[MS-GPSCR]:
Group Policy:
Scripts Extension Encoding

Revision Summary

Date	Revision History	Revision Class	Comments
03/02/2007	1.0	Major	Updated and revised the technical content.
04/03/2007	1.1	Minor	Updated the technical content.
05/11/2007	2.0	Major	New format
06/01/2007	2.0.1	Editorial	Revised and edited the technical content.
07/03/2007	2.0.2	Editorial	Revised and edited the technical content.
08/10/2007	2.0.3	Editorial	Revised and edited the technical content.
09/28/2007	2.0.4	Editorial	Revised and edited the technical content.
10/23/2007	2.1	Minor	Updated a reference to MS-PROTO.
01/25/2008	2.1.1	Editorial	Revised and edited the technical content.
03/14/2008	2.1.2	Editorial	Revised and edited the technical content.
06/20/2008	2.1.3	Editorial	Revised and edited the technical content.
07/25/2008	2.1.4	Editorial	Revised and edited the technical content.
08/29/2008	2.2	Minor	Added section references.
10/24/2008	3.0	Major	Updated and revised the technical content.
12/05/2008	4.0	Major	Updated and revised the technical content.
01/16/2009	4.0.1	Editorial	Revised and edited the technical content.
02/27/2009	4.0.2	Editorial	Revised and edited the technical content.
04/10/2009	4.0.3	Editorial	Revised and edited the technical content.
05/22/2009	4.1	Minor	Updated the technical content.
07/02/2009	5.0	Major	Updated and revised the technical content.
08/14/2009	5.1	Minor	Updated the technical content.
09/25/2009	5.2	Minor	Updated the technical content.
11/06/2009	5.2.1	Editorial	Revised and edited the technical content.
12/18/2009	5.3	Minor	Updated the technical content.
01/29/2010	5.4	Minor	Updated the technical content.
03/12/2010	5.5	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
04/23/2010	6.0	Major	Updated and revised the technical content.
06/04/2010	6.1	Minor	Updated the technical content.
07/16/2010	7.0	Major	Significantly changed the technical content.
08/27/2010	8.0	Major	Significantly changed the technical content.
10/08/2010	9.0	Major	Significantly changed the technical content.
11/19/2010	10.0	Major	Significantly changed the technical content.
01/07/2011	11.0	Major	Significantly changed the technical content.
02/11/2011	12.0	Major	Significantly changed the technical content.
03/25/2011	13.0	Major	Significantly changed the technical content.
05/06/2011	14.0	Major	Significantly changed the technical content.
06/17/2011	15.0	Major	Significantly changed the technical content.
09/23/2011	15.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	16.0	Major	Significantly changed the technical content.
03/30/2012	16.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	16.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	17.0	Major	Significantly changed the technical content.
01/31/2013	18.0	Major	Significantly changed the technical content.
08/08/2013	19.0	Major	Significantly changed the technical content.
11/14/2013	19.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 6
1.2.1 Normative References 7
1.2.2 Informative References 7
1.3 Overview 7
1.3.1 Background 7
1.3.2 Scripts Extension Encoding Overview 8
1.4 Relationship to Other Protocols 10
1.5 Prerequisites/Preconditions 10
1.6 Applicability Statement 10
1.7 Versioning and Capability Negotiation 11
1.8 Vendor-Extensible Fields 11
1.9 Standards Assignments 11

2 Messages 12
2.1 Transport 12
2.2 Message Syntax 12
2.2.1 Common Message Requirements 12
2.2.2 Scripts.ini Syntax 12
2.2.3 Psscripts.ini Syntax 13

3 Protocol Details 15
3.1 Administrative Tool Plug-in Details 15
3.1.1 Abstract Data Model 15
3.1.1.1 Scripts.ini 15
3.1.1.2 PSScripts.ini 15
3.1.2 Timers 15
3.1.3 Initialization 15
3.1.4 Higher-Layer Triggered Events 15
3.1.5 Message Processing Events and Sequencing Rules 15
3.1.6 Timer Events 17
3.1.7 Other Local Events 17
3.2 Client Plug-in Details 17
3.2.1 Abstract Data Model 17
3.2.1.1 Command Execution Subsystem 18
3.2.1.1.1 Abstract Interface of Command Execution Subsystem 19
3.2.1.1.2 Abstract Interface of Executable Group 19
3.2.1.1.3 Abstract Interface of Executable List 19
3.2.2 Timers 20
3.2.3 Initialization 20
3.2.4 Higher-Layer Triggered Events 20
3.2.4.1 Process Group Policy 20
3.2.5 Message Processing Events and Sequencing Rules 20
3.2.6 Timer Events 22
3.2.7 Other Local Events 22

4 Protocol Examples 23

5 Security 24
5.1 Security Considerations for Implementers 24

5.2 Index of Security Parameters 24
6 Appendix A: Product Behavior 25
7 Change Tracking 27
8 Index 28

1 Introduction

This document specifies the Group Policy: Scripts Extension Encoding protocol, which provides a mechanism to communicate script information from a **Group Policy server** to a Group Policy client. The Group Policy client will use this information to ensure that administrative-defined scripts are available to execute at specific events such as Logon and Logoff.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

client-side extension GUID (CSE GUID)
computer policy mode
domain
domain controller (DC)
globally unique identifier (GUID)
Group Policy Object (GPO)
Group Policy Object (GPO) distinguished name (DN)
Group Policy Object (GPO) path
policy application
policy target
tool extension GUID or administrative plug-in GUID
UncPath
Unicode
user policy mode

The following terms are specific to this document:

Group Policy (GP) server: A server that holds a database of Group Policy Objects (GPOs) that other machines can retrieve. The **GP server** must be a **domain controller (DC)**.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

1.2.1 Normative References

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

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

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

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

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, http://www.ietf.org/rfc/rfc4234.txt

1.2.2 Informative References

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

[MS-GPOD] Microsoft Corporation, "Group Policy Protocols Overview".

[MS-WPO] Microsoft Corporation, "Windows Protocols Overview".

[MSFT-PROFSCR] Microsoft Corporation, "Assign a logon script to a user or group", January 21, 2005, http://technet2.microsoft.com/windowsserver/en/library/e9028566-1be7-45f8-a219-6b09dce34f8d1033.mspx?mfr=true

1.3 Overview

Group Policy: Scripts Extension Encoding provides a mechanism for an administrator to instruct an arbitrarily large group of clients to execute administrator-specified code at computer start, computer shut-down, user log-on, and user log-off. The code executed by clients is in the form of a command-line tool or batch-processing script that is present either on the client's local file system or at a network file system location.

This mechanism allows administrators to perform various maintenance and management tasks on client computers, including (but not limited to) collecting diagnostic information, invoking security scans, cleaning or resetting system state, and installing tools.

The protocol allows for administration of up to two separate groups of scripts. These two groups correspond to logon/logoff scripts and startup/shutdown scripts. The grouping provides an organization of scripts that will execute during different system events.

User-logon scripts configured using this protocol differ from user-logon scripts configured as part of user-object scripts [MSFT-PROFSCR].

An overview of the timeline when user and computer policies are applied to a client is described in [MS-GPOD] section 3.1.

1.3.1 Background

The Group Policy: Core Protocol (as specified in [MS-GPOL]) enables clients to discover and retrieve policy settings created by administrators of a **domain**. These settings are persisted within **group**

policy objects (GPOs), which are assigned to **policy target** accounts in Active Directory directory service. Policy target accounts are either computer accounts or user accounts in Active Directory. Each client uses the Lightweight Directory Access Protocol (LDAP), as specified in [RFC2251], to determine what GPOs are applicable to it by consulting Active Directory objects corresponding to its computer account and the user accounts of any users logging on to the client computer.

On each client, each GPO is interpreted and acted on by software components known as client plug-ins. The client plug-ins responsible for a given GPO are specified using an attribute of the GPO. This attribute specifies a list of **GUID** pairs. The first GUID of each pair is referred to as a **client-side extension GUID (CSE GUID)**. The second GUID of each pair is referred to as a **tool extension GUID**.

For each GPO applied to a client, the client consults the CSE GUIDs listed in the GPO to determine what client plug-ins on the client should handle the GPO. The client then invokes the client plug-ins to handle the GPO.

A client plug-in uses the contents of the GPO to retrieve relevant settings in a manner specific to the plug-in. After its settings are retrieved, the client plug-in uses those settings to perform plug-in-specific processing.

1.3.2 Scripts Extension Encoding Overview

The following diagram depicts the entities that participate in Group Policy: Scripts Extension Encoding:

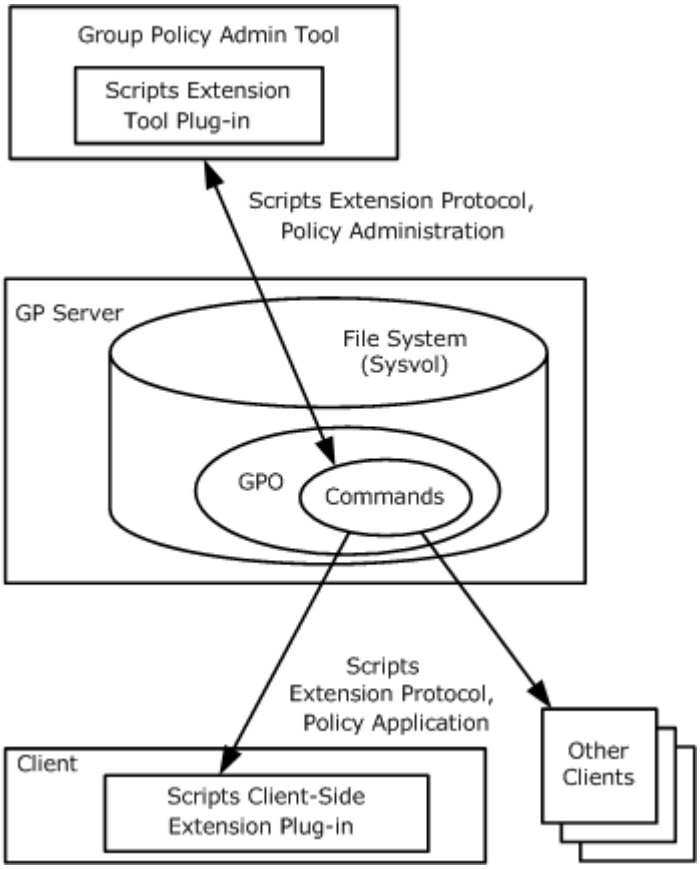

Figure 1: Group Policy: Scripts Extension Encoding entities

Clients can use either or both of the following modes for this protocol because they address different issues:

- **Computer Policy Mode**

  In this mode, Group Policy Objects (GPOs) are applied for the computer on which the client is running.

  The following sequence of operations occurs from both policy administration and **policy application** modes:

  1. An administrator invokes the Group Policy Administrative tool to administer a GPO (as specified in [MS-GPOL]), using the policy administration mode (as specified in [MS-GPOL] section 1.3.4). Through Group Policy: Scripts Extension Encoding, the presence of the tool extension GUID for computer policy settings for Group Policy: Scripts Extension Encoding is retrieved, and it indicates that the GPO contains policy settings that should be administered through the policy administration portion of Group Policy: Scripts Extension Encoding. The administrative tool invokes a plug-in specific to Group Policy: Scripts Extension Encoding so that the administrator can administer Group Policy: Scripts Extension Encoding settings. This results in the storage and retrieval of metadata inside a GPO on a Group Policy server. This metadata describes commands that the administrator wants to execute on a client that is affected by the GPO. The administrator views the data and updates it to add a directive to run a command when the client computer starts. The directive can be any action that can be run locally on the client computer.

  2. A client computer affected by that GPO is started (or is connected to the network, if this happens after the client starts), and the Group Policy: Core Protocol is invoked by the client to retrieve policy settings from the Group Policy server. As part of the processing of the Group Policy: Core Protocol (as specified in [MS-GPOL] section 3.2.5.1.10), the Group Policy: Scripts Extension Encoding CSE GUID is read from this GPO, and this instructs the client to invoke a Group Policy: Scripts Extension Encoding plug-in component for policy application.

  3. In processing the policy application portion of Group Policy: Scripts Extension Encoding, the client identifies the directive to run the administrator's command at computer start and configures a command execution subsystem of the underlying operating system on the client computer (logically not a part of Group Policy: Scripts Extension Encoding or the Group Policy: Core Protocol) with this directive. When the computer is in the process of starting, the command execution subsystem invokes the command as required by the administrator. Similarly, when the client later shuts down, the command execution subsystem executes any shutdown commands.

- **User Policy Mode**

  In this mode, GPOs are applied for the user who is logged on to the computer on which the client is running.

  The following sequence of operations is performed from the policy administration and policy application mode:

  1. Step 1 is the same as the preceding step 1 for computer policy mode, except that a separate tool extension GUID for Group Policy: Scripts Extension Encoding is used, and the administrator can specify commands that are to run at the time a user logs on or off.

2. Step 2 is the same as the preceding step 2 for computer policy mode, except that it occurs when a user logs on (or when the computer is connected to the network, if this happens after the user logs on).

3. In processing the policy application portion of Group Policy: Scripts Extension Encoding, the client identifies the directive to run the administrator's command at user logon time and configures the command execution subsystem with this directive. Because the user is in the process of logging on while the protocol is executing, the command execution subsystem invokes the command as needed by the administrator. When the user later logs off, any logoff commands are then executed.

1.4 Relationship to Other Protocols

This protocol depends on the Group Policy: Core Protocol specified in [MS-GPOL] to provide a list of applicable GPOs. It also transmits Group Policy settings and instructions between the **client** and the **GP server** by reading and writing files using remote file access.

See [MS-WPO] section 6.4 for an overview of remote file access.

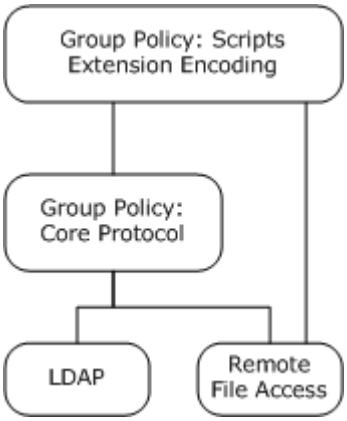

Figure 2: Group Policy: Scripts Extension Encoding protocol relationship diagram

1.5 Prerequisites/Preconditions

The prerequisites for this protocol are the same as those for the Group Policy: Core Protocol (as specified in [MS-GPOL] section 1.5).

In addition, a client is required to have a system/subsystem capable of executing commands at startup and shutdown times if computer policy mode is used, and at user logon and logoff times if user policy mode is used.

1.6 Applicability Statement

Group Policy: Scripts Extension Encoding is applicable only within the Group Policy: Core Protocol framework specified in [MS-GPOL]. Group Policy: Scripts Extension Encoding is used to run short-lived administrative automation tasks against groups of client computers in a domain. It should not be used to remotely execute interactive applications or long-lived background tasks.

This protocol is appropriate for use only when the same executable commands are relevant to all clients.<1>

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

This protocol defines CSE GUID and tool extension GUID standards assignments, as specified in [MS-GPOL] section 1.8. The following table shows the assignments.

Parameter	Value
CSE GUID	{42B5FAAE-6536-11D2-AE5A-0000F87571E3}
Tool extension GUID (user policy mode settings)	{40B66650-4972-11D1-A7CA-0000F87571E3}
Tool extension GUID (computer policy mode settings)	{40B6664F-4972-11D1-A7CA-0000F87571E3}

2 Messages

2.1 Transport

The Group Policy: Scripts Extension Encoding transports messages by reading and writing remote files.

The Group Policy: Core Protocol uses Group Policy: Scripts Extension Encoding client-side extension GUID (CSE GUID) and tool extension GUID values to invoke Group Policy: Scripts Extension Encoding only to access GPOs from which messages of this protocol can be generated.

This protocol enables the client to identify scripts and other executable code that it invokes. Therefore, the client must be able to validate that the source of the script's location (that is, the Group Policy server) has not been spoofed by a malicious user. If the source can be spoofed, the malicious user can cause the client to execute arbitrary code using high privileges on the client. This requirement to validate the Group Policy server is the reason mutual authentication is required for this protocol's use of remote file access against the GP server.

2.2 Message Syntax

2.2.1 Common Message Requirements

Messages exchanged in this protocol allow the client to discover settings in the GPOs that instruct clients to execute arbitrary commands. After interpreting the settings, the client attempts to execute the scripts according to the settings.

The following definitions aid in understanding this section:

Computer-scoped GPO path: A scoped GPO path that ends in "\Machine".

Scoped GPO path: A GPO path that is appended with "\User" for the user policy mode of a policy application (or "\Machine" for the computer policy mode).

User-scoped GPO path: A scoped GPO path that ends in "\User".

Messages of the protocol are transferred as files using remote file access. The files MUST be named as "<*gpo path*>\scripts\scripts.ini" or "<*gpo path*>\scripts\psscripts.ini", where <*gpo path*> is a scoped GPO path.

2.2.2 Scripts.ini Syntax

Scripts.ini is a text file encoded in UTF-16LE with Byte Order Mark (0xFFFE) that conforms to the following Augmented Backus-Naur Form (ABNF) [RFC4234].

```
IniFile =          WhiteSpace Sections WhiteSpace
Sections =         1*Section
WhiteSpaceClass = CR / LF / WSP
WhiteSpace =       *WhiteSpaceClass
SpaceDelimiter =   1*WhiteSpaceClass

Section =          SectionHeader Keys

SectionHeader =    WhiteSpace "[" SectionName "]"   SpaceDelimiter
SectionName =      TokLogon / TokLogoff / TokStartup / TokShutdown
```

```
Keys =              1*Key
Key =               TokKey TokIs TokValue
TokKey =            WhiteSpace 1*(ALPHA / DIGIT)
TokIs =             WhiteSpace "="
TokValue =          WhiteSpace 1*(ALPHA / "_" / DIGIT )   SpaceDelimiter

TokLogon =          WhiteSpace "Logon"                    WhiteSpace
TokLogoff =         WhiteSpace "Logoff"                   WhiteSpace
TokStartup =        WhiteSpace "Startup"                  WhiteSpace
TokShutdown =       WhiteSpace "Shutdown"                 WhiteSpace
```

The specific format of scripts.ini MUST be as follows:

Sections: When used in computer policy mode (that is, with a computer-scoped GPO path), sections Startup and Shutdown are optional. The sections Logon and Logoff MUST NOT exist.

When used in user policy mode (that is, with a user-scoped GPO path), sections Logon and Logoff are optional. The sections Startup and Shutdown MUST NOT exist.

Any sections not valid for a particular mode MUST be ignored and do not invalidate the file.

Keys: Keys in the Startup, Shutdown, Logon, and Logoff sections MUST be named with the syntax "<integer>CmdLine" and "<integer>Parameters", where <integer> is the text representation of an integer value greater than or equal to zero and less than 2^31. If any key in the file begins with <integer>, both keys ("<integer>CmdLine" and "<integer>Parameters") MUST be present and come in pairs, though the order in which they appear can be interchanged. The <integer> value MUST start from 0 and MUST be in ascending order incremented by one.

TokValue: The values in the Startup, Shutdown, Logon, and Logoff sections are text strings. The text values of "<integer>CmdLine" keys MUST be file system paths that are specified by using any valid syntax for the client file systems that may reference files on the local computer or on a network location. The lengths of these paths MUST be fewer than 260 (**Unicode**) characters. Each path MUST be the path of an executable program that can be invoked by clients. The text values of "<integer>Parameters" keys can be any string (this is the string that is passed as command-line parameters to the executable program as part of its invocation by the client).<2>

2.2.3 Psscripts.ini Syntax

Psscripts.ini is a text file encoded in UTF-16LE with Byte Order Mark (0xFFFE) that conforms to the following Augmented Backus-Naur Form (ABNF) [RFC4234].

```
IniFile =           WhiteSpace Sections WhiteSpace
Sections =          1*Section
WhiteSpaceClass =   CR / LF / WSP
WhiteSpace =        *WhiteSpaceClass
SpaceDelimiter =    1*WhiteSpaceClass

Section =           SectionHeader Keys

SectionHeader =     WhiteSpace "[" SectionName "]"   SpaceDelimiter
SectionName =       TokLogon / TokLogoff / TokStartup / TokShutdown / TokScriptsConfig
```

```
Keys =             1*Key
Key =              TokKey TokIs TokValue
TokKey =           WhiteSpace 1*(ALPHA / DIGIT)
TokIs =            WhiteSpace "="
TokValue =         WhiteSpace 1*(ALPHA / "_" / DIGIT )   SpaceDelimiter

TokLogon =         WhiteSpace "Logon"                    WhiteSpace
TokLogoff =        WhiteSpace "Logoff"                   WhiteSpace
TokStartup =       WhiteSpace "Startup"                  WhiteSpace
TokShutdown =      WhiteSpace "Shutdown"                 WhiteSpace
TokScriptsConfig = WhiteSpace "ScriptsConfig"            WhiteSpace
```

The specific format of psscripts.ini MUST be the same as described above for scripts.ini with the following additional Sections, Keys and TokValue elements:

Sections: The psscripts.ini file MUST contain the section ScriptsConfig if at least one of its keys is present; otherwise the section SHOULD be omitted.

Keys: Keys in the optional ScriptsConfig section MUST be named **StartExecutePSFirst** or **EndExecutePSFirst**.

The **StartExecutePSFirst** key indicates whether the computer startup and user logon scripts listed in psscripts.ini are to be executed before or after the scripts listed in scripts.ini. If unspecified, the order is implementation-dependent.

The **EndExecutePSFirst** key indicates whether the computer shutdown and user logoff scripts listed in psscripts.ini are to be executed before or after the scripts listed in scripts.ini. If unspecified, the order is implementation-dependent.

TokValue: The values of the **StartExecutePSFirst** and **EndExecutePSFirst** keys in the optional ScriptsConfig section MUST have the text value of case-insensitive "true" or "false". If "true", scripts listed in psscripts.ini MUST be executed before the scripts listed in scripts.ini. If "false", scripts listed in psscripts.ini MUST be executed after the scripts listed in scripts.ini.

3 Protocol Details

3.1 Administrative Tool Plug-in Details

3.1.1 Abstract Data Model

The administrative tool has a user interface that allows an administrator to author scripts.ini and psscripts.ini files.

3.1.1.1 Scripts.ini

The scripts.ini file (as specified in section 2.2.2) contains the settings for the Scripts Executable group defined in the client abstract data model (section 3.2.1). These settings are:

- Script Type that identifies when the script is to be executed. Values can be one of the following: Startup, Logon, Shutdown, or Logoff.
- Executable Item is the command line and its parameters.

3.1.1.2 PSScripts.ini

The psscripts.ini file (as specified in section 2.2.3) contains the settings for the PSScripts Executable group defined in the client abstract data model (section 3.2.1). These settings are:

- Script Type that identifies when the script is to be executed. Values can be one of the following: Startup, Logon, Shutdown, or Logoff.
- Executable Item is the command line and its parameters.
- Script order that indicates whether the scripts in PSScripts.ini run before or after the scripts in Scripts.ini.

3.1.2 Timers

None.

3.1.3 Initialization

When the administrative-side plug-in starts, it gets a scoped GPO path from the Group Policy: Core Protocol, as specified in [MS-GPOL] section 2.2.4. The plug-in then processes the GPO path as specified in Message Processing Events and Sequencing Rules (section 3.1.5).

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The administrative-side plug-in MUST attempt to use remote file access to retrieve any existing scripts.ini file from "<gpo path>\scripts\scripts.ini", where <gpo path> is the scoped GPO path. The plug-in SHOULD also attempt to use remote file access to retrieve any existing psscripts.ini file from "<gpo path>\scripts\psscripts.ini".<3>

The processing for reading and writing the settings from the GPO for administrative purposes is as follows:

To create the Group Policy: Scripts Extension Encoding settings, the administrative tool plug-in MUST perform these steps for each GPO:

1. Perform a remote file open on the file specified by "<gpo path>\scripts\scripts.ini", where <gpo path> is the scoped GPO path in the group policy object. If this operation fails, go to step 3.

2. Perform one or more remote file reads to read the entire contents of the opened file until the entire file has been read or an error in reading occurs.

3. If the file "<gpo path>\scripts\scripts.ini" is present, display the settings that were read in step 2. If no file was found, display an empty list.

4. The administrator configures new Group Policy: Scripts Extension Encoding settings by specifying the CmdLine and Parameters values for the scripts in scripts.ini.

5. If scripts are configured for scripts.ini group, use remote file write sequences to create a new scripts.ini file in the "<gpo path>\scripts\" directory if no file existed. Write the administrator-configured Group Policy: Scripts Extension Encoding settings to the scripts.ini file, overwriting the old content with updated content according to the format specified in section 2.2.2.

6. If opened, perform a remote file close to close the scripts.ini file.

7. After every creation, modification, or deletion that affects the scripts.ini file, the administrative tool MUST invoke the Group Policy Extension Update task as specified in [MS-GPOL] section 3.3.4.4.

Additionally, the administrative tool plug-in SHOULD perform these steps for each GPO:<4>

1. Perform a remote file open on the file specified by "<gpo path>\scripts\psscripts.ini", where <gpo path> is the scoped GPO path in the group policy object. If this operation fails, go to step 3.

2. Perform one or more remote file reads to read the entire contents of the opened file until the entire file has been read or an error in reading occurs.

3. If the file, "<gpo path>\scripts\psscripts.ini", is present, display the settings that were read in step 2. If no file was found, display an empty list.

4. The administrator configures new Group Policy: Scripts Extension Encoding settings by specifying the CmdLine and Parameters values for the scripts in psscripts.ini.

5. The administrator optionally configures whether scripts listed in psscripts.ini are to be executed before or after the scripts listed in scripts.ini by specifying the StartExecutePSFirst (for startup, logon scripts) and EndExecutePSFirst (for shutdown, logoff scripts) values in the ScriptConfig section of the psscript.ini file. A value of case-insensitive "true" means scripts listed in psscripts.ini MUST be executed before the scripts listed in scripts.ini in the GPO. A value of case-insensitive "false" means scripts listed in psscripts.ini MUST be executed after the scripts listed in scripts.ini in the GPO.

6. If scripts are configured for psscripts.ini group, use remote file write sequences to create a new psscripts.ini file in the "<gpo path>\scripts\" directory if no file existed. Write the administrator-configured Group Policy: Scripts Extension Encoding settings to the pscripts.ini file, overwriting the old content with updated content according to the format specified in section 2.2.3.

7. If opened, perform a remote file close to close the psscripts.ini file.

8. After every creation, modification, or deletion that affects the psscripts.ini file, the administrative tool MUST invoke the Group Policy Extension Update task as specified in [MS-GPOL] section 3.3.4.4.

When an administrator specifies a command to be executed under a given condition using the administrative tool, the Group Policy: Scripts Extension Encoding plug-in MUST put the commands into a scripts.ini or psscripts.ini file, as specified in section 2.2, and copy it to "<gpo path>\scripts\scripts.ini" or "<gpo path>\scripts\psscripts.ini", specified as follows, where <gpo path> is the scoped GPO path obtained from the Group Policy: Core Protocol part of the administrative tool. If this fails, the administrator MUST be informed, and the scripts.ini and psscripts.ini files SHOULD be reverted to the state in which it existed prior to the protocol sequence.<5>

To update the scripts.ini or psscripts.ini files in a GPO, the state of that GPO on the GP server MUST be updated with the following message sequence:

1. A remote file open from client to server: The file name used MUST be "<gpo path>\scripts\scripts.ini" or "<gpo path>\scripts\psscripts.ini", where <gpo path> is the user-scoped GPO path (if the GPO user settings are being updated) or the computer-scoped GPO path (if the computer settings are being updated). The remote file open MUST request write permission and request that if the file does not exist it will be created. If the open request returns a failure status, the Group Policy: Scripts Extension Encoding sequence MUST be terminated.

2. The tool MUST perform one or more remote file writes to overwrite the contents of the opened file with new settings. These writes MUST continue until the entire file is copied or an error is encountered.

3. File close: The tool MUST issue a remote file close operation.

The two files, scripts.ini and psscripts.ini, correspond to the two separate groups of scripts supported. Depending on the group of script, the administrative tool updates either scripts.ini or psscripts.ini.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Plug-in Details

During policy application, the protocol is invoked after the Group Policy: Core Protocol, as specified in [MS-GPOL] section 3.2.1.4, has computed a list of GPOs for which Group Policy: Scripts Extension Encoding is to be invoked.

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The Group Policy: Scripts Extension Encoding plug-in itself maintains no state.

3.2.1.1 Command Execution Subsystem

The command execution subsystem of the client computer maintains the following state:

- Two executable groups. Each group has **Executable Lists**, one per **Execution Context**.
- **Executable Group Order**.
- **Default Executable Group Order**.

The **Executable Groups** and **Executable Group Order** can be updated by the client plug-in.

The command execution subsystem invokes the updated list of executable programs at the appropriate time (logon, logoff, startup, shutdown) in the right order (as specified by **Executable Order**):

Executable Group: There are two groups: the Scripts Executable group and the PSScripts Executable group. Each group contains **Executable Lists**, one per **Execution Context**.

Executable List: Each list contains **Execution Context** and a list of **Executable Items**.

Execution Context: Indicates when the **Executable Items** inside **Executable List** are to be invoked.

Value	Meaning
Log on	**Executable List** to be invoked when a user logs on. In computer policy mode, this list is to be ignored.
Log off	**Executable List** to be invoked when a user logs off. In computer policy mode, this list is to be ignored.
Start up	**Executable List** to be invoked at computer startup. In user policy mode, this list is to be ignored.
Shut down	**Executable List** to be invoked at computer shutdown. In user policy mode, this list is to be ignored.

Executable Item:

Executable Order: The order in which an **Executable Item** is to be executed inside an **Executable List**.

Executable Path: A file system path to a file that can be accessed and executed.

Executable Parameters: A string containing space-separated parameters to be passed to the executable program when it is executed.

Executable Group Order: The order in which the scripts in the two groups are to be executed. PSFirst indicates the PSScripts group executes before the Scripts group, PSLast indicates the PSScripts group executes after the Scripts group.

Default Executable Group Order: The **Executable Group Order** for scripts when it is not otherwise specified.<6>

There are three abstract interfaces for this component, which are defined in the subsections that follow.

3.2.1.1.1 Abstract Interface of Command Execution Subsystem

The **command execution subsystem** abstract interface provides the following methods and parameters.

Retrieve Executable Group: This method is used to retrieve from the command execution subsystem an **Executable Group** given the name of the **Executable Group**.

The Group Policy client extension provides the following:

Executable Group Name: Name of the **Executable Group** (Scripts Executable Group or PSScripts Executable Group).

Executable Group: Output data structure representing **Executable Group**.

Retrieve Executable Group Order: This method is used to retrieve from the command execution subsystem the **Executable Group Order** for the command execution subsystem.

The Group Policy client extension provides the following:

Executable Group Order: Output data structure representing **Executable Group Order**.

Retrieve Default Executable Group Order: This method is used to retrieve from the command execution subsystem the **Default Executable Group Order** for the command execution subsystem.

The Group Policy client extension provides the following:

Default Executable Group Order: Output data structure representing **Default Executable Group Order**.

3.2.1.1.2 Abstract Interface of Executable Group

The **Executable Group** abstract interface provides the following method and parameters that are operations on an **Executable Group**.

Retrieve Executable List: The Group Policy client extension provides the following:

Execution Context: The context to which this **Executable List** belongs.

Executable List: Output data structure representing **Executable List**.

3.2.1.1.3 Abstract Interface of Executable List

The **Executable List** abstract interface provides the following methods and parameters that are operations on an **Executable List**.

Insert Program Into Executable List: The Group Policy client extension provides the following:

order: Position in the list at which insertion of a new executable program occurs.

list item: Input data structure contains the components **Executable Path** and **Executable Parameters**.

Remove Program From Executable List: The Group Policy client extension provides the following:

order: Position in the list from which removal of an executable program occurs.

Retrieve Program From Executable List: The Group Policy client extension provides the following:

order: Position in the list from which retrieval of an executable program occurs.

list item: Output data structure that is comprised of the components **Executable Path** and **Executable Parameters**.

Retrieve Next Program From Executable List: The Group Policy client extension provides the following to store the first item in the list:

list item: Output data structure that is comprised of the components **Executable Path** and **Executable Parameters**.

The first item is automatically removed upon return from this function.

Empty Executable List: This function empties the entire list.

Retrieve Size Of Executable List: This function returns the number of items in the list.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Process Group Policy

This extension is launched by the Group Policy: Core Protocol, which invokes this Process Group Policy event, whose abstract interface is specified in [MS-GPOL] section 3.2.4.1, to apply policies handled by this extension.

3.2.5 Message Processing Events and Sequencing Rules

For each GPO in the New or Changed GPO list, one file with the format specified in section 2.2 is read from the GP server, as specified later in this section. If any file fails to be read, the plug-in MUST ignore the failure and continue to read files for other GPOs.

Using the SecurityToken passed by GPOL, any remote file access in this section SHOULD be done under impersonation of the policy target as described in [MS-DTYP] section 2.7, Impersonation Abstract Interfaces.

For each GPO in the New or Changed GPO list, the Group Policy: Scripts Extension Encoding client plug-in MUST do the following to process the Scripts Group:

1. Perform a remote file open on the file specified by "<gpo path>\scripts\scripts.ini", where <gpo path> is the scoped GPO path derived from the **gPCFileSysPath** attribute of the GPO. If this

operation fails due to File Not Found, attempt to process the psscripts.ini in the following sequence. If the operation fails for any other reason, abort processing this GPO and continue with the next GPO.

2. Perform one or more remote file reads to read the entire contents of the opened file until the entire file has been read or an error in reading occurs.

3. Perform a remote file close to close the file.

4. The file is then parsed according to the format in section 2.2.2 to create the Scripts group. If the file does not conform to that format, parsing of the file MUST resume after the next end-of-line character (%0A or %0D in ABNF notation). If the file does conform to that format, the settings MUST be applied to the corresponding parameters in the abstract data model of the command execution subsystem. If the file does not conform to that format, the file MUST NOT be processed further by the client.

   Note that the <integer> specified under Keys in section 2.2.2 specifies an order; lower integers indicate that executable paths specified in the same section are to be invoked before those with higher values. The value of <integer>Cmdline becomes the executable path of the executable program, with <integer>Parameters becoming the parameters passed to the executable program.

For each GPO in the New or Changed GPO list, the Group Policy: Scripts Extension Encoding client plug-in SHOULD do the following to process the PSScripts Group:<7>

1. Perform a remote file open on the file specified by "<gpo path>\scripts\psscripts.ini", where <gpo path> is the scoped GPO path in the GPO.

```
If this operation fails due to File Not Found,
      If Scripts Group processing also failed due to File Not Found,
              abort processing.
      Else
              Proceed to step 7 assuming empty PSScripts Executable
              Group, PSLast Computer Executable order and PSLast User
              Executable order.
If this operation fails for any other reason,
      abort processing.
```

2. Perform one or more remote file reads to read the entire contents of the opened file until the entire file has been read or an error in reading occurs.

3. Perform a remote file close to close the file.

4. The file is then parsed according to the format in section 2.2.3 to create the PSScripts group. If the file does not conform to that format, parsing of the file MUST resume after the next end-of-line character (%0A or %0D in ABNF notation). If the file does conform to that format, the settings MUST be applied to the corresponding parameters in the abstract data model of the command execution subsystem.

5. If the <gpo path> is a Computer-scoped GPO path, determine the Computer Executable order as follows:

1. If the StartExecutePSFirst key is present in the ScriptsConfig section of the file, get its value. If the value is case-insensitive "true", the Computer Executable Group order is PSFirst. If the value is case-insensitive "false", the Computer Executable Group order is PSLast.

2. If the StartExecutePSFirst key is not present, the Default Computer Executable Group order is examined. If it equals 1, the Computer Executable Group order is PSFirst.

3. Otherwise, the Computer Executable Group order is PSLast.

6. If the <gpo path> is a User-scoped GPO path, determine the User Executable order as follows:

   1. If the StartExecutePSFirst key is present in the ScriptsConfig section of the file, get its value. If the value is case-insensitive "true", the User Executable Group order is PSFirst. If the value is case-insensitive "false", the User Executable Group order is PSLast.

   2. If the StartExecutePSFirst key is not present, the Default User Executable Group order is examined. If it equals 1, the User Executable Group order is PSFirst.

   3. Otherwise, the User Executable Group order is PSLast.

7. Process scripts in the Scripts and the PSScripts Executable Groups as follows:

   1. If the <gpo-path> is a Computer-scoped GPO path, process the Start up and Shut down scripts in the Scripts and the PSScripts Executable Groups following the Computer Executable Group order.

   2. If the <gpo path> is a User-scoped GPO path, process the Log on and Log off scripts in the Scripts and the PSScripts Executable Groups following the User Executable Group order.

      Note that the <integer> specified under Keys in section 2.2.2 specifies an order; lower integers indicate that executable paths specified in the same section are to be invoked before those with higher values. The value of <integer>Cmdline becomes the executable path of the executable program with <integer>Parameters becoming the parameters passed to the executable program.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

In the following example, when specific users log out, the command "\\managementserver\scripts\logtime.exe users \\archiveserver\logshare" is run followed by \\managementserver\scripts\OnLogoff.ps1 users \\archiveserver\logshare.

Also, when those users log on, the commands "\\managementserver\scripts\OnLogon.ps1 users -verbose, "defrag.exe systemdrive" and "\\managementserver\scripts\logstart.exe users -verbose" are run, in that sequence.

The following sequence of events occurs in this example:

1. The administrator invokes the administrative tool and uses the Group Policy: Core Protocol subsystem, which specifies to create settings for a given set of users, using the Group Policy: Scripts Extension Encoding plug-in.
2. The Group Policy: Scripts Extension Encoding plug-in to the administrative tool is invoked with a GPO path that it uses to construct a path to scripts.ini and psscripts.ini files. The plug-in tries to read these files and finds that they do not exist.
3. Because the files do not exist, the plug-in allows the administrator to enter commands to be used in new settings. When the administrator is done, the plug-in creates the following scripts.ini file:

```
[Logoff]
0CmdLine=\\managementserver\scripts\logtime.exe
0Parameters=users \\archiveserver\logshare
[Logon]
0CmdLine=defrag.exe
0Parameters=systemdrive
1CmdLine=\\managementserver\scripts\logstart.exe
1Parameters=users -verbose
```

The plug-in also creates the following psscripts.ini file:

```
[ScriptConfig]
StartExecutePSFirst=true
 EndExecutePSFirst=false
[Logoff]
0CmdLine=\\managementserver\scripts\OnLogoff.ps1
0Parameters=users \\archiveserver\logshare
[Logon]
0CmdLine=\\managementserver\scripts\OnLogon.ps1
0Parameters=users -verbose
```

The plug-in then copies the scripts.ini and psscripts.ini files to the remote location.

When the user logs on to a computer, the Group Policy: Core Protocol finds a GPO with the Group Policy: Scripts Extension Encoding CSE GUID, invokes the client plug-in, and gives it a GPO path. The client plug-in uses the GPO path to construct the path to the scripts.ini and psscripts.ini files, reads and parses the files, and then configures the command execution subsystem to execute the commands at the specified times.

5 Security

5.1 Security Considerations for Implementers

The key security issues are as follows:

- Implementers should help to ensure that the executable files run under the security context of the policy target.
- Implementers should prevent spoofing that might allow a non-administrator of the computer to alter the behavior of the executable file.
- Implementers should take into account that the data stored at the file system path of a script should be secured to be writable only to GPO administrators. For scripts that are stored inside the GPO's file system path, this is covered by the security measures used to secure the GPO itself. If scripts are stored in user-defined locations outside the GPO, the administrator that configures the Group Policy: Scripts Extension Encoding is responsible for securing the script. Implementers can encourage the user to be mindful of this consideration through the user interface of administrative tools.
- Implementers should note that any scripts or executable code configured to be executed by this protocol allow the administrators of the GPO from which the scripts were configured to become administrators on the computer or to invoke code in the context of a user that logs in to the client. The functionality of this protocol is one of the reasons that any administrators of a GPO have the capability of becoming administrators of the client computer.
- When an executable file (as specified by <integer>CmdLine) has no path specified, the implementer should search for the executable file in trusted locations. An example, using Defrag.exe, is presented in section 4.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

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

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 1.6: The earliest client version of Windows that supports this protocol is Windows 2000. The first server version of Windows that supports this protocol is Windows 2000 Server.

<2> Section 2.2.2: For the **TokValue** field, Windows systems support file system paths in the **UncPath** format and in the format of a local Windows file system path. No other syntaxes are supported for Windows systems.

<3> Section 3.1.5: Valid throughout the document: Reference to psscripts.ini file - the earliest client and server versions of Windows that support protocols and messages involving psscripts.ini file are Windows 7 and Windows Server 2008 R2 respectively.

<4> Section 3.1.5: Information about the Windows implementation of the administrative tool plug-in that is used to perform these steps is applicable to the following product versions of Windows:

- Windows 7
- Windows Server 2008 R2
- Windows 8

- Windows Server 2012
- Windows 8.1
- Windows Server 2012 R2

<5> Section 3.1.5: The Windows implementation of the administrative tool displays an error to the user if any errors in the protocol sequence in this section occur, which indicates that the GPO cannot be updated with the intentions specified through the protocol. The Windows implementation does not update the contents of the scripts.ini or psscripts.ini file if any of the protocol sequences in this section fail.

<6> Section 3.2.1.1: In Windows, the value of the **Default Executable Group Order** for Startup or Shutdown scripts is read from the registry location **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\RunComputerPSScriptsFirst**. It is of type REG_DWORD. The value 1 indicates PSFirst.

In Windows, the value of the **Default Executable Group Order** for Logon or Logoff scripts is read from these two registry locations, in order of priority:

1. **HKEY_LOCAL_MACHINE**

   \Software\Microsoft\Windows\CurrentVersion\Policies\System\RunUserPSScriptsFirst. It is of type REG_DWORD, a 32-bit number. The value 1 indicates PSFirst.

2. **HKEY_CURRENT_USER**

   \Software\Microsoft\Windows\CurrentVersion\Policies\System\RunUserPSScriptsFirst. It is of type REG_DWORD, a 32-bit number. The value 1 indicates PSFirst.

<7> Section 3.2.5: Information about the PSScripts group in Windows implementations is applicable to the following product versions of Windows:

- Windows 7
- Windows Server 2008 R2
- Windows 8
- Windows Server 2012
- Windows 8.1
- Windows Server 2012 R2

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

A

Abstract data model
  administrative tool plug-in
    overview 15
    PSScripts.ini 15
    Scripts.ini 15
  client plug-in
    command execution subsystem 18
    overview 17
Administrative tool plug-in
  abstract data model
    overview 15
    PSScripts.ini 15
    Scripts.ini 15
  higher-layer triggered events 15
  initialization 15
  local events 17
  message processing 15
  sequencing rules 15
  timer events 17
  timers 15
Applicability 10

B

Background 7

C

Capability negotiation 11
Change tracking 27
Client plug-in
  abstract data model
    command execution subsystem 18
    overview 17
  higher-layer triggered events - process group policy 20
  initialization 20
  local events 22
  message processing 20
  overview 17
  sequencing rules 20
  timer events 22
  timers 20

D

Data model - abstract
  administrative tool plug-in
    overview 15
    PSScripts.ini 15
    Scripts.ini 15
  client plug-in
    command execution subsystem 18
    overview 17

E

Examples 23

F

Fields - vendor-extensible 11

G

Glossary 6

H

Higher-layer triggered events
  administrative tool plug-in 15
  client plug-in - process group policy 20

I

Implementer - security considerations 24
Index of security parameters 24
Informative references 7
Initialization
  administrative tool plug-in 15
  client plug-in 20
Introduction 6

L

Local events
  administrative tool plug-in 17
  client plug-in 22

M

Message processing
  administrative tool plug-in 15
  client plug-in 20
Messages
  requirements - common 12
  transport 12

N

Normative references 7

O

Overview (synopsis) 7

P

Parameters - security index 24
Preconditions 10
Prerequisites 10
Product behavior 25
Psscripts.ini syntax 13

R

References
informative 7
normative 7
Relationship to other protocols 10

S

Scripts extension overview 8
Scripts.ini syntax 12
Security
implementer considerations 24
parameter index 24
Sequencing rules
administrative tool plug-in 15
client plug-in 20
Standards assignments 11
Syntax
Psscripts.ini 13
Scripts.ini 12

T

Timer events
administrative tool plug-in 17
client plug-in 22
Timers
administrative tool plug-in 15
client plug-in 20
Tracking changes 27
Transport 12
Triggered events - higher-layer
administrative tool plug-in 15
client plug-in - process group policy 20

V

Vendor-extensible fields 11
Versioning 11