[MS-OXDISCO]:

Autodiscover HTTP Service 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.
- **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.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability.
6/27/2008	1.0	Major	Initial Release.
8/6/2008	1.01	Minor	Updated references to reflect date of initial release.
9/3/2008	1.02	Minor	Revised and edited technical content.
10/1/2008	1.03	Minor	Revised and edited technical content.
12/3/2008	1.04	Minor	Revised and edited technical content.
4/10/2009	2.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	4.0.0	Major	Updated and revised the technical content.
2/10/2010	4.1.0	Minor	Updated the technical content.
5/5/2010	4.1.1	Editorial	Revised and edited the technical content.
8/4/2010	4.2	Minor	Clarified the meaning of the technical content.
11/3/2010	4.2	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	5.0	Major	Significantly changed the technical content.
8/5/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
10/7/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	6.0	Major	Significantly changed the technical content.
4/27/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.0	Major	Significantly changed the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	9.0	Major	Significantly changed the technical content.
11/18/2013	9.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	10.0	Major	Significantly changed the technical content.
5/26/2015	10.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	11.0	Major	Significantly changed the technical content.
9/14/2015	12.0	Major	Significantly changed the technical content.
6/13/2016	13.0	Major	Significantly changed the technical content.
9/14/2016	13.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 8
1.2.1 Normative References 8
1.2.2 Informative References 9
1.3 Overview 9
1.4 Relationship to Other Protocols 9
1.5 Prerequisites/Preconditions 10
1.6 Applicability Statement 10
1.7 Versioning and Capability Negotiation 10
1.8 Vendor-Extensible Fields 10
1.9 Standards Assignments 10

2 Messages 11
2.1 Transport 11
2.2 Message Syntax 11
2.2.1 Service Connection Point Publication Service Objects 11
2.2.1.1 Service Connection Point Object Syntax 11
2.2.1.2 Searching for Service Connection Point Objects 11
2.2.1.3 Creating Service Connection Point Objects 12
2.2.2 DNS SRV Queries 12
2.2.3 HTTP 302 Redirection 12
2.2.4 Email Addresses 13
2.2.5 Autodiscover Server URI Results 13

3 Protocol Details 14
3.1 Client Details 14
3.1.1 Abstract Data Model 14
3.1.2 Timers 14
3.1.3 Initialization 14
3.1.4 Higher-Layer Triggered Events 14
3.1.5 Message Processing Events and Sequencing Rules 14
3.1.5.1 Query a Well-Known LDAP Server for Service Connection Point Objects 15
3.1.5.2 Locations Found Directly From the Email Domain 15
3.1.5.3 Locations Found from SRV DNS Records 15
3.1.5.4 Locations Found by an HTTP Redirect 16
3.1.6 Timer Events 16
3.1.7 Other Local Events 16
3.2 Server Details 16
3.2.1 Abstract Data Model 16
3.2.2 Timers 16
3.2.3 Initialization 16
3.2.3.1 Locations Published in LDAP via Service Connection Point Objects with an HTTP URI 16
3.2.3.2 Locations Published in LDAP via Service Connection Point objects with an LDAP URI 17
3.2.3.3 Locations Published in DNS as Autodiscover.<Domain> and <Domain> 17
3.2.3.4 Locations Published in DNS By Using SRV Records 17
3.2.3.5 Locations Published Through an HTTP GET 17
3.2.4 Higher-Layer Triggered Events 18
3.2.5 Message Processing Events and Sequencing Rules 18
3.2.6 Timer Events 18
3.2.7 Other Local Events 18

4 Protocol Examples 19

4.1 Publishing an Autodiscover Server Location 19
4.2 Autodiscover Client Querying for Autodiscover Servers 20

5 Security 22
5.1 Security Considerations for Implementers 22
5.2 Index of Security Parameters 22

6 Appendix A: Product Behavior 23

7 Change Tracking 24

8 Index 25

1 Introduction

The Autodiscover HTTP Service Protocol provides a way for **Autodiscover clients** to find **Autodiscover servers**. This protocol extends the **Domain Name System (DNS)** and directory services to make the location and settings of mail servers available to clients. This enables the clients to use the functionality specified in the Autodiscover Publishing and Lookup Protocol [MS-OXDSCLI].

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:

Active Directory: A general-purpose network directory service. **Active Directory** also refers to the Windows implementation of a directory service. **Active Directory** stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in **Active Directory**. **Active Directory** is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, **Lightweight Directory Access Protocol (LDAP)** versions 2 and 3, Kerberos, and **DNS**.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

Autodiscover client: A client that queries for a set of server locations where setup and configuration information for an [RFC2821]-compliant email address is stored.

Autodiscover server: A server in a managed environment that makes setup and configuration information available to **Autodiscover clients**. The location of Autodiscover servers is made available via the Autodiscover HTTP Service Protocol, as described in [MS-OXDISCO].

distinguished name (DN): (1) A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

(2) In **Lightweight Directory Access Protocol (LDAP)**, an LDAP Distinguished Name, as described in [RFC2251] section 4.1.3. The DN of an object is the DN of its parent, preceded by the RDN of the object. For example: CN=David Thompson, OU=Users, DC=Microsoft, DC=COM. For definitions of CN and OU, see [RFC2256] sections 5.4 and 5.12, respectively.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the **Active Directory** service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names (1) to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

email address: A string that identifies a user and enables the user to receive Internet messages.

fully qualified domain name (FQDN): An unambiguous domain name (2) that gives an absolute location in the **Domain Name System's (DNS)** hierarchy tree, as defined in [RFC1035] section 3.1 and [RFC2181] section 11.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

LDAP Data Interchange Format (LDIF): A standard that defines how to import and export directory data between directory servers that use the **Lightweight Directory Access Protocol (LDAP)**, as described in [RFC2849].

Lightweight Directory Access Protocol (LDAP): The primary access protocol for **Active Directory**. Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a directory service (DS), as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

port: A TCP/IP numbered connection point that is used to transfer data.

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data-a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication (2) using X.509 certificates (2). For more information, see [X509]. The SSL protocol is precursor to **Transport Layer Security (TLS)**. The TLS version 1.0 specification is based on SSL version 3.0 [SSL3].

service binding information: The **URIs** that are needed to bind to a service.

service connection point: An object that is made available by a directory service and that clients can use to discover **Autodiscover servers**.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group. See [RFC4346].

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

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 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-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol".

[RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987, http://www.ietf.org/rfc/rfc1034.txt

[RFC1823] Howes, T., and Smith, M., "The LDAP Application Program Interface", RFC 1823, August 1995, http://www.rfc-editor.org/rfc/rfc1823.txt

[RFC1960] Howes, T., "A String Representation of LDAP Search Filters", RFC 1960, June 1996, http://www.rfc-editor.org/rfc/rfc1960.txt

[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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.rfc-editor.org/rfc/rfc2396.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

[RFC2782] Gulbrandsen, A., Vixie, P., and Esibov, L., "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000, http://www.ietf.org/rfc/rfc2782.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.txt

[RFC2849] Good, G., "The LDAP Data Interchange Format (LDIF) - Technical Specification", RFC 2849, June 2000, http://www.ietf.org/rfc/rfc2849.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[RFC4210] Adams, C., Farrell, S., Kause, T., and Mononen, T., "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, September 2005, http://www.rfc-editor.org/rfc/rfc4210.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

None.

1.3 Overview

The Autodiscover HTTP Service Protocol allows a managed network (**domain**) to expose Autodiscover servers to clients that are configured with an email address.

This protocol provides a way for Autodiscover clients to find Autodiscover servers. The client starts with an email address of the form <local-part>@<domain> and expands it to a list of **Uniform Resource Identifiers (URIs)**, any of which can be Autodiscover servers.

URIs for Autodiscover server locations can be published by using the following methods:

- Service connection point objects, which can be queried by using the **Lightweight Directory Access Protocol (LDAP)**
- Direct DNS configuration
- DNS service (SRV) record configuration
- **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)** 302 redirection

1.4 Relationship to Other Protocols

This protocol requires an Autodiscover server and an Autodiscover client that implement the Autodiscover Publishing and Lookup Protocol, as specified in [MS-OXDSCLI]. This protocol relies on HTTPS, as specified in [RFC2818], for data protection services and it relies on [RFC1034] for DNS services. It also relies on [MS-ADTS] and [RFC1823] for the **service connection point** object and LDAP, respectively.

The following data flow diagram shows a client querying the directory and DNS for an Autodiscover server, and the server publishing its location in the directory and DNS.

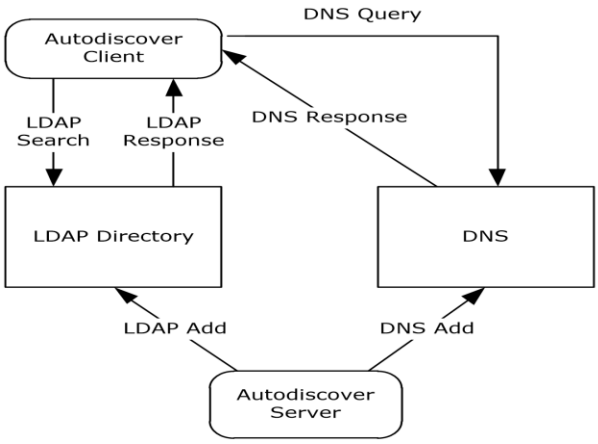

Figure 1: Autodiscover client and server interactions

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

1.5 Prerequisites/Preconditions

The Autodiscover client has to be configured with an LDAP directory and base **distinguished name (DN) (2)** that is well-known to the Autodiscover server administrator.

The Autodiscover server has to be configured to provide its services over **HTTP** secured with **Transport Layer Security (TLS)**, as specified in [RFC2818].

1.6 Applicability Statement

This protocol is applicable in scenarios in which an email client makes an Autodiscover request to discover email server settings and email servers respond with their locations and settings.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

For the purposes of this protocol, an Autodiscover client and an Autodiscover server do not communicate directly. Instead, the Autodiscover client communicates with common well-known data sources that the Autodiscover server administrator has preconfigured.<1>

The following transports and data sources are used:

- LDAP and LDAP directories. For more details, see [RFC1823].
- The DNS and DNS SRV records. For more details, see [RFC1034] and [RFC2782].
- Hypertext Transfer Protocol (HTTP) and HTTP 302 redirection. For more details, see [RFC2616].

2.2 Message Syntax

2.2.1 Service Connection Point Publication Service Objects

The service connection point allows clients to use the Autodiscover service. The service connection point connects the network to the server that performs the service of locating Autodiscover servers. Clients of the Autodiscover service use the service connection point publication service object to locate, connect to, and authenticate an instance of the service. The service connection point publication service object contains the service binding information that includes the **FQDN** of the server.

2.2.1.1 Service Connection Point Object Syntax

Using the formal syntax definition of the **LDAP Data Interchange Format (LDIF)**, as specified in [RFC2849], a service connection point object can be expressed as the following.

```
DN: <distinguishedName>
Objectcategory: serviceConnectionPoint
ChangeType: add
serviceBindingInformation: <serviceBindingInformationValue>
Keywords: <KeywordsValue>
[Keywords: <KeywordsValue>]
```

That is, a service connection point object MUST have a **distinguishedName** attribute, one or more **KeywordsValue** attributes, and one **serviceBindingInformationValue** attribute.

A **distinguishedName** attribute is a multipart name that uniquely identifies a node in a tree-structured directory database.

The KeywordsValue attribute is specified in section 3.1.5.1.

The serviceBindingInformationValue attribute is the URI that is needed to bind to a service.

2.2.1.2 Searching for Service Connection Point Objects

The following LDAP elements and operations are used to search for a service connection point object:

- The hostname parameter specified in [RFC1823] section 4.1, hereafter referred as host. host is a server running LDAP. This value SHOULD be well-known to the Autodiscover client and the Autodiscover server administrator.
- The portno parameter specified in [RFC1823] section 4.1, hereafter referred as port. port is the **port** of the LDAP service on the host. This value is commonly 389. This value SHOULD be well-known to the Autodiscover client and Autodiscover server administrator.
- The base parameter specified in [RFC1823] section 4.4, hereafter referred as base. base is the distinguished name (DN) (2) to base the search on. This value SHOULD be well-known to the Autodiscover server and the Autodiscover client.
- The scope parameter specified in [RFC1823] section 4.4, hereafter referred as scope. scope is the search scope. For Autodiscover clients, the value MUST be LDAP_SCOPE_SUBTREE. This is a constant specified in [RFC1823] section 4.4.
- The attrs parameter specified in [RFC1823] section 4.4. This value is the list of attributes to query. For the purposes of this protocol, the list MUST contain "serviceBindingInformation", and "Keywords".
- The filter parameter specified in [RFC1823] section 4.4, hereafter referred as filter. This parameter is an LDAP search filter, as specified in [RFC1960]. For the purposes of this protocol, filter is (&(objectcategory=serviceConnectionPoint)(|(keywords=67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68)(keywords=77378F46-2C66-4aa9-A6A6-3E7A48B19596))).

2.2.1.3 Creating Service Connection Point Objects

Service connection point objects can be created in an LDAP directory. To do so, the administrator needs the following data elements:

- host: This value SHOULD be well-known to the Autodiscover client and Autodiscover server administrator.
- port: This value is typically 389. This value SHOULD be well-known to the Autodiscover client and Autodiscover server administrator.
- The dn parameter specified in [RFC1823] section 4.9. This value is a DN of the service connection point object to create. This value SHOULD be well-known to the Autodiscover server administrator and the Autodiscover client.
- The attrs parameter specified in [RFC1823] section 4.9. This value is the list of attributes to write. For the purposes of this protocol, the list MUST contain "Objectcategory", "serviceBindingInformation", and "Keywords". The value of "Objectcategory" MUST be "serviceConnectionPoint". For details, see sections 3.1.5.1 and 3.2.3.1.

2.2.2 DNS SRV Queries

To query for Autodiscover servers, the Autodiscover client SHOULD use the following data elements specified by the usage rules in [RFC2782]:

- _service is "_Autodiscover"
- _protocol is "_tcp"
- The target is supplied by the Autodiscover client.

2.2.3 HTTP 302 Redirection

The following section uses **Augmented Backus-Naur Form (ABNF)** notation. For more details, see [RFC5234].

The Autodiscover client can send an HTTP GET request to retrieve the Autodiscover server URI. The request URI has the following format:

```
<RequestUri> = HTTP COLON SLASH SLASH AUTODISCOVERDOT <target> AUTODISCOVERSUFFIX

HTTP = "http"
COLON = ":"
SLASH = %2f ; forward slash or "/"
AUTODISCOVERDOT = "Autodiscover."
AUTODISCOVERSUFFIX = SLASH "Autodiscover" SLASH "Autodiscover.xml"
<target> = targetDomain ; The email domain that the Autodiscover client wishes to query.
```

The above strings are not case sensitive.

2.2.4 Email Addresses

All email addresses are assumed to be in the format specified in [RFC2822] section 3.4.1. That is, they follow the format <local-part>@<domain>.

2.2.5 Autodiscover Server URI Results

The result of an Autodiscover query is a list of possible Autodiscover server URIs, as specified in [RFC3986]. These URIs are the servers that are pinged until a match is found for the autodiscovery.

3 Protocol Details

3.1 Client 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 main data elements required by any implementation are:

- **Email address:** An email address of the form <local-part>@<domain>. This is the email address for which the corresponding Autodiscover server URI is being located.
- **LDAP directories and** service connection point **objects:** LDAP directories contain published server locations in service connection point objects. The service connection point object can be used to identify Autodiscover server URIs.
- **DNS & DNS SRV records:** DNS can contain SRV records for the Autodiscover service. The SRV records can then be used to find the Autodiscover server URI.

3.1.2 Timers

None.

3.1.3 Initialization

The client requires an email address of the form <local-part>@<domain>.

3.1.4 Higher-Layer Triggered Events

The Autodiscover publishing and lookup services are triggered by a user action, or optionally a timer.

3.1.5 Message Processing Events and Sequencing Rules

The Autodiscover client expands the **email address** provided during initialization into a list of URIs, as specified in [RFC2396]. Because Autodiscover server URIs can be acquired in different ways, to create a fully populated list, the Autodiscover client SHOULD<2> do all of the following:

- Query a well-known LDAP server for service connection point objects, as specified in section 3.1.5.1.
- Perform text manipulations on the domain of the email address, as specified in section 3.1.5.2.
- Search the DNS for Autodiscover SRV records, as specified in section 3.1.5.3.
- Perform an HTTP **GET** request to determine whether redirects to other Autodiscover servers exist, as specified in section 3.1.5.4.

Note that a client can acquire the URI of an Autodiscover server without a fully populated list of Autodiscover server URIs.

3.1.5.1 Query a Well-Known LDAP Server for Service Connection Point Objects

Autodiscover server locations can be published in LDAP directories via service connection point objects.

A service connection point object can be created by using the LDAP API specified in [RFC1823] section 4.9.

To discover these servers, Autodiscover clients execute a client search as specified in section 2.2.1.2 and as specified in [RFC1823] section 4.4.

For each of the entries returned, if the **serviceBindingInformation** attribute is an LDAP URI (a text string of the form "LDAP://"<host>[:<port>]) and the **Keywords** attribute contains a string of the form "domain="<domain>, then the client repeats the search as specified in section 2.2.1.2 with the new **host** element and **port** element values.

If the **serviceBindingInformation** attribute is an LDAP URI (a text string of the form "LDAP://"<host>[:<port>]), but the **Keywords** attribute does not contain a string of the form "domain="<domain>, then the client repeats the search as specified in section 2.2.1.2 with the new **host** element and **port** element values after all other entries have been evaluated.

If the **serviceBindingInformation** attribute is an "http://" or "https://" URI then the client has found a URI that is possibly an Autodiscover server and the client SHOULD add this to the list of possible Autodiscover servers.

If the Autodiscover directory service map GUID is found in the **Keywords**, then the **serviceBindingInformation** is an LDAP URI.

If the Autodiscover URI map GUID is found in the **Keywords**, then the **serviceBindingInformation** is an HTTP URI.

3.1.5.2 Locations Found Directly From the Email Domain

The following two URIs MUST be added to the list of possible Autodiscover server URIs:

```
http://<Domain>/Autodiscover/Autodiscover.xml
https://Autodiscover.<Domain>/Autodiscover/Autodiscover.xml
```

If an **HTTP POST** to either of the above URIs results in an HTTP 302 redirect, then the redirect as found in the location field of the response is added to the list of possible Autodiscover server URIs. For more details, see section 2.2.3. For more details about Autodiscover client requests, see [MS-OXDSCLI] section 3.1.5.2.

3.1.5.3 Locations Found from SRV DNS Records

An Autodiscover client can query DNS to obtain SRV records for the Autodiscover service by using the following query. For more information, see section 2.2.2. The query produces an ordered list of hosts. If no valid entries are found, then the query will return an empty list.

```
_autodiscover._tcp.<domain>
```

If the result is <host>, add "https://"<host>"/Autodiscover/Autodiscover.xml" to the list of possible Autodiscover URIs.

3.1.5.4 Locations Found by an HTTP Redirect

An Autodiscover client can also issue an HTTP **GET** method with the URI set to "http://Autodiscover.<domain>/Autodiscover/Autodiscover.xml".

The **RequestUri** element can be processed as specified in [RFC2616] section 9.3. If the response is a 302 redirection (as specified in [RFC2616] section 10.3.3), the Autodiscover client uses the value of the redirection **URL**. Note that if the response is not a 302 redirection, then the expected response is an Autodiscover server URI.

If this URI results in an HTTP 302 redirect, prompt the user to warn them of the redirection. If the user accepts, the new location is added to the list of possible Autodiscover server URIs.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

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.

Any implementation requires a published Autodiscover server URI. This enables clients to find servers by using the Autodiscover service.

3.2.2 Timers

None.

3.2.3 Initialization

Autodiscover servers do not automatically publish all their locations. Administrators can manually publish Autodiscover server locations.

3.2.3.1 Locations Published in LDAP via Service Connection Point Objects with an HTTP URI

An administrator can publish a service connection point object by using the following values:

```
DN:<AdministratorChosenDN>
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: <AutodiscoverServerURI>
Keywords: "77378F46-2C66-4aa9-A6A6-3E7A48B19596"
<Extensions>

<Extensions> := NULL |
```

```
            Extension | Extensions

Extension := "Domain=" <AuthoritativeDomain> |
       "Site=" <ADSite>
```

The **AuthoritativeDomain** element is a domain that the Autodiscover server can provide information about.

The **ADSite** element is the Active Directory site, as specified in [MS-ADTS].

3.2.3.2 Locations Published in LDAP via Service Connection Point objects with an LDAP URI

If the administrator of the Autodiscover server knows that Autodiscover clients prefer a different LDAP server than the Autodiscover server, then the administrator can manually publish a service connection point object in the client's preferred LDAP server. A client querying for service connection point objects can then learn about the Autodiscover server's preferred LDAP server.

```
DN: <AdministratorChosenDN>
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: <LDAPURI>
Keywords: "67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68"
<Extensions>

<Extensions> := NULL |
          Extension | Extensions

Extension := "Domain=" <AuthoritativeDomain>
```

The **AuthoritativeDomain** element is a domain that the Autodiscover server can provide information about.

3.2.3.3 Locations Published in DNS as Autodiscover.<Domain> and <Domain>

An administrator who wants to publish an Autodiscover server for <Domain> can configure DNS and **SSL** such that "https://Autodiscover.<Domain>/Autodiscover/Autodiscover.xml" and "https://<Domain>/Autodiscover/Autodiscover.xml" are URIs that are serviced by Autodiscover servers.

This is configured manually.

3.2.3.4 Locations Published in DNS By Using SRV Records

If "https://Server/Autodiscover/Autodiscover.xml" can serve Autodiscover clients for the given **Domain** element, an administrator can publish the following SRV record:

```
SRV _autodiscover._tcp.<DOMAIN> = <AutodiscoverServer>
```

This is configured manually. For details, see section 2.2.2.

3.2.3.5 Locations Published Through an HTTP GET

If "https://Server/Autodiscover/Autodiscover.xml" can serve Autodiscover clients for <Domain>, an administrator can configure the following HTTP redirect:

"http://Autodiscover.<Domain>/Autodiscover/Autodiscover.xml"

to 302 redirect to

"https://Server/Autodiscover/Autodiscover.xml".

Non-secure HTTP URIs SHOULD NOT be used to query settings, as specified in [MS-OXDSCLI]. They SHOULD only be used for redirections.

This is configured manually. For more details, see [RFC2616].

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Publishing an Autodiscover Server Location

The following topology is used in this example:

- The DNS name of the mail server is Mail.Contoso.com.
- The DNS name of the Web service computer is WebService.Contoso.com. It has a valid SSL certificate.
- Autodiscover Web services are available at: https://WebService.Contoso.com/Autodiscover/Autodiscover.xml.
- The mailbox server and Web services server are configured to use MailLdap.Contoso.com as their LDAP server.
- Clients are configured to use ClientLdap.Contoso.com.

The following figure illustrates this topology.

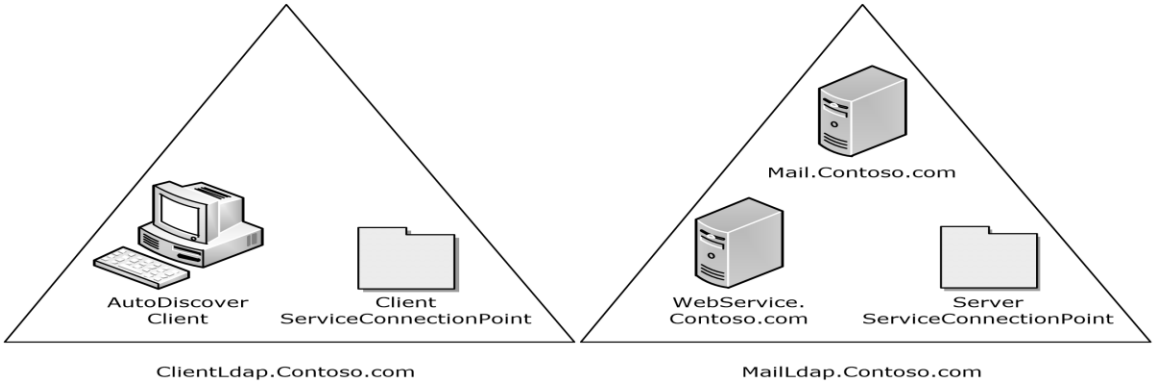

Figure 2: Topology of the Autodiscover client and server

An administrator wants to publish Autodiscover services for mailboxes on Mail.Contoso.com. For various reasons, the administrator is unable to configure "https://Contoso.com/Autodiscover.Autodiscover.xml" to respond to Autodiscover requests. Instead, the administrator uses [RFC1034] and [RFC4210] to create SSL certificates that allow the Autodiscover server to HTTP 302 redirect "https://Autodiscover.Contoso.com/Autodiscover/Autodiscover.xml" to "https://WebService.Contoso.com/Autodiscover/Autodiscover.xml".

Also, the administrator creates and publishes two LDAP objects to help clients find the Autodiscover server.

For MailLdap.Contoso.com, the administrator publishes the following:

```
DN: CN=WebServices,CN=Autodiscover,CN=Protocols,CN=WebServices,CN=Servers,CN=Exchange
Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=Contoso,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=Contoso,DC=com
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: https://WebService.Contoso.com/Autodiscover/Autodiscover.xml
Keywords: "77378F46-2C66-4aa9-A6A6-3E7A48B19596"
```

On the client LDAP server, the administrator publishes the following:

```
DN: CN=mail.contoso.com,CN=Microsoft Exchange
Autodiscover,CN=Services,CN=Configuration,DC=Users,DC=Contoso,DC=com
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: LDAP://MailLdap.Contoso.com
Keywords: "67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68"
```

4.2 Autodiscover Client Querying for Autodiscover Servers

This example uses the following configuration:

- A mail client is configured to use the email address User@Contoso.com.
- The mail client is configured to use ClientLdap.Contoso.com as its LDAP server.
- Servers are configured as specified in section 3.2 of this document.

The client wants to construct a list of URIs of possible Autodiscover server locations. First the client executes the steps specified in section 3.1.5.1. The client searches its LDAP server on ClientLdap.Contoso.com for a service connection point object that has the following **GUIDs**: 67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68 or 77378F46-2C66-4aa9-A6A6-3E7A48B19596.

The client performs the search by constructing the following URI:

```
LDAP//ClientLdap.Contoso.com
"/?cn,serviceBindingInformation,Keywords?sub?(&(objectcategory=serviceConnectionPoint)(|(keyw
ords=67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68)( keywords=77378F46-2C66-4aa9-A6A6-3E7A48B19596)))"
```

After evaluating that query, the following service connection point object is returned to the client:

```
DN: CN=mail.contoso.com,CN=Microsoft Exchange
Autodiscover,CN=Services,CN=Configuration,DC=Users,DC=Contoso,DC=com
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: LDAP://MailLdap.Contoso.com
Keywords: "67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68"
```

Seeing that the **service binding information** is provided in an LDAP URI, the Autodiscover client then proceeds to construct the following:

```
LDAP//MailLdap.Contoso.Com
"/?cn,serviceBindingInformation,Keywords?sub?(&(objectcategory=serviceConnectionPoint)(|(keyw
ords=67661D7F-8FC4-4fa7-BFAC-E1D7794C1F68)( keywords=77378F46-2C66-4aa9-A6A6-3E7A48B19596)))"
```

This query returns the following object:

```
DN: CN=WebServices,CN=Autodiscover,CN=Protocols,CN=WebServices,CN=Servers,CN=Exchange
Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=Contoso,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=Contoso,DC=com
Changetype: add
Objectcategory: serviceConnectionPoint
serviceBindingInformation: https://WebService.Contoso.com/Autodiscover/Autodiscover.xml
Keywords: "77378F46-2C66-4aa9-A6A6-3E7A48B19596"
```

From this, the client adds "https://WebService.Contoso.com/Autodiscover/Autodiscover.xml" to the list of possible Autodiscover Web services.

The communication is shown in the following figure.

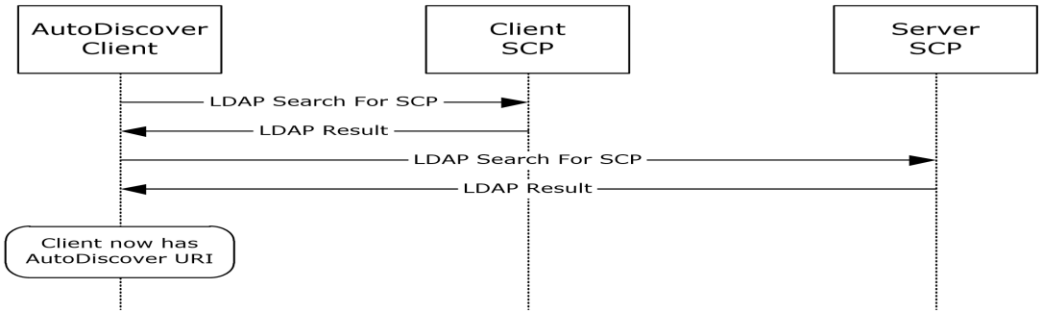

Figure 3: Communication required to find an Autodiscover server URI by using service connection point objects

Next, the client adds "https://Autodiscover.Contoso.com/Autodiscover/Autodiscover.xml" and "https://Contoso.com/Autodiscover/Autodiscover.xml" to the list of possible email addresses based on the information specified in section 3.1.5.2.

As specified in section 2.2.1.2, the client executes a DNS search for the SRV record "_autodiscover._tcp.Contoso.com". No records are returned. This is expected behavior, as no records were created.

5 Security

5.1 Security Considerations for Implementers

There are many possible DNS spoofing attacks. For this reason, clients are strongly advised against using non-SSL URIs unless they have the consent of the user. Administrators are strongly advised to provide Autodiscover data only via HTTPS.

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.

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Windows 8.1
- Windows Communication Apps
- Windows 10 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 2.1: Microsoft Windows automatically pre-configures domain-joined computers with an **Active Directory** server. Office Outlook 2007 uses this LDAP server as the well-known LDAP server. Office Outlook 2007 uses the value of the **configurationNamingContext** attribute, as described in in [MS-ADTS], of the preconfigured Active Directory server as the well-known DN for service connection point objects.

<2> Section 3.1.5: Windows Communication Apps do not query for service connection point objects and do not search DNS for Autodiscover SRV records.

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
  client 14
  server 16
Applicability 10
Autodiscover client querying for Autodiscover servers example 20
Autodiscover Server URI Results message 13

C

Capability negotiation 10
Change tracking 24
Client
  abstract data model 14
  higher-layer triggered events 14
  initialization 14
  message processing 14
  other local events 16
  sequencing rules 14
  timer events 16
  timers 14

D

Data model - abstract
  client 14
  server 16
DNS SRV Queries message 12

E

Email Addresses message 13
Examples
  Autodiscover client querying for Autodiscover servers 20
  publishing an Autodiscover server location 19

F

Fields - vendor-extensible 10

G

Glossary 6

H

Higher-layer triggered events
  client 14
  server 18
HTTP 302 Redirection message 12

I

Implementer - security considerations 22
Index of security parameters 22
Informative references 9
Initialization
  client 14
  server 16
Introduction 6

M

Message processing
  client 14
  server 18
Messages
  Autodiscover Server URI Results 13
  DNS SRV Queries 12
  Email Addresses 13
  HTTP 302 Redirection 12
  Service Connection Point Publication Service Objects 11
  transport 11

N

Normative references 8

O

Other local events
  client 16
  server 18
Overview (synopsis) 9

P

Parameters - security index 22
Preconditions 10
Prerequisites 10
Product behavior 23
Publishing an Autodiscover server location example 19

R

References 8
  informative 9
  normative 8
Relationship to other protocols 9

S

Security
  implementer considerations 22
  parameter index 22
Sequencing rules
  client 14
  server 18
Server
  abstract data model 16
  higher-layer triggered events 18
  initialization 16
  message processing 18
  other local events 18
  sequencing rules 18
  timer events 18
  timers 16
Service Connection Point Publication Service Objects message 11

Standards assignments 10

T

Timer events
  client 16
  server 18
Timers
  client 14
  server 16
Tracking changes 24
Transport 11
Triggered events - higher-layer
  client 14
  server 18

V

Vendor-extensible fields 10
Versioning 10