

[MS-OAUTH2EX]: OAuth 2.0 Authentication Protocol Extensions

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
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	Major	Significantly changed 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	5
1.6	Applicability Statement	5
1.7	Versioning and Capability Negotiation	5
1.8	Vendor-Extensible Fields	5
1.9	Standards Assignments	5
2	Messages	6
2.1	Transport	6
2.2	Message Syntax	6
3	Protocol Details	7
3.1	Common Details	7
3.1.1	Abstract Data Model	7
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
3.1.5.1	Protected Resource	8
3.1.5.1.1	Request Access	8
3.1.5.1.1.1	Request Body	8
3.1.5.1.1.2	Response Body	8
3.1.5.1.1.3	Processing Details	8
3.1.6	Timer Events	8
3.1.7	Other Local Events	9
4	Protocol Examples	10
4.1	Example request for token to access a resource	10
4.2	Response with token to access requested resource	10
5	Security	11
5.1	Security Considerations for Implementers	11
5.2	Index of Security Parameters	11
6	Appendix A: Full JSON Schema	12
7	Appendix B: Product Behavior	13
8	Change Tracking	14
9	Index	16

1 Introduction

OAuth 2.0 Authentication Protocol Extensions describes extensions to the OAuth 2.0 Authentication Protocol. These extensions consist of additional parameters in the request **URI** and the **JSON** objects returned in the **HTTP** response body.

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

Coordinated Universal Time (UTC)
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
principal
realm
X.509

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

JavaScript Object Notation (JSON)
Representational State Transfer (REST)
security token service (STS)
tenant
token
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)

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.

[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>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

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

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

1.3 Overview

This document describes extensions to the OAuth 2.0 Authentication Protocol. These extensions consist of additional parameters in the request URI and the JSON objects returned in the HTTP response body. These extensions provide additional functionality, such as finer granularity of token expiry periods, which is incremental over what is provided by the base specification.

1.4 Relationship to Other Protocols

1.5 Prerequisites/Preconditions

Applications using these extensions must have their **principal** identifiers registered as members of the **realm** managed by the **security token service (STS)** and are able to authenticate to it by exchanging secrets (public key material carried in **X.509** certificates). The applications must also trust tokens issued by the STS. These preconditions are established during deployment of new **tenants**.

1.6 Applicability Statement

These extensions are used in OAuth 2.0 Authentication requests for obtaining an access **token** for use with **REST** bases service requests.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

No specific parameters are passed to the transport layer. The transport layer, **HTTPS**, is used for securing messages and tokens sent over the wire.

2.2 Message Syntax

Parameters in these extensions are either transmitted in the request body in **URL**-encoded form or in the response as part of a JSON object. Tokens returned in conjunction with use of this protocol conform to the JWT specification.

XML based serialization is not used with these extensions or underlying protocols.

3 Protocol Details

3.1 Common Details

This specification defines additional parameters for use with OAuth 2.0 Authentication Protocol messages. These parameters convey additional semantics related to the access token being requested and about the access token returned as part of these messages. This specification does not define any additional messages over the base specification.

3.1.1 Abstract Data Model

The following parameters are used in conjunction with the OAuth 2.0 Authentication Protocol.

Mandatory parameters when using the extensions defined in this specification.

Parameter name: resource

Parameter usage location: Request body, URL encoded

Type: string, URI value expected

Semantics: This parameter indicates the target resource that the caller wants to talk to. Value is expected to be URI format as defined in RFC 3986

Parameter name: not_before

Parameter usage location: Response body in JSON object

Type: string, **UTC** value expected

Semantics: Token returned in the OAuth parameter access_token as part of this response is not valid before the time specified in this parameter.

Parameter name: expires_on

Parameter usage location: Response body in JSON object

Type: string, UTC value expected

Semantics: Token returned in the OAuth parameter access_token as part of this response will no longer be valid at the time specified in this parameter.

Optional parameters when using the extensions defined in this specification.

Parameter name: realm

Parameter usage location: Request body, URL encoded or Response body in JSON object

Type: string

Semantics: Indicates the realm that the caller is part of.

Parameter name: created_on

Parameter usage location: Request body, URL-encoded or Response body in JSON object

Type: string, UTC value expected

Semantics: Token returned in the OAuth parameter `access_token` as part of this was created at the time specified in this parameter.

Parameter name: `expires_in`

Parameter usage location: Response body in JSON object

Type: string, int value expected.

Semantics: Lifetime of the token returned in the OAuth parameter `access_token` as part of this response specified in seconds.

3.1.2 Timers

None.

3.1.3 Initialization

Connections made using the underlying OAuth 2.0 Authentication Protocol are what initiate use of the extensions defined in this specification.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The token request contains the elements indicated in section [3.1.1](#).

3.1.5.1 Protected Resource

This resource indicates the protected resource that the user is trying to get access to.

3.1.5.1.1 Request Access

For gaining access to a resource, an access token with the elements mentioned in section [3.1.1](#) need to be presented.

3.1.5.1.1.1 Request Body

Refer to the section [4.1](#).

3.1.5.1.1.2 Response Body

Refer to the section [4.2](#).

3.1.5.1.1.3 Processing Details

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

5 Security

5.1 Security Considerations for Implementers

Security considerations described in underlying specifications and in profiles referencing this extension specification should be considered when using these extensions.

5.2 Index of Security Parameters

None.

6 Appendix A: Full JSON Schema

For ease of implementation, the following is the full JSOM schema for this protocol.

```
"not_before" : {
  "type" : "string",
  "format" : "date-time"
},
"expires_on" : {
  "type" : "string",
  "format" : "date-time"
},
"realm" : {
  "type" : "string",
  "format" : "uri"
},
"created_on" : {
  "type" : "string",
  "format" : "date-time"
},
"expires_in" : {
  "type" : "integer"
}
}
```

7 Appendix B: 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® Lync® 2013
- Microsoft® Lync® Server 2013
- Microsoft® Exchange Server 2013
- Microsoft® SharePoint® Server 2013
- Microsoft® SharePoint® Foundation 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.

8 Change Tracking

This section identifies changes that were made to the [MS-OAUTH2EX] protocol document between the September 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 Introduction	Updated all preliminary information in the document.	Y	Content updated.

9 Index

A

[Applicability](#) 5

[Vendor-extensible fields](#) 5

[Versioning](#) 5

C

[Capability negotiation](#) 5

[Change tracking](#) 14

F

[Fields - vendor-extensible](#) 5

G

[Glossary](#) 4

I

[Implementer - security considerations](#) 11

[Index of security parameters](#) 11

[Informative references](#) 5

[Introduction](#) 4

N

[Normative references](#) 4

O

[Overview \(synopsis\)](#) 5

P

[Parameters - security index](#) 11

[Preconditions](#) 5

[Prerequisites](#) 5

[Product behavior](#) 13

R

References

[informative](#) 5

[normative](#) 4

S

Security

[implementer considerations](#) 11

[parameter index](#) 11

[Standards assignments](#) 5

T

[Tracking changes](#) 14

V