[MS-NTHT-Diff]:

NTLM Over HTTP Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
10/22/2006	0.01	New	Version 0.01 release
1/19/2007	1.0	Major	Version 1.0 release
3/2/2007	1.1	Minor	Version 1.1 release
4/3/2007	1.2	Minor	Version 1.2 release
5/11/2007	1.3	Minor	Version 1.3 release
6/1/2007	1.3.1	Editorial	Changed language and formatting in the technical content.
7/3/2007	1.3.2	Editorial	Changed language and formatting in the technical content.
7/20/2007	1.3.3	Editorial	Changed language and formatting in the technical content.
8/10/2007	1.4	Minor	Clarified the meaning of the technical content.
9/28/2007	1.4.1	Editorial	Changed language and formatting in the technical content.
10/23/2007	2.0	Major	Updated and revised the technical content.
11/30/2007	2.1	Minor	Clarified the meaning of the technical content.
1/25/2008	2.1.1	Editorial	Changed language and formatting in the technical content.
3/14/2008	2.1.2	Editorial	Changed language and formatting in the technical content.
5/16/2008	2.1.3	Editorial	Changed language and formatting in the technical content.
6/20/2008	3.0	Major	Updated and revised the technical content.
7/25/2008	3.1	Minor	Clarified the meaning of the technical content.
8/29/2008	3.1.1	Editorial	Changed language and formatting in the technical content.
10/24/2008	3.1.2	Editorial	Changed language and formatting in the technical content.
12/5/2008	4.0	Major	Updated and revised the technical content.
1/16/2009	4.0.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	4.0.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	4.0.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	4.0.4	Editorial	Changed language and formatting in the technical content.
7/2/2009	4.1	Minor	Clarified the meaning of the technical content.
8/14/2009	4.2	Minor	Clarified the meaning of the technical content.
9/25/2009	4.2.1	Editorial	Changed language and formatting in the technical content.
11/6/2009	4.3	Minor	Clarified the meaning of the technical content.
12/18/2009	4.3.1	Editorial	Changed language and formatting in the technical content.
1/29/2010	4.3.2	Editorial	Changed language and formatting in the technical content.

Date	Revision History	Revision Class	Comments
3/12/2010	4.3.3	Editorial	Changed language and formatting in the technical content.
4/23/2010	4.3.4	Editorial	Changed language and formatting in the technical content.
6/4/2010	4.3.5	Editorial	Changed language and formatting in the technical content.
7/16/2010	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	4.3.5	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	4.4	Minor	Clarified the meaning of the technical content.
9/23/2011	4.4	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	5.0	Major	Updated and revised the technical content.
3/30/2012	5.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	6.0	Major	Updated and revised the technical content.
10/25/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	7.0	Major	Updated and revised the technical content.
11/14/2013	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	8.0	Major	Significantly changed the technical content.
10/16/2015	8.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
7/14/2016	8.0	None	No changes to the meaning, language, or formatting of the technical content.
6/1/2017	8.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	9.0	Major	Significantly changed the technical content.
9/12/2018	10.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

Microsoft provides support for NT LAN Manager (NTLM) (as specified in [MS-NLMP]) authentication in Microsoft Internet Explorer and Microsoft Internet Information Services (IIS) that uses the HTTP Protocol (for more information, see [RFC2616]) in addition to other standard authentication mechanisms. This provides the benefits of the NTLM Authentication Protocol for web applications when other authentication mechanisms (such as those specified in [RFC4559] and [RFC2617]) are not available.

Support for NTLM authentication is as specified in [RFC4559], using native NTLM Authentication Protocol (as specified in [MS-NLMP]) data units instead of encoded tokens (as specified in [RFC4178]). The tokens are still transmitted using base64 encoding. This document calls out the differences in the Microsoft implementation from what is specified in [RFC4559], where applicable.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Backus-Naur Form (BNF): A syntax used to describe context-free grammars, which is a prescribed way to describe languages ([RFC2616] section 2.1). See also "Augmented Backus-Naur Form (ABNF)".

client: A program that establishes connections for the purpose of sending requests (see [RFC2616] section 1.3).

proxy: An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients (see [RFC2616] section 1.3).

server: An application program that accepts connections in order to service requests by sending back responses (see [RFC2616] section 1.3).

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 (Updated Section) 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.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, http://www.rfc-editor.org/rfc/rfc2617.txt

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and Ingersoll, W., "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005, ~~http~~https://www.rfc-editor.org/rfc/rfc4178.txt

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.rfc-editor.org/rfc/rfc4559.txt

1.2.2 Informative References

None.

1.3 Overview

The NTLM over HTTP Protocol authentication variant is similar to the SPNEGO HTTP (as specified in [RFC4559]) authentication mechanism. Both are used to authenticate a web client to a web server. Although SPNEGO HTTP (as specified in [RFC4559]) works with both Kerberos and NTLM authentication, the NTLM over HTTP Protocol authentication variant only works with NTLM. The Kerberos protocol is not supported.

1.4 Relationship to Other Protocols

This document is a companion to the SPNEGO HTTP authentication document, as specified in [RFC4559]. It uses the augmented Backus-Naur Form (BNF), as specified in [RFC4559] section 4, and relies on both the non-terminals defined in that document and other aspects of the specification HTTP/1.1, as specified in [RFC2617]. For more information, see [RFC2616].

1.5 Prerequisites/Preconditions

NTLM over HTTP Protocol authentication assumes the following in addition to any assumptions specified in [MS-NLMP].

1. The web server is operating in an environment with a database of user identities, and the NT LAN Manager (NTLM) Authentication Protocol, as specified in [MS-NLMP], is available to authenticate those users.

2. The web client has implemented the NT LAN Manager (NTLM) Authentication Protocol, as specified in [MS-NLMP], so that it can participate in user authentication to the web server.

1.6 Applicability Statement

NTLM HTTP authentication is used in environments where SPNEGO-based Kerberos and NTLM HTTP authentication, as specified in [RFC4559], are not available, and the web client and server support NTLM authentication, as specified in [MS-NLMP].

1.7 Versioning and Capability Negotiation

Versioning and capability negotiation is handled by the HTTP protocols specified in [RFC2617] (for more information, see [RFC2616]). This protocol has no additional versioning or capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

Parameter	Value	Reference
HTTP auth-scheme	NTLM	[RFC2617] section 1.2

2 Messages

2.1 Transport

NTLM over HTTP Protocol messages are carried in the HTTP authentication exchanges as authentication data (auth-data), as specified in [RFC4559] sections 4.1 and 4.2.

2.2 Message Syntax

The use of NTLM over HTTP Protocol authentication is indicated by an HTTP authentication scheme (auth-scheme) NTLM. The authentication parameters (auth-params) that are exchanged are base64-encoded messages. For more details about auth-scheme and auth-params, see [RFC2617] section 1.2.

2.2.1 WWW-Authenticate Response Header

If the server receives a request for an access-protected object and an acceptable Authorization Request Header has not been sent, the server MUST respond with a "401 Unauthorized" status code and a WWW-Authenticate Response Header, per the framework in [RFC2616]. The initial WWW-Authenticate Response Header MUST NOT carry any auth-data. For more details about the text in this section, see [RFC2616], and specifically for the 401 status code, see [RFC2616] section 10.4.2.

The NTLM scheme operates as follows.

```
challenge= "NTLM" auth-data
auth-data = 1#( [ntlm-data] )
```

The meaning of the value of the directive used above is as follows:

```
ntlm-data
```

The ntlm-data directive contains the base64 encoding of a CHALLENGE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.2.

2.2.2 Authorization Request Header

Upon receipt of the response containing a WWW-Authenticate header from the server, the client is expected to retry the HTTP request with the authorization header, per the framework in [RFC2616] in the following.

```
credentials= "NTLM" auth-data2

auth-data2= 1#( [ntlm-data] )
```

The meaning of the value of the directive used above is as follows:

```
ntlm-data
```

The ntlm-data directive contains the base64 encoding of either an AUTHENTICATE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.3, or a NEGOTIATE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.1.

Any return code other than a client error HTTP 401 status code (for more information, see [RFC2616] section 10.4.2), represents successful authentication. If the client is not able to access the requested resource and the response status code is not HTTP 401, the problem is HTTP protocol-specific (for more information, see [RFC2616] section 10).

2.2.3 Proxy-Authenticate Response Header

If the client must authenticate itself to a proxy and an acceptable proxy-authorization header has not been sent, the proxy MUST respond with a "407 Proxy Authentication Required" status code (for more information, see [RFC2616] section 10.4.8) and a "Proxy-Authenticate" header, per the framework in [RFC2616]. The initial proxy-authenticate header MUST NOT carry any auth-data.

The NTLM scheme operates as follows.

```
challenge= "NTLM" auth-data3

auth-data3= 1#( [ntlm-data] )
```

The meanings of the values of the directives used above are as follows:

```
ntlm-data
```

The ntlm-data directive contains the base64 encoding of a CHALLENGE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.2.

2.2.4 Proxy-Authorization Request Header

Upon receipt of the response containing a proxy-authenticate header from the proxy, the client is expected to retry the HTTP request with the proxy-authorization header, per the framework in [RFC2616].

```
credentials= "NTLM" auth-data4

auth-data4= 1#( [ntlm-data] )
```

The meanings of the values of the directives used above are as follows:

```
ntlm-data
```

The ntlm_data directive contains the base64 encoding of either an AUTHENTICATE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.3, or a NEGOTIATE_MESSAGE, as specified in [MS-NLMP] section 2.2.1.1.

Any return code other than a client error HTTP 407 status code ([RFC2616] section 10.4.2), represents successful authentication. If the client is not able to access the requested resource and the response status code is not HTTP 407, the reason is HTTP protocol-specific. For details, see [RFC2616] section 10.

3 Protocol Details

3.1 Common Details

3.1.1 Abstract Data Model

The abstract data model follows the specifications in [RFC4559].

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The WWW-Authenticate header is only sent from the server. The Authorization header is only sent by the client. (For details, see [RFC2617] and also see [RFC2616] sections 14.47 and 14.8.) Clients, servers, and proxies MUST be compliant with [RFC2617] and [RFC2616].

The Proxy-Authenticate header is only sent from the proxy. The Proxy-Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] sections 14.33 and 14.34.) Clients, servers, and proxies MUST be compliant with [RFC2617] and [RFC2616].

3.1.5.1 Unexpected Messages

Unexpected messages cause the authentication to fail.

- If the server receives an unexpected message, it sends an HTTP 401 message to the client.
- If the client receives an unexpected message, it does not send a new request to the server.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

There are no local events other than those specified in [RFC4559].

3.2 Server Details

3.2.1 Abstract Data Model

The abstract data model follows the specifications in [RFC4559].

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The WWW-Authenticate header is only sent from the server. The Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] section 14.47.) Servers MUST be compliant with [RFC2617] and [RFC2616].

The Proxy-Authenticate header is only sent from the proxy. The Proxy-Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] sections 14.33 and 14.34.) Servers MUST be compliant with [RFC2617] and [RFC2616].

3.2.6 Timer Events

None.

3.2.7 Other Local Events

There are no local events other than those specified in [RFC4559].

3.3 Client Details

3.3.1 Abstract Data Model

The abstract data model follows the specifications in [RFC4559].

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

The WWW-Authenticate header is only sent from the server. The Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] section 14.47.) Servers MUST be compliant with [RFC2617] and [RFC2616].

The Proxy-Authenticate header is only sent from the proxy. The Proxy-Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] sections 14.33 and 14.34.) Servers MUST be compliant with [RFC2617] and [RFC2616].

3.3.6 Timer Events

None.

3.3.7 Other Local Events

There are no local events other than those specified in [RFC4559].

3.4 Proxy Details

3.4.1 Abstract Data Model

The abstract data model follows the specifications in [RFC4559].

3.4.2 Timers

None.

3.4.3 Initialization

None.

3.4.4 Higher-Layer Triggered Events

None.

3.4.5 Message Processing Events and Sequencing Rules

The WWW-Authenticate header is only sent from the server. The Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] section 14.47.) Servers MUST be compliant with [RFC2617] and [RFC2616].

The Proxy-Authenticate header is only sent from the proxy. The Proxy-Authorization header is only sent by the client. (For more information, see [RFC2617] and [RFC2616] sections 14.33 and 14.34.) Servers MUST be compliant with [RFC2617] and [RFC2616].

3.4.6 Timer Events

None.

3.4.7 Other Local Events

There are no local events other than those specified in [RFC4559].

4 Protocol Examples

4.1 Server Examples

This scenario shows the messages exchanged when a web client requests an access-protected document from a web server using a GET method request at the URL: http://www.nowhere.org/dir/index.html.

```
C: GET dir/index.html
```

The first time the client requests the document, no Authorization header is sent; so the server responds with the following.

```
S: HTTP/1.1 401 Unauthorized
S: WWW-Authenticate: NTLM
```

The client obtains the local user credentials by using the [MS-NLMP] security package and then generates a new GET request to the server. The request contains an Authorization header with an NTLM NEGOTIATE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.1) in ntlm-data.

```
C: GET dir/index.html
C: Authorization: NTLM tESsBmE/yNY3lb6a0L6vVQEZNqwQn0s8Unew
```

The server decodes the ntlm-data that is contained in the auth-data2 base64-encoded data and passes this to its implementation of [MS-NLMP]. If the server accepts this authentication data from the client, it responds with an HTTP 401 code (for more information, see [RFC2616] section 10.2) and a WWW-Authenticate header with an NTLM CHALLENGE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.2) in ntlm-data.

```
S: HTTP/1.1 401 Unauthorized
S: WWW-Authenticate: NTLM yNY3lb6a0L6vVQEZNqwQn0s8UNew33KdKZvG+Onv
```

The client decodes the ntlm-data that is contained in the auth-data base64-encoded data and passes this to its implementation of [MS-NLMP]. If this authentication data is valid, the client responds by reissuing the GET request with an Authorization header that contains an NTLM AUTHENTICATE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.3) in ntlm-data.

```
C: GET dir/index.html
C: Authorization: NTLM kGaXHz6/owHcWRlvGFk8ReUZKHo=QEZNqwQn0s8U
```

The server decodes the ntlm-data that is contained in the auth-data2 base64-encoded data and passes this to its implementation of [MS-NLMP]. If the server accepts this authentication data from the client, it responds with an HTTP 2xx code (for more information, see [RFC2616] section 10.2) indicating success. The requested content is also included in the server response.

Note The base64 values used previously are for illustrative purposes only and do not represent valid base64-encoded NTLM messages.

4.2 Proxy Examples

This scenario shows the messages that are exchanged when a web client requests an access-protected document from a proxy using a GET method request at the URL: http://www.nowhere.org/dir/index.html.

```
C: GET dir/index.html
```

The first time the client requests the document, no Proxy-Authorization header is sent; so the proxy responds with the following.

```
S: HTTP/1.1 407 Proxy Authentication Required
S: Proxy-Authenticate: NTLM
```

The client obtains the local user credentials using the [MS-NLMP] security package and then generates a new GET request to the proxy. The request contains a Proxy-Authorization header with an NTLM NEGOTIATE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.1) in ntlm-data.

```
C: GET dir/index.html
C: Proxy-Authorization: NTLM tESsBmE/yNY3lb6a0L6vVQEZNqwQn0s8Unew
```

The proxy decodes the ntlm-data that is contained in the auth-data2 base64-encoded data and passes this to its implementation of [MS-NLMP]. If the proxy accepts this authentication data from the client, it responds with an HTTP 407 code (for more information, see [RFC2616] section 10.2) and a Proxy-Authenticate header with an NTLM CHALLENGE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.2) in ntlm-data.

```
S: HTTP/1.1 407 Proxy Authentication Required
S: Proxy-Authenticate: NTLM yNY3lb6a0L6vVQEZNqwQn0s8UNew33KdKZvG+Onv
```

The client decodes the ntlm-data that is contained in the auth-data base64-encoded data and passes this to its implementation of [MS-NLMP]. If this authentication data is valid, the client responds by reissuing the GET request with a Proxy-Authorization header that contains an NTLM AUTHENTICATE_MESSAGE (as specified in [MS-NLMP] section 2.2.1.3) in ntlm-data.

```
C: GET dir/index.html
C: Proxy-Authorization: NTLM kGaXHz6/owHcWRlvGFk8ReUZKHo=QEZNqwQn0s8U
```

The proxy decodes the ntlm-data that is contained in the auth-data2 base64-encoded data and passes this to its implementation of [MS-NLMP]. If the proxy accepts this authentication data from the client, it responds with an HTTP 2xx code (for more information, see [RFC2616] section 10.2) indicating success. The requested content is also included in the proxy response.

Note The base64 values used previously are for illustrative purposes only and do not represent valid base64-encoded NTLM messages.

5 Security

5.1 Security Considerations for Implementers

The NTLM Authentication Protocol (see [MS-NLMP]) does not provide any facilities for mutual authentication; therefore, server identities cannot be verified. Other security considerations are as specified in [RFC4559] section 6.

5.2 Index of Security Parameters

None.

6 (Updated Section) Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

Windows Releases

- Windows NT operating system
- 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
- Windows 10 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Windows Server 2019 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

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

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

Section	Description	Revision class
6 Appendix A: Product Behavior	Added Windows Server 2019 to the list of applicable products.	Major

8 Index

S

T

V

W