

[MS-DPWSSN]:

Devices Profile for Web Services (DPWS): Size Negotiation Extension

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. 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
12/5/2008	0.1	Major	Initial Availability.
1/16/2009	0.1.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	0.1.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	0.1.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	0.1.4	Editorial	Changed language and formatting in the technical content.
7/2/2009	1.0	Major	Updated and revised the technical content.
8/14/2009	1.0.1	Editorial	Changed language and formatting in the technical content.
9/25/2009	1.1	Minor	Clarified the meaning of the technical content.
11/6/2009	1.1.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	2.0	Major	Updated and revised the technical content.
1/29/2010	2.1	Minor	Clarified the meaning of the technical content.
3/12/2010	2.1.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	2.1.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	2.1.3	Editorial	Changed language and formatting in the technical content.
7/16/2010	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	2.2	Minor	Clarified the meaning of the technical content.
9/23/2011	2.2	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	3.0	Major	Updated and revised the technical content.
3/30/2012	3.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
7/12/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	3.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	4.0	Major	Updated and revised 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	7
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.....	11
2.2.3	Elements	11
2.2.3.1	Ims:LargeMetadataSupport	11
2.2.4	Complex Types.....	11
2.2.5	Simple Types	11
2.2.6	Attributes	11
2.2.7	Groups	12
2.2.8	Attribute Groups.....	12
3	Protocol Details	13
3.1	Server Details.....	13
3.1.1	Abstract Data Model.....	13
3.1.2	Timers	13
3.1.3	Initialization.....	13
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	14
3.2.1	Abstract Data Model.....	14
3.2.2	Timers	14
3.2.3	Initialization.....	14
3.2.4	Message Processing Events and Sequencing Rules	14
3.2.5	Timer Events.....	14
3.2.6	Other Local Events.....	14
4	Protocol Examples	15
4.1	Request From a Client Using This Protocol Extension	15
4.2	Request From a Client Without This Protocol Extension	15
4.3	Response Message from DPWS	15
4.4	Using the Ims:LargeMetadataSupport Element.....	15
5	Security.....	17
5.1	Security Considerations for Implementers	17
5.2	Index of Security Parameters	17
6	Appendix A: Full WSDL	18
7	Appendix B: Product Behavior	19
8	Change Tracking.....	20

1 Introduction

This document specifies an extension to the Devices Profile for Web Services (DPWS) to allow the negotiation of message sizes between a client and a service for a specific message transaction. This extension to an existing protocol does not define new operations, but instead defines **XML Schema** that may be added to existing messages to allow clients and services to properly configure themselves.

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:

client: A computer on which the remote procedure call (RPC) client is executing.

device: The Devices Profile for Web Services (DPWS) term for a special instance of a service that is discoverable and contains other services with metadata describing those services.

endpoint: In the context of a web service, a network target to which a SOAP message can be addressed. See [\[WSADDR\]](#).

service: The receiving endpoint of a **web services** request message, and sender of any resulting **web services** response message.

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a URI value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [\[SOAP1.2-1/2007\]](#) section 5.2 for more information.

SOAP message: An XML document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory SOAP body. See [\[SOAP1.2-1/2007\]](#) section 5 for more information.

SOAP Message: The data encapsulated in a SOAP envelope that flows back and forth between a protocol client and a web service, as described in [\[SOAP1.1\]](#).

web service: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as HTTP, Simple Mail Transfer Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML Schema (XSD): A language that defines the elements, attributes, namespaces, and data types for XML documents as defined by [\[XMLSCHEMA1/2\]](#) and [\[W3C-XSD\]](#) standards. An XML schema uses XML syntax for its language.

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.

[DPWS] Chans, S., Conti, D., Schlimmer, J., et al., "Devices Profile for Web Services", February 2006, <http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf>

[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>

[SOAP1.2-1/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[WSAddressing] Box, D., et al., "Web Services Addressing (WS-Addressing)", August 2004, <http://www.w3.org/Submission/ws-addressing/>

[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>

[WSMETA] Ballinger, K., Bissett, B., Box, D., et al., "Web Services Metadata Exchange (WS-MetadataExchange)", Version 1.1, August 2006, <http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf>

[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

Devices Profile for Web Services (DPWS) specifies a device friendly, well-structured messaging model providing basic functionality such as discovery of an endpoint, metadata for that endpoint, and request/response messaging. This model is built using core **Web Services** specifications as building blocks, and assembled with explanatory text. DPWS identifies the roles of clients, which discover device endpoints and communicate with **devices** and **services**; devices, which can be discoverable service endpoints that host other services, and the services hosted within the device. Additionally, DPWS defines metadata for both devices and the service endpoints hosted by devices.

This model maps to the requirements of modern home computers in most cases. Home computers are often set to be discoverable and provide metadata describing themselves and their endpoints (such as file shares) and resources to clients on a network.

DPWS has one key restriction, in that it recommends clients and services limit their messages to 32,767 octets in length (See [\[DPWS\]](#) Appendix I, R0003, and R0026).

Windows leverages the DPWS model for describing home computers and their metadata. In DPWS terminology, a home computer is a device, and services and resources available from the home computer are described by services in metadata.

The metadata provided may be quite large, beyond the size originally envisioned by DPWS for resource constrained devices. This large size is always due to metadata describing resource or **endpoints** hosted within the device. This document describes an extension to DPWS that allows a DPWS-based client and service to negotiate a larger acceptable message size.

A client that is capable of accepting large message responses may send a request with the large message support indicator. The service, upon receiving a request, looks for the large message support indicator and:

- If the indicator is present, the service constructs a response which includes all configured metadata and endpoints available from the home computer.
- If the indicator is not present, the service determines what the size of the response will be, and either:
 - If the size of the response is less than or equal to the 32,767 octet limit defined in [\[DPWS\]](#), the response is constructed and sent to the client.
 - If the size of the response is more than the 32,767 octet limit defined in [\[DPWS\]](#), the response is changed to contain only metadata describing the computer, and is then sent.

The logic in this protocol is structured in this way to help preserve network behavior with clients that do not support the large message support indicator. These clients will still see the existence and basic description of the device, but will not see all of the associated resources and services, as these clients expect services to manage their metadata to within the 32,767 octet limit.

1.4 Relationship to Other Protocols

This extension is built on DPWS and relies on DPWS functionality to work. This extension is designed to allow DPWS compliant implementations to negotiate larger message sizes, as supported by standard Web Services; that is, normal Web Services not governed by DPWS do not need such a protocol extension.

This extension does not define new **SOAP messages** or patterns, but instead defines XML Schema that may be added to existing schema extensibility points.

1.5 Prerequisites/Preconditions

Both the client and the device in this exchange must be DPWS compliant.

The client must accept SOAP messages larger than 32,767 octets.

Services must be capable of sending as much data as they can prepare.

1.6 Applicability Statement

Use of this protocol is appropriate for client implementations if:

- The client is a DPWS compliant implementation.
- The client intends to communicate with the DPWS representation of a computer.
- The client supports receiving DPWS response messages larger than 32,767 octets in length.

Use of this protocol is appropriate for service implementations if:

- The service intends to represent itself as a DPWS compliant computer on the network.
- The service supports sending messages larger than 32,767 in length.

1.7 Versioning and Capability Negotiation

This protocol extension covers a single version of the extension as specified in section [2.1](#). There are no previous versions. There are no special transport requirements or other restrictions placed on DPWS resulting from use of this extension.

1.8 Vendor-Extensible Fields

The schema for this protocol extension does define additional vendor extensions.

1.9 Standards Assignments

There are no specific standards assignments for this protocol extension.

2 Messages

2.1 Transport

Use of this protocol extension requires support for the underlying specifications:

- **Clients** and devices implementing this protocol extension **MUST** be compliant with DPWS as specified in [\[DPWS\]](#).
- DPWS-compliant endpoints **MUST** be fully compatible with the February 2006 revision of DPWS.

DPWS compliance ensures clients and device support the consistent set of protocol versions.

This protocol extension is limited to these existing messages:

- Devices implementing this protocol extension **MUST** support receiving the extension in the **SOAP Header** of a **SOAP Message** with the **SOAP Action** <http://schemas.xmlsoap.org/ws/2004/09/transfer/Get>
- Devices implementing this protocol extension **MUST** support sending a SOAP Message as a response with a length greater than 32,767 octets.
- Clients implementing this protocol extension **MUST** support sending the extension in the SOAP Header of a SOAP Message with the SOAP Action <http://schemas.xmlsoap.org/ws/2004/09/transfer/Get>
- Clients implementing this protocol extension **MUST** support receiving a SOAP Message as a response with a length greater than 32,767 octets.

Not all SOAP implementations properly support SOAP Header extensions. This support is required for use of this extension.

This extension does not define a finite boundary for the size of the SOAP Message response.

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#) and **Web Services Description Language (WSDL)** as defined in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific XML namespace prefix for each 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
s	http://www.w3.org/2003/05/soap-envelope	[SOAP1.2-1/2003]
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	[WSAddressing]
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	[WSMETA]
lms	http://schemas.microsoft.com/windows/dpws/LargeMetadataSupport/2007/08	This document

2.2.2 Messages

This protocol does not define any messages.

2.2.3 Elements

The following table summarizes the set of common XML Schema element definitions defined by this specification. XML Schema element definitions that are specific to a particular operation are described with the operation.

Element	Description
lms:LargeMetadataSupport	Indicator for support of large metadata messages.

2.2.3.1 lms:LargeMetadataSupport

The lms:LargeMetadataSupport element is used to indicate client side support for large metadata responses. The normative form for this element is as follows.

```
<lms:LargeMetadataSupport/>
```

The element is not extensible and does not support attributes.

The schema for this element is as follows.

```
<xs:element name='LargeMetadataSupport'>
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

This element MUST NOT be used outside of the top level of the SOAP header in a SOAP Message. If the element is used outside of the top level of the SOAP header where it is defined as semantically meaningful, then it will be ignored. Standard exceptions to this apply (for example, if the header is inserted in a location that violates the schema). This element MUST NOT be used in SOAP headers where the wsa:Action is not http://schemas.xmlsoap.org/ws/2004/09/transfer/Get. If the element is used in a SOAP header when the action is not http://schemas.xmlsoap.org/ws/2004/09/transfer/Get and the service understands the action, then the element will be ignored.

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

This protocol extension governs the behavior of a device's use of the data model specified in [\[DPWS\]](#) depending on whether a client supports large metadata responses.

3.1 Server Details

3.1.1 Abstract Data Model

This protocol extension does not describe a separate data model.

Devices that support this protocol extension MUST support the data model described in [\[DPWS\]](#) section 5.

Data specified in [\[DPWS\]](#) section 5.2 is sent in part or in full depending on client support for large metadata responses (as indicated by the presence of the `<Ims:LargeMetadataSupport>` element as defined in section [2.2.3.1](#)) and the size of the metadata response.

3.1.2 Timers

This protocol extension requires no new timers.

3.1.3 Initialization

This protocol extension requires no new initialization.

3.1.4 Message Processing Events and Sequencing Rules

If the data specified in [\[DPWS\]](#) section 5.2 is sent with a full response exceeding 32,767 octets in length and the client supports large metadata responses, the full response

- MUST be compliant with [\[DPWS\]](#) section 3, except R0003 and R0026,
- MUST be compliant with [\[DPWS\]](#) section 5, and
- MUST be sent in its entirety.

If the data specified in [\[DPWS\]](#) section 5.2 is sent in part due to a full response exceeding 32,767 octets in length and the client not supporting large metadata responses, the `wsdp:Relationship/wsdp:Host` element and its children MUST be included in the response, and the `wsdp:Relationship/wsdp:Hosted` element MAY be included, to the extent that its inclusion does not cause the response to exceed 32,767 octets in length.

This specification does not define any new Web Services Description Language (WSDL) operations.

3.1.5 Timer Events

This protocol extension defines no timer events.

3.1.6 Other Local Events

This protocol extension defines no local events and is not subject to any new local events.

3.2 Client Details

3.2.1 Abstract Data Model

This protocol extension does not describe a separate data model.

Clients that support this protocol extension MUST support the data model described in [\[DPWS\]](#) section 5.

The client side of this protocol extension is simply a pass-through. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

Clients supporting large metadata responses MUST include the <Ims:LargerMetadataSupport> element as defined in section [2.2.3.1](#).

3.2.2 Timers

This protocol extension requires no new timers.

3.2.3 Initialization

This protocol extension requires no new initialization.

3.2.4 Message Processing Events and Sequencing Rules

This protocol extension describes no separate message processing events and sequencing rules.

This specification does not define any new WSDL operations.

3.2.5 Timer Events

This protocol extension defines no timer events.

3.2.6 Other Local Events

This protocol extension defines no local events and is not subject to any new local events.

4 Protocol Examples

4.1 Request From a Client Using This Protocol Extension

The following is an abbreviated example request from a client using this protocol extension:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:To>xs:anyURI</wsa:To>
    <lms:LargeMetadataSupport/>
  </s:Header>
  <s:Body/>
</s:Envelope>
```

4.2 Request From a Client Without This Protocol Extension

The following is an abbreviated example request from a client without this protocol extension:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:To>xs:anyURI</wsa:To>
  </s:Header>
  <s:Body/>
</s:Envelope>
```

4.3 Response Message from DPWS

The following is an abbreviated example response message from DPWS (see [\[DPWS\]](#) section 5.1).

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
    </wsa:Action>
    <wsa:RelatesTo>xs:RequestURI</wsa:RelatesTo>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
  </s:Header>
  <s:Body ...>
    <wsx:Metadata>
      ...
    </wsx:Metadata>
  </s:Body>
</s:Envelope>
```

4.4 Using the lms:LargeMetadataSupport Element

A usage example of the lms:LargeMetadataSupport element is as follows.

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:To>xs:anyURI</wsa:To>
    <lms:LargeMetadataSupport/>
  </s:Header>
  <s:Body ...>
    ...
  </s:Body>
</s:Envelope>
```


5 Security

5.1 Security Considerations for Implementers

Clients indicating they support large metadata response may use this in a DDoS attack on services . The nature of the attack is to have a large number of distributed clients issue requests indicating they support large responses. Assuming TCP is in use, the clients would then slow their processing of data and cause TCP backoffs to slow data transmission to 100 bytes/second. The large number of clients would result in a significant number of active connections and potentially a large amount of in memory state on the service. It is not clear that this attack is significantly more damaging than having the clients aggressively download, which would instead exhaust bandwidth but would have similar external consequences.

Similarly, clients indicating they support large metadata responses may end up receiving a large response at a very slow rate, and may be impacted by the same in-memory state concerns as services.

5.2 Index of Security Parameters

There are no security parameters for this protocol extension.

6 Appendix A: Full WSDL

This protocol extension does not define a WSDL. It extends the WSDL defined and referenced by [\[DPWS\]](#). The following XSD contains a definition of the `Ims:LargeMetadataSupport` element (section [2.2.3.1](#)).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schemas.microsoft.com/windows/dpws/LargeMetadataSupport/2007/08"
  xmlns:lms=http://schemas.microsoft.com/windows/dpws/LargeMetadataSupport/2007/08
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name='LargeMetadataSupport'>
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

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.

- 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
- Windows Server 2016 Technical Preview 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.

8 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 Windows 10 to applicability list.	Y	Content update.

9 Index

A

Abstract data model
[client](#) 14
[server](#) 13
[Abstract data model - server](#) 13
[Applicability](#) 9
[Attribute groups](#) 12
[Attributes](#) 11

C

[Capability negotiation](#) 9
[Change tracking](#) 20
Client
[abstract data model](#) 14
[initialization](#) 14
[local events](#) 14
[message processing](#) 14
[sequencing rules](#) 14
[timer events](#) 14
[timers](#) 14
Common message syntax
[attribute groups](#) 12
[attributes](#) 11
[complex types](#) 11
[elements](#) 11
[groups](#) 12
[messages](#) 11
[namespaces](#) 10
[overview](#) 10
[simple types](#) 11
[Complex types](#) 11

D

[Data model - abstract](#) 13
[client](#) 14
[server](#) 13

E

Elements
[Ims:LargeMetadataSupport](#) 11
Events
[local - client](#) 14
[local - server](#) 13
[timer - client](#) 14
[timer - server](#) 12
Examples
[Ims:LargeMetadataSupport element](#) 15
[request from a client using this protocol extension](#)
15

[request from a client without this protocol extension](#) 15
[response message - from \[DPWS\]](#) 15

F

[Fields - vendor-extensible](#) 9
[Full WSDL](#) 18

G

[Glossary](#) 6
[Groups](#) 12

I

[Implementer - security considerations](#) 17
[Index of security parameters](#) 17
[Informative references](#) 7
Initialization
[client](#) 14
[server](#) 13
[Initialization - server](#) 13
[Introduction](#) 6

L

[Ims:LargeMetadataSupport element](#) 11
[Ims:LargeMetadataSupport element example](#) 15
Local events
[client](#) 14
[server](#) 13
[Local events - server](#) 13

M

Message processing
[client](#) 14
[server](#) 13
[Message processing - server](#) 13
Messages
[attribute groups](#) 12
[attributes](#) 11
[complex types](#) 11
[elements](#) 11
[enumerated](#) 11
[groups](#) 12
[Ims:LargeMetadataSupport element](#) 11
[namespaces](#) 10
[simple types](#) 11
[syntax](#) 10
[attribute groups](#) 12

- [attributes](#) 11
- [complex types](#) 11
- [elements](#) 11
- [groups](#) 12
- [messages](#) 11
- [namespaces](#) 10
- [overview](#) 10
- [simple types](#) 11
- [transport](#) 10

N

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

O

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

P

- [Parameters - security index](#) 17
- [Preconditions](#) 9
- [Prerequisites](#) 9
- [Product behavior](#) 19
- Protocol Details
 - [overview](#) 13

R

- [References](#) 7
 - [informative](#) 7
 - [normative](#) 7
- [Relationship to other protocols](#) 8
- [Request from a client using this protocol extension example](#) 15
- [Request from a client without this protocol extension example](#) 15
- [Response message example](#) 15

S

- Security
 - [implementer considerations](#) 17
 - [parameter index](#) 17
- Sequencing rules
 - [client](#) 14
 - [server](#) 13
 - [Sequencing rules - server](#) 13
- Server
 - [abstract data model](#) 13
 - [initialization](#) 13
 - [local events](#) 13
 - [message processing](#) 13
 - [sequencing rules](#) 13

- [timer events](#) 13
- [timers](#) 13
- [Simple types](#) 11
- [Standards assignments](#) 9
- Syntax
 - [attribute groups](#) 12
 - [attributes](#) 11
 - [complex types](#) 11
 - elements
 - [lms:LargeMetadataSupport](#) 11
 - [overview](#) 11
 - [groups](#) 12
 - [messages](#) 11
 - [messages - overview](#) 10
 - [namespaces](#) 10
 - [overview](#) 10
 - [simple types](#) 11

T

- Timer events
 - [client](#) 14
 - [server](#) 13
 - [Timer events - server](#) 13
- Timers
 - [client](#) 14
 - [server](#) 13
 - [Timers - server](#) 13
 - [Tracking changes](#) 20
 - [Transport](#) 10
- Types
 - [complex](#) 11
 - [simple](#) 11

V

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

W

- [WSDL](#) 18