

[MS-OXWSAUTID]: Authentication Identification Extension

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

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

Revision Summary

Date	Revision History	Revision Class	Comments
02/10/2010	1.0.0	Major	Initial Availability.
05/05/2010	1.1.0	Minor	Updated the technical content.
08/04/2010	1.2	Minor	Clarified the meaning of the technical content.
11/03/2010	1.2	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.0	Major	Significantly changed the technical content.
08/05/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/07/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	3.0	Major	Significantly changed the technical content.
04/27/2012	3.1	Minor	Clarified the meaning of the technical content.
07/16/2012	3.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	3.2	Minor	Clarified the meaning of the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References.....	4
1.2.1	Normative References.....	4
1.2.2	Informative References	5
1.3	Overview	5
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions	6
1.6	Applicability Statement.....	6
1.7	Versioning and Capability Negotiation.....	6
1.8	Vendor-Extensible Fields.....	6
1.9	Standards Assignment.....	6
2	Messages.....	7
2.1	Transport.....	7
2.2	Message Syntax	7
2.2.1	X-Nego-Capability Header	7
3	Protocol Details	8
3.1	Server Details	8
3.1.1	Abstract Data Model	8
3.1.2	Timers	8
3.1.3	Initialization	8
3.1.4	Higher-Layer Triggered Events.....	8
3.1.5	Message Processing Events and Sequencing Rules.....	8
3.1.6	Timer Events	8
3.1.7	Other Local Events	9
3.2	Client Details.....	9
3.2.1	Abstract Data Model	9
3.2.2	Timers	9
3.2.3	Initialization	9
3.2.4	Higher-Layer Triggered Events.....	9
3.2.5	Message Processing Events and Sequencing Rules.....	9
3.2.6	Timer Events	9
3.2.7	Other Local Events	9
4	Protocol Examples.....	10
5	Security.....	11
5.1	Security Considerations for Implementers.....	11
5.2	Index of Security Parameters	11
6	Appendix A: Product Behavior.....	12
7	Change Tracking.....	13
8	Index	15

1 Introduction

The Authentication Identification Extension enables a client to communicate the client-supported authentication schemes to a server so that the server can identify shared authentication schemes in a response back to the client.

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 [\[MS-GLOS\]](#):

Augmented Backus-Naur Form (ABNF)
Hypertext Transfer Protocol (HTTP)
NT LAN Manager (NTLM) Authentication Protocol

The following terms are defined in [\[MS-OXGLOS\]](#):

Transport Layer Security (TLS)

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described 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 technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

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-N2HT] Microsoft Corporation, "[Negotiate and Nego2 HTTP Authentication Protocol Specification](#)".

[MS-NTHT] Microsoft Corporation, "[NTLM Over HTTP Protocol Specification](#)".

[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>

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, <http://www.ietf.org/rfc/rfc2246.txt>

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

[RFC4121] Zhu, L., Jaganathan, K., and Hartman, S., "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005, <http://www.ietf.org/rfc/rfc4121.txt>

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

1.2.2 Informative References

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

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol](#)".

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)".

[MS-OXPROTO] Microsoft Corporation, "[Exchange Server Protocols System Overview](#)".

[MS-OXWAVLS] Microsoft Corporation, "[Availability Web Service Protocol](#)".

[MS-OXWCONFIG] Microsoft Corporation, "[Web Service Configuration Protocol](#)".

[MS-OXWMT] Microsoft Corporation, "[Mail Tips Web Service Extensions](#)".

[MS-OXWOAB] Microsoft Corporation, "[Offline Address Book \(OAB\) Retrieval File Format](#)".

[MS-OXWOOF] Microsoft Corporation, "[Out of Office \(OOO\) Web Service Protocol](#)".

[MS-OXWSMSHR] Microsoft Corporation, "[Folder Sharing Web Service Protocol](#)".

[MS-OXWSMTRK] Microsoft Corporation, "[Message Tracking Web Service Protocol](#)".

[MS-OXWUMS] Microsoft Corporation, "[Voice Mail Settings Web Service Protocol](#)".

[MS-RPCH] Microsoft Corporation, "[Remote Procedure Call over HTTP Protocol Specification](#)".

1.3 Overview

In most implementations, clients that support the Negotiate and Nego2 HTTP Authentication Protocol, as described in [\[MS-N2HT\]](#), support either the Kerberos or **NTLM** authentication providers. In cases where servers do support the use of custom Negotiate-capable authentication providers without fallback to Kerberos or NTLM, an additional client capability protocol must be indicated to the server.

The **X-Nego-Capability** header is used to specify the authentication providers that are supported by a client within the Negotiate or Nego2 HTTP Authentication Protocol. The server compares the authentication providers received in the **X-Nego-Capability** header to Negotiate-capable authentication providers that are available to the server. The server sends a response to the client that specifies the Negotiate or Nego2 HTTP Authentication Protocol in **WWW-Authenticate** response headers if there is at least one capable authentication service provider that is shared by both the server and client. The client uses the values from the **WWW-Authenticate** headers to determine which authentication scheme to use.

1.4 Relationship to Other Protocols

The **X-Nego-Capability** header is an extension to the **Hypertext Transfer Protocol (HTTP)**, as described in [\[RFC2616\]](#).

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [\[MS-OXPROTO\]](#).

1.5 Prerequisites/Preconditions

The successful use of the **X-Nego-Capability** header is based on the assumption that both the client and the server can authenticate users and that the client and the server can identify the available authentication services.

1.6 Applicability Statement

This extension is applicable to clients and server applications that dynamically determine the best authentication scheme that is shared by both the client and server. The **X-Nego-Capability** header is applicable to environments where the client and server support the Negotiate and Nego2 HTTP Authentication Protocol, as described in [\[MS-N2HT\]](#).

This extension is used to identify client authentication providers when the following are used:

- Autodiscover Publishing and Lookup Protocol, as described in [\[MS-OXDCLI\]](#)
- Availability Web Service Protocol, as described in [\[MS-OXWAVLS\]](#)
- Web Service Configuration Protocol, as described in [\[MS-OXWCONFIG\]](#)
- Mail Tips Web Service Extensions, as described in [\[MS-OXWMT\]](#)
- Offline Address Book (OAB) Retrieval File Format, as described in [\[MS-OXWOAB\]](#)
- Out of Office (OOF) Web Service Protocol, as described in [\[MS-OXWOOF\]](#)
- Folder Sharing Web Service Protocol, as described in [\[MS-OXWSMSHR\]](#)
- Message Tracking Web Service Protocol, as described in [\[MS-OXWSMTRK\]](#)
- Voice Mail Settings Web Service Protocol, as described in [\[MS-OXWUMS\]](#)

This protocol is not applicable to the Remote Procedure Call over HTTP Protocol, as described in [\[MS-RPCH\]](#).

1.7 Versioning and Capability Negotiation

Versioning and capability negotiation is handled at the HTTP layer of the protocol stack, as described in [\[RFC2616\]](#). Values for capability negotiation of the actual supported authentication schemes come from system registry entries on both the client and server.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignment

None.

2 Messages

2.1 Transport

The **X-Nego-Capability** header is carried in HTTP requests (as specified in [\[RFC2616\]](#)).

2.2 Message Syntax

Negotiation capability information is indicated in HTTP requests by the **X-Nego-Capability** header.

2.2.1 X-Nego-Capability Header

The **X-Nego-Capability** header specifies the Negotiate-capable authentication schemes that are supported by a client.

The format for the **X-Nego-Capability** HTTP request header in **Augmented Backus-Naur Form (ABNF)**, as specified in [\[RFC5234\]](#), is as follows.

```
X-Nego-Capability = "X-Nego-Capability:" SP Nego-scheme-range
Nego-scheme-range = Negotiable-security-package *( "," SP Negotiable-security-package)
Negotiable-security-package = 1*CHAR
```

3 Protocol Details

3.1 Server Details

3.1.1 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 abstract data model for negotiate capability has one abstract data element that lists the authentication schemes that are supported by the client. Additionally, the intersection of the supported client and server authentication schemes results in a list of **WWW-Authenticate** headers in the response that is received by the client.

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 server MUST return **WWW-Authenticate** headers that identify each authentication scheme the server and the client have in common.

The server SHOULD respond to a client request that contains the **X-Nego-Capability** header with a list of tokens that represent the common HTTP authentication schemes that are shared by the client and server if the credentials sent in the initial request prompt an HTTP 401 response. The authentication scheme tokens are contained in **WWW-Authenticate** headers that are sent by the server. The server SHOULD include the **WWW-Authenticate: Nego2** or the **WWW-Authenticate: Negotiate** response headers when the **X-Nego-Capability** header that is received from the client includes at least one Negotiate-capable authentication provider that is mutually supported by the server, and the response is an HTTP 401 error, as specified in [\[RFC2616\]](#).

The server can return **WWW-Authenticate** tokens other than **Negotiate** and **Nego2** regardless of the **X-Nego-Capability** header value sent by the client.

The server currently only expects the **MsoidSSP X-Nego-Capability** header token from the client.

The server will ignore the **X-Nego-Capability** header if the header does not conform to the syntax specified in section [2.2.1](#).

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

The Authentication Extension client sends an **X-Nego-Capability** header that specifies the authentication schemes supported by the client. The client receives a response from the server that indicates the common authentication schemes supported by both the client and server.

The **X-Nego-Capability** header MUST be populated with tokens that identify the Negotiate-capable authentication schemes that are supported by the client. The format for the **X-Nego-Capability** header is specified in section [2.2.1](#). The client's initial request SHOULD NOT attempt authorization. The initial request to obtain shared authentication schemes MUST include the **X-Nego-Capability** header.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

The client initializes the **X-Nego-Capability** header with a list of tokens that represent the authentication schemes that are supported by the client.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

Clients SHOULD NOT include the **X-Nego-Capability** header on requests that are not protected by **Transport Layer Security (TLS)** (as specified in [\[RFC2246\]](#)), because this can allow a man-in-the-middle to downgrade the authentication scheme that is exposed to clients. [<1>](#)

The **X-Nego-Capability** header MUST be sent in a request to the server before the client attempts authorization so that the server can indicate to the client which authentication schemes it can use.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

An HTTP 1.1 client requests a resource from a server by sending an HTTP **GET** request, as shown in the following example.

```
GET /autodiscover/autodiscover.xml HTTP/1.1
Host: autodiscover.contoso.com
Content-Type: text/xml
User-Agent: mso/14.0 (win NT 6.1; Microsoft Office Outlook 14.0.1234
X-Nego-Capability: Nego2, Negotiate, Kerberos, NTLM
```

In this request, the **X-Nego-Capability** header identifies four supported authentication schemes: Nego2, Negotiate, Kerberos, and NTLM.

The server is expected to respond to this request with an HTTP 401 with a set of **WWW-Authenticate** headers that identify the shared authentication schemes. The client responds with the best available authentication scheme by using the **Authorization** header.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

The following table lists the authentication mechanisms that are supported with the **X-Nego-Capability** header.

Security parameter	Reference
NTLM	[MS-NHTT]
Kerberos	[RFC4121]
Negotiate	[MS-N2HT]
Nego2	[MS-N2HT]

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:

- Microsoft Outlook 2010
- Microsoft Outlook 2013

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 3.2.5:](#) Outlook 2010 and Outlook 2013 do not send the **X-Nego-Capability** header unless the request is protected by Transport Layer Security (TLS), as described in [\[RFC2246\]](#).

7 Change Tracking

This section identifies changes that were made to the [MS-OXWSAUTID] protocol document between the July 2012 and October 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

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

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

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

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

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

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

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.1 Glossary	Added "Augmented Backus-Naur Form (ABNF)" to the list of terms defined in [MS-GLOS].	N	Content updated.
2.2.1 X-Nego-Capability Header	Added a link to the glossary for the term "ABNF".	N	Content updated.
6 Appendix A: Product Behavior	Removed Outlook 2007 from the list of applicable products.	Y	Content updated.

8 Index

A

Abstract data model
[client](#) 9
[server](#) 8
[Applicability](#) 6

C

[Capability negotiation](#) 6
[Change tracking](#) 13
Client
[abstract data model](#) 9
[higher-layer triggered events](#) 9
[initialization](#) 9
[message processing](#) 9
[other local events](#) 9
[overview](#) 9
[sequencing rules](#) 9
[timer events](#) 9
[timers](#) 9

D

Data model - abstract
[client](#) 9
[server](#) 8

F

[Fields - vendor-extensible](#) 6

G

[Glossary](#) 4

H

Higher-layer triggered events
[client](#) 9
[server](#) 8

I

[Implementer - security considerations](#) 11
[Index of security parameters](#) 11
[Informative references](#) 5
Initialization
[client](#) 9
[server](#) 8
[Introduction](#) 4

M

Message processing
[client](#) 9
[server](#) 8
Messages

[transport](#) 7
[X-Nego-Capability Header](#) 7

N

[Normative references](#) 4

O

Other local events
[client](#) 9
[server](#) 9
[Overview \(synopsis\)](#) 5

P

[Parameters - security index](#) 11
[Preconditions](#) 6
[Prerequisites](#) 6
[Product behavior](#) 12

R

[References](#) 4
[informative](#) 5
[normative](#) 4
[Relationship to other protocols](#) 5

S

Security
[implementer considerations](#) 11
[parameter index](#) 11
Sequencing rules
[client](#) 9
[server](#) 8
Server
[abstract data model](#) 8
[higher-layer triggered events](#) 8
[initialization](#) 8
[message processing](#) 8
[other local events](#) 9
[sequencing rules](#) 8
[timer events](#) 8
[timers](#) 8

T

Timer events
[client](#) 9
[server](#) 8
Timers
[client](#) 9
[server](#) 8
[Tracking changes](#) 13
[Transport](#) 7
Triggered events - higher-layer
[client](#) 9

[server](#) 8

V

[Vendor-extensible fields](#) 6

[Versioning](#) 6

X

[X-Nego-Capability Header message](#) 7