

[MS-WSTC]:

WS-Discovery: Termination Criteria 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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **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
5/22/2009	0.1	Major	First Release.
7/2/2009	0.1.1	Editorial	Changed language and formatting in the technical content.
8/14/2009	0.1.2	Editorial	Changed language and formatting in the technical content.
9/25/2009	0.2	Minor	Clarified the meaning of the technical content.
11/6/2009	1.0	Major	Updated and revised the technical content.
12/18/2009	1.0.1	Editorial	Changed language and formatting in the technical content.
1/29/2010	1.0.2	Editorial	Changed language and formatting in the technical content.
3/12/2010	2.0	Major	Updated and revised the technical content.
4/23/2010	2.0.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	2.0.2	Editorial	Changed language and formatting in the technical content.
7/16/2010	3.0	Major	Updated and revised the technical content.
8/27/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	3.1	Minor	Clarified the meaning of the technical content.
9/23/2011	3.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	4.0	Major	Updated and revised the technical content.
3/30/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	4.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
8/8/2013	4.0	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	4.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	5.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport	10
2.2	Common Message Syntax	10
2.2.1	Namespaces	10
2.2.2	Messages.....	10
2.2.3	Elements	10
2.2.3.1	<MaxResults>	10
2.2.3.2	<Duration>	11
2.2.4	Complex Types.....	11
2.2.5	Simple Types	11
2.2.6	Attributes	11
2.2.7	Groups	11
2.2.8	Attribute Groups.....	11
3	Protocol Details	12
3.1	Server Details.....	12
3.1.1	Abstract Data Model.....	12
3.1.2	Timers	12
3.1.3	Initialization.....	12
3.1.3.1	Initialization Due to Receipt of a Probe Message	12
3.1.3.2	Initialization Due to Receipt of a Resolve Message.....	12
3.1.4	Message Processing Events and Sequencing Rules	13
3.1.5	Timer Events.....	13
3.1.6	Other Local Events.....	13
3.2	Client Details	13
3.2.1	Abstract Data Model.....	13
3.2.2	Timers	13
3.2.3	Initialization	13
3.2.4	Message Processing Events and Sequencing Rules	14
3.2.4.1	Probe Operation.....	14
3.2.4.2	Resolve Operation.....	14
3.2.5	Timer Events.....	15
3.2.6	Other Local Events.....	15
4	Protocol Examples	16
4.1	Probe Message Example.....	16
4.2	Resolve Message Example.....	16
5	Security	18
5.1	Security Considerations for Implementers	18
5.2	Index of Security Parameters	18
6	Appendix A: Full WSDL	19

7	Appendix B: Product Behavior	20
8	Appendix C: Protocol Schema	21
9	Change Tracking.....	22
10	Index.....	24

1 Introduction

The WS-Discovery: Termination Criteria Protocol Extensions specify an extension to the **WS-Discovery** protocol for sending and receiving a **termination criterion** as part of WS-Discovery **Probe** and **Resolve** messages. The termination criterion allows a **Target Service host** to be aware of constraints that the **client** is operating under when it is looking for a **Target Service**. A Target Service host can use this information to determine whether a response is sent to the client, thus saving network bandwidth by not transmitting a response that the client will ignore.

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 [\[RFC2119\]](#). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

.NET Framework: An integral Windows component that supports building and running applications and XML web services. The Microsoft .NET Framework has two main components: the common language runtime and the .NET Framework class library. For more information about the .NET Framework, see [\[MSDN-.NET-FRAMEWORK\]](#). The following versions of the .NET Framework are available in the following released Windows products or as supplemental software. Microsoft .NET Framework 1.0: Windows NT 4.0 operating system, Microsoft Windows 98 operating system, Windows 2000 operating system, Windows Millennium Edition operating system, Windows XP operating system, and Windows Server 2003 operating system. Microsoft .NET Framework 1.1: Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Server 2003 R2 operating system, Windows Vista operating system, and Windows Server 2008 operating system. Microsoft .NET Framework 2.0: Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7 operating system, Windows Server 2008 R2 operating system, Windows 8 operating system, Windows Server 2012 operating system, Windows 8.1 operating system, Windows Server 2012 R2 operating system, Windows 10 operating system, and Windows Server 2016 Technical Preview operating system. Microsoft .NET Framework 3.0: Windows XP, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. Microsoft .NET Framework 3.5: Windows XP, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. Microsoft .NET Framework 4.0: Windows XP, Windows Server 2003, Windows Server 2003 R2, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. Microsoft .NET Framework 4.5: Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, and Windows 10. Microsoft .NET Framework 4.6: Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, and Windows 10.

client: The term "Client" that is defined in [\[WS-Discovery1.1\]](#).

find criteria: The criterion that is sent in a **Probe** message consisting of **Type** and Scopes, and that is used by **clients** to locate **Target Services**.

probe: The Web Services Dynamic Discovery (WS-Discovery) protocol message sent by a **client** to discover content, as defined in [\[WS-Discovery1.1\]](#).

Probe Match: A **WS-Discovery** message, as defined in [WS-Discovery1.1].

Resolve: A **WS-Discovery** message, as defined in [WS-Discovery1.1].

Resolve Match: A **WS-Discovery** message, as defined in [WS-Discovery1.1].

Target Service: The term "Target Service" that is defined in [WS-Discovery1.1].

Target Service host: A process or agent that has knowledge of multiple **Target Services**. A **Target Service host** is responsible for the discovery of these **Target Services**; it implements the **WS-Discovery** protocol and responds to discovery messages on behalf of these **Target Services**.

termination criterion: A set of constraints placed on a search operation by the **client**. These constraints can include the duration within which the search operation should complete, and the maximum number of responses the **client** is looking for.

Type: The term "Type" that is defined in [WS-Discovery1.1].

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. 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 UUID.

WS-Discovery: This term refers to the specific version of the **WS-Discovery** protocol that the implementer is taking a dependency on. This term can refer to any of the supported protocol versions that are specified in section 1.7.

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.

[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>

[WS-Discovery1.1] Modi, V., and Kemp, D., "Web Services Dynamic Discovery (WS-Discovery) Version 1.1", OASIS Status: Public Review, January 2009, <http://docs.oasis-open.org/ws-dd/discovery/1.1/pr-01/wsdd-discovery-1.1-spec-pr-01.pdf>

[WS-Discovery] Beatty, J., Kakivaya, G., Kemp D., et al., "Web Services Dynamic Discovery (WS-Discovery)", April 2005, <http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS-2ED] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

[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., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "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

None.

1.3 Overview

This document specifies an extension to the WS-Discovery protocol for sending and receiving termination criteria as part of the WS-Discovery Probe and Resolve messages. The termination criterion defines the constraints placed on a Probe operation executed by a client and sent as part of the WS-Discovery message to a Target Service host. If the Target Service host implements this extension, then it can act on the termination criterion specified. The criterion allows the Target Service host to determine whether sending a response back to the client is necessary.

The extension is limited to two XML message elements, conforming to the XML schema, that are encapsulated in WS-Discovery Probe and Resolve messages. The motivation behind this extension is to allow clients to limit the number of WS-Discovery responses received, based on duration or number of services. A scenario that illustrates the use of this extension follows.

A client is looking for a Target Service that uses the WS-Discovery protocol. When sending out the Probe message, the client is interested only in information regarding one Target Service, and needs this information within a certain period of time. The client inserts the appropriate termination criteria information into the Probe message, as specified in this document. The WS-Discovery protocol specifies the semantics of the way in which a client sends a Probe message.

When a Target Service host receives the Probe message, it determines which Target Services match the Probe as specified in the WS-Discovery protocol. Since this Target Service host also implements this extension, it applies the termination criterion. If the Target Service host is able to meet the time criterion, then it responds back to the client with a **Probe Match** message containing one Target Service that matches the original Probe message. The semantics of the Probe Match message transmission are defined in the WS-Discovery protocol.

1.4 Relationship to Other Protocols

This protocol is an extension of the WS-Discovery protocol. Its dependency is described in the following layering diagram. The extension defines XML elements encapsulated in the "extensions" section of the WS-Discovery Probe and Resolve messages. The protocol relationships pertaining to the WS-Discovery protocol, such as the protocols layered above it and the transport protocols below it, are not covered in this document.

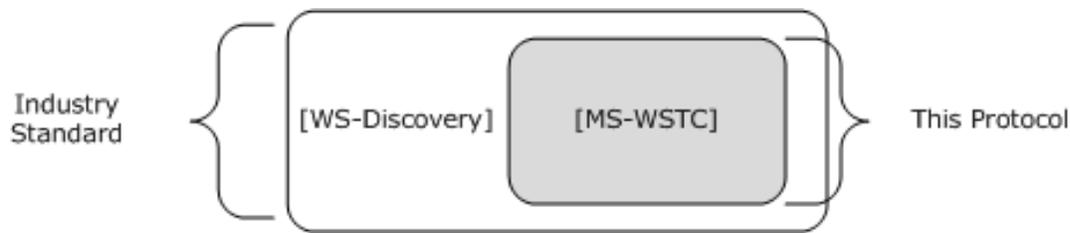


Figure 1: Layering diagram for the WS-Discovery: Termination Criteria Protocol Extensions

1.5 Prerequisites/Preconditions

In order to implement this protocol extension, an implementation of WS-Discovery, as specified in [\[WS-Discovery1.1\]](#) or [\[WS-Discovery\]](#), is required.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This extension does not specify any supported transports. The transport support is defined and constrained by the WS-Discovery protocol.
- **Protocol Versions:** This protocol extension does not have a specific WSDL declaration [\[WSDL\]](#), nor does it modify the WSDL of the encapsulating protocol. This extension is applicable to the following versions of the WS-Discovery protocols:
 - [\[WS-Discovery1.1\]](#)
 - [\[WS-Discovery\]](#)
- **Security and Authentication Methods:** This protocol does not specify any particular security and authentication methods. All applicable security and authentication methods to the WS-Discovery protocol are defined in the WS-Discovery specification.
- **Localization:** This protocol extension does not have any localization considerations.
- **Capability Negotiation:** This protocol does not require any specific configurations or interface versions.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol extension can be used over any transport protocol allowed by the WS-Discovery specification.

2.2 Common Message Syntax

This protocol extension defines only elements that are inserted into existing WS-Discovery Probe and Resolve messages. The syntax of the message elements defined in this protocol extension uses XML schema as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#).

2.2.1 Namespaces

This specification defines and references two XML namespaces using the mechanisms specified in [\[XMLNS-2ED\]](#). Although this specification associates a specific XML namespace prefix for one XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
	http://schemas.microsoft.com/ws/2008/06/discovery	Section 8
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA2]

This document does not use a prefix for the <http://schemas.microsoft.com/ws/2008/06/discovery> namespace that is defined above.

2.2.2 Messages

None.

2.2.3 Elements

The following table summarizes the set of common XML schema element definitions defined by this specification in the <http://schemas.microsoft.com/ws/2008/06/discovery> namespace.

Element	Description
<MaxResults>	This element contains a positive integer value, specifying the maximum number of Target Services a client is looking for.
<Duration>	This element contains a value of duration type, specifying the maximum amount of time the client is willing to wait to get a response for a particular request.

2.2.3.1 <MaxResults>

```
<MaxResults xmlns="http://schemas.microsoft.com/ws/2008/06/discovery">  
  xs:positiveInteger  
</MaxResults>
```

The value of the MaxResults element MUST be greater than 0 and less than or equal to 2,147,483,647. A value of MaxResults equal to 2,147,483,647 is treated as a special case, implying an infinite number of responses. Client and server implementations handle out of bounds values differently. These cases are described in the respective protocol details sections.

2.2.3.2 <Duration>

```
<Duration xmlns="http://schemas.microsoft.com/ws/2008/06/discovery">  
  xs:duration  
</Duration>
```

The value of the Duration element MUST be greater than P0Y0M0DT0H0M0S and MUST be less than or equal to P0Y0M0DT0H0M2147483.647S. In addition to this, a special-case value of P10675199DT2H48M05.4775807S implies an infinite duration. Client and server implementations handle out-of-bounds values differently. These cases are described in the respective protocol details sections.

2.2.4 Complex Types

This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

A server implementation must maintain the following data element:

MaxResponses: This value indicates the maximum number of Target Services about which a Target Service host should respond to a client in a Probe Match message. The constraint placed by the **MaxResponses** value is not applicable to **Resolve Match** messages.

3.1.2 Timers

DurationTimer: This timer regulates the maximum time allocated for the Target Service host to process the Probe or Resolve message and respond to a client with a Probe Match or Resolve Match message.

There is no default value for the **DurationTimer**. The value is set as defined in the initialization (section [3.1.3](#)).

3.1.3 Initialization

If any of the data elements are not conformant to the constraints defined in section [2.2.3](#), then the Target Service MUST drop the incoming message and take no further action.

3.1.3.1 Initialization Due to Receipt of a Probe Message

Upon receipt of a Probe message, the following initialization MUST occur:

If a <Duration> element described in section [2.2.3.2](#) is specified and is equal to P10675199DT2H48M05.4775807S, then the **DurationTimer** MUST NOT be started; this value implies an infinite timeout. Otherwise, the **DurationTimer** MUST be started with a value equal to the value of the <Duration> element.

If a <MaxResults> element described in section [2.2.3.1](#) is specified, then **MaxResponses** MUST be set to a value equal to the value of the <MaxResults> element.

If the <Duration> element is set to P10675199DT2H48M05.4775807S and if the <MaxResults> element is set to 2,147,483,647, then the implementation MUST drop the Probe message; the implementation MUST NOT respond to the client.

3.1.3.2 Initialization Due to Receipt of a Resolve Message

Upon receipt of a Resolve message, the following initialization MUST occur:

If a <MaxResults> element described in section [2.2.3.1](#) is specified, then the Target Service MUST ignore this value.

If a <Duration> element described in section [2.2.3.2](#) is specified and is equal to P10675199DT2H48M05.4775807S, then the **DurationTimer** MUST NOT be started; this value implies an infinite timeout.

Otherwise, the **DurationTimer** MUST be started with a value equal to the value of the <Duration> element.

3.1.4 Message Processing Events and Sequencing Rules

If the protocol is initialized due to the receipt of a Probe message and the **MaxResponses** value is set, then the following rules MUST be followed. If any Target Services match the **find criteria**, then the Target Service host MUST respond to the client with information about a number of matching Target Services that MUST be less than or equal to the value of **MaxResponses**. If the number of matching Target Services is greater than the value of **MaxResponses**, the Target Services that the Target Service host responds to the client about is implementation-specific. The semantics of the construction of a Probe Match message depends on the version of WS-Discovery that is being extended.

If the protocol is initialized due to the receipt of a Resolve message, then there are no specific message processing events or sequencing rules to follow.

This protocol extension does not define any new operations beyond those defined in the WS-Discovery protocol, as specified in [\[WS-Discovery1.1\]](#) or [\[WS-Discovery\]](#).

3.1.5 Timer Events

If the **DurationTimer** is not set as part of the initialization, then the server does not have any constraints placed by this timer; the operation has an infinite amount of time to execute.

If the **DurationTimer** is set and expires, then the Target Service host MUST NOT send any more replies to the client.

3.1.6 Other Local Events

None.

3.2 Client Details

3.2.1 Abstract Data Model

A client implementation must maintain the following data elements:

MaxResults: This value indicates the maximum number of Target Services that a client requires.

3.2.2 Timers

DurationTimer: This timer regulates the maximum time a client is willing to wait to get a response for a particular Probe or Resolve request.

The default value for this timer is 20 seconds.

3.2.3 Initialization

An application developer initializes the value of **DurationTimer** and the value of **MaxResults** by providing these values to appropriate methods that result in the transmission of Probe and Resolve messages.

If any of the values do not conform to the constraints defined in section [2.2.3](#), then the client MUST NOT send the Probe or Resolve message to the recipient.

If a value for **DurationTimer** is set and is equal to P10675199DT2H48M05.4775807S, then the timer MUST NOT be instantiated; this value implies that the operation has an infinite amount of time to execute. Otherwise, the **DurationTimer** is initialized with the value specified by the developer and then started.

3.2.4 Message Processing Events and Sequencing Rules

3.2.4.1 Probe Operation

These message processing and sequencing rules are to be followed when sending a Probe message.

If a valid value for a **DurationTimer** is specified by the application developer that is not equal to P10675199DT2H48M05.4775807S, then the client MUST send this value as part of the <Duration> element specified in section [2.2.3.2](#).

If the value for the **DurationTimer** is specified to be P10675199DT2H48M05.4775807S, then the client MUST NOT add the <Duration> element to the Probe envelope defined by WS-Discovery.

If the value for the **DurationTimer** is not specified by the application developer, then the client MUST send the default value for the **DurationTimer**, as specified in section [3.2.2](#), as part of the <Duration> element specified in section [2.2.3.2](#), and set the **DurationTimer** to this value.

The <Duration> element MUST be inserted into the xs:any blocks of the Probe envelope defined by WS-Discovery.

If a value for **MaxResults** is specified by the application developer to be 2,147,483,647 and the **DurationTimer** is specified to be P10675199DT2H48M05.4775807S, then the client MUST NOT send the Probe envelope defined by WS-Discovery.

Otherwise, if a value for **MaxResults** is specified by the application developer, then the client MUST send this value as part of the <MaxResults> element, as specified in section [2.2.3.1](#).

The <MaxResults> element MUST be inserted into the xs:any blocks of the Probe message defined by WS-Discovery.

An example of a Probe message with these elements inserted is provided in section [4](#).

3.2.4.2 Resolve Operation

If a value for **DurationTimer** is specified by the application developer for a Resolve operation and the value is less than P10675199DT2H48M05.4775807S, then the client MUST send this value as part of the <Duration> element specified in section [2.2.3.2](#).

If the value for the **DurationTimer** is not specified, then the client MUST send the default value for the **DurationTimer**, as specified in section [3.2.2](#), as part of the <Duration> element.

If the value for the **DurationTimer** is specified by the application developer to be P10675199DT2H48M05.4775807S, then the client MUST NOT add the <Duration> element to the Resolve envelope defined by WS-Discovery and MUST NOT set the **DurationTimer**.

The <Duration> element MUST be inserted into the xs:any blocks of the Resolve envelope defined by WS-Discovery.

If a value for **MaxResults** is specified by the application developer for a Resolve operation, then the client MUST NOT add the <MaxResults> element to the Resolve envelope defined by WS-Discovery.

After the client receives a number of responses greater than the value of **MaxResults**, it MUST ignore the remaining responses.

An example of a Resolve message with these elements inserted is provided in section [4](#).

3.2.5 Timer Events

If the **DurationTimer** is not set as part of the initialization, then the server does not have any constraints placed by this timer; the operation has an infinite amount of time to execute.

If the **DurationTimer** is set and expires, then the client MUST ignore any additional responses.

3.2.6 Other Local Events

None.

4 Protocol Examples

4.1 Probe Message Example

A client performing stock trades needs to locate a Target Service of **Type** `i:StatisticalAnalysis` and requires that service in the next 5 seconds. Using an implementation of WS-Discovery, it multicasts the following Probe message, which contains the `<MaxResults>` and `<Duration>` elements, as defined in this document. The `<MaxResults>` and `<Duration>` elements contain the termination criteria that limit the number of responses to one and the duration to 5 seconds. This following Probe message is sent from a client that is implementing [\[WS-Discovery1.1\]](#).

Example 1: Probe message sent using [WS-Discovery1.1]

```
(01) <s:Envelope
(02)   xmlns:a="http://schemas.xmlsoap.org/ws/2005/08/addressing"
(03)   xmlns:d="http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01"
(04)   xmlns:i="http://statistics.example.org/2003/stats"
(05)   xmlns:s="http://www.w3.org/2003/05/soap-envelope" >
(06) <s:Header>
(07)   <a:Action>
(08)     http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/Probe
(09)   </a:Action>
(10)   <a:MessageID>
(11)     urn:uuid:3afd5e21-de48-4277-8119-f25ebdd9890a
(12)   </a:MessageID>
(13)   <a:To>urn:docs-oasis-open-org:ws-dd:discovery:2009:01</a:To>
(14) </s:Header>
(15) <s:Body>
(16)   <d:Probe>
(17)     <d:Types>i:StatisticalAnalysis</d:Types>
(18)     <MaxResults
xmlns="http://schemas.microsoft.com/ws/2008/06/discovery">1</MaxResults>
(19)     <Duration
xmlns="http://schemas.microsoft.com/ws/2008/06/discovery">PT5S</Duration>
(20)   </d:Probe>
(21) </s:Body>
(22) </s:Envelope>
(23)
```

Lines 18 and 19 show the termination criterion inserted into the Probe message.

A Target Service host is aware of several Target Services that match the Type the client is looking for. However, it is limited to responding with a Probe Match containing information about only one Target Service. If it is able to respond within 5 seconds, it replies back to the client with a Probe Match message containing information about one such service.

4.2 Resolve Message Example

A client on a factory floor needs to resolve a tracking service located on a barcode scanning device. Since the device is mobile, it is using a **universally unique identifier (UUID)** for its address. The client needs to find the service in the next 10 seconds and sends out the following Resolve message. This Resolve message is sent from a client that is implementing [\[WS-Discovery\]](#).

Example 2: Resolve message sent using [WS-Discovery]

```
(01) <s:Envelope
(02)   xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(03)   xmlns:d="http://schemas.xmlsoap.org/ws/2005/04/discovery"
(04)   xmlns:s="http://www.w3.org/2003/05/soap-envelope" >
(05) <s:Header>
```

```
(06) <a:Action>
(07)   http://schemas.xmlsoap.org/ws/2005/04/discovery/Resolve
(08) </a:Action>
(09) <a:MessageID>
(10)   urn:uuid:3a2886e0-0ab0-44ff-8a61-1ceb223be3ec
(11) </a:MessageID>
(12) <a:To>urn:schemas-xmlsoap-org:ws:2005:04:discovery</a:To>
(13) </s:Header>
(14) <s:Body>
(15)   <d:Resolve>
(16)     <a:EndpointReference>
(17)       <a:Address>
(18)         urn:uuid:9dec7471-e559-4dc5-ba85-50b68bb8d938
(19)       </a:Address>
(20)     </a:EndpointReference>
(21)     <Duration xmlns="http://schemas.microsoft.com/ws/2008/06/discovery">
(22)       PT10S
(23)     </Duration>
(24)   </d:Resolve>
(25) </s:Body>
(26) </s:Envelope>
```

Lines 21 through 23 show the Duration element inserted into the Resolve message.

A Target Service host is aware of the Target Service that matches the address the client is looking for. If it is able to respond within 10 seconds, it replies back to the client with a Resolve Match message containing information about that Target Service.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

This protocol extension does not have a specific WSDL; rather it uses the WSDL declaration of the WS-Discovery protocol that it extends.

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.

This document specifies version-specific details in the Microsoft .NET Framework. For information about which versions of .NET Framework are available in each released Windows product or as supplemental software, see [.NET Framework](#).

- Microsoft .NET Framework 4.0
- Microsoft .NET Framework 4.5
- Microsoft .NET Framework 4.6

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 Appendix C: Protocol Schema

The following is the XSD schema for the `http://schemas.microsoft.com/ws/2008/06/discovery` namespace.

```
<xs:schema targetNamespace="http://schemas.microsoft.com/ws/2008/06/discovery"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MaxResults" type="xs:positiveInteger" />
  <xs:element name="Duration" type="xs:duration" />
</xs:schema>
```

9 Change Tracking

This section identifies changes that were made to this document since the last release. 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- 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 dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Z Appendix B: Product Behavior	Added .NET Framework 4.6 to the applicability list.	Y	Content update.

10 Index

<

[<Duration> element](#) 11
[<MaxResults> element](#) 10

A

Abstract data model
[client](#) 13
[server](#) 12
[Applicability](#) 9
[Attribute groups](#) 11
[Attributes](#) 11

C

[Capability negotiation](#) 9
[Change tracking](#) 22
Client
[abstract data model](#) 13
[initialization](#) 13
[local events](#) 15
[message processing](#) 14
[sequencing rules](#) 14
[timer events](#) 15
[timers](#) 13
[Complex types](#) 11

D

Data model - abstract
[client](#) 13
[server](#) 12
[Duration element](#) 11

E

Elements
[<Duration>](#) 11
[<MaxResults>](#) 10
Events
local
[client](#) 15
[server](#) 13
[local - client](#) 15
[local - server](#) 13
[timer - client](#) 15
[timer - server](#) 13
Examples
[Probe message](#) 16
[Resolve message](#) 16

F

[Fields - vendor-extensible](#) 9
[Full WSDL](#) 19

G

[Glossary](#) 6
[Groups](#) 11

I

[Implementer - security considerations](#) 18
[Index of security parameters](#) 18
[Informative references](#) 8
Initialization
[client](#) 13
[server](#) 12
[Introduction](#) 6

L

Local events
[client](#) 15
[server](#) 13

M

[MaxResults element](#) 10
Message processing
[client](#) 14
[server](#) 13
Messages
[<Duration> element](#) 11
[<MaxResults> element](#) 10
[attribute groups](#) 11
[attributes](#) 11
[complex types](#) 11
[elements](#) 10
[enumerated](#) 10
[groups](#) 11
[namespaces](#) 10
[simple types](#) 11
[syntax](#) 10
[transport](#) 10

N

[Namespaces](#) 10
[Normative references](#) 7

O

[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 18
[Preconditions](#) 9
[Prerequisites](#) 9
[Probe message example](#) 16
[Product behavior](#) 20

R

[References](#) 7
 [informative](#) 8
 [normative](#) 7
[Relationship to other protocols](#) 8
[Resolve message example](#) 16

S

Security
 [implementer considerations](#) 18
 [parameter index](#) 18
Sequencing rules
 [client](#) 14
 [server](#) 13
Server
 [abstract data model](#) 12
 [initialization](#) 12
 [local events](#) 13
 [message processing](#) 13
 [sequencing rules](#) 13
 [timer events](#) 13
 [timers](#) 12
[Simple types](#) 11
[Standards assignments](#) 9
Syntax
 [attribute groups](#) 11
 [attributes](#) 11
 [complex types](#) 11
 elements
 [Duration](#) 11
 [MaxResults](#) 10
 [overview](#) 10
 [groups](#) 11
 [messages](#) 10
 [messages - overview](#) 10
 [namespaces](#) 10
 [overview](#) 10
 [simple types](#) 11

T

Timer events
 [client](#) 15
 [server](#) 13
Timers
 [client](#) 13
 [server](#) 12
[Tracking changes](#) 22
[Transport](#) 10
Types
 [complex](#) 11
 [simple](#) 11

V

[Vendor-extensible fields](#) 9
[Versioning](#) 9

W

[WSDL](#) 19