

[MS-OXDISCO]: Autodiscover HTTP Service Protocol Specification

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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). 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
04/04/2008	0.1		Initial Availability.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Updated references to reflect date of initial release.
09/03/2008	1.02		Revised and edited technical content.
10/01/2008	1.03		Revised and edited technical content.
12/03/2008	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	4.1.1	Editorial	Revised and edited the technical content.
08/04/2010	4.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2010	4.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Overview	6
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Message Syntax	9
2.2.1 SCP Publication Service Objects	9
2.2.1.1 Service Connection Point Object Syntax	9
2.2.1.2 Searching for Service Connection Point Objects	9
2.2.1.3 Creating Service Connection Point Objects	10
2.2.2 DNS SRV Queries	10
2.2.3 HTTP 302 Redirection	11
2.2.4 E-mail Addresses	11
2.2.5 Autodiscover Server URI Results	11
3 Protocol Details	12
3.1 Client Details	12
3.1.1 Abstract Data Model	12
3.1.2 Timers	12
3.1.3 Initialization	12
3.1.4 Higher-Layer Triggered Events	12
3.1.5 Message Processing Events and Sequencing Rules	12
3.1.5.1 Query a Well-Known LDAP Server for Service Connection Point Objects	13
3.1.5.2 Locations Found Directly From the E-mail Domain	13
3.1.5.3 Locations Found from SRV DNS Records	13
3.1.5.4 Locations Found by an HTTP Redirect	14
3.1.6 Timer Events	14
3.1.7 Other Local Events	14
3.2 Server Details	14
3.2.1 Abstract Data Model	14
3.2.2 Timers	14
3.2.3 Initialization	14
3.2.3.1 Locations Published in LDAP via Service Connection Point Objects with an HTTP URI	14
3.2.3.2 Locations Published in LDAP via Service Connection Point objects with an LDAP URI	15
3.2.3.3 Locations Published in DNS as Autodiscover.<Domain> and <Domain>	15
3.2.3.4 Locations Published in DNS By Using SRV Records	15
3.2.3.5 Locations Published Through an HTTP GET	16
3.2.4 Higher-Layer Triggered Events	16

3.2.5	Message Processing Events and Sequencing Rules	16
3.2.6	Timer Events	16
3.2.7	Other Local Events	16
4	Protocol Examples	17
4.1	Publishing an Autodiscover Server Location	17
4.2	Autodiscover Client Querying for Autodiscover Servers	18
5	Security	20
5.1	Security Considerations for Implementers	20
5.2	Index of Security Parameters	20
6	Appendix A: Product Behavior	21
7	Change Tracking	22
8	Index	24

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-OXDCLI](#)].

1.1 Glossary

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

Active Directory
Augmented Backus-Naur Form (ABNF)
distinguished name (DN)
domain
Domain Name System (DNS)
fully qualified domain name (FQDN)
GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Lightweight Directory Access Protocol (LDAP)
Secure Sockets Layer (SSL)

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

Autodiscover client
Autodiscover server
service connection point
Transport Layer Security (TLS)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)

The following terms are specific to this document:

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].

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

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

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

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-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](#)", June 2007.

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol Specification](#)", June 2008.

[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.ietf.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.ietf.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.ietf.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.ietf.org/rfc/rfc2818.txt>

[RFC2822] Resnick, P., Ed., "Internet Message Format", STD 11, 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.ietf.org/rfc/rfc5234.txt>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)", April 2008.

1.3 Overview

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

Uniform Resource Identifiers (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 [\[MS-OXDCLI\]](#). 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.

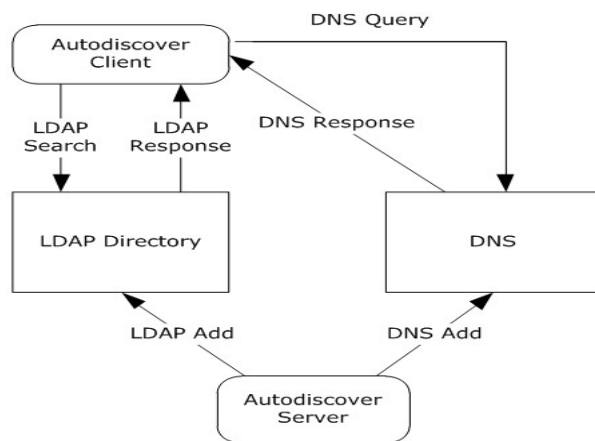


Figure 1: Autodiscover client and server interactions

1.5 Prerequisites/Preconditions

The Autodiscover client has to be configured with an LDAP directory and base **distinguished name (DN)** 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 e-mail client makes an Autodiscover request to discover e-mail server settings and e-mail 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 SCP 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 **SCP Publication Service** object to locate, connect to, and authenticate an instance of the service. The **SCP 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) 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 e-mail domain that the Autodiscover client wishes to query.
```

The above strings are not case sensitive.

2.2.4 E-mail Addresses

All e-mail 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

This protocol specifies a way for Autodiscover clients to find Autodiscover servers. The client starts with an e-mail address of the form <local-part>@<domain> and expands it to a list of URIs any of which can be Autodiscover servers.

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:

- **E-mail address:** An e-mail address of the form <local-part>@<domain>. This is the e-mail 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 e-mail 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 e-mail address provided during initialization into a list of URIs as specified in [\[RFC2396\]](#). Since Autodiscover server URIs can be acquired in different ways, to create a fully-populated list, the Autodiscover client SHOULD do all of the following:

- Query a well-known LDAP server for service connection point objects, as described in section [3.1.5.1](#).
- Perform text manipulations on the domain of the email address, as described in section [3.1.5.2](#).
- Search the DNS for Autodiscover SRV records, as described in section [3.1.5.3](#).

- Perform an HTTP GET request to determine whether there are redirects to other Autodiscover servers, as described 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 E-mail 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 3.1.8.2.

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.

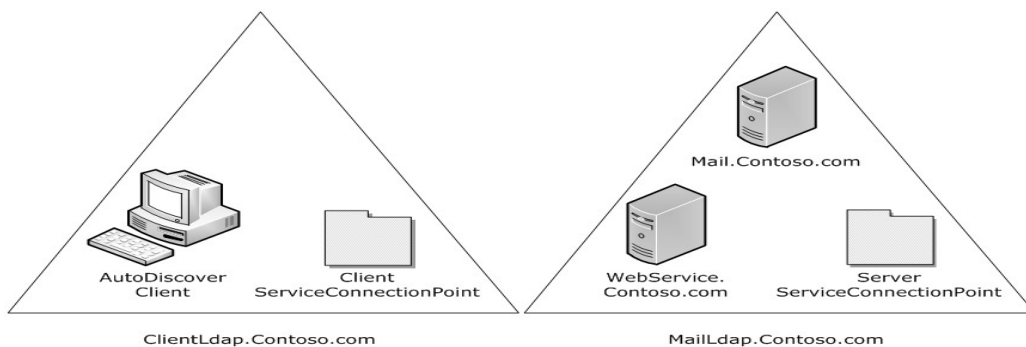


Figure 2: Topology of the Autodiscover client and server

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 e-mail 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.

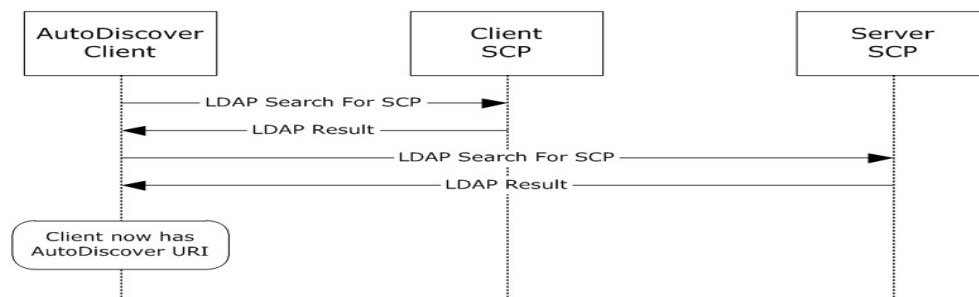


Figure 3: Communication required to find an Autodiscover server URI by using service connection point objects

Next, the client adds "https://Autodiscover.Contoso.com/Autodiscover/Autodiscover.xml" and

"https://Contoso.com/Autodiscover/Autodiscover.xml" to the list of possible e-mail 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® Office Outlook® 2007
- Microsoft® Outlook® 2010

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:](#) 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 **ConfigurationNamingContext** of the pre-configured Active Directory server as the well-known DN for service connection point objects.

7 Change Tracking

This section identifies changes that were made to the [MS-OXDISCO] protocol document between the November 2010 and March 2011 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 content for compliance with revised Block template.	Y	Content updated for template compliance.
1.4 Relationship to Other Protocols	Updated the content for compliance with revised Block template.	Y	Content updated for template compliance.
3.1.1 Abstract Data Model	Bug # 56792 - made URI into glossary term.	N	Content updated.

8 Index

A

Abstract data model
 [client](#) 12
 [server](#) 14
[Applicability](#) 8
[Autodiscover Server URI Results message](#) 11

C

[Capability negotiation](#) 8
[Change tracking](#) 22
Client
 [abstract data model](#) 12
 [higher-layer triggered events](#) 12
 [initialization](#) 12
 [message processing](#) 12
 [other local events](#) 14
 [sequencing rules](#) 12
 [timer events](#) 14
 [timers](#) 12

D

Data model - abstract
 [client](#) 12
 [server](#) 14
[DNS SRV Queries message](#) 10

E

[E-mail Addresses message](#) 11

F

[Fields - vendor-extensible](#) 8

G

[Glossary](#) 5

H

Higher-layer triggered events
 [client](#) 12
 [server](#) 16
[HTTP 302 Redirection message](#) 11

I

[Implementer - security considerations](#) 20
[Index of security parameters](#) 20
[Informative references](#) 6
Initialization
 [client](#) 12
 [server](#) 14
[Introduction](#) 5

M

Message processing
 [client](#) 12
 [server](#) 16
Messages
 [Autodiscover Server URI Results](#) 11
 [DNS SRV Queries](#) 10
 [E-mail Addresses](#) 11
 [HTTP 302 Redirection](#) 11
 [SCP Publication Service Objects](#) 9
 [transport](#) 9

N

[Normative references](#) 5

O

Other local events
 [client](#) 14
 [server](#) 16
[Overview](#) 6

P

[Parameters - security index](#) 20
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 21

R

References
 [informative](#) 6
 [normative](#) 5
[Relationship to other protocols](#) 7

S

[SCP Publication Service Objects message](#) 9
Security
 [implementer considerations](#) 20
 [parameter index](#) 20
Sequencing rules
 [client](#) 12
 [server](#) 16
Server
 [abstract data model](#) 14
 [higher-layer triggered events](#) 16
 [initialization](#) 14
 [message processing](#) 16
 [other local events](#) 16
 [sequencing rules](#) 16
 [timer events](#) 16
 [timers](#) 14
[Standards assignments](#) 8

T

Timer events

[client](#) 14
[server](#) 16

Timers

[client](#) 12
[server](#) 14

[Tracking changes](#) 22

[Transport](#) 9

Triggered events - higher-layer

[client](#) 12
[server](#) 16

V

[Vendor-extensible fields](#) 8

[Versioning](#) 8