,DC=Domain"`

The parameters in this syntax statement are defined as follows:

DomainController

Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of My Computer.

GUID

Type the GUID number that corresponds to the GPO you want to import.

Domain

Type the name of the domain to which the GPO belongs. The domain name should be in the distinguished name format. For example, to specify the domain name, `mydomain.com`, the syntax should be `DC=MYDOMAIN, DC=COM`.

ADSIGPOObject

Specifies and ADSI pointer to the GPO in Active Directory.

OverwriteFlag

Specifies the overriding condition for the import. The values are `False` and `True`. `False` denotes *do not override if the GPO already exists in the domain*. `True` denotes *override the existing GPO*.

A.9.6   Read Name

Read the name property of a category.

Syntax (Visual Basic Script)

CategoryObject.Name

Sample Code (Visual Basic Script)

The following code prints all domain-level categories for each domain in the GP Repository.

```
Dim oGPRroot, oCategory, oDomain
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
For Each oDomain in oGPRroot
     Wscript.Echo oDomain.Name
     For Each oCategory in oDomain
          Wscript.Echo oCategory.Name
     Next
Next
```

Syntax (C# Method)

CategoryObject.Name

Sample Code (C# Method)

The following code prints all domain-level categories for each domain in the GP Repository.

```
public static void ReadCategoryName()
        {
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            foreach (IfaGPRDomain gprDomain in oGPRroot)
            {
                Console.WriteLine(gprDomain.Name);
                foreach (IfaGPRCategory gprCategory in gprDomain)
                {
                    Console.WriteLine(gprCategory.Name);
                }
            }
            Console.ReadKey();
        }
```

A.10   GPO Node Operations

The following sections provide the scriptable operations that can be carried out on the GPO scripting object.

A.10.1 Approve GPO

Approve a GPO to be exported to Active Directory or unapprove a GPO. If you set the value of the parameter to `True`, the method approves the GPO. Else, if the value of the parameter is `False`, the method unapproves the GPO.

Syntax (Visual Basic Script)

```
GPOObject.Approve True|False
```

Sample Code (Visual Basic Script)

The following sample approves all GPOs in a category.

```
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     oGPO.Approve True
Next
wscript.echo "All GPOs have been approved."
```

Syntax (C# Method)

```
GPOObject.Approve(True|False)
```

Sample Code (C# Method)

The following sample approves all GPOs in a category.

```
public static void ApproveGPO()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach(IfaGPRGpo oGPO in oCategory)
            {
                oGPO.Approve (true);
            }
            Console.WriteLine("All GPOs have been approved.");
            Console.ReadKey();
        }
```

A.10.2 Approve GPO with Comments

Approve a GPO to be exported to Active Directory, or unapprove a GPO, and include comments in the history view. If you set the value of the parameter to `True`, the method approves the GPO. Else, if the value of the parameter is `False`, the method unapproves the GPO.

Syntax (Visual Basic Script)

```
GPOObject.ApproveWithComment True|False , "Comment"
```

Sample Code (Visual Basic Script)

The following sample approves all GPOs in a category and adds a comment.

```
Dim oGPRroot, oCategory, oGPO, sCategory, sComment
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     oGPO.ApproveWithComment True , "This GPO is approved for export."
Next
wscript.echo "All GPOs have been approved."
```

Syntax (C# Method)

```
GPOObject.ApproveWithComment(True|False, "Comment")
```

Sample Code (C# Method)

The following sample approves all GPOs in a category and adds a comment.

```
public static void ApproveGPOwithComments()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach(IfaGPRGpo5 oGPO in oCategory)
            {
                oGPO.ApproveWithComment(true, "Approved by .Net application");
            }
            Console.WriteLine("All GPOs have been approved.");
            Console.ReadKey();
        }
```

A.10.3 Check In GPO

Check in a GPO.

Syntax (Visual Basic Script)

```
GPOObject.CheckIn "Comment"
```

Sample Code (Visual Basic Script)

The following sample allows you to check in all checked-out GPOs in a category.

```
REM Check in all checked out GPOs in a category
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
    If oGPO.StatusCheckedOut = True then
        oGPO.CheckIn "Checked in from script"
    end if
Next
```

Syntax (C# Method)

```
GPOObject.CheckIn("Comment")
```

Sample Code (C# Method)

The following sample allows you to check in all checked-out GPOs in a category.

```
public static void CheckinGPOs()
    {
        string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
        IfaGPRRoot oGPRroot = new faGPRRoot();
        oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
        IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
        oCategory.EnumType = "GPO";
        foreach (IfaGPRGpo oGPO in oCategory)
        {
            if (oGPO.StatusCheckedOut == true)
            {
                oGPO.CheckIn("Checked in from .Net application");
            }
        }
        Console.WriteLine("All GPOs have been checked in.");
        Console.ReadKey();
    }
```

A.10.4   Check Out GPO

Check out a GPO.

Syntax (Visual Basic Script)

```
GPOObject.CheckOut "Comment"
```

Sample Code (Visual Basic Script)

The following sample allows you to check out all checked-in GPOs in a category.

```
REM Check out all GPOs in a category
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     If oGPO.StatusCheckedOut = False then
         oGPO.CheckOut "Checked out from script"
     end if
Next
```

Syntax (C# Method)

```
GPOObject.CheckOut("Comment")
```

Sample Code (C# Method)

The following sample allows you to check out all checked-in GPOs in a category.

```
public static void CheckOutGPOs()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo oGPO in oCategory)
            {
                if (oGPO.StatusCheckedOut == false)
                {
                    oGPO.CheckOut("Checked out from .Net application");
                    Console.WriteLine("GPO: " + oGPO.Name + " was checked out.");
                }
            }
            Console.WriteLine("All GPOs have been checked out.");
            Console.ReadKey();
        }
```

A.10.5   Create a GPO Link to a SOM Object

Links the GPO to a Scope of Management (SOM) object.

Syntax (Visual Basic Script)

```
GPOObject.CreateLink "SOM LDAP path"
```

Sample Code (Visual Basic Script)

The following code links a GPO to a SOM object.

```
Dim oGPRroot, oGPO, sGPOPath
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=MYREPOSITORYDB;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=MYWORKSTATION;Use Encryption for
Data=False;Tag with column collation when possible=False")
sGPOPath = "FAGPR://CN={3D6843CF-FB04-4AC6-9B47-1E6CE974D2F4}, CN=cat1,
DC=MYDOMAIN,DC=COM"
Set oGPO = oGPRroot.GetObject(sGPOPath)
' First check out the GPO
oGPO.CheckOut "Checkout comment"
' Call the CreateLink API with the LDAP path of the SOM object as input
oGPO.CreateLink "LDAP://OU=MyOU,DC=MYDOMAIN,DC=COM"
' Finally check in the GPO
oGPO.CheckIn "Checkin comment"
wscript.echo "Operation Completed"
```

Syntax (C# Method)

```
GPOObject.CreateLink("SOM LDAP path")
```

Sample Code (C# Method)

The following code links a GPO to a SOM object.

```
public static void CreateLinkToSOM()
        {
                string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
                string sGPOPath = "FAGPR://CN={C104C9C7-9355-4FEC-8824-22D7BF4797A9},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
                string sContainer = "LDAP://OU=MyOU,DC=MYDOMAIN,DC=LAB";
                IfaGPRRoot oGPRroot = new faGPRRoot();
                oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
                IfaGPRGpo5 oGpo = oGPRroot.GetObject(sGPOPath);
                oGpo.CheckOut("Checkout comment");
                oGpo.CreateLink(sContainer);
                oGpo.CheckIn("Checkin comment");
                IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
                Console.WriteLine("GPO was linked successfully");
                Console.ReadKey();
        }
```

A.10.6   Delete a GPO Link from a SOM Object

Deletes a GPO link from a Scope of Management (SOM) object.

Syntax (Visual Basic Script)

```
GPOObject.DeleteLink "SOM LDAP path"
```

Sample Code (Visual Basic Script)

The following sample removes a GPO link from a SOM object.

```
Dim oGPRroot, oGPO, sGPOPath
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;IntegratedSecurity=SSPI;Initial
Catalog=GPO_REPOSITORY;DataSource=MYREPOSITORYDB;Use Procedure for
Prepare=1;AutoTranslate=True;Packet Size=4096;Workstation
ID=MYWORKSTATION;UseEncryption for Data=False;Tag with column collation
whenpossible=False")
sGPOPath = "FAGPR://CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Cat1,
DC=MYDOMAIN,DC=COM"
Set oGPO = oGPRroot.GetObject(sGPOPath)
' First check out the GPO
oGPO.CheckOut "Checkout comment"
' Call the DeleteLink API with the LDAP path of the SOM object as input
oGPO.DeleteLink "LDAP://OU=MyOU,DC=MYDOMAIN,DC=COM"
' Finally check in the GPO
oGPO.CheckIn "Checkin comment"
wscript.echo "Operation Completed"
```

Syntax (C# Method)

```
GPOObject.DeleteLink("SOM LDAP path")
```

Sample Code (C# Method)

The following sample removes a GPO link from a SOM object.

```
public static void DeleteLinkToSOM()
        {
                string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
                string sGPOPath = "FAGPR://CN={C104C9C7-9355-4FEC-8824-22D7BF4797A9},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
                string sContainer = "LDAP://OU=Bolivia,DC=MYDOMAIN,DC=LAB";
                IfaGPRRoot oGPRroot = new faGPRRoot();
                oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
                IfaGPRGpo5 oGpo = oGPRroot.GetObject(sGPOPath);
                oGpo.CheckOut("Checkout comment");
                oGpo.DeleteLink(sContainer);
                oGpo.CheckIn("Checkin comment");
                IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
                Console.WriteLine("GPO was unlinked successfully");
                Console.ReadKey();
        }
```

A.10.7   Delete GPO

Delete a GPO.

Syntax (Visual Basic Script)

```
GPOObject.Delete
```

Sample Code (Visual Basic Script)

The following sample allows you to delete a GPO.

```
Dim oGPRroot, oCategory, oGPO, sGPO
sGPO = "FAGPR://CN={6E936ED3-00C8-4FE7-95A1-803874AB7EA0},
CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;s")
Set oGPO = oGPRroot.GetObject(sGPO)
oGPO.Delete
```

Syntax (C# Method)

```
GPOObject.Delete()
```

Sample Code (C# Method)

The following sample allows you to delete a GPO.

```
public static void DeleteGPO()
        {
            string sGPOPath = "FAGPR://CN={C104C9C7-9355-4FEC-8824-22D7BF4797A9},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRGpo5 oGPO = oGPRroot.GetObject(sGPOPath);
            oGPO.Delete();
            Console.WriteLine("GPO deleted successfully");
            Console.ReadKey();
        }
```

A.10.8   Export GPO

Export approved GPO to live Active Directory domain.

Syntax (Visual Basic Script)

```
GPOObject.Export "ExportParameter"
```

Sample Code (Visual Basic Script)

Export all GPOs in a category. This sample works for all GPOs with an approved status.

```
Dim oGPRroot, oCategory, oGPO, sCategory, sExportOpt
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sExportOpt = "NoBackUpOverwrite"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     oGPO.Export sExportOpt
Next
wscript.echo "All GPOs have been exported."
```

Syntax (C# Method)

```
GPOObject.Export("ExportParameter")
```

Sample Code (C# Method)

Export all GPOs in a category. This sample works for all GPOs with an approved status.

```
public static void ExportGPOs()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sExportOpt = "NoBackUpOverwrite";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo gprGpo in oCategory)
            {
                if (gprGpo.StatusApproved == true)
                    gprGpo.Export(sExportOpt);
            }
            Console.WriteLine("All GPOs have been exported.");
            Console.ReadKey();
        }
```

Export Parameter

You can specify one of the following export parameters:

BackUpOverwrite

> If the GPO already exists in Active Directory, overwrite it and back up the live Active Directory GPO into the GP Repository prior to overwriting it. You can also use an integer value of 14 instead of BackUpOverwrite.

NoBackUpOverwrite

> If the GPO already exists in Active Directory, overwrite it. The live GPO is not backed up prior to import. You can also use an integer value of 13 instead of NoBackUpOverwrite.

`DoNotOverwrite`

> Export fails if the GPO already exists in Active Directory. You can also use an integer value of `12` instead of `DoNotOverwrite`.

A.10.9 Export GPO with Comments

Export approved GPO to live Active Directory domain and include comments in the history view.

NOTE: This script can take several seconds or longer to complete when you run it for the first time.

Syntax (Visual Basic Script)

GPOObject.ExportWithComment "*ExportParameter*" , "*Comment*"

Sample Code (Visual Basic Script)

Export all GPOs in a category and include a comment. This sample works for all GPOs with an approved status.

```
Dim oGPRroot, oCategory, oGPO, sCategory, sExportOpt, sComment
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sExportOpt = "NoBackUpOverwrite"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     oGPO.ExportWithComment sExportOpt , "This GPO has been exported to Active
Directory."
Next
wscript.echo "All GPOs have been exported."
```

Syntax (C# Method)

GPOObject.ExportWithComment("*ExportParameter*", "*Comment*")

Sample Code (C# Method)

Export all GPOs in a category and include a comment. This sample works for all GPOs with an approved status.

```
public static void ExportGPOsWithComments()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            string sExportOpt = "NoBackUpOverwrite";
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo5 gprGpo in oCategory)
            {
                if (gprGpo.StatusApproved == true)
                  gprGpo.ExportWithComment(sExportOpt, "This GPO has been exported
to Active Directory.");
            }
            Console.WriteLine("All GPOs have been exported.");
            Console.ReadKey();
        }
```

Export Parameter

You can specify one of the following export parameters:

BackUpOverwrite

If the GPO already exists in Active Directory, Overwrite it and backup the live Active Directory GPO into the GP Repository prior to overwriting it. You can also use an integer value of 14 instead of BackUpOverwrite.

NoBackUpOverwrite

If the GPO already exists in the Active Directory overwrite it. The live GPO is not backed up prior to Import. You can also use an integer value of 13 instead of NoBackUpOverwrite.

DoNotOverwrite

Export fails if the GPO already exists in Active Directory. You can also use an integer value of 12 instead of DoNotOverwrite.

The Export Batch File

This batch file uses the GPAExportUtil.exe tool to create an entry for each approved GPO you have selected to export. If you want to export all approved GPOs in the selected domains, the batch file uses the GPAExportUtil.exe tool to create an entry for each selected domain.

Syntax

```
"<product installation path>\GPAExportUtil.exe" {{/g:<guid of GPO> |

/d:<DNS name of AD domain> | /a}

{/C:"<SQL Connection string>" | {/SQLS:<repository_server>
```

```
/SQLD:<rep_database_name> [/U:<SQL username> /P:<SQL password>]}} |
```

```
[/?|/H]
```

Options

The following table describes the command-line parameters and variables.

Variable name	Replace with
/g:<guid of GPO>	The GUID of the approved GPO you want to export using GPAExportUtil.exe. Use along with the /d option when you want to export two or more GPOs with the same GUID, but from different domains (required when exporting individual GPOs)
/d:<DNS name of AD domain>	The DNS name of the domain where approved GPOs will be exported. When this parameter is not specified, approved GPOs will be exported to the domain of the user performing the export. You can use this parameter when exporting any built-in domain policy GPOs or GPOs with same GUID.
/a	All approved GPOs in all domains of the specified GP Repository will be exported (optional).
/C:"<SQL Connection string>"	Full SQL Server connection string to the GP Repository database, in double quotes. Use instead of the other SQL Server parameters (required).
/SQLS:<repository_server>	Name of the GP Repository SQL Server (optional).
/SQLD:<rep_database_name>	Name of the GP Repository SQL Server database (optional).
/U:<SQL username>	SQL Server account name to use for SQL Authentication (optional).
/P:<SQL password>	SQL Server account password to use for SQL Server Authentication. Use caution when specifying this parameter in batch files (optional).

Sample Code

To export two selected GPOs from the domain, the export batch file contains the following entries:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{1FEB5933-DA75-49BC-A63F-
FA86C7CA9E20} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{F94F2CF6-0264-4DA6-B76C-
7C920360894D} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
```

```
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

To export all GPOs in a domain, the export batch file contains the following entry:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /d:usregion.com                    /
Connect:"Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=TREK02;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=TREK02;Use Encryption for
Data=False;Tag with column collation when possible=False"
```

To export two GPOs (in this case, the default domain policy) with the same GUID, but from different domains, the export batch file contains the following entries:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{31B2F340-016D-11D2-945F-
00C04FB984F9} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{31B2F340-016D-11D2-945F-
00C04FB984F9} /d:nordicregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

A.10.10 Get GPO Check Out Status

Allows you to view whether a GPO is checked out. This operation returns a `True` or `False` value.
`True` indicates a GPO is checked out and `False` indicates that the GPO is checked in.

Syntax (Visual Basic Script)

GPOObject.StatusCheckedOut

Sample Code (Visual Basic Script)

The following sample displays whether a GPO is checked out.

```
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
    If oGPO.StatusCheckedOut = True then
        Wscript.Echo oGPO.Name + "is checked out"
    else
        Wscript.Echo oGPO.Name + "is checked in"
    end if
Next
```

Syntax (C# Method)

GPOObject.StatusCheckedOut

Sample Code (C# Method)

The following sample displays whether a GPO is checked out.

```
public static void GetGPOCheckoutStatus()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo gprGpo in oCategory)
            {
                if (gprGpo.StatusCheckedOut == false)
                {
                Console.WriteLine(gprGpo.Name + "is checked in");
                }
                else
                {
                Console.WriteLine(gprGpo.Name + "is checked out");
                }
            }
            Console.ReadKey();
        }
```

A.10.11   Get GPO Approval Status

Allows you to read the approval status of a GPO. This operation returns a `True` or `False` value. A
`True` value denotes Approved status and a `False` value denotes Unapproved status.

Syntax (Visual Basic Script)

GPOObject.StatusApproved

Sample Code (Visual Basic Script)

The following sample displays the approval status of a GPO.

```
Dim oGPRroot, oCategory, oGPO, sCategory, sExportOpt
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sExportOpt = "NoBackUpOverwrite"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
    If oGPO.StatusApproved = True then
        Wscript.Echo oGPO.Name + " is approved"
    else
        Wscript.Echo oGPO.Name + " is not approved"
    end if
Next
```

Syntax (C# Method)

```
GPOObject.StatusApproved
```

Sample Code (C# Method)

The following sample displays the approval status of a GPO.

```
public static void GetGPOApprovalStatus()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo gprGpo in oCategory)
            {
                if (gprGpo.StatusApproved == true)
                {
                Console.WriteLine(gprGpo.Name + " is approved");
                }
                else
                {
                Console.WriteLine(gprGpo.Name + " is not approved");
                }
            }
            Console.ReadKey();
        }
```

A.10.12   Lock or Mask GPO

Sets or gets status of locking or masking of a GPO for a user or group.

Syntax (Visual Basic Script)

Sets locking or masking of a GPO for a user or group.

```
Int Result = GPOObject.SetGPOSecurityFilterInfo(string AccountName, Int
OpType, Int Overwrite)
```

Gets status of locking or masking of a GPO for a user or group.

```
Int Result = GPOObject.GetGPOSecurityFilterInfo(string AccountName, Int
OpType)
```

Parameter	Value
AccountName	User or Group
OpType	0 – Lock
	1 – Mask
	2 – Unlock
	3 – Unmask
Overwrite	0 – No overwrite
	1 – overwrite
Result	1 – Operation successful
	0 – Operation unsuccessful

Sample Code (Visual Basic Script)

The following code locks a GPO for a user or group object.

```
Dim oGPRroot, oCategory, oGPO, sGPOPath, result
sGPOPath =  "FAGPR://CN={7DEE509A-2817-416F-B969-DDCEA57FE6A3},
CN=MyCategory, DC=MYDOMAIN,DC=LAB"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("PROVIDER = SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data
Source=MABOSLPT03;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=MABOSLPT03;Use
Encryption for Data=False;Tag with column collation when
possible=False")
Set oGPO = oGPRroot.GetObject(sGPOPath)
result = oGPO.SetGPOSecurityFilterInfo("MYDOMAIN\<UserorGroup>", 0, 1)
Wscript.Echo result
```

The following code checks to see if a GPO is unmasked for a user or group object.

```
Dim oGPRroot, oCategory, oGPO, sGPOPath, result
sGPOPath =  "FAGPR://CN={7DEE509A-2817-416F-B969-DDCEA57FE6A3},
CN=MyCategory, DC=MYDOMAIN,DC=LAB"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("PROVIDER = SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data
Source=MABOSLPT03;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=MABOSLPT03;Use
Encryption for Data=False;Tag with column collation when
possible=False")
Set oGPO = oGPRroot.GetObject(sGPOPath)
result = oGPO.GetGPOSecurityFilterInfo("MYDOMAIN\<UserorGroup>", 3)
Wscript.Echo result
```

Syntax (C# Script)

Sets locking or masking of a GPO for a user or group.

```
Int Result = GPOObject.SetGPOSecurityFilterInfo(string AccountName, Int
OpType, Int OverWrite);
```

Gets status of locking or masking of a GPO for a user or group.

```
Int Result = GPOObject.GetGPOSecurityFilterInfo(string AccountName, Int
OpType);
```

Parameter	Value
AccountName	User or Group
OpType	0 – Lock
	1 – Mask
	2 – Unlock
	3 – Unmask
Overwrite	0 – No overwrite
	1 – overwrite
Result	1 – Operation successful
	0 – Operation unsuccessful

Sample Code (C# Script)

The following code locks a GPO for a user or group object.

```
public static void LockGPO()
{
string sGPOPath =  "FAGPR://CN={7DEE509A-2817-416F-B969-DDCEA57FE6A3},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
IfaGPRRoot oGPRroot = new faGPRRoot();
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure for
Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use Encryption
for
Data=False;Tag with column collation when possible=False");
IfaGPRGpo8 oGPO = (IfaGPRGpo8)oGPRroot.GetObject(sGPOPath);
int result = oGPO.SetGPOSecurityFilterInfo("MYDOMAIN\\<UserorGroup>", 0,
1);
Console.WriteLine(result.ToString());
Console.ReadKey();
}
```

The following code checks to see if a GPO is unmasked for a user or group object.

```
public static void UnLockGPO()
{
string sGPOPath =  "FAGPR://CN={7DEE509A-2817-416F-B969-DDCEA57FE6A3},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
IfaGPRRoot oGPRroot = new faGPRRoot();
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure for
Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use Encryption
for
Data=False;Tag with column collation when possible=False");
IfaGPRGpo8 oGPO = (IfaGPRGpo8)oGPRroot.GetObject(sGPOPath);
int result = oGPO.GetGPOSecurityFilterInfo("MYDOMAIN\\<UserorGroup>", 3);
Console.WriteLine(result.ToString());
Console.ReadKey();
}
```

A.10.13   Read GPO CN Name

Retrieves the CN name (GUID) of a GPO.

Syntax (Visual Basic Script)

StrName = GPOObject.CNName

Sample Code (Visual Basic Script)

The following sample lists the CN names of all GPOs in a category.

```
REM List names of GPOs in a Category
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     Wscript.Echo oGPO.Name & ", " & oGPO.CNName
Next
```

Syntax (C# Method)

```
StrName = GPOObject.CNName
```

Sample Code (C# Method)

The following sample lists the CN names of all GPOs in a category.

```
public static void ReadGPOCNName()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo2 oGPO in oCategory)
            {
                Console.WriteLine(oGPO.Name + ", " + oGPO.CNName);
            }
            Console.ReadKey();
        }
```

A.10.14   Generate GPO Report

Generate an HTML report for a specific version of a GPO. Specify the version number of the GPO that you want to generate a report for. If you want to generate a Report for the latest version of the GPO then pass the value as 0.

Syntax (Visual Basic Script)

```
GPOObject.ReportHtml VersionNumber, HTMLFile
```

Sample Code (Visual Basic Script)

The following sample generates reports for all GPOs in a category.

```
Dim oGPRroot, oCategory, oGPO, sCategory, sPath, i, sCurrentFile
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sPath = "C:\Diffreport_User\"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
i= 1
for each oGPO in oCategory
    sCurrentFile = sPath + oGPO.Name + cstr(i) + ".htm"
    oGPO.ReportHtml 0, sCurrentFile
    i = i +1
Next
```

Syntax (C# Method)

```
GPOObject.ReportHtml(VersionNumber, HTMLFile)
```

Sample Code (C# Method)

The following sample generates reports for all GPOs in a category.

```
public static void GenerateGPOReport()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/";
            string sCurrentFile = "";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            int i = 1;
            foreach (IfaGPRGpo gprGpo in oCategory)
            {
                sCurrentFile = sPath + gprGpo.Name + Convert.ToString(i) + ".htm";
                gprGpo.ReportHtml(0, sCurrentFile);
                i++;
            }
            Console.WriteLine("GPOs reports were generated successfully");
            Console.ReadKey();
        }
```

A.10.15   Compare or Differentiate Two GPOs

Compare two GPOs and generate an HTML report of the comparison including the similarities and differences. The two GPOs must exist before running this operation. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax (Visual Basic Script)

GPOObject1.Compare2GPOsReportHTML *GPOObject2*, *HTMLfile*, *DiffParameter*

Sample Code (Visual Basic Script)

The following sample generates an HTML report that compares two GPOs.

```
Dim oGPRroot, oGPO1, sGPO1, oGPO2, sGPO2
sGPO1 = "FAGPR://CN={B64E5669-C0BB-4549-BEF0-
E9E3554AA70A},CN=cat1,DC=rootdev2,DC=com"
sGPO2 = "FAGPR://CN={B6F9BDBA-BF2B-4973-83C3-
FA07236B6BF8},CN=cat1,DC=rootdev2,DC=com"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oGPO1 = oGPRroot.GetObject(sGPO1)
Set oGPO2 = oGPRroot. GetObject(sGPO2)
oGPO1.Compare2GPOsReportHtml oGPO2, "c:\\report\diff.htm", FALSE
```

Syntax (C# Method)

GPOObject1.Compare2GPOsReportHTML(*GPOObject2*, *HTMLfile*, *DiffParameter*)

Sample Code (C# Method)

The following sample generates an HTML report that compares two GPOs.

```
public static void ComparisionReportOfTwoGPOs()
        {
            string sGPO1 = "FAGPR://CN={E9DAE4E3-1D76-46EA-8B06-37B30D80E764},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sGPO2 = "FAGPR://CN={4F246D45-332E-45AC-B728-B7A0A612C61E},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/comparisionTwoGPOs.htm";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRGpo oGPO1 = oGPRroot.GetObject(sGPO1);
            IfaGPRGpo oGPO2 = oGPRroot.GetObject(sGPO2);
            oGPO1.Compare2GPOsReportHtml(oGPO2, sPath, false);
            Console.WriteLine("GPO comparision Report was generated successfully");
            Console.ReadKey();
        }
```

A.10.16   Compare GPO GP Repository Versions

Generate a comparison HTML report with two different GP Repository versions of the GPO. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax (Visual Basic Script)

GPOObject.CompareVersionReportHtml *Version1*, *Version2*, "*HTML Report name*", *DiffParameter*

Sample Code (Visual Basic Script)

The following sample generates an HTML report that compares two GP Repository versions of a GPO.

```
Dim oGPRroot, oCategory, oGPO, sGPO
sGPO = "FAGPR://CN={6E936ED3-00C8-4FE7-95A1-
803874AB7EA0},CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oGPO = oGPRroot.GetObject(sGPO)
oGPO.CompareVersionReportHtml 1,2, "c:\report.htm", True
```

Syntax (C# Method)

GPOObject1.CompareVersionReportHTML(*Version1, Version2*, *HTMLfile*, *DiffParameter*)

Sample Code (C# Method)

The following sample generates an HTML report that compares two GP Repository versions of a GPO.

```
public static void ComparisionReportOfGPORepositoryVersions()
       {
           string sGPO1 = "FAGPR://CN={E9DAE4E3-1D76-46EA-8B06-37B30D80E764},
CN=MyCategory, DC=MYDomain,DC=COM";
           string sPath = "C://Folder/ComparisionRepGPOVersions.htm";
           IfaGPRRoot oGPRroot = new faGPRRoot();
           oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use Encryption for
Data=False;Tag with column collation when possible=False");
           IfaGPRGpo oGPO1 = oGPRroot.GetObject(sGPO1);
           oGPO1.CompareVersionReportHtml(1, 2, sPath, false);
           Console.WriteLine("Comparision report was generated successfully");
           Console.ReadKey();
       }
```

A.10.17   Compare and Differentiate Active Directory GPO Versions

Generate a comparison HTML report with GPOs from Active Directory. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax (Visual Basic Script)

GPOObject1.CompareADReportHTML *VersionNumber*, *HTMLfile*, *DiffParameter*

Sample Code (Visual Basic Script)

Generate a report that compares all GPOs in a category with the Active Directory versions of the GPOs.

```
Dim oGPRroot, oCategory, oGPO, sConnect, sCategory, sPath
Dim i, sCurrentFile
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sPath = "C:\Diffreport_User\"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
i = 1
For Each oGPO in oCategory
     sCurrentFile = sPath + oGPO.Name + cstr(i) + ".htm"
     oGPO.CompareADReportHtml 0, sCurrentFile, FALSE
     i = i + 1
Next
wscript.echo "Operation Completed"
```

Syntax (C# Method)

GPOObject1.CompareADReportHTML(*VersionNumber*, *HTMLfile*, *DiffParameter*)

Sample Code (C# Method)

Generate a report that compares all GPOs in a category with the Active Directory versions of the GPOs.

```
public static void ComparisionReportOfGPOActiveDirectoryVersions()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sPath = "C://Folder/";
            string sCurrentFile;
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            int i = 1;
            foreach (IfaGPRGpo oGPO in oCategory)
            {
                sCurrentFile = sPath + oGPO.Name + Convert.ToString(i) + ".htm";
                oGPO.CompareADReportHtml(0, sCurrentFile, true);
            }
            Console.WriteLine("Comparison report of GPOs in a category with the AD
versions generated successfully");
            Console.ReadKey();
        }
```

A.10.18   Copy a GPO

Creates a copy of a GPO in the target category in the GP Repository.

Syntax (Visual Basic Script)

GPOObject.CopyTo *TargetContainerObject*

or

Set *NewGPOObject* = *OriginalGPOObject*.CopyTo(*TargetContainerObject*)

Sample Code (Visual Basic Script)

The following code creates a copy of the GPO in the target category.

```
Dim oGPRroot, sCategory, sGPO, oCategory, oGPO
sCategory = "FAGPR://CN=ExampleOU,CN=RELEASE,DC=Repository,DC=Net"
sGPO = "FAGPR://CN={6E936ED3-00C8-4FE7-95A1-803874AB7EA0},
CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
Set oGPO = oGPRroot.GetObject(sGPO)
oGPO.CopyTo oCategory
```

Syntax (C# Method)

```
GPOObject.CopyTo(TargetContainerObject)
or
MyGPOObject = OriginalGPOObject.CopyTo(TargetContainerObject)
```

Sample Code (C# Method)

The following code creates a copy of the GPO in the target category.

```
public static void CopyGPO()
        {
            string sCategoryTarget = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sGPO = "FAGPR://CN={C1AF6C94-7738-4E7A-83CB-4EA154B2F2D5},
CN=NewCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            IfaGPRGpo oGPO1 = oGPRroot.GetObject(sGPO1);
            oGPO1.CopyTo(oCategoryTarget);
            Console.WriteLine("GPO copied successfully");
            Console.ReadKey();
        }
```

A.10.19   Enumerate GPO Links

Enumerate a list of GPO links within a GPO and generates a list of Link Status and Link Order
properties for each GPO link.

Syntax (Visual Basic Script)

```
oGPO.EnumerateLinks p_pvarLinks, p_pvarLinkStatus, p_pvarLinkOrder
```

Sample Code (Visual Basic Script)

The following code enumerates the GPO links for a GPO.

```
Dim oGPRroot, oGPO, sGPOPath
Dim p_pvarLinks, p_pvarLinkStatus, p_pvarLinkOrder, linkldap
Set oGPRroot = Wscript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=MYREPOSITORYDB;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=MYWORKSTATION;Use Encryption for
Data=False;Tag with column collation when possible=False")
sGPOPath = "FAGPR://CN={6A63640A-09E5-4833-B43F-BEB96DE47AC8}, CN=cat2,
DC=MYDOMAIN,DC=COM"
Set oGPO = oGPRroot.GetObject(sGPOPath)
oGPO.EnumerateLinks p_pvarLinks, p_pvarLinkStatus, p_pvarLinkOrder
For Each linkLdap in p_pvarLinks
    WScript.Echo linkLdap
Next
```

Syntax (C# Method)

```
oGPO.EnumerateLinks(p_pvarLinks, p_pvarLinkStatus, p_pvarLinkOrder)
```

Sample Code (C# Method)

The following code enumerates the GPO links for a GPO.

```
public static void Enumerate_GPOLinks()
        {
            string sGPOPath = "FAGPR://CN={4F246D45-332E-45AC-B728-B7A0A612C61E},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            dynamic p_pvarLinks;
            dynamic p_pvarLinkStatus;
            dynamic p_pvarLinkOrder;
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRGpo3 oGPO = oGPRroot.GetObject(sGPOPath);
            oGPO.EnumerateLinks(out p_pvarLinks, out p_pvarLinkStatus, out
p_pvarLinkOrder);
            foreach (object linkLdap in p_pvarLinks)
            {
                Console.WriteLine(linkLdap.ToString());
            }
            Console.ReadKey();
        }
```

Parameters

p_pvarLinks

> Returns a collection of GPO Links.

p_pvarLinkStatus

> Returns a collection of Link Status.

p_pvarLinkOrder

> Returns a collection of Link Order.

A.10.20 Link a GPO to a Category

Links a GPO to a category.

Syntax (Visual Basic Script)

```
GPOObject.LinkGPO "GPOGUID"
```

Sample Code (Visual Basic Script)

The following code links a GPO to a category.

```
Dim oGPRroot, oCategory
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject("FAGPR://CN=Desktop,DC=NetIQLabs,DC=com")
oCategory.LinkGPO "{31B2F340-016D-11D2-945F-00C04FB984F9}"
```

Syntax (C# Method)

```
GPOObject.LinkGPO("GPOGUID")
```

Sample Code (C# Method)

The following code links a GPO to a category.

```
public static void LinkGPOtoCategory()
        {
            string sCategory = "FAGPR://CN=NewCategory, DC=MYDOMAIN,DC=LAB";
            string sGPO = "{4F246D45-332E-45AC-B728-B7A0A612C61E}, CN=MyCategory,
DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.LinkGPO(sGPO);
            Console.WriteLine("GPO was linked to category successfully");
            Console.ReadKey();
        }
```

A.10.21 Migrate GPO

Migrate a GPO across different domains or to a different category within the same domain. You can specify either a category or a GPO as the target. In either case, specify the GP Repository path of the target object.

Syntax for Migrating a New GPO

GPOObject.MigrateTo *TargetCategory*

Syntax for Migrating an Existing GPO

GPOObject.MigrateToEx *TargetGPO,* True│False

Sample Code, Scenario 1 (Visual Basic Script)

The following sample migrates all GPOs in a category to another category across domains in the same database.

```
Dim oGPRroot, oSourceCat, oGPO, sSourceCat, oTargetCat, sTargetCat
sSourceCat = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sTargetCat = "FAGPR://CN=UserOU,CN=RELEASE,DC=USA,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oSourceCat = oGPRroot.GetObject(sSourceCat)
Set oTargetCat = oGPRroot.GetObject(sTargetCat)
oSourceCat.EnumType = "GPO"
For Each oGPO in oSourceCat
     oGPO.MigrateTo(oTargetCat)
Next
wscript.echo "Operation completed."
```

Sample Code, Scenario 1 (C# Method)

The following sample migrates all GPOs in a category to another category across domains in the same database.

```
public static void MigrateGPO_CattoCatAcrossDomainsSameDB()
        {
             string sCategorysource = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
             string sCategorytarget = "FAGPR://CN=MyCategory,
DC=MYTARGETDOMAIN,DC=LAB";
             IfaGPRRoot oGPRroot = new faGPRRoot();
             oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
             IfaGPRCategory oCategorysource = oGPRroot.GetObject(sCategorysource);
             IfaGPRCategory oCategorytarget = oGPRroot.GetObject(sCategorytarget);
             oCategorysource.EnumType = "GPO";
             foreach (IfaGPRGpo oGPO in oCategorysource)
             {
                 oGPO.MigrateTo(oCategorytarget);
             }
        Console.WriteLine("GPOs were migrated successfully.");
        Console.ReadKey();
        }
```

Sample Code, Scenario 2 (Visual Basic Script)

The following sample migrates all GPOs in a category to another category across domains in a different database.

```
Dim oSourceGPRroot, oTargetGPRroot
Dim sSourceConnect, sTargetConnect
Dim oSourceCat, oGPO, sSourceCat, oTargetCat, sTargetCat
sSourceConnect = "DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;"
sTargetConnect = "DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;"
sSourceCat = "FAGPR://CN=cat1,CN=ImportedGPOs,DC=rootdev2,DC=com"
sTargetCat = "FAGPR://CN=cat2,CN=ImportedGPOs,DC=rootdev22,DC=com2"
Set oSourceGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oSourceGPRroot.ConnectTo(sSourceConnect)
Set oTargetGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oTargetGPRroot.ConnectTo(sTargetConnect)
Set oSourceCat = oSourceGPRroot.GetObject(sSourceCat)
Set oTargetCat = oTargetGPRroot.GetObject(sTargetCat)
oSourceCat.EnumType = "GPO"
For Each oGPO in oSourceCat
     oGPO.MigrateTo oTargetCat
Next
wscript.echo "Operation completed."
```

Sample Code, Scenario 2 (C# Method)

The following sample migrates all GPOs in a category to another category across domains in a different database.

```
public static void MigrateGPO_CattoCatAcrossDomainsDiffDB()
       {
             string sCategorysource = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
             string sCategorytarget = "FAGPR://CN=MyCategory,
DC=MYTARGETDOMAIN,DC=LAB";
             IfaGPRRoot oGPRrootSource = new faGPRRoot();
             IfaGPRRoot oGPRrootTarget = new faGPRRoot();
             oGPRrootSource.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
             oGPRrootTarget.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data
Source=JALQEGP611\test_instance;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=JALQEGP611;Use Encryption for
Data=False;Tag with column collation when possible=False");
             IfaGPRCategory oCategorySource =
oGPRrootSource.GetObject(sCategorysource);
             IfaGPRCategory oCategoryTarget =
oGPRrootTarget.GetObject(sCategorytarget);
             oCategorySource.EnumType = "GPO";
             foreach (IfaGPRGpo oGPO in oCategorySource)
             {
                 oGPO.MigrateTo(oCategoryTarget);
             }
             Console.WriteLine("GPOs were migrated successfully.");
             Console.ReadKey();
       }
```

Sample Code, Scenario 3 (Visual Basic Script)

The following sample migrates a GPO in a category to another category that already contains the same GPO. If the value is set to True, the target GPO will be renamed to match the source GPO. If the value is set to False, the target GPO name will be retained.

```
Dim oGPRroot, oGPO, zGPO, sGPOTarget, sGPOSource
sGPOSource="FAGPR://CN={251C91F3-F547-415F-BCA9-3B349B916E8D},
CN=Target, DC=GPDOM700,DC=LAB"
sGPOTarget="FAGPR://CN={251C91F3-F547-415F-BCA9-3B349B916E8D},
CN=Target2, DC=GPDOM7002,DC=LAB2"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oGPO = oGPRroot.GetObject(sGPOSource)
Set zGPO = oGPRroot.GetObject(sGPOTarget)
oGPO.MigrateToEx zGPO, false
wscript.echo "Operation completed."
```

Sample Code, Scenario 3 (C# Method)

The following sample migrates a GPO in a category to another category that already contains the same GPO. If the value is set to True, the target GPO will be renamed to match the source GPO. If the value is set to False, the target GPO name will be retained.

```
public static void MigrateGPO_CattoCatwhichAlreadyContainsaGPO()
        {
            string sGPOsource = "FAGPR://CN={4F246D45-332E-45AC-B728-B7A0A612C61E},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sGPOtarget = "FAGPR://CN={C1AF6C94-7738-4E7A-83CB-4EA154B2F2D5},
CN=NewCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRGpo3 oGPOSource = oGPRroot.GetObject(sGPOsource);
            IfaGPRGpo3 oGPOtarget = oGPRroot.GetObject(sGPOtarget);
            oGPOSource.MigrateToEx(oGPOtarget, false);
            Console.WriteLine("GPOs were migrated successfully.");
            Console.ReadKey();
        }
```

A.10.22   Paste to an Existing GPO

Copies information from a GPO to a different GPO. The `PasteOptions` parameter allows you to specify which information should be copied. `PasteOptions` can contain any combination of the following values: `DATA`, `Links`, `Name`, `Security`, `WMI`. To specify multiple values, separate them with the pipe symbol (|), such as "`DATA|Links|Name|Security|WMI`". When you specify the `Name` value, `PasteOptions` adds the prefix, "Copy of" to the GPO name. For example, `PasteOptions` copies `MyGPO` as `Copy of MyGPO`. You can also specify `ALL` to copy all information.

Syntax (Visual Basic Script)

GPOObject.PasteToGpo *TargetGPOObject*, *PasteOptions*

Sample Code (Visual Basic Script)

The following sample allows you to copy the name and links from the source GPO to the target GPO.

```
Dim oGPRroot, sGPO, sGPOTarget, oGPO, oGPOTarget
sGPO = "FAGPR://CN={6E936ED3-00C8-4FE7-95A1-803874AB7EA0},
CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
sGPOTarget = "FAGPR:// CN={B0DF1662-1F2A-4A4A-8073-357E138AB148},
CN=ExampleOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oGPO = oGPRroot.GetObject(sGPO)
Set oGPOTarget = oGPRroot.GetObject(sGPOTarget)
oGPO.PasteToGpo oGPOTarget, "Name|Links"
```

Syntax (C# Method)

```
GPOObject.PasteToGpo(TargetGPOObject, PasteOptions)
```

Sample Code (C# Method)

The following sample allows you to copy the name and links from the source GPO to the target GPO.

```
public static void PastetoanExistingGPO()
        {
            string sGPO = "FAGPR://CN={4F246D45-332E-45AC-B728-B7A0A612C61E},
CN=MyCategory, DC=MYDOMAIN,DC=LAB";
            string sGPOTarget = "FAGPR://CN={1E28B502-B8B2-4957-9C94-D39B1BD8F18A},
CN=NewCategory, DC=MYDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRGpo oGPO = oGPRroot.GetObject(sGPO);
            IfaGPRGpo oGPOTarget = oGPRroot.GetObject(sGPOTarget);
            oGPO.PasteToGpo(oGPOTarget, "ALL");
            Console.WriteLine("GPO pasted successful");
            Console.ReadKey();
        }
```

A.10.23   Read GPO Name

Retrieves the name of a GPO.

Syntax (Visual Basic Script)

```
StrName = GPOObject.Name
```

Sample Code (Visual Basic Script)

The following sample allows you to list GPO names in a category.

```
REM List names of GPOs in a Category
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
    Wscript.Echo oGpo.Name
Next
```

Syntax (C# Method)

```
StrName = GPOObject.Name
```

Sample Code (C# Method)

The following sample allows you to list GPO names in a category.

```
public static void ReadGPOName()
        {
            string sCategory = "FAGPR://CN=MyCategory, DC=MYTARGETDOMAIN,DC=LAB";
            IfaGPRRoot oGPRroot = new faGPRRoot();
            oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
            IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
            oCategory.EnumType = "GPO";
            foreach (IfaGPRGpo2 oGPO in oCategory)
            {
                Console.WriteLine(oGPO.Name);
            }
            Console.ReadKey();
        }
```

A.10.24   Synchronize GPO Link Order

The Synchronize GPO Link Order tool (NqGPASyncLinkOrder.exe) allows you to synchronize the GPO link order in the GP Repository to match the GPO link order in Active Directory. GPA automatically runs this tool when you indicate for it to run from the Offline Mirror wizard, when you create an offline mirror from the command line, or when you execute this tool from the command line. Using the Offline Mirror wizard simplifies synchronizing GPO link order. For more information on synchronizing link order using the Offline Mirror wizard, see Section 5.9.3, "Synchronizing GPO Link Order Using the Offline Mirror Wizard," on page 94. For general information about synchronizing GPO link orders, see Section 5.8.4, "Synchronizing GPOs with AD Before Export," on page 90.

NOTE

◆ To synchronize all GPOs under all the GP Repository domains, do not specify a domain (/D) or Active Directory (AD) container (/ADContainer).

◆ The Synchronize GPO Link Order tool also synchronizes the block inheritance settings in the GP Repository to match the block inheritance settings in Active Directory during the upgrade process.

To synchronize GPO link order between the GP Repository and AD using the command-line tool:

1 Log on to a GPA Console computer with an account that has permissions to modify GPOs in the GP Repository and to read Active Directory in the domain where you want to synchronize GPO link order.

2 Open a command prompt window.

3 Navigate to the `\Tools` folder under the product installation path. If you used the default installation path, navigate to `C:\Program Files\NetIQ\Group Policy Administrator\Tools`.

4 Run the Synchronize GPO Link Order tool, `NqGPASyncLinkOrder.exe`. For general information about the synchronizing link order, see Section 5.8.4, "Synchronizing GPOs with AD Before Export," on page 90.

Your command may be similar to one of the following examples:

```
NqGPASyncLinkOrder /ADContainer:OU=DV - Link Order, DC=GPDOM800,
DC=Lab"
/S:. /DB:GPO_REPOSITORY
NqGPASyncLinkOrder /D:OU=DV - LO - LEVEL 2, OU=DV - LO - LEVEL 1, OU=DV
- LO - LINK ORDER, DC=GPDOM800, DC=Lab" /S:. /DB:GP_REPOSITORY
```

5 The Synchronize GPO Link Order tool displays a completion status in the command prompt window.

NOTE: For more information about using the Offline Mirror wizard to synchronize link order from the GPA console, see Section 5.9.3, "Synchronizing GPO Link Order Using the Offline Mirror Wizard," on page 94.

6 The Synchronize GPO Link Order tool creates a log file in the `%Temp%` folder for the current user. Refer to this file for before and after details about link order changes.

Syntax

```
NqGPASyncLinkOrder [/D:MyDomain.com | /ADContainer:LDAPPath][/S:MyDBServer
/DB:MyDataBase /U:MyUser /P:MyPassword]
```

Options

The following table describes the command-line parameters and variables.

Variable name	Replace with
`/D:`*Domain_DNS_Name*	DNS name of the source domain, such as `domainname.local, which synchronizes all GPOs in the domain.` If you do not specify this parameter, the tool synchronizes the GPO link order of all GPOs in all domains. (optional)
`/ADContainer:`*LDAPPath*	LDAP path of the AD Container (Domain, Site, or OU) for the sync operation, which links the GPOs directly under the specified domain.(optional). For example: `LDAP://OU=OUname, DC=DomainName, D=local` (optional).
`/S:`*Repository_Server*	Name of the Microsoft SQL Server where you installed the GP Repository. The default value, `period` (.), indicates you installed the GP Repository on the local Microsoft SQL Server. (mandatory)
`/DB:`*DatabaseName*	Name of the GP Repository database (mandatory).
`/U:`*SQLUserName*	Microsoft SQL Server logon user name used by SQL Authentication to access the GP Repository (optional).
`/P:`*SQLUserPassword*	Microsoft SQL Server logon password used by SQL Authentication to access the GP Repository (optional).

Sample Code

```
NqGPASyncLinkOrder /S:GPDOM800 /DB:MyDataBase /U:MyUser /P:MyPassword
or
NqGPASyncLinkOrder /D:MyDomain.com /S:GPDOM800 /DB:MyDataBase /U:MyUser /
P:MyPassword
or
NqGPASyncLinkOrder /ADContainer:LDAPPath /S:GPDOM800 /DB:MyDataBase /U:MyUser /
P:MyPassword
```

A.10.25   Undo Check Out GPO

Undo a checkout without saving any changes to the GP Repository.

Syntax (Visual Basic Script)

```
GPOObject.UndoCheckOut
```

Sample Code (Visual Basic Script)

The following sample allows you to undo the checkout of all GPOs checked out in a category. When you undo a checkout, GPA discards any changes you have made to the GPOs.

```
REM Undo Checkout for all checked out GPOs in a category
Dim oGPRroot, oCategory, oGPO, sCategory
sCategory = "FAGPR://CN=UserOU,CN=RELEASE,DC=Repository,DC=Net"
Set oGPRroot = WScript.CreateObject("faGPRRoot.faGPRRoot")
oGPRroot.ConnectTo("DRIVER={ODBC Driver 13 for SQL Server};SERVER="<SQL Server
Instance Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;")
Set oCategory = oGPRroot.GetObject(sCategory)
oCategory.EnumType = "GPO"
For Each oGPO in oCategory
     If oGPO.StatusCheckedOut = True then
          oGPO.UndoCheckOut
     end if
Next
```

Syntax (C# Method)

```
GPOObject.UndoCheckOut()
```

Sample Code (C# Method)

The following sample allows you to undo the checkout of all GPOs checked out in a category. When you undo a checkout, GPA discards any changes you have made to the GPOs.

```
public static void UndoGPO()
        {
             string sCategory = "FAGPR://CN=MyCategory, DC=MYDOMAIN,DC=LAB";
             IfaGPRRoot oGPRroot = new faGPRRoot();
             oGPRroot.ConnectTo("Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=GPA_SERVER;Use Procedure
for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=GPA_SERVER;Use
Encryption for Data=False;Tag with column collation when possible=False");
             IfaGPRCategory oCategory = oGPRroot.GetObject(sCategory);
             oCategory.EnumType = "GPO";
             foreach (IfaGPRGpo oGPO in oCategory)
             {
                 oGPO.UndoCheckOut();
             }
             Console.WriteLine("Undo GPO checkout successful");
             Console.ReadKey();
        }
```

A.11   Search Operations

You can create .NET applications to perform searches using GPA. If you are running GPA on a 64-bit platform, run these applications using a 32-bit command prompt window. On a 64-bit computer, you can access the 32-bit command prompt window from the `%WINDIR%\SysWOW64` folder.

To create a .NET application:

1 Create a console application in Microsoft Visual Studio.

NOTE: Ensure the logged on user account has `Write` permission on the *GPA install location*/`Bin` folder.

2 In Solution Explorer, right-click on your project name and select **Properties**.

3 Select the Build tab and change the output path to *GPA install location*/`Bin`.

4 In **Solution Explorer** select **References**, then right-click on **Reference** and select **Add Reference**.

5 Select the Browse tab, navigate to your GPA install location, and add the following assemblies:

```
NetiQ.GPA.SearchAPI.dll
NetiQ.GPA.SearchHelpers.dll
```

6 Add the following libraries at the beginning of each application:

```
using NetIQ.GPA.SearchComponentHelper;
using NetiQ.GPA.SearchAPI;
```

7 *If you are using multiple search criteria,* include the following library at the beginning of the application:

```
using System.Collections;
```

The following sections contain samples of searching using a single criterion and searching using multiple criteria.

NOTE: GPA returns these search results in XML format.

A.11.1 GPO Name

The following sample allows you to search for GPOs by name using the `contains/does not contain/is exactly` condition.

Note that `Is Exactly = opEquals`

```
public void SearchForGPONAME()
        {
            try
            {
                //Instantiate the Search Criteria Builder.
                IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                if (null == gpaSearchCriteriaBuilder)
                {
                    Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                }
                //Search Item like GPO Name.
                GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoDisplayName;
                //Search Conditions like contains, Does not contains, Is exactly.
                GPASearchOperatorTypes oTypes =
GPASearchOperatorTypes.opContains;
                //Create Search query for example (GPO Name Contains 'gpo')
                IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "gpo");
                //Create the search scope
                IGPASearchScope GPASearchScope = new GpaSearchScope();
```

```
                        //Assign valid AD domain, where intended to do search.
                        GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
    //Create scope for repository domain search .
                        GPASearchScope.AddRepositoryDomain();
    //Search for AD GPOs, REP GPOs or both.
                        GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                        //Instantiate the GPAsearchAPI
                        IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                        //Initialize the Search API.
                        bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                        if (bstatus)
                        {
                            // Get the Search Request ID.
                            string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                            if (strSearchRequestID.Length > 0)
                            {
                                // Save the search xml.
                                oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                            }
                        }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
                }
```

A.11.2   GPO Links

The following sample allows you to search for GPOs by links using the `Exist in/Does not exist
in` condition

```
public void SearchForGPOLinks()
            {
                try
                {
                    //Instantiate the Search Criteria Builder.
                    IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                    if (null == gpaSearchCriteriaBuilder)
                    {
                        Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                    }
                    //Search Item like GPO Links.
                    GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gposomLinks;
                    //Search Conditions like Exist in, Does not exist in.
                    GPASearchOperatorTypes oTypes =
GPASearchOperatorTypes.opExistIn;
                    //Create Search query for example (GPO Links Exist in '[All
Sites]')
                    IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "[All Sites]");
                    //Create the search scope
                IGPASearchScope GPASearchScope = new GpaSearchScope();
                    //Assign valid AD domain, where intended to do search.
                    GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
    //Create scope for repository domain search .
```

```
                        GPASearchScope.AddRepositoryDomain("MYDOMAIN.LAB", "");
                        //Search for AD GPOs, REP GPOs or both.
                        GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                        //Instantiate the GPAsearchAPI
                        IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                        //Initialize the Search API.
                        bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                        if (bstatus)
                        {
                            // Get the Search Request ID.
                            string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                            if (strSearchRequestID.Length > 0)
                            {
                                // Save the search xml.
                                oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                            }
                        }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
                }
```

A.11.3   Security Group

The following sample allows you to search for GPOs by security group using the `Has this explicit permission/Does not have this explicit permission/Has this effective permission/Does not have this effective permission` condition.

```
public void SearchForSecurityGroup()
        {
            try
            {
                //Instantiate the Search Criteria Builder.
                IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                if (null == gpaSearchCriteriaBuilder)
                {
                    Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                    return;
                }
                //Search Item like Security Group.
                GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoSecurityGroup;
                //Search Conditions like Has this explicit permission, Does not
have this explicit permission,
                //Has this effective permission, Does not have this effective
permission.
                GPASearchOperatorTypes oTypes =
GPASearchOperatorTypes.opHasExplicitPermission;
                //Create Search query for example (NetIQLabs.com\JSmith Has this
explicit permission 'Apply settings')
                //'Apply settings' = 1
                //'Edit settings' = 2
                //'Edit settings, delete, modify security' = 3
                //'Read settings' = 4
                IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "1","MYDOMAIN.LAB/
```

```
user|group");
                    //Create the search scope
                    IGPASearchScope GPASearchScope = new GpaSearchScope();
                    //Assign valid domain, where intended to do search.
                    GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                    //Search for AD GPOs.
                    GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                    //Instantiate the GPAsearchAPI
                    IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                    //Initialize the Search API.
                    bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                    if (bstatus)
                    {
                        // Get the Search Request ID.
                        string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                        if (strSearchRequestID.Length > 0)
                        {
                            // Save the search xml.
                            oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                        }
                    }
                }
                catch (Exception genExp)
                {
                    Console.WriteLine("Failed to Search." + genExp.Message);
                }
            }
```

A.11.4 Linked WMI Filter

The following sample allows you to search for GPOs by linked WMI filter using the `Is/Is not` condition.

Note that `Is = opEquals`

`Is Not = opNotEquals`

```
public void SearchForWMIFilter()
        {
            try
            {
                //Instantiate the Search Criteria Builder.
                IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                if (null == gpaSearchCriteriaBuilder)
                {
                    Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                    return;
                }
                //Search Item like WMI Filter.
                GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoWMIFilter;
                //Search Conditions like Is, Is Not.
                GPASearchOperatorTypes oTypes = GPASearchOperatorTypes.opEquals;
                //Create Search query for example (WMI Filter is 'operating
system')
                IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes,oTypes,"operating system");
                //Create the search scope
```

```
                        IGPASearchScope GPASearchScope = new GpaSearchScope();
                        //Assign valid AD domain, where intended to do search.
                        GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                        //Create scope for repository domain search .
                        GPASearchScope.AddRepositoryDomain("MYDOMAIN.LAB", "");
                        //Search for AD GPOs and REP GPOs.
                        GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInADAndREP);
                        //Instantiate the GPAsearchAPI
                        IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                        //Initialize the Search API.
                        bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                        if (bstatus)
                        {
                            // Get the Search Request ID.
                            string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                            if (strSearchRequestID.Length > 0)
                            {
                                // Save the search xml.
                                oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                            }
                        }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
                }
```

A.11.5   User Configuration

The following sample allows you to search for GPOs by user configuration using the `contains/
does not contain` condition.

```
public void SearchUserConfigurationSetting()
        {
            try
            {
                //Instantiate the Search Criteria Builder.
                IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                if (null == gpaSearchCriteriaBuilder)
                {
                    Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                    return;
                }

    //For User Configuration.
                //Search query : (User Configuration Contains 'Deployed Printer
Connections')
                IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(GPASearchCriterionTypes.gpoUserExtensi
ons,

GPASearchOperatorTypes.opContains,

"Deployed Printer Connections");
                //Create the search scope
                IGPASearchScope GPASearchScope = new GpaSearchScope();
```

```
                        //Assign valid domain, where intended to do search.
                        GPASearchScope.AddRepositoryDomain("MYDOMAIN.LAB", "");

                        //Search for AD GPOs.
                        GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInREP);

        //Instantiate the GPAsearchAPI
          IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                        if (null == oGPASearchAPI)
                        {
                            Console.WriteLine("Failed to instance of GpaSearchAPI");
                            return;
                        }

                        //Initialize the Search API.
                      bool bstatus = oGPASearchAPI.Initialize("Server_host_name", "http",
63847);
                        if (bstatus)
                        {
                            // Get the Search Request ID.
                            string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                            if (strSearchRequestID.Length > 0)
                            {
                                oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                            }

                        }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
            }
```

A.11.6   Computer Configuration

The following sample allows you to search for GPOs by computer configuration using the
contains/does not contain condition.

```
public void SearchComputerConfigurationSetting()
        {
                try
                {
                    //Instantiate the Search Criteria Builder.
                    IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                    if (null == gpaSearchCriteriaBuilder)
                    {
                        Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                    }

                    //Computer Configuration
                    //Sample Search query : (User Configuration Contains '802.3 Group
Policy')
                    IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(GPASearchCriterionTypes.gpoComputerExt
ensions,

GPASearchOperatorTypes.opContains,
```

```
Group Policy");
                //Create the search scope
                IGPASearchScope GPASearchScope = new GpaSearchScope();

                //Assign valid domain, where intended to do search.
                GPASearchScope.AddRepositoryDomain("MYDOMAIN.LAB", "");

                //Search for AD GPOs.
                GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInREP);

    //Instantiate the GPAsearchAPI
                IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                if (null == oGPASearchAPI)
                {
                    Console.WriteLine("Failed to instance of GpaSearchAPI");
                    return;
                }

                //Initialize the Search API.
              bool bstatus = oGPASearchAPI.Initialize("Server_host_name", "http",
63847);
                if (bstatus)
                {
                    // Get the Search Request ID.
                    string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                    if (strSearchRequestID.Length > 0)
                    {
                        oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                    }

                }
            }
            catch (Exception genExp)
            {
                Console.WriteLine("Failed to Search." + genExp.Message);
            }
        }
```

A.11.7   GUID Configuration

The following sample allows you to search for GPOs by GUID configuration using the equals condition.

```
public void SearchForGUID()
        {
                try
                {
                    //Instantiate the Search Criteria Builder.
                    IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                    if (null == gpaSearchCriteriaBuilder)
                    {
                        Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                    }
                    //Search Item like GUID.
                    GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoGUID;
                    //Search Conditions like Equals.
```

```
                        GPASearchOperatorTypes oTypes = GPASearchOperatorTypes.opEquals;
                        //Create Search query for example (GUID Equals '{DC047537-A1C0-
4212-81D7-D2674D98AA89}')
                        IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "{DC047537-A1C0-4212-
81D7-D2674D98AA89}");
                        //Create the search scope
                        IGPASearchScope GPASearchScope = new GpaSearchScope();
                        //Assign valid domain, where intended to do search.
                        GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                        //Search for AD GPOs.
                        GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                        //Instantiate the GPAsearchAPI
                        IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                        //Initialize the Search API.
                        bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                        if (bstatus)
                        {
                            // Get the Search Request ID.
                            string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                            if (strSearchRequestID.Length > 0)
                            {
                                // Save the search xml.
                                oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                            }
                        }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
                }
```

A.11.8 Keyword

The following sample allows you to search for GPOs by keyword using the Contains condition.

```
public void SearchForKeywordSearch()
        {
            try
            {
                //Instantiate the Search Criteria Builder.
                IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                if (null == gpaSearchCriteriaBuilder)
                {
                    Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                    return;
                }
                //Search Item like Keyword.
                GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoKeyword;
                //Search Conditions like Contains.
                GPASearchOperatorTypes oTypes =
GPASearchOperatorTypes.opContains;
                //Create Search query for example (Keyword Contains 'Certificate
Services Client - Auto-Enrollment')
                IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "Certificate Services
```

```
Client - Auto-Enrollment");
                    //Create the search scope
                    IGPASearchScope GPASearchScope = new GpaSearchScope();
                    //Assign valid domain, where intended to do search.
                    GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                    //Search for AD GPOs.
                    GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                    //Instantiate the GPAsearchAPI
                    IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                    //Initialize the Search API.
                    bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                    if (bstatus)
                    {
                        // Get the Search Request ID.
                        string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                        if (strSearchRequestID.Length > 0)
                        {
                            // Save the search xml.
                            oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                        }
                    }
                }
                catch (Exception genExp)
                {
                    Console.WriteLine("Failed to Search." + genExp.Message);
                }
            }
```

A.11.9 Advanced Keyword

The following sample allows you to search for GPOs by advanced keyword using the `Contains/Begins with/Ends with/Equals` condition.

```
public void SearchForAdvanceKeywordSearch()
            {
                try
                {
                    //Instantiate the Search Criteria Builder.
                    IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                    if (null == gpaSearchCriteriaBuilder)
                    {
                        Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                    }
                    //Advance Keyword search Item.
                    GPASearchCriterionTypes eSearchCriteriaType =
GPASearchCriterionTypes.gpoAdvancedKeyword;
                    //Setting Name Operator like Contains, Begins with, Ends with,
Equals.
                    GPASearchOperatorTypes eSettingNameOperator =
GPASearchOperatorTypes.opContains;
                    //Setting Name Operand.
                    string strSettingNameOperand = "Configure Automatic Updates";
                    //Setting Value Operator like Contains, Begins with, Ends with,
Equals.
                    GPASearchOperatorTypes eSettingValueOperator =
GPASearchOperatorTypes.opBeginsWith;
                    //Setting Value Oprand.
```

```
                            string strSettingValueOprand = "Enabled";
                            //Create Search query for example (Advanced Keyword Search
Setting Name Contains 'Configure Automatic Updates' And Setting Value Begins with
'Enabled')
                            IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(eSearchCriteriaType,

eSettingNameOperator,

strSettingNameOperand,

eSettingValueOperator,

strSettingValueOprand);
                            //Create the search scope
                            IGPASearchScope GPASearchScope = new GpaSearchScope();
                            //Assign valid domain where intended to do search.
                            GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                            //Search for AD GPOs.
                            GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                            //Instantiate the GPAsearchAPI
                            IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                            //Initialize the Search API.
                            bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                            if (bstatus)
                            {
                                // Get the Search Request ID.
                                string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                                if (strSearchRequestID.Length > 0)
                                {
                                    oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                                }
                            }
                    }
                    catch (Exception genExp)
                    {
                        Console.WriteLine("Failed to Search." + genExp.Message);
                    }
                }
```

A.11.10   GPO Search Using Multiple Criteria

The following sample allows you to search for GPOs using multiple criteria.

```
public void SearchForMultipleSearchCriteria()
            {
                try
                {
                    IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                    if (null == gpaSearchCriteriaBuilder)
                    {
                        Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                        return;
                    }
                    IList participantArrayList = new ArrayList();
                    IGPASearchParticipant Participant;
                    //First Search Item: GPO Name
                    //Search query : (GPO Name Contains 'rc')
```

```
                                Participant =
gpaSearchCriteriaBuilder.CreateSearchFilter(GPASearchCriterionTypes.gpoDisplayName
,

GPASearchOperatorTypes.opContains,
                                                              "rc");
                    //Add the Search query to the array list.
                    participantArrayList.Add(Participant);

                    //Second Search Item: GPO Links
                    //Search query : (GPO Links Exist In 'MYDOMAIN.LAB')
                    Participant =
gpaSearchCriteriaBuilder.CreateSearchFilter(GPASearchCriterionTypes.gposomLinks,

GPASearchOperatorTypes.opExistIn,

"MYDOMAIN.LAB");
                    //Add the Search query to the array list.
                    participantArrayList.Add(Participant);

                    //Third Search Item: Keyword search
                    //Search query : (Keyword Contains 'Enabled')
                    Participant =
gpaSearchCriteriaBuilder.CreateSearchFilter(GPASearchCriterionTypes.gpoKeyword,

GPASearchOperatorTypes.opContains,
                                                            "Enabled");
                    //Add the Search query to the array list.
                    participantArrayList.Add(Participant);
                    //List of search queries.
                    IGPASearchParticipant gpaSearchParticipantList;
                    //Final Search queries : (GPO Name Contains 'rc')AND(GPO Links
Exist In 'MYDOMAIN.LAB')AND(Keyword Contains 'Enabled')
                    //Relation Type of search: Match All Criteria = AND, Match Any
Criteria = OR
                    gpaSearchParticipantList =
gpaSearchCriteriaBuilder.LinkSearchCriteriaEx(participantArrayList,
GPASearchRelationshipType.AND/*Relation Type*/);
                    //Create the search scope
                    IGPASearchScope GPASearchScope = new GpaSearchScope();
                    GPASearchScope.AddDomain("MYDOMAIN.LAB", "");
                    //Create scope for repository domain search .
                    GPASearchScope.AddRepositoryDomain("MYDOMAIN.LAB", "");
                    //Search for AD GPOs.
                    //GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                    //Search for REP GPOs.
                    //GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInREP);
                    //Search for AD GPOs and REP GPOs.
                    GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInADAndREP);
                    //Instantiate the GPAsearchAPI
                    IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                    //Initialize the Search API.
                    bool bresult = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                    if (bresult)
                    {
```

```
                          // Get the Search Request ID.
                          string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipantList, GPASearchScope);
                          if (strSearchRequestID.Length > 0)
                              // Save the search xml.
                              oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                      }
                  }
              catch (Exception genExp)
              {
                  Console.WriteLine("Failed to Search." + genExp.Message);
              }
          }
```

A.11.11   GPO Search Using Multiple Domains

The following sample allows you to search for GPOs in multiple domains.

```
public void SearchForMultipleDomain()
          {
              try
              {
                  //Instantiate the Search Criteria Builder.
                  IGPASearchCriteriaBuilder gpaSearchCriteriaBuilder = new
CGPASearchCriteriaBuilder();
                  if (null == gpaSearchCriteriaBuilder)
                  {
                      Console.WriteLine("Failed to create
NetIQGPASearchHelpers.CGPASearchCriteriaBuilder");
                      return;
                  }
                  //Search Item like GPO Name.
                  GPASearchCriterionTypes cTypes =
GPASearchCriterionTypes.gpoDisplayName;
                  //Search Conditions like Contains.
                  GPASearchOperatorTypes oTypes =
GPASearchOperatorTypes.opContains;
                  //Create Search query for example (GPO Name Contains 'rc_')
                  IGPASearchParticipant gpaSearchParticipant =
gpaSearchCriteriaBuilder.CreateSearchFilter(cTypes, oTypes, "rc_");
                  //Create the search scope
                  IGPASearchScope GPASearchScope = new GpaSearchScope();
                  //Assign valid AD domains, where intended to do search.
                  GPASearchScope.AddDomain("AD_Domain A", "");
                  GPASearchScope.AddDomain("AD_Domain B", "");
                 // Assign valid Repository domains, where intended to do search.
                  GPASearchScope.AddRepositoryDomain("REP_Domain A", "");
                  GPASearchScope.AddRepositoryDomain("REP_Domain B", "");
                  //Search for AD GPOs.
                      //GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInAD);
                  //Search for REP GPOs.
                      //GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInREP);
                  //Search for AD GPOs and REP GPOs.
                      GPASearchScope.AddOptions(SearchOptions.SearchType,
SearchType.SearchInADAndREP);
                  //Instantiate the GPAsearchAPI
                  IGPASearchAPIs oGPASearchAPI = new GpaSearchAPI();
                  //Initialize the Search API.
                  bool bstatus = oGPASearchAPI.Initialize("Server_host_name",
"http", 63847);
                  if (bstatus)
```

```
            {
                // Get the Search Request ID.
                string strSearchRequestID =
oGPASearchAPI.SearchGPOs(gpaSearchParticipant, GPASearchScope);
                if (strSearchRequestID.Length > 0)
                {
                    // Save the search xml.
                    oGPASearchAPI.GetSearchResult(strSearchRequestID,
"File_Location\\Filename.xml");
                }
            }
        }
        catch (Exception genExp)
        {
            Console.WriteLine("Failed to Search." + genExp.Message);
        }
    }
```

B Automating GPA Operations with PowerShell Cmdlets

The Windows PowerShell command-line and scripting language can be used to automate many Group Policy tasks, including configuring registry-based policy settings and various Group Policy Management Console (GPMC) tasks. To help you perform these tasks, the Group Policy module for Windows PowerShell provides the cmdlets covered in the following sections:

- Section B.1, "Connect to GP Repository," on page 243
- Section B.2, "Get Object," on page 244
- Section B.3, "Domain Operations," on page 245
- Section B.4, "Category Operations," on page 249
- Section B.5, "GPO Node Operations," on page 251

NOTE: If you are running GPA on a 64-bit platform, you need to run PowerShell commands using a 32-bit command prompt window. On a 64-bit computer, you can access the 32-bit command prompt window from the `%WINDIR%\SysWOW64` folder.

B.1 Connect to GP Repository

Every GP Repository script must connect to a GP Repository to obtain data.

B.1.1 Obtaining the Connection String Value

The connection string is the parameter required to connect to the database.

To obtain this connection string:

1 Launch the GPA Console from the computer where you are going to execute the script or application.

2 After connecting to the GP Repository, select the **GP Repository** node.

3 On the Action menu, click **Properties**. The connection string is displayed in the Properties window.

4 Copy the connection string and paste it into your script or method file.

NOTE: The connection to the GP Repository is also based on security permissions for the user account in whose context the script or application is executed. Hence, if that user does not have permission to connect to the database, the connection command returns an error.

If you connect to the database with SQL Server authentication by providing a SQL Server user name and password, then the Connect String window displays the password as "<Password>". You need to replace this variable with the actual password.

B.1.2 Syntax

```
Set-GPRConnection [-ConnectionString] <String>
```

Parameter	Value
ConnectionString	The SQL connection string to the GPA Repository database.

Sample Code

The following sample establishes a connection to the GP Repository database named "GPO_REPOSITORY" on MABOSLPT03.

```
PS C:\>Set-GPRConnection -ConnectionString "DRIVER={ODBC Driver 13 for SQL
Server};SERVER="<SQL Server Instance
Name>";Trusted_Connection=Yes;DATABASE=GPO_REPOSITORY;"
```

B.2   Get Object

Initiate instance of an object, such as a GPO, category, or domain, in the GP Repository. The object should exist in the GP Repository.

B.2.1   Repository Object Path

The *Repository object path* is the location of a node for a GPO, category, or domain in the GP Repository. The format for the Repository object path is similar to an LDAP path with the following exceptions:

- ◆ Use `faGPR://` for the GP Repository path instead of `LDAP://`.
- ◆ Category names are preceded by `CN=` and each element of the domain name is preceded by `DC=`.

The simplest method for viewing the Repository object path is to right-click the node for the appropriate GPO, category, or domain, and then click **Properties**. Use the **Path** property.

B.2.2   Syntax

```
Get-GPRObject [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
	NOTE: To see the complete list of available properties, run `Get-Member` on the object returned from the `Get-GPRObject` command.

Sample Code 1

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample gets the NetIQLabs.com domain GPR object for the specified FAGPRPath.

```
PS C:\>Get-GPRObject  -FAGPRPath "FAGPR://DC=NetIQLabs,DC=com"
```

Sample Code 2

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample gets the desktop category GPR object for the specified FAGPRPath.

```
PS C:\>Get-GPRObject -FAGPRPath "FAGPR://CN=Desktop,DC=NetIQLabs,DC=com"
```

Sample Code 3

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample gets the GPO GPR object for the specified FAGPRPath.

```
PS C:\>Get-GPRObject -FAGPRPath "FAGPR://CN={31B2F340-016D-11D2-945F-
00C04FB984F9},CN=Desktop,DC=NetIQLabs,DC=com"
```

B.3   Domain Operations

The following sections provide the scriptable operations that can be carried out on the domain object.

B.3.1   Create Offline Policy Container Hierarchy

Run the `NQCreateOfflinePolicyContainerHierarchy.exe` file to create a temporary copy of the settings information of all GPOs in the GP Repository. GPA automatically creates the offline policy container hierarchy when you add domains to the GP Repository. This only works for domains that have a trust relationship with the repository member domain.

To run the `NQCreateOfflinePolicyContainerHierarchy.exe` file, you should have Domain Admin permissions in the domain for which you want to create the offline policy container hierarchy.

The `NQCreateOfflinePolicyContainerHierarchy.exe` file displays a status report in the command prompt window as it runs. After execution, the `NQCreateOfflinePolicyContainerHierarchy.exe` file creates a log that lists the domains it successfully recreated and those domains it failed to recreate. The log displays the "`ATTENTION REQUIRED`" text next to the domain name of any domain the tool failed to recreate.

Syntax

```
NQCreateOfflinePolicyContainerHierarchy /D:Domain_DNS_Name
/S:Repository_Server /DB:DatabaseName
```

Options

The following table describes the command-line parameters and variables.

Variable Name	Replace With
/D:*Domain_DNS_Name*	DNS name of the evaluation domain, such as `abc.xyz`. If you specify the domain name, then GPA creates the offline policy container hierarchy for only that domain. If you do not specify the domain name, then GPA creates an offline policy container hierarchy for each domain in the GP Repository (optional).
/S:*Repository_Server*	Name of the Microsoft SQL Server where you have installed the GP Repository. The default value, `period` (.), indicates the local Microsoft SQL Server.
/DB:*DatabaseName*	Name of the GP Repository database. The default value of the GP Repository database name is GPO_REPOSITORY. If the database name is different, specify the correct database name (optional, if you specify the domain name).
/?	Command-line Help for the tool.

Sample Code

```
NQCreateOfflinePolicyContainerHierarchy /D:ABC.com /S:ABCSQLServer /
DB:ABCDatabaseName
```

B.3.2   Create Category

Create a new category.

Syntax

```
New-GPRCategory [-FAGPRPath] <String> [-Name] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the new category.
Name	The name of the new category.

Sample Code 1

The following sample creates a child category named "NewCategory" under the parent category "Desktop".

> **NOTE:** Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

```
PS C:\>New-GPRCategory  -FAGPRPath "FAGPR://
CN=Desktop,DC=NetIQLabs,DC=com" -Name "NewCategory"
```

Sample Code 2

The following sample creates a Domain-level category named "NewCategory" on the domain "NetIQLabs.com".

> **NOTE:** Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

```
PS C:\>New-GPRCategory  -FAGPRPath "FAGPR://DC=NetIQLabs,DC=com" -Name
"NewCategory"
```

B.3.3 Set Default User Map

Updates the target domain map for the source domain (the map to target domain from source domain). For each user in the source domain's map, this operation adds a map entry from the source account to the target account with the same account name (if any).

Syntax

```
Set-GPRDefaultUserMap [-SourceDomain] <String> [-TargetDomain] <String>
```

Parameter	Value
SourceDomain	The FAGPR path of the source domain.
TargetDomain	The FAGPR path of the target domain.

Sample Code

The following sample maps each user account from the source domain, Org1.com, to the corresponding target account with the same name in Test.Org1.com.

```
PS C:\>Set-DefaultUserMap -SourceDomain "FAGPR://DC=Org1,DC=com" -
TargetDomain "FAGPR://DC=Test,DC=Org1,DC=com"
```

B.3.4 Merge GPOs

Merges the settings from two GPOs into a new GPO in the same domain.

Syntax

```
Merge-GPRGpo [-SourceFAGPRPath] <String[]> [-TargetFAGPRPath] <String> [[-
TargetGPOName] <String>] [[DeleteAllSource] <SwitchParameter>]
```

Parameter	Value
SourceFAGPRPath	The FAGPR paths of the source GPOs.
TargetFAGPRPath	The target category path where the new merged GPO will be created.
TargetGPOName	The name of the new GPO.
DeleteAllSource (optional)	Indicates whether to delete the source GPOs after the merge. You don't have to specify a value with this parameter.

Sample Code 1

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample merges two GPOs and creates a new GPO in the specified category.

```
PS C:\>Merge-GPRGpo -SourceFAGPRPath ("FAGPR://CN={6BE2D70A-46B5-4287-
A88F-8C31C2E73586}, CN=cat, DC=gpdom150,DC=lab","FAGPR://CN={8674177A-
6A00-4F89-8CDD-1C075E137572}, CN=cat, DC=gpdom150,DC=lab") -
TargetFAGPRPath "FAGPR://CN=cat, DC=gpdom150,DC=lab" -TargetGPOName
"MergedGPO1"
```

Sample Code 2

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample merges two GPOs and creates a new GPO in the specified category. The example also deletes the source GPOs.

```
PS C:\>Merge-GPRGpo -SourceFAGPRPath ("FAGPR://CN={6BE2D70A-46B5-4287-
A88F-8C31C2E73586}, CN=cat,DC=gpdom150,DC=lab","FAGPR://CN={8674177A-6A00-
4F89-8CDD-1C075E137572}, CN=cat, DC=gpdom150,DC=lab") -TargetFAGPRPath
"FAGPR://CN=cat, DC=gpdom150,DC=lab" -TargetGPOName "MergedGPO1" -
DeleteAllSource
```

Sample Code 3

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample merges two GPOs and creates a new GPO, overwriting one of the source GPOs.

```
PS C:\>Merge-GPRGpo -SourceFAGPRPath ("FAGPR://CN={6BE2D70A-46B5-4287-
A88F-8C31C2E73586}, CN=cat,DC=gpdom150,DC=lab","FAGPR://CN={8674177A-6A00-
4F89-8CDD-1C075E137572}, CN=cat, DC=gpdom150,DC=lab") -TargetFAGPRPath
"FAGPR://CN={6BE2D70A-46B5-4287-A88F-8C31C2E73586}, CN=cat,
DC=gpdom150,DC=lab"
```

B.4 Category Operations

The following sections provide the scriptable operations that can be carried out on the category object.

B.4.1 Create GPO

Create a GPO under a category.

Syntax

```
New-GPRGpo [-FAGPRPath] <String> [-Name] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the new GPO.
Name	The name of the new GPO.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample creates a repository GPO in GPDOM.LAB.

```
PS C:\>New-GPRGpo -FAGPRPath "FAGPR://CN=Meger Scripts, DC=GPDOM,DC=LAB" -
Name "SoftwarePolicy"
```

B.4.2 Delete Category

Deletes a category. This operation would delete all GPOs and subcategories. To use this command, all GPOs in the category must be checked in.

Syntax

```
Remove-GPRCategory [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the category.

Sample Code

> **NOTE:** Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample deletes the category 'Test' from the GPDOM.LAB domain.

```
PS C:\>Remove-GPRCategory -FAGPRPath "FAGPR://CN=Test, DC=GPDOM,DC=LAB"
```

B.4.3 Import GPO from Active Directory

Import an existing Active Directory GPO into a category.

Understanding LDAP Path and Overwrite Flag

This operation includes parameters that are defined as follows:

GPOLDAPPath

Specifies the LDAP path of the GPO that needs to be imported from a live Active Directory domain. To obtain the LDAP path of the GPO (`LDAP://...`), use the ADSI Edit tool. If the ADSI Edit tool is not available, substitute the following information in the LDAP path:

`"LDAP://DomainController/CN={GUID},CN=Policies,CN=System,DC=Domain"`

The parameters in this syntax statement are defined as follows:

DomainController

Type the name of the primary domain controller of the domain. Provide the full computer name, which has the actual computer name along with the domain to which it belongs. You can find the full name on the Network Identification tab of the Property page of **My Computer**.

GUID

Type the GUID number that corresponds to the GPO you want to import.

Domain

Type the name of the domain to which the GPO belongs. The domain name should be in the distinguished name format. For example, to specify the domain name, `mydomain.com`, the syntax should be `DC=MYDOMAIN, DC=COM`.

ADSIGPOObject

Specifies and ADSI pointer to the GPO in Active Directory.

OverwriteFlag

Specifies the overriding condition for the import. The values are `False` and `True`. `False` denotes *do not override if the GPO already exists in the domain*. `True` denotes *override the existing GPO*.

Syntax

```
Import-GPRADGpo [-FAGPRPath] <String> [-GpoLdapPath] <String> [-Overwrite]
<SwitchParameter>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the new GPO.
GpoLdapPath	The LDAP path of the GPO that you want to import from a live Active Directory domain.
	To obtain the LDAP path of the GPO, use the ADSI Edit tool. If it is not available, substitute the following information for the LDAP path: `LDAP:// DomainController/ CN={GUID},CN=Policies,CN=System,DC=Dom ain`.
Overwrite	Specifies whether or not to overwrite the GPO if it already exists in the GP Repository. You don't have to specify a value with this parameter.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample imports an Active Directory GPO from GPDOM.LAB to the GP Repository.

```
PS C:\>Import-GPRADGpo  -FAGPRPath "FAGPR://CN=MyCategory,
DC=GPDOM,DC=LAB" -GPOLDAPPath "LDAP://[DomainControler]/CN={000344FD-1494-
45A4-BF39-5022C4B4741A},CN=Policies,CN=System,DC=GPDOM,DC=LAB" -Overwrite
```

B.5   GPO Node Operations

The following sections provide the scriptable operations that can be carried out on the GPO scripting object.

B.5.1   Approve GPO

Approve a GPO to be exported to Active Directory or unapprove a GPO. If you set the value of the parameter to `True`, the method approves the GPO. Else, if the value of the parameter is `False`, the method unapproves the GPO.

Syntax

```
Set-GPRApproveGpo [-FAGPRPath] <String> [-Approve] <Boolean>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Approve	Specifies whether or not the GPO can be exported.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample approves a repository GPO to be exported to AD from NetIQLabs.LAB.

```
PS C:\>Set-GPRApproveGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=NetIQLabs,DC=LAB" -Approve $true
```

B.5.2 Approve GPO with Comments

Approve a GPO to be exported to Active Directory, or unapprove a GPO, and include comments in the history view. If you set the value of the parameter to `True`, the method approves the GPO. Else, if the value of the parameter is `False`, the method unapproves the GPO.

Syntax

```
Set-GPRApproveGpo [-FAGPRPath] <String> [-Approve] <Boolean> [-Comment]
<String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Approve	Specifies whether or not the GPO can be exported.
Comment	Allows you to make a statement regarding the approval of the GPO.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample approves a repository GPO to be exported to AD from NetIQLabs.LAB and adds a comment.

```
PS C:\>Set-GPRApproveGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=NetIQLabs,DC=LAB" -Approve $true -
Comment "Approved by PowerShell script"
```

B.5.3 Check In GPO

Check in a GPO.

Syntax

```
Close-GPRCheckInGpo [-FAGPRPath] <String> [-Comment] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Comment	Allows you to make a statement regarding the check-in porcess.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample checks in a repository GPO from GPDOM.LAB and adds an optional comment.

```
PS C:\>CheckInGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -Comment "Mycomment"
```

B.5.4 Check Out GPO

Check out a GPO.

Syntax

```
Open-GPRCheckOutGpo [-FAGPRPath] <String> [-Comment] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Comment	Allows you to make a statement regarding the check-out porcess.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample checks out a repository GPO from GPDOM.LAB and adds an optional comment.

```
PS C:\>Open-GPRCheckOutGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-
A471-F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -Comment
"Mycomment"
```

B.5.5   Create a GPO Link to a SOM Object

Links the GPO to a Scope of Management (SOM) object.

Syntax

```
Set-GPRGpoLink [-FAGPRPath] <String> [-SomLdapPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
SomLdapPath	The LDAP path of the SOM object from AD.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample links the GPO to an SOM object named "TestOU".

```
PS C:\>Set-GPRGpoLink -FAGPRPath "FAGPR://CN={8BBF2488-6BED-410A-8B9B-
174B0D6F63AA}, CN=TEST, DC=NetIQLabs,DC=com" -SomLdapPath "LDAP://
OU=TestOU,DC=NetIQLabs,DC=lab"
```

B.5.6   Delete a GPO Link from a SOM Object

Deletes a GPO link from a Scope of Management (SOM) object.

Syntax

```
Remove-GPRGpoLink [-FAGPRPath] <String> [-SomLdapPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
SomLdapPath	The LDAP path of the SOM object from AD.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample deletes a link from a GPO to an SOM object named "TestOU".

```
PS C:\>Remove-GPRGpoLink -FAGPRPath "FAGPR://CN={8BBF2488-6BED-410A-8B9B-
174B0D6F63AA}, CN=TEST, DC=NetIQLabs,DC=com" -SomLdapPath DAP://
OU=TestOU,DC=NetIQLabs,DC=lab"
```

B.5.7 Delete GPO

Delete a GPO.

Syntax

```
Remove-GPRGpo [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample deletes a repository GPO from GPDOM.LAB.

```
PS C:\>Remove-GPRGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB"
```

B.5.8 Export GPO

Export approved GPO to live Active Directory domain.

Export Parameter

You can specify one of the following export parameters:

BackUpOverwrite

   If the GPO already exists in Active Directory, overwrite it and back up the live Active Directory GPO into the GP Repository prior to overwriting it. You can also use an integer value of `14` instead of `BackUpOverwrite`.

NoBackUpOverwrite

If the GPO already exists in Active Directory, overwrite it. The live GPO is not backed up prior to import. You can also use an integer value of `13` instead of `NoBackUpOverwrite`.

DoNotOverwrite

Export fails if the GPO already exists in Active Directory. You can also use an integer value of `12` instead of `DoNotOverwrite`.

Syntax

```
Export-GPRGpo [-FAGPRPath] <String> [-Overwrite] <SwitchParameter>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Overwrite	If the GPO already exists in the AD, it will be overwritten. You don't have to specify a value with this parameter.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample exports a repository GPO to an Active Directory domain named "GPDOM.LAB".

```
PS C:\>Export-GPRGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -Overwrite
```

B.5.9 Export GPO with Comments

Export approved GPO to live Active Directory domain and include comments in the history view.

NOTE: This script can take several seconds or longer to complete when you run it for the first time.

Export Parameter

You can specify one of the following export parameters:

BackUpOverwrite

If the GPO already exists in Active Directory, Overwrite it and backup the live Active Directory GPO into the GP Repository prior to overwriting it. You can also use an integer value of `14` instead of `BackUpOverwrite`.

NoBackUpOverwrite

If the GPO already exists in the Active Directory overwrite it. The live GPO is not backed up prior to Import. You can also use an integer value of `13` instead of `NoBackUpOverwrite`.

`DoNotOverwrite`

> Export fails if the GPO already exists in Active Directory. You can also use an integer value of `12` instead of `DoNotOverwrite`.

Syntax

```
Export-GPRGpo [-FAGPRPath] <String> [-Overwrite] <SwitchParameter> [-
Comment] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Overwrite	If the GPO already exists in the AD, it will be overwritten. You don't have to specify a value with this parameter.
Comment	You can make a statement regarding the export process.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample exports a repository GPO to an Active Directory domain named "GPDOM.LAB" and adds a comment.

```
PS C:\>Export-GPRGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -Comment "This GPO has
been exported to Active Directory." -Overwrite
```

The Export Batch File

This batch file uses the `GPAExportUtil.exe` tool to create an entry for each approved GPO you have selected to export. If you want to export all approved GPOs in the selected domains, the batch file uses the `GPAExportUtil.exe` tool to create an entry for each selected domain.

Syntax

```
"<product installation path>\GPAExportUtil.exe" {{/g:<guid of GPO> |

/d:<DNS name of AD domain> | /a}

{/C:"<SQL Connection string>" | {/SQLS:<repository_server>

/SQLD:<rep_database_name> [/U:<SQL username> /P:<SQL password>]}} |

[/?|/H]
```

Options

The following table describes the command-line parameters and variables.

Variable name	Replace with
`/g:<guid of GPO>`	The GUID of the approved GPO you want to export using `GPAExportUtil.exe`. Use along with the `/d` option when you want to export two or more GPOs with the same GUID, but from different domains (required when exporting individual GPOs)
`/d:<DNS name of AD domain>`	The DNS name of the domain where approved GPOs will be exported. When this parameter is not specified, approved GPOs will be exported to the domain of the user performing the export. You can use this parameter when exporting any built-in domain policy GPOs or GPOs with same GUID.
`/a`	All approved GPOs in all domains of the specified GP Repository will be exported (optional).
`/C:"<SQL Connection string>"`	Full SQL Server connection string to the GP Repository database, in double quotes. Use instead of the other SQL Server parameters (required).
`/SQLS:<repository_server>`	Name of the GP Repository SQL Server (optional).
`/SQLD:<rep_database_name>`	Name of the GP Repository SQL Server database (optional).
`/U:<SQL username>`	SQL Server account name to use for SQL Authentication (optional).
`/P:<SQL password>`	SQL Server account password to use for SQL Server Authentication. Use caution when specifying this parameter in batch files (optional).

Sample Code

To export two selected GPOs from the domain, the export batch file contains the following entries:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{1FEB5933-DA75-49BC-A63F-
FA86C7CA9E20} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{F94F2CF6-0264-4DA6-B76C-
7C920360894D} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

To export all GPOs in a domain, the export batch file contains the following entry:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /d:usregion.com /
Connect:"Provider=SQLOLEDB.1;Integrated Security=SSPI;Initial
Catalog=GPO_REPOSITORY;Data Source=TREK02;Use Procedure for Prepare=1;Auto
Translate=True;Packet Size=4096;Workstation ID=TREK02;Use Encryption for
Data=False;Tag with column collation when possible=False"
```

To export two GPOs (in this case, the default domain policy) with the same GUID, but from different domains, the export batch file contains the following entries:

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{31B2F340-016D-11D2-945F-
00C04FB984F9} /d:usregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

```
"C:\Program Files\NetIQ\Group Policy
Administrator\tools\GPAExportUtil.exe" /g:{31B2F340-016D-11D2-945F-
00C04FB984F9} /d:nordicregion.com /Connect:"Provider=SQLOLEDB.1;Integrated
Security=SSPI;Initial Catalog=GPO_REPOSITORY;Data Source=TREK02;Use
Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
ID=TREK02;Use Encryption for Data=False;Tag with column collation when
possible=False"
```

B.5.10 Get GPO Check Out Status

Allows you to view whether a GPO is checked out. This operation returns a `True` or `False` value. `True` indicates a GPO is checked out and `False` indicates that the GPO is checked in.

Syntax

```
Get-GPRCheckOutStatus [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample checks the check out status of a repository GPO from GPDOM.LAB.

```
PS C:\>CheckOutStatus -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-A471-
F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB"
```

B.5.11 Get GPO Approval Status

Allows you to read the approval status of a GPO. This operation returns a `True` or `False` value. A `True` value denotes Approved status and a `False` value denotes Unapproved status.

Syntax

```
Get-GPRGpoApproveStatus [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample checks the approval status of a repository GPO from GPDOM.LAB.

```
PS C:\>Get-GPRGpoApproveStatus -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-
A471-F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB"
```

B.5.12 Lock or Mask GPO

Sets or gets status of locking or masking of a GPO for a user or group.

Syntax

```
Get-GPRGpoSecurityFilter [-FAGPRPath] <String> [-AccountName] <String> [-
OpType] <String>
```

```
Set-GPRGpoSecurityFilter [-FAGPRPath] <String> [-AccountName] <String> [-
OpType] <String> [-Force] <SwitchParameter>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
AccountName	User or Group
OpType	Lock Mask Unlock Unmask
Force (optional)	Allows or prevents the deletion of the security filter options.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample locks a GPO for a user object.

```
PS C:\>Set-GPRGpoSecurityFilter -FAGPRPath "FAGPR://CN={8BBF2488-6BED-
410A-8B9B-174B0D6F63AA}, CN=TEST, DC=NetIQLabs,DC=com" -AccountName
"MYDOMAIN\MyUser" -OpType "Lock"
```

The following sample checks if the GPO is masked for the user 'MyUser' from the domain 'MYDOMAIN' and returns the result.

```
PS C:\>Get-GPRGpoSecurityFilter -FAGPRPath "FAGPR://CN={8BBF2488-6BED-
410A-8B9B-174B0D6F63AA}, CN=TEST, DC=NetIQLabs,DC=com" -AccountName
"MYDOMAIN\MyUser" -OpType "Mask"
```

B.5.13 Generate GPO Report

Generate an HTML report for a specific version of a GPO. Specify the version number of the GPO that you want to generate a report for. If you want to generate a Report for the latest version of the GPO then pass the value as 0.

Syntax

```
Get-GPRGpoSettingsReport [-FAGPRPath] <String> [-Version] <Int32> [-
HtmlFileName] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.
Version	Specifies the version number of the GPO for which you want to run a report.
HtmlFileName	Specifies the target location of the settings report.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample generates a settings report for a repository GPO from NetIQLabs.com.

```
PS C:\>Get-GPRGpoSettingsReport -FAGPRPath "FAGPR://CN={8BBF2488-6BED-
410A-8B9B-174B0D6F63AA}, CN=TEST, DC=NetIQLabs,DC=com" -Version 0 -
HtmlFileName "c:\test.html"
```

B.5.14   Compare or Differentiate Two GPOs

Compare two GPOs and generate an HTML report of the comparison including the similarities and differences. The two GPOs must exist before running this operation. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax

```
Compare-GPRGpoSettingsReport [-FAGPRPath] <String> [-GPOtoCompare]
<String> [-HTMLFileName] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the first GPO in the comparison.
GPOtoCompare	Specifies the second GPO in the comparison.
HtmlFileName	Specifies the target location for the settings report.
DiffOnly (optional)	Specifies the type of comparison report. If you include this parameter, the report includes only the differences. Otherwise, the report includes both the similarities and differences.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample compares two GPOs and stores the report at "C:\ComparisonReport.html".

```
PS C:\>Compare-GPRGpoSettingsReport -FAGPRPath "FAGPR://CN={36553F1C-6D5D-
48E0-A471-F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -GPOToCompare
"FAGPR://CN={8BBF2488-6BED-410A-8B9B-174B0D6F63AA}, CN=TEST,
DC=GPDOM,DC=LAB" -HTMLFileName "C:\ComparisonReport.html"
```

B.5.15   Compare GPO GP Repository Versions

Generate a comparison HTML report with two different GP Repository versions of the GPO. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax

```
Compare-GPRGpoSettingsReportVersion [-FAGPRPath] <String> [-RepVersion1]
<Int32> [-RepVersion2] <Int32> [-HTMLFileName] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository GPO.
RepVersion1	Specifies the first version of the GPO that you want to compare.
RepVersion2	Specifies the second version of the GPO that you want to compare.
HtmlFileName	Specifies the target location for the HTML report.
DiffOnly (optional)	Specifies the type of comparison report. If you include this parameter, the report includes only the differences. Otherwise, the report includes both the similarities and differences.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample compares two versions of a repository GPO and stores the resulting report at "C:\ComparisonReport.html".

```
PS C:\>Compare-GPRGpoSettingsReportVersion -FAGPRPath "FAGPR://
CN={36553F1C-6D5D-48E0-A471-F42EB87E25C2}, CN=Meger Scripts,
DC=GPDOM,DC=LAB" -RepVersion1 1 -RepVersion2 2 -HtmlFileName
"C:\ComparisonReport.html"
```

B.5.16 Compare and Differentiate Active Directory GPO Versions

Generate a comparison HTML report with GPOs from Active Directory. The *DiffParameter* indicates the type of comparison report. A `True` value includes only the differences in the report. A `False` value includes both the similarities and differences in the report.

Syntax

```
Compare-GPRGpoSettingsReportAD [-FAGPRPath] <String> [-RepVersion] <Int32>
[-HTMLFileName] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository GPO.
RepVersion	Specifies the version of the GPO that you want to compare.
HtmlFileName	Specifies the target location for the HTML report.

Parameter	Value
DiffOnly (optional)	Specifies the type of comparison report. If you include this parameter, the report includes only the differences. Otherwise, the report includes both the similarities and differences.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample compares the repository version of a GPO with its Active Directory version and stores the resulting report at "C:\ComparisonReport.html".

```
PS C:\>Compare-GPRGpoSettingsReportAD -FAGPRPath "FAGPR://CN={36553F1C-
6D5D-48E0-A471-F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB" -
RepVersion 1 -HTMLFileName "C:\ComparisonReport.html"
```

B.5.17   Migrate GPO

Migrate a GPO across different domains or to a different category within the same domain. You can specify either a category or a GPO as the target. In either case, specify the GP Repository path of the target object.

Syntax for Migrating a New GPO

GPOObject.MigrateTo *TargetCategory*

Syntax for Migrating an Existing GPO

GPOObject.MigrateToEx *TargetGPO,* True|False

Syntax

```
Move-GPRMigrateGpo [-TargetFAGPRPath] <String> [-FAGPRPath] <String> [-
RenameGpo] <SwitchParameter>
```

Parameter	Value
TargetFAGPRPath	The GPR path of the target category or the target GPO.
FAGPRPath	The FAGPRPath of the Repository GPO.
RenameGpo (optional)	Renames the target GPO. You don't have to specify a value with this parameter.

Sample Code, Scenario 1

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample moves a GPO to a different category.

```
PS C:\>Move-GPRMigrateGpo -TargetFAGPRPath "FAGPR://
CN=CATTEST,DC=NetIQLabs,DC=COM" -FAGPRPath "FAGPR://CN={5EEBEF0F-5304-
4FCC-83C7-835EFBB72CCC},CN=CATTEST,DC=NetIQLabs,DC=COM"
```

Sample Code, Scenario 2

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the PowerShell cmdlet sample (page 244) for more information.

The following sample moves a GPO to another GPO and renames the target GPO to match the source GPO.

```
PS C:\>Move-GPRMigrateGpo -TargetFAGPRPath "FAGPR://CN={7A1741A8-ECF7-
42D8-9A02-C96B15F1FAF5},CN=CATTEST, DC=NetIQLabs,DC=Com" -FAGPRPath
"FAGPR://CN={BFD3C60B-D2F9-49F9-A80E-5B1C17D5AD25},CN=CAT_TEST,
DC=NetIQLabs,DC=com" -RenameGPO
```

B.5.18   Get-GPRGPOPath

The `Get-GPRGPOPath` cmdlet gets the GPOs for the specified Name or GUID. Before executing this cmdlet, use the `Set-GPRConnection` cmdlet to establish a connection to the GPA Repository database.

Gets the GPOs for the specified Name or GUID

PARAMETERS

Name <String>

The Name of the Repository GPO.

GUID <String>

The GUID of the Repository GPO

Domain [<String>]

The parameter is to specify the FAGPR path of the source domain.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information about_CommonParameters, see (http://go.microsoft.com/fwlink/?LinkID=113216).

SYNTAX

```
Get-GPRGPOPath [-Name] <String> [[-Domain] <String>] [<CommonParameters>]
Get-GPRGPOPath [-Guid] <String> [[-Domain] <String>] [<CommonParameters>]
```

NOTE: For examples, type `"get-help Get-GPRGPOPath -examples"`. For more information, type `"get-help Get-GPRGPOPath -detailed"`. For technical information, type `"get-help Get-GPRGPOPath -full"`.

Example B-1 1

```
PS C:\>Get-GPRGPOPath -Name GPO1
```

This example gets all the GPOs with Name GPO1 in all the managed domains.

Example B-2 1

```
PS C:\>Get-GPRGPOPath -Guid "{7803F7D2-4B6C-4627-94E8-A1E0A4BAC97A}"
```

This example gets all the GPOs having GUID {7803F7D2-4B6C-4627-94E8-A1E0A4BAC97A}.

Example B-3 1

```
PS C:\>Get-GPRGPOPath -Name GPO1 -Domain NetIQLabs.COM
```

This example gets all the GPOs with Name GPO1 in the domain NetIQLabs.COM.

Example B-4 1

```
PS C:\>Get-GPRGPOPath -Guid "{7803F7D2-4B6C-4627-94E8-A1E0A4BAC97A}" -
Domain NetIQLabs.COM
```

This example gets all the GPOs having GUID {7803F7D2-4B6C-4627-94E8-A1E0A4BAC97A} in the domain NetIQLabs.COM.

B.5.19 Read GPO Name

Retrieves the name of a GPO.

Syntax

```
Get-GPRGpo [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the category.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the for more information.

The following sample displays the list of GPOs in the category name desktop in the domain NetIQLabs.com.

```
PS C:\>Get-GPRGpo -FAGPRPath "FAGPR://CN=Desktop,DC=NetIQLabs,DC=com"
```

B.5.20   Undo Check Out GPO

Undo a checkout without saving any changes to the GP Repository.

Syntax

```
Undo-GPRCheckOutGpo [-FAGPRPath] <String>
```

Parameter	Value
FAGPRPath	The FAGPRPath of the Repository object.

Sample Code

NOTE: Before executing this cmdlet, run `Set-GPRConnection` to establish a connection to the GPA Repository database. See the for more information.

The following sample cancels the check out of a repository GPO from the GPDOM.LAB domain.

```
PS C:\>Undo-GPRCheckOutGpo -FAGPRPath "FAGPR://CN={36553F1C-6D5D-48E0-
A471-F42EB87E25C2}, CN=Meger Scripts, DC=GPDOM,DC=LAB"
```

C   Grooming the Database

Group Policy Administrator (GPA) includes SQL Server stored procedures you can execute to delete (groom) old versions of GPOs from the GPO_REPOSITORY database on the GP Repository computer. You can groom GPOs based on the following criteria:

- Version number
- Date

For more information see the following sections:

- Section C.1, "Executing the Grooming SQL Server Stored Procedures," on page 269
- Section C.2, "Grooming GPOs by Version Stored Procedure," on page 270
- Section C.3, "Grooming GPOs by Date Stored Procedure," on page 271
- Section C.4, "Scheduling Database Grooming," on page 272

C.1   Executing the Grooming SQL Server Stored Procedures

To immediately groom the GPO_REPOSITORY database, you can run a query to execute a Grooming SQL Server stored procedure.

NOTE: You can also groom the database on a schedule. For more information, see Section C.4, "Scheduling Database Grooming," on page 272.

To execute a SQL Server stored procedure:

1 Start **SQL Server Management Studio** in the Microsoft SQL Server program folder.

2 Connect to the SQL Server instance containing the GPO_REPOSITORY database with a login having db_owner privileges on the GPO_REPOSITORY database.

3 Expand the **Database** folder and select **GPO_REPOSITORY**.

4 On the toolbar, click **New Query**.

5 In the right pane, type the query by using the syntax described in one of the following sections:

- Section C.2, "Grooming GPOs by Version Stored Procedure," on page 270
- Section C.3, "Grooming GPOs by Date Stored Procedure," on page 271

6 On the SQL Editor toolbar, click Execute.

C.2   Grooming GPOs by Version Stored Procedure

You can execute the `fa_rep_groomRepositorybyVersion` stored procedure to groom GPOs based on version number. Grooming based on a version number grooms all GPOs with a version lower than you specify. You can also limit the criteria to a specific GPO or to all GPOs within a specific domain.

For more information about executing stored procedures, see Section C.1, "Executing the Grooming SQL Server Stored Procedures," on page 269.

> **NOTE:** If you specify a version higher than the latest GPO version, then GPA grooms all but the latest GPO version.

C.2.1   Syntax

```
exec fa_rep_groomRepositorybyVersion '{GPO_GUID}','version','domain'
```

C.2.2   Parameters

Specify the following parameters.

GPO_GUID	**GUID of the GPO. Specify the GUID to groom only GPOs with the same GUID in the specified domain. If you want to groom all GPOs within the specified domain, regardless of GUID, type `null`. Note: If you specify the GUID, you must also specify the domain.**
domain	Name of the domain that contains the GPO. Specify the domain name to groom only GPOs within the same domain. If you want to groom GPOs within all domains, type `null`. Use a fully qualified domain name (FQDN) format (domain.com).
version	Version of the GPO. The stored procedure grooms all GPOs lower than the version you specify that match the GUID and domain or domain, if specified.

C.2.3   Examples

Grooming versions lower than 7 of a specific GPO within a domain named MyDomain.com:

```
exec fa_rep_groomRepositorybyVersion '{40addbf7-9d0e-412a-9507-
3e381fb5c707}','7','MyDomain.com'
```

Grooming versions lower than 7 of all GPOs within a domain named MyDomain.com:

```
exec fa_rep_groomRepositorybyVersion null,'7','MyDomain.com'
```

Grooming versions lower than 7 of all GPOs within all domains:

```
exec fa_rep_groomRepositorybyVersion null,'7',null
```

C.3 Grooming GPOs by Date Stored Procedure

You can execute the `fa_rep_groomRepositorybyDate` stored procedure to groom GPOs based on date. Grooming based on date enables you to groom all GPO versions with a modified date that is older than the date you specify. You can also limit the criteria to a specific GPO or to all GPOs within a specific domain.

For more information about executing stored procedures, see Section C.1, "Executing the Grooming SQL Server Stored Procedures," on page 269.

NOTE: Note
If you specify a date that is more recent than the last modified date, then GPA grooms GPOs that are older than the last modified GPO.

C.3.1 Syntax

```
exec fa_rep_groomRepositorybyDate '{GPO_GUID}','modified date','domain'
```

C.3.2 Parameters

Specify the following parameters.

GPO_GUID	**GUID of the GPO. Specify the GUID to groom only GPOs with the same GUID in the specified domain. If you want to groom all GPOs within the specified domain, regardless of GUID, type `null`. Note: If you specify the GUID, you must also specify the domain.**
domain	Name of the domain that contains the GPO. Specify the domain name to groom only GPOs within the same domain. If you want to groom GPOs within all domains, type `null`. Use a fully qualified domain name (FQDN) format (domain.com).
modified date	Modified date of the GPO in the following format: yyyy-mm-dd. The stored procedure grooms all GPOs older than the date you specify that match the GUID and domain or domain, if specified.

C.3.3 Examples

Grooming all versions older than March 10, 2009, of a specific GPO within a domain named MyDomain.com

```
exec fa_rep_groomRepositorybyDate '{40addbf7-9d0e-412a-9507-
3e381fb5c707}','2009-03-10','MyDomain.com'
```

Grooming all versions older than March 10, 2009, of all GPOs within a domain named MyDomain.com

```
exec fa_rep_groomRepositorybyDate null,'2009-03-10','MyDomain.com'
```

Grooming all versions older than March 10, 2009, of all GPOs within all domains

```
exec fa_rep_groomRepositorybyDate null,'2009-03-10',null
```

C.4   Scheduling Database Grooming

You can also schedule database grooming with Microsoft Windows Tasks Scheduler (previously called Scheduled Tasks). You can create a batch file on a computer that has the SQLCMD Utility installed to execute a SQL Server stored procedure, and then use the Microsoft Windows Tasks Scheduler to schedule the batch file.

Ensure the login account you use to connect to the SQL Server instance has db_owner privileges on the GPO_REPOSITORY database.

To execute the SQL Server stored procedure on a schedule:

1 Create a text file to contain the SQL Server script that executes the SQL Server stored procedure as appropriate:

 ◆ Section C.2, "Grooming GPOs by Version Stored Procedure," on page 270

 ◆ Section C.3, "Grooming GPOs by Date Stored Procedure," on page 271

2 Create a batch file that uses the SQLCMD utility to run the script by using the appropriate syntax:

 SQLCMD –s *ServerName* –d *DatabaseName* –i *path\TextFileName* (for Windows authentication)

 SQLCMD –U *SQLLoginID* –P *Password* –s *ServerName* –d *DatabaseName* –i *path\TextFileName* (for SQL authentication)

 For more information about the SQL Server syntax, see the Microsoft documentation.

3 Use the Microsoft Windows Task Scheduler to schedule the batch file. For more information, see the Microsoft documentation.

D Ports Used by GPA

Group Policy Administrator (GPA) requires the following ports to be open:

Ports	Computers	Notes
389/636	GPA Console > Domain Controller	The GPA Console communicates with the domain controller using LDAP over TCP/IP through port 389 (or port 636 for communication via SSL) to perform GP Explorer operations and to import, create, export, check out, and check in GPOs.
389/636	GPA Server > Domain Controller	The Export Only Account on the GPA Server exports GPOs over TCP/IP through port 389 (or port 636 for communication via SSL).
135	GPA Console > Domain Controller	DCOM port used for all remote procedure calls. In GPA, used to connect to Sysvol share folder.
135	GPA Server > Domain Controller	Used to connect to the Sysvol share folder.
135	GPA Console > Local GPOs folder	Used by the RPC Endpoint Mapper to check out GPOs.
135	GPA Console > client computer	Used to generate Diagnostic reports.
445	GPA Console > Domain Controller	Used by Microsoft-DS service to share resources across computers.
445	GPA Server > Domain Controller	Used by Microsoft-DS service.
49156	GPA Console > Domain Controller	Used by LSASS.exe to authenticate credentials.
49156	GPA Server > Domain Controller	Used by LSASS.exe to authenticate credentials.
49000-65535	GPA Console > Domain Controller	Dynamic return ports used by GPA console for RPC.
1433	GPA Console > GP Repository	Used by Microsoft SQL Server for all GPA operations that involve the GP Repository.
63847	GPA Console > GPA Server	Used by GPA Server service to send notifications and by the Export Only Account to export GPOs. Also used by GPA Console to connect to GPA Server published in AD to use that server to search GPOs in AD or GP Repository domains.
64000	GPA Server > GPA Server	Used by NetIQ.GPA.SettingsReport.exe process to generate the GPO Settings report for AD or Repository GPOs in order to be indexed.
25	GPA Server > Mail Server	Used by the GPA Server to send emails by SMTP to the mail server.

The following image illustrates the ports, protocols, and services used by GPA.
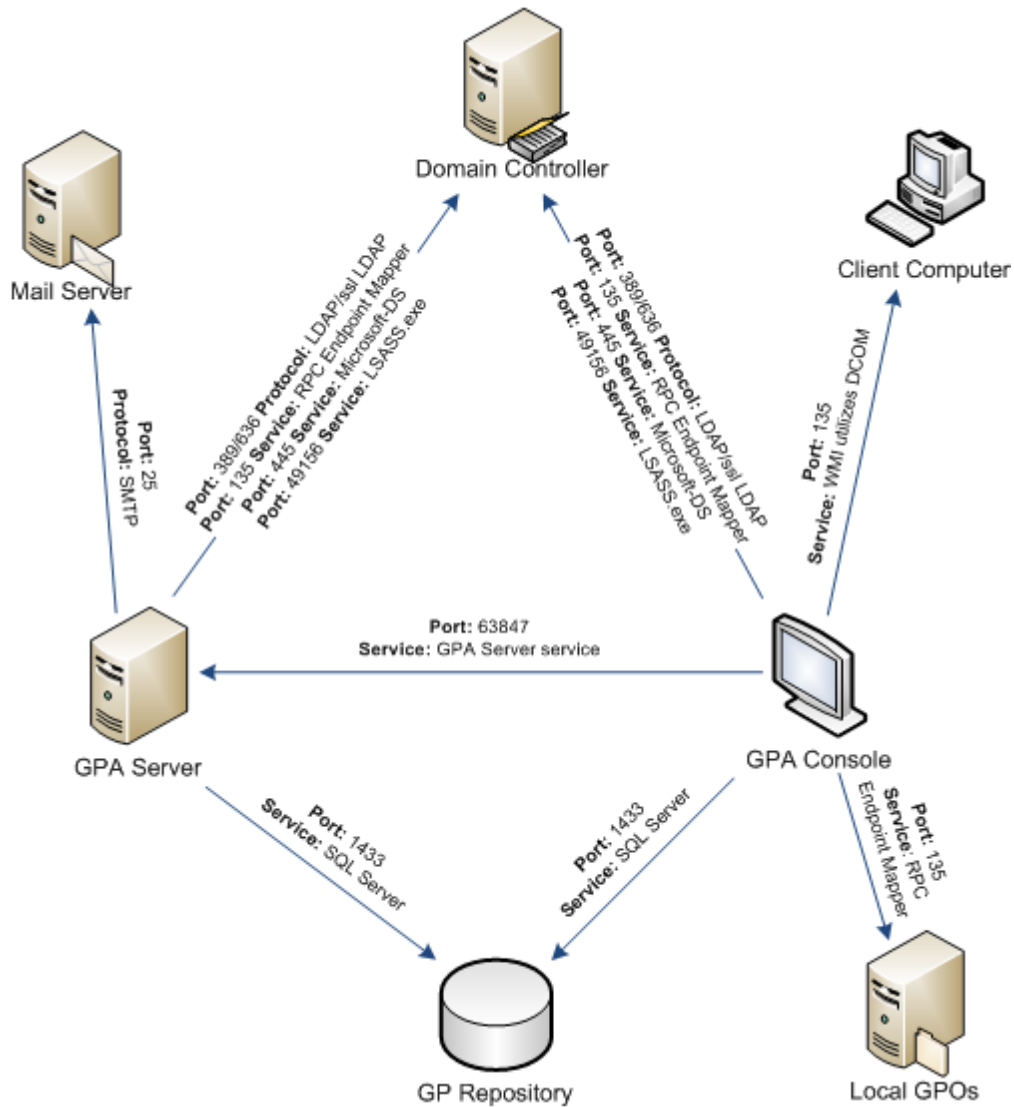
E Detailed Security Requirements

This appendix explains the security permissions you need to work with and analyze GPOs in Active Directory and in GPA. For more information about GPA security, including how to define security for GPA users, see Chapter 4, "Configuring Security and Permissions," on page 53.

E.1 GP Repository Task Specifics

The following table lists these tasks and indicates at what levels in the GP Repository you can set permissions for each task.

Number	Task	GP Repository	Domain	Category	GPO	Notes
1	Full Control	☑				Sets permissions for all tasks at all levels
2	Full GP Repository Server Control	☑				Sets permissions for tasks 3-5
3	Add GP Repository User	☑				
4	Add Remote User	☑				
5	Customize Deployment Options	☑				
6	Full Domain Control	☑	☑			Sets all domain-level permissions for tasks 8, 9, 10, 12, and 14
7	Create New Domain	☑				
8	Delete Domain	☑	☑			
9	Migrate GPO	☑	☑			
10	Import GPO from Active Directory	☑	☑			

Number	Task	GP Repository	Domain	Category	GPO	Notes
11	Synchronize ADMX from the Central Store	☑	☑			This task is directly associated with the Import GPO from Active Directory task at the GP Repository and domain levels. You cannot set permissions for this task directly. When you enable the Import GPO from Active Directory task, you also set permissions for this task.
12	Export GPO to Active Directory	☑	☑			
13	Export ADMX to the Central Store	☑	☑			This task is directly associated with the Export GPO to Active Directory and Modify Export Status tasks at the GP Repository and domain levels. You cannot set permissions for this task directly. When you enable the Export GPO to Active Directory and Modify Export Status tasks, you also set permissions for this task.
14	Edit Domain Maps	☑	☑			

Number	Task	GP Repository	Domain	Category	GPO	Notes
15	Full Category Control	☑	☑	☑		Sets all category-level permissions for tasks 16-19
16	Create Category	☑	☑	☑		
17	Delete Category	☑	☑	☑		
18	Paste GPO Category Link	☑	☑	☑		
19	Rename Category	☑	☑	☑		
20	Full GPO Control	☑	☑	☑	☑	Sets all permissions below this level.
21	Create GPO	☑	☑	☑	☑	
22	Add ADMX	☑	☑			This task is directly associated with the Create GPO task at the GP Repository and domain levels. You cannot set permissions for this task directly. When you enable the Create GPO task, you also set permissions for this task.
23	Modify GPO	☑	☑	☑	☑	Sets permissions for tasks 24-27
24	Modify GPO Settings	☑	☑	☑	☑	
25	Modify GPO Links	☑	☑	☑	☑	
26	Modify GPO Security	☑	☑	☑	☑	
27	Rename GPO	☑	☑	☑	☑	
28	Delete GPO	☑	☑	☑	☑	

Number	Task	GP Repository	Domain	Category	GPO	Notes
29	Remove ADMX	☑	☑			This task is directly associated with the Delete GPO task at the GP Repository and domain levels. You cannot set permissions for this task directly. When you enable the Delete GPO task, you also set permissions for this task.
30	Check Out GPO	☑	☑	☑	☑	
31	Override Check Out	☑	☑	☑	☑	
32	Rollback	☑	☑	☑	☑	
33	Approve/Reject/ Unapprove GPO	☑	☑	☑	☑	
34	Approve/ Unapprove ADMX Files	☑	☑			This task is directly associated with the Approve/ Unapprove GPO task at the GP Repository and domain levels. You cannot set permissions for this task directly. When you enable the Approve/ Unapprove GPO task, you also set permissions for this task.
35	Modify Export Status	☑	☑	☑	☑	
36	Modify GPO Security Filters	☑	☑	☑	☑	

Number	Task	GP Repository	Domain	Category	GPO	Notes
37	Modify GPO Enterprise Sync	☑	☑	☑	☑	Enables user to designate master and controlled GPOs
38	GPO Synchronizer		☑	☑	☑	Enables user to modify GPOs using Enterprise Synchronization

E.2   GP Explorer Requirements

The following table lists the various security requirements for using the GP Explorer Console.

Task	Security Requirement
Launch console View GPO	1. GPA uses the current user account to connect to the domain. To use another user account for the connection: Save the console as an MMC file. Right-click on the console file. Click **Run As**. Every user account is a member of Authenticated Users by default. Therefore, GPA displays all GPOs that have Read permission set for the current user or authenticated user account.
Create GPO	User account must be a member of one of the following groups: ◆ Domain Administrators ◆ Enterprise Administrators ◆ Group Policy Creator Owners
Delete GPO	User account must have **Delete all child objects** setting on the GPO.
Search GPO	Result of the search displays only those GPOs that have the Read permission set for the current user account.
Backup GPO	User account must have Read permissions on the GPOs and the LSDOU associated with the GPOs.
Restore GPO	User account must be a member of one of the following groups: ◆ Domain Administrators ◆ Enterprise Administrators ◆ Group Policy Creator Owners

Task	Security Requirement
Link GPO to OU Modify security filters	Domain Administrator and Enterprise Administrator accounts have permission to modify OU links and security filters. Other user accounts must have Delegated permission. To assign Delegated permission, use the Delegation of Control wizard in the Active Directory Users and Computers console.
Copy, paste, import GPO	User account must be a member of one of the following groups: ◆ Domain Administrators ◆ Enterprise Administrators ◆ Group Policy Creator Owners
GPO report	User account must have Read permission to the GPOs.
Set indexing properties	User account must have Full Domain Control (6) in the domain.
GP Repository permissions	User account must have Full Domain Control in the domain.

E.3 GP Repository Requirements

The following table lists the various security requirements for using the GP Repository Console.

Task	Security Requirement
Launch console	SQL permissions: User account must have connect permissions to the Microsoft SQL Server Database containing the GP Repository.
GP Repository Tasks	
Connect to Database	SQL permissions: Current user must have SQL user account to connect to GP Repository
Disconnect from Database	None
Compare GPOs	None
Generate Activity Report	None
Add the GP Repository User	SQL permissions: User account must have Security Admin and Database Owner permissions to the Microsoft SQL Server Database containing the GP Repository.
Add Remote User	SQL permissions: Current user must have SQL privilege to create new SQL user
New Domain	Active Directory permissions: Must have permissions in Active Directory to create container

Task	Security Requirement
Customize Deployment Options	GP Repository permissions: **Customize Deployment Options** permission
Domain Level Tasks	
Delete Domain	GP Repository permissions: **Delete Domain** permission
Compare GPOs	None
Create Category	GP Repository permissions: **Create Category** permission
Edit Domain Maps	GP Repository permissions: **Edit Domain Map** permission
Compare GPOs	None
Set indexing properties	User account must have Full Domain Control (6) in the domain.
GP Repository permissions	User account must have Full Domain Control in the domain.
Category Level Tasks	
Create Category	GP Repository permissions: **Create Category** permission
Delete Category	GP Repository permissions: **Delete Category** permission
Rename Category	GP Repository permissions: **Rename Category** permission
New GPO	GP Repository permissions: **Create GPO** permission
Paste as New GPO	GP Repository permissions: **Create GPO** permission
Paste GPO Category Link	GP Repository permissions: **Paste GPO Category Link** permission
Import GPO from Active Directory (GPO does not exist in GP Repository)	GP Repository permissions: ◆ **Import GPO from AD** permission at Domain level ◆ **Create GPO** permission at Category level
Import GPO from Active Directory (GPO already exists in GP Repository)	GP Repository permissions: ◆ **Import GPO from AD** permission at Domain level ◆ **Modify GPO** permission at Category level

Task	Security Requirement
GPO Level Tasks	

Check Out	GP Repository permissions:
	
	Check Out permission and one or more of the following:
	
	    • Modify GPO Setting
	    • Modify GPO Security
	    • Modify GPO Links
	    • Rename GPO

Check In	GP Repository permissions: **Check Out** permission
Override Check Out	GP Repository permissions: **Override Check Out** permission
View History	None
Approve Version	GP Repository permissions: **Approve/Unapprove** permission
Undo Approve Version	GP Repository permissions: **Approve/Unapprove** permission
Send for Approval	GP Repository permissions: **Check Out GPO** permission **Modify GPO Settings** permission
Reject Version	GP Repository permissions: **Approve/Unapprove** permission
Compare Active Directory Version	Active Directory permissions: Read permission on GPO in Active Directory
Differentiate Active Directory Version	Active Directory permissions: Read permission on GPO in Active Directory
Rollback GPO Version	GP Repository permissions: **Rollback** permission

Task	Security Requirement
Export GPO to Active Directory (GPO does not exist in Active Directory or GPO already exists in Active Directory)	The export override account must be a domain user and have the following permissions: Domain SYSVOL permissions and Active Directory Permissions: Full Control. For more information, see Step 3b on page 31 and Step 3c on page 32. GP Repository permissions: Full Control
Synchronize GPO	GP Repository permissions: **Modify GPO** permission
Migrate to Category	GP Repository permissions: ◆ **Migrate GPO** permission at Domain level ◆ **Create GPO** permission at Category level
Migrate to GPO	GP Repository permissions: ◆ **Migrate GPO** permission at Domain level ◆ **Modify GPO** permission at GPO level
Delete GPO	GP Repository permissions: **Delete GPO** permission in Repository Domain Property page

E.4 GP Analysis Requirements

The following table lists the various security requirements for using the GP Analysis Console.

Task	Security Requirement
Launch console	Active Directory permissions: User account must have Read permission to the GPOs on the domain under analysis.
Perform RSoP	SQL permissions Active Directory permissions: User account must have Read permission to all GPOs and SDOU hierarchies.
Perform Search	SQL permissions Active Directory permissions: User account must have Read permission to all GPOs and LSDOU hierarchies.

Task	Security Requirement
Perform Compare or Differentiate	SQL permissions: SQL connect permissions to GP Repository. Active Directory permissions: User account must have Read permission to all GPOs and SDOU hierarchies.
Perform remote diagnostics	OS permissions: User account that runs remote diagnostics must have local Administrator rights on that remote computer.
Perform client-side auditing	OS permissions: User account must have Read permission to the registry on the remote computer.

E.5   Non-Domain Admin AD Rights Needed for the Export Only Account:

Since these permissions are native to Microsoft AD, we recommend that you configure the permissions using Group Policy PowerShell scripts and Microsoft GPMC.

Task	Microsoft Native Option	Example
Create GPOs in the domain	Add the Export Only account to the global Group Policy Creator Owners group.	None
Modify GPO Link and GP Option	Grant Link GPOs permission using GPMC. For more information, see Microsoft Documentation.	None

Task	Microsoft Native Option	Example

Full Edit permission in the GPO (for existing GPOs in AD)	Execute PS Set-GPPermission as per Microsoft Documentation.	Import-Module GroupPolicy
		
		$params = @{
		
		    All = $true
		
		    TargetName = "AccountName"
		
		    TargetType = 'User'
		
		    PermissionLevel = 'GpoEditDeleteModifySecurity'
		
		    Replace = $true
		
		    }
		
		    Set-GPPermission @params
		
		NOTE: Replace "AccountName" with Export Only Account name.