

[MS-SPTWS]: Service Platform Topology Web 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 [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.06	Editorial	Changed language and formatting in the technical content.
03/18/2011	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	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	6
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	10
2.2.3 Elements	10
2.2.4 Complex Types	10
2.2.4.1 SPTopologyWebServiceApplicationFault	10
2.2.5 Simple Types	10
2.2.5.1 guid	11
2.2.6 Attributes	11
2.2.7 Groups	11
2.2.8 Attribute Groups	11
3 Protocol Details	12
3.1 ITopologyWebServiceApplication Server Details	12
3.1.1 Abstract Data Model	12
3.1.2 Timers	13
3.1.3 Initialization	13
3.1.4 Message Processing Events and Sequencing Rules	13
3.1.4.1 EnumerateSharedServiceApplications	13
3.1.4.1.1 Messages	14
3.1.4.1.1.1 ITopologyWebServiceApplication_EnumerateSharedServiceApplicationns_InputMessage	14
3.1.4.1.1.2 ITopologyWebServiceApplication_EnumerateSharedServiceApplicationns_OutputMessage	14
3.1.4.1.2 Elements	14
3.1.4.1.2.1 EnumerateSharedServiceApplications	14
3.1.4.1.2.2 EnumerateSharedServiceApplicationsResponse	14
3.1.4.1.3 Complex Types	15
3.1.4.1.3.1 ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg	15
3.1.4.1.3.2 ArrayOfSPSharedServiceApplicationInfo	15
3.1.4.1.3.3 SPSharedServiceApplicationInfo	16
3.1.4.1.3.4 Version	17
3.1.4.1.4 Simple Types	17

3.1.4.1.5	Attributes.....	17
3.1.4.1.6	Groups.....	17
3.1.4.1.7	Attribute Groups	17
3.1.4.2	GetEndpoints	17
3.1.4.2.1	Messages	18
3.1.4.2.1.1	ITopologyWebServiceApplication_GetEndpoints_InputMessage	18
3.1.4.2.1.2	ITopologyWebServiceApplication_GetEndpoints_OutputMessage	18
3.1.4.2.2	Elements.....	18
3.1.4.2.2.1	GetEndpoints	18
3.1.4.2.2.2	GetEndpointsResponse	19
3.1.4.2.3	Complex Types	19
3.1.4.2.3.1	ReadOnlyCollectionOfanyURI.....	19
3.1.4.2.3.2	ArrayOfanyURI.....	19
3.1.4.2.4	Simple Types.....	20
3.1.4.2.5	Attributes.....	20
3.1.4.2.6	Groups.....	20
3.1.4.2.7	Attribute Groups	20
3.1.5	Timer Events	20
3.1.6	Other Local Events	20
4	Protocol Examples.....	21
4.1	EnumerateSharedServiceApplications Operation	21
4.2	GetEndpoints Operation.....	22
5	Security.....	25
5.1	Security Considerations for Implementers.....	25
5.2	Index of Security Parameters	25
6	Appendix A: Full WSDL.....	26
7	Appendix B: Product Behavior.....	31
8	Change Tracking.....	32
9	Index	34

1 Introduction

This document specifies the Service Platform Topology Web Service Protocol. This protocol allows a protocol client to request a list of the service applications that are known to a protocol server, and to retrieve information, such as endpoint URIs, about those service applications.

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

1.1 Glossary

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

Augmented Backus-Naur Form (ABNF)
GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)

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

build number
class module
display name
endpoint
major version
minor version
revision
service application
Simple Object Access Protocol (SOAP)
SOAP action
SOAP body
SOAP fault
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
WSDL operation
XML schema definition (XSD)

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-SPSTWS] Microsoft Corporation, "[SharePoint Security Token Service Web Service Protocol Specification](#)".

[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>

[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>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

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

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624>

[WSA1.0 Metadata] Gudgin, M., Ed., Hadley, M., Ed., Rogers, T., Ed., Yalcinalp, U., Ed., "Web Services Addressing 1.0 - Metadata", W3C Recommendation, September 2007, <http://www.w3.org/TR/2007/REC-ws-addr-metadata-20070904>

[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>

[WS-Policy1.5] W3C, "Web Services Policy 1.5 - Framework", 04 September 2007, <http://www.w3.org/TR/ws-policy/ws-policy-framework.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.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

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

1.2.2 Informative References

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol allows a protocol client to obtain a list of and information about **service applications** that are known to a protocol server. It also allows the protocol client to retrieve a list of **endpoint (4) URIs** for a service application and to communicate with that service application by using a protocol supported by that service application.

This protocol supports two major types of operations:

- Enumerating service applications - A protocol client can request a list of service applications that are known to a protocol server. The protocol client can display this list to a user and enable the user to select and perform additional operations with a service application that is in the list.
- Retrieving endpoints (4) for a service application - A protocol client can request a list of endpoint (4) URIs for a specified service application. Protocol clients typically use this type of operation to identify the endpoints (4) of the protocols that they are using.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [\[SOAP1.1\]](#), [\[SOAP1.2/1\]](#) and [\[SOAP1.2/2\]](#). It transmits those messages by using **HTTPS**, as described in [\[RFC2818\]](#).

The following diagram shows the underlying messaging and transport stack used by the protocol:

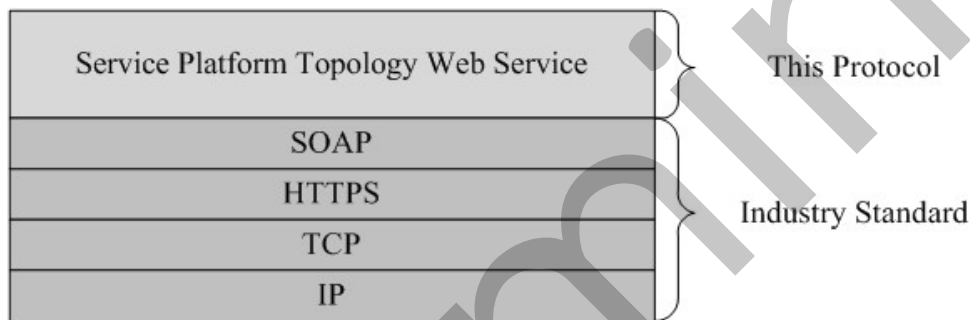


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more endpoint (4) URIs that are known by protocol clients. The protocol server endpoint (4) URI is known by the protocol client.

This protocol requires that the protocol client has the requisite permission to call methods that are implemented on the protocol server.

1.6 Applicability Statement

This protocol is intended for use by protocol clients and protocol servers that are connected by high-bandwidth, low-latency network connections.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol can be implemented by using transports that support sending Simple Object Access Protocol (SOAP) messages, as specified in section 2.1.
- **Protocol Versions:** This protocol is not versioned.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Preliminary

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **present**, and **not null**.

2.1 Transport

Protocol servers **MUST** support SOAP over HTTPS.

All protocol messages **MUST** be transported by using HTTPS bindings at the transport level and a server certificate **MUST** be deployed.

Protocol messages **MUST** be formatted as specified in either [\[SOAP1.1\]](#) section 4 or [\[SOAP1.2/1\]](#) section 5. Protocol server faults **MUST** be returned by using either **HTTP** status codes, as specified in [\[RFC2616\]](#) section 10, or **SOAP faults**, as specified in [\[SOAP1.1\]](#) section 4.4 or [\[SOAP1.2/1\]](#) section 5.4.

This protocol **MAY** transmit an additional SOAP header, the **ServiceContext** header, as specified in [\[MS-SPSTWS\]](#).

This protocol does not define any means for activating a protocol server or protocol client. The protocol server **MUST** be configured and begin listening in an implementation-specific way.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses XML Schema, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and WSDL, as specified 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
wsp	http://www.w3.org/ns/ws-policy	[WS-Policy1.5]
ser	http://schemas.microsoft.com/2003/10/Serialization/	
xsd	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soapenc	http://schemas.xmlsoap.org/soap/encoding/	[SOAP1.2/1] [SOAP1.2/2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]

Prefix	Namespace URI	Reference
tns	http://tempuri.org/	
wSDL	http://schemas.xmlsoap.org/wSDL/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
wsam	http://www.w3.org/2007/05/addressing/metadata	[WSA1.0 Metadata]
soap	http://schemas.xmlsoap.org/wSDL/soap/	[SOAP1.1]

2.2.2 Messages

None.

2.2.3 Elements

This specification does not define any common XML Schema element definitions.

2.2.4 Complex Types

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

Complex Type	Description
SPTopologyWebServiceApplicationFault	The SPTopologyWebServiceApplicationFault element specifies a SOAP fault.

2.2.4.1 SPTopologyWebServiceApplicationFault

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.SharePoint

The **SPTopologyWebServiceApplicationFault** element specifies a SOAP fault.

```
<xs:complexType name="SPTopologyWebServiceApplicationFault">
  <xs:sequence>
    <xs:element minOccurs="0" name="FaultReason" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

FaultReason: A string specifying the reason for the fault. This value MUST be ignored by the protocol client.

2.2.5 Simple Types

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

Simple Type	Description
guid	The guid simple type defines a unique identifier for a service application class module . This simple type is defined by the namespace at http://schemas.microsoft.com/2003/10/Serialization/ .

2.2.5.1 guid

Namespace: <http://schemas.microsoft.com/2003/10/Serialization/>

The **guid** simple type defines a unique identifier for a service application class module. This simple type is defined by the namespace at <http://schemas.microsoft.com/2003/10/Serialization/>.

```
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="\da-fA-F}{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>
```

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

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **present**, and **not null**.

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. 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.

This protocol allows protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults by using HTTP status codes or SOAP faults. Except where specified otherwise, protocol clients SHOULD interpret HTTP status codes as specified in [\[RFC2616\]](#) section 10. Protocol servers can notify protocol clients of application-level faults by using SOAP faults. Except where specified otherwise, these SOAP faults are not significant for interoperability and protocol clients can interpret them in an implementation-specific manner.

3.1 ITopologyWebServiceApplication 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.

The protocol server maintains a list of service applications and, for each of those service applications, the following properties:

- **GUID** - A unique identifier that represents the service application.
- **ApplicationClassId** - A unique identifier that represents the service application class module.
- **ApplicationVersion** - The version number of the service application class module.
- **Comments** - A string that contains descriptive information about the service application.
- **DisplayName** - A string that specifies the **display name** of the service application.
- **TermsOfServiceUri** - A **URL** for a document that contains additional, descriptive information about the service application.
- **Uri** - A **URI** that specifies a single address that logically represents the service application.
- **endpoint (4) URIs** - A list of all the endpoint (4) URIs for the service application. The protocol server maintains a list of endpoint (4) URIs for the default transport that the service application supports.

The format of these properties is specified in section [3.1.4.1.3.3](#).

The protocol server also maintains information about its endpoint (4) URIs in this list.

In response to messages from a protocol client, the protocol server sends a list of service applications and their properties to the protocol client. The addition, deletion, and modification of the list of service applications and their properties is not specified by this protocol and is implementation-specific.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

This specification includes the following **WSDL operations**:

WSDL Operation	Description
EnumerateSharedServiceApplications	The EnumerateSharedServiceApplications operation returns a collection of information about all of the service applications on the protocol server.
GetEndPoints	The GetEndPoints message returns a collection of all the endpoint (4) URIs for a service application that is specified by a GUID in an input message.

3.1.4.1 EnumerateSharedServiceApplications

The **EnumerateSharedServiceApplications** operation returns a collection of information about all of the service applications on the protocol server.

```
<wsdl:operation name="EnumerateSharedServiceApplications">
  <wsdl:input
    wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplications"
    message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage"/
  >
  <wsdl:output
    wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicationsResponse"
    message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage"
  />
  <wsdl:fault
    wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicationsSPTopologyWebServiceApplicationFaultFault"
    name="SPTopologyWebServiceApplicationFaultFault"
    message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebServiceApplicationFaultFault_FaultMessage"/>
</wsdl:operation>
```

The protocol client sends a **ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage** request **WSDL message** and the protocol server MUST respond with a **ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage** response WSDL message, as follows:

- If one or more service applications exist on the protocol server, the protocol server MUST return a collection of information about all of those service applications. The information is contained in an **EnumerateSharedServiceApplicationsResult** element, as specified in section [3.1.4.1.2.2](#).
- If no service applications exist on the protocol server, the protocol server MUST return an empty collection.

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.1.1.1

ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage

The request WSDL message for the **EnumerateSharedServiceApplications** WSDL operation.

The **SOAP action** value is:

```
http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplications
```

The **SOAP body** contains the **EnumerateSharedServiceApplications** element.

3.1.4.1.1.2

ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage

The response WSDL message for the **EnumerateSharedServiceApplications** WSDL operation.

The SOAP body contains the **EnumerateSharedServiceApplicationsResponse** element.

3.1.4.1.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.1.2.1 EnumerateSharedServiceApplications

The input data for the **EnumerateSharedServiceApplications** WSDL operation.

```
<xs:element name="EnumerateSharedServiceApplications">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
```

3.1.4.1.2.2 EnumerateSharedServiceApplicationsResponse

The result data for the **EnumerateSharedServiceApplications** WSDL operation.

```
<xs:element name="EnumerateSharedServiceApplicationsResponse">
  <xs:complexType>
    <xs:sequence>
```

```

    <xs:element minOccurs="0" name="EnumerateSharedServiceApplicationsResult"
nillable="true"
xmlns:q3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
type="q3:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

EnumerateSharedServiceApplicationsResult: A collection of information about all of the service applications that are on the protocol server. It MUST conform to the **XML schema definition (XSD)** of the **ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg** complex type, as specified in section [3.1.4.1.3.1](#). This element MUST be present and MUST NOT be empty.

3.1.4.1.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

3.1.4.1.3.1 ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg

Namespace: <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>

The **ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg** complex type contains the data that results from a **EnumerateSharedServiceApplications** WSDL operation. It is defined by the namespace at <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>.

```

<xs:complexType name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg">
  <xs:sequence>
    <xs:element name="list" nillable="true"
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
type="q2:ArrayOfSPSharedServiceApplicationInfo"/>
  </xs:sequence>
</xs:complexType>

```

list: An array of **SPSharedServiceApplicationInfo**. It MUST conform to the XSD of the **ArrayOfSPSharedServiceApplicationInfo** complex type, as specified in section [3.1.4.1.3.2](#).

3.1.4.1.3.2 ArrayOfSPSharedServiceApplicationInfo

Namespace: <http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration>

The **ArrayOfSPSharedServiceApplicationInfo** complex type is a list of objects, each of which contains information about a specific service application. It is defined by the namespace at <http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration>. If there are no service applications on the server, the returned list MUST be empty.

```

<xs:complexType name="ArrayOfSPSharedServiceApplicationInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="SPSharedServiceApplicationInfo"
nillable="true" type="tns:SPSharedServiceApplicationInfo"/>
  </xs:sequence>
</xs:complexType>

```

SPSharedServiceApplicationInfo: A structure that contains information about the service application. It MUST conform to the XSD of the **SPSharedServiceApplicationInfo** complex type, as specified in section [3.1.4.1.3.3](#).

3.1.4.1.3.3 SPSharedServiceApplicationInfo

Namespace: <http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration>

The **SPSharedServiceApplicationInfo** complex type contains information about a service application. It is defined by the namespace at <http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration>.

```
<xs:complexType name="SPSharedServiceApplicationInfo">
  <xs:sequence>
    <xs:element minOccurs="0" name="ApplicationClassId" type="ser:guid"/>
    <xs:element minOccurs="0" name="ApplicationVersion" nillable="true"
      xmlns:q1="http://schemas.datacontract.org/2004/07/System" type="q1:Version"/>
    <xs:element minOccurs="0" name="Comments" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="DisplayName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="TermsOfServiceUri" nillable="true" type="xs:anyURI"/>
    <xs:element minOccurs="0" name="Uri" nillable="true" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

ApplicationClassId: A GUID that identifies the service application class module. It MUST conform to the XSD of the **guid** simple type, as specified in section [2.2.5.1](#). This element MUST be present.

ApplicationVersion: The version of the service application class module. It MUST conform to the XML Schema (1) of the **Version** complex type, as specified in section [3.1.4.1.3.4](#). This element MUST be present.

Comments: A string that contains a description of the service application. This element MUST be present.

DisplayName: A string that specifies the display name of the service application. This element MUST be present.

TermsOfServiceUri: A URL that specifies a document containing additional information about the service application. This element MUST be present.

Uri: A URI that specifies the logical address of the service application. It MUST be either a hardware load-balancer address of the service or conform to the following **ABNF**. This element MUST be present.

```
logicaladdress = "urn:schemas-microsoft-
com:sharepoint:service:"serviceid"#authority=urn:uuid:"topologyserviceid"&authority="endpoint
uri
topologyserviceid = guid
serviceid = guid
guid = 32*hexdig
hexdig = %x30-39/"A"/"B"/"C"/"D"/"E"/"F"/"a"/"b"/"c"/"d"/"e"/"f"
```

endpointuri: The endpoint (4) URI of the service that is providing the implementation of this protocol.

topologyserviceid: A GUID that specifies the service that is providing the implementation of this protocol.

serviceid: A GUID that identifies the service application about which information is being returned.

3.1.4.1.3.4 Version

Namespace: http://schemas.datacontract.org/2004/07/System

The **Version** complex type is a structure that contains version information for a service application class module. It is defined by the namespace at http://schemas.datacontract.org/2004/07/System.

```
<xs:complexType name="Version">
  <xs:sequence>
    <xs:element name="_Build" type="xs:int"/>
    <xs:element name="_Major" type="xs:int"/>
    <xs:element name="_Minor" type="xs:int"/>
    <xs:element name="_Revision" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

_Build: An integer that specifies the **build number** of the class module. This value MUST be greater than or equal to 0.

_Major: An integer that specifies the **major version** number of the class module. This value MUST be greater than or equal to 0.

_Minor: An integer that specifies the **minor version** number of the class module. This value MUST be greater than or equal to 0.

_Revision: An integer that specifies the **revision** number of the class module. This value MUST be greater than or equal to 0.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 GetEndpoints

The **GetEndpoints** message returns a collection of all the endpoint (4) URIs for a service application that is specified by a GUID in an input message.

```
<wsdl:operation name="GetEndpoints">
```

```

    <wsdl:input wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints"
    message="tns:ITopologyWebServiceApplication_GetEndPoints_InputMessage"/>
    <wsdl:output
    wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsResponse"
    message="tns:ITopologyWebServiceApplication_GetEndPoints_OutputMessage"/>
    <wsdl:fault
    wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndPointsSPTopologyWebServi
    ceApplicationFaultFault" name="SPTopologyWebServiceApplicationFaultFault"
    message="tns:ITopologyWebServiceApplication_GetEndPoints_SPTopologyWebServiceApplicationFault
    Fault_FaultMessage"/>
  </wsdl:operation>

```

The protocol client sends a **ITopologyWebServiceApplication_GetEndPoints_InputMessage** request WSDL message and the protocol server MUST respond with a **ITopologyWebServiceApplication_GetEndPoints_OutputMessage** response WSDL message, as follows:

- If a service application with the specified GUID exists, the protocol server MUST return a collection of all the endpoint (4) URIs for the service application.
- If a service application with the specified GUID does not exist, the protocol server MUST return an **SPTopologyWebServiceApplicationFault** SOAP fault with the **FaultReason** indicating that the service application was not found on the server.

3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.1.1 ITopologyWebServiceApplication_GetEndPoints_InputMessage

The request WSDL message for the **GetEndPoints** WSDL operation.

The SOAP action value is:

```
http://tempuri.org/ITopologyWebServiceApplication/GetEndPoints
```

The SOAP body contains the **GetEndPoints** element.

3.1.4.2.1.2 ITopologyWebServiceApplication_GetEndPoints_OutputMessage

The response WSDL message for the **GetEndPoints** WSDL operation.

The SOAP body contains the **GetEndPointsResponse** element.

3.1.4.2.2 Elements

The following XML Schema element definitions are specific to this operation.

3.1.4.2.2.1 GetEndPoints

The input data for the **GetEndPoints** WSDL operation.

```

<xs:element name="GetEndPoints">
  <xs:complexType>
    <xs:sequence>

```

```

    <xs:element minOccurs="0" name="serviceId"
      xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/" type="q1:guid"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

serviceId: A GUID that identifies the service application. This value **MUST** be present and **MUST NOT** be empty.

3.1.4.2.2.2 GetEndpointsResponse

The result data for the **GetEndpoints** WSDL operation.

```

<xs:element name="GetEndpointsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetEndpointsResult" nillable="true"
        xmlns:q2="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
        type="q2:ReadOnlyCollectionOfanyURI"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

GetEndpointsResult: A collection of endpoint (4) URIs for the specified service application. It **MUST** conform to the XSD of the **ReadOnlyCollectionOfanyURI** complex type, as specified in section [3.1.4.2.3.1](#). This element **MUST** be present and **MUST NOT** be empty.

3.1.4.2.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

3.1.4.2.3.1 ReadOnlyCollectionOfanyURI

Namespace: <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>

The **ReadOnlyCollectionOfanyURI** complex type is a collection of endpoint (4) URIs for a service application. It is defined by the namespace at <http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel>.

```

<xs:complexType name="ReadOnlyCollectionOfanyURI">
  <xs:sequence>
    <xs:element name="list" nillable="true"
      xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      type="q1:ArrayOfanyURI"/>
  </xs:sequence>
</xs:complexType>

```

list: An array of endpoint (4) URIs for the service application. It **MUST** conform to the XSD of the **ArrayOfanyURI** complex type, as specified in section [3.1.4.2.3.2](#).

3.1.4.2.3.2 ArrayOfanyURI

Namespace: <http://schemas.microsoft.com/2003/10/Serialization/Arrays>

The **ArrayOfanyURI** complex type is an array of endpoint (4) URIs for a service application. It is defined by the namespace at <http://schemas.microsoft.com/2003/10/Serialization/Arrays>. If a service application does not have any endpoint (4) URIs, the **ArrayOfanyURI** element MUST be empty.

```
<xs:complexType name="ArrayOfanyURI">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="anyURI" nillable="true"
      type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

anyURI: A specific endpoint (4) URI for the service application.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 EnumerateSharedServiceApplications Operation

In this example, a protocol client requests a list of service applications that are on a protocol server and the protocol server responds to that request.

The protocol client constructs the following WSDL request message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplications xmlns="http://tempuri.org/">
      </EnumerateSharedServiceApplications>
    </s:Body>
  </s:Envelope>
```

There are three service applications on the protocol server. Therefore, the protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplicationsResponse xmlns="http://tempuri.org/">
      <EnumerateSharedServiceApplicationsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list
xmlns:c="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration">
          <c:SPSharedServiceApplicationInfo>
            <c:ApplicationClassId>e8479529-b61f-410a-a631-11e577975716</c:ApplicationClassId>
            <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
              <d:_Build>0</d:_Build>
              <d:_Major>1</d:_Major>
              <d:_Minor>0</d:_Minor>
              <d:_Revision>0</d:_Revision>
            </c:ApplicationVersion>
            <c:Comments>Service1 provides functionality1</c:Comments>
            <c:DisplayName>Service1App</c:DisplayName>
            <c:TermsOfServiceUri
i:nil="true">http://ServerA/Service1Help.html</c:TermsOfServiceUri>
            <c:Uri>urn:schemas-microsoft-com:sharepoint:service:deae7d4345744b6bc7f8370ff1c8801#authority=urn:uuid:e3f6969562ad47ea9122638e9ab488b7&amp;authority=https%3A%2F%2FServerA%3A32844%2FTopology%2FTopology.svc</c:Uri>
          </c:SPSharedServiceApplicationInfo>
          <c:SPSharedServiceApplicationInfo>
            <c:ApplicationClassId>8801a560-b15b-41e5-a661-elac85bdad9e</c:ApplicationClassId>
            <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
              <d:_Build>0</d:_Build>
              <d:_Major>1</d:_Major>
              <d:_Minor>0</d:_Minor>
              <d:_Revision>0</d:_Revision>
            </c:ApplicationVersion>
            <c:Comments>Service2 provides functionality2</c:Comments>
            <c:DisplayName>Service2App</c:DisplayName>
            <c:TermsOfServiceUri i:nil="true">http://ServerA/Service2Help.html
</c:TermsOfServiceUri>
```

```

    <c:Uri>urn:schemas-microsoft-com:sharepoint:service:
7b59f9413c534e6883d76eee5ee0125c#authority=urn:uuid:e3f6969562ad47ea9122638e9ab488b7&amp;auth
ority=https%3A%2F%2FServerA%3A32844%2FTopology%2FTopology.svc</c:Uri>
  </c:SPSharedServiceApplicationInfo>
  <c:SPSharedServiceApplicationInfo>
    <c:ApplicationClassId>9901a560-b15b-41e5-a661-e1ac85bdad9e</c:ApplicationClassId>
    <c:ApplicationVersion xmlns:d="http://schemas.datacontract.org/2004/07/System">
      <d:_Build>0</d:_Build>
      <d:_Major>1</d:_Major>
      <d:_Minor>0</d:_Minor>
      <d:_Revision>0</d:_Revision>
    </c:ApplicationVersion>
    <c:Comments>Service3 provides functionality3</c:Comments>
    <c:DisplayName>Service3App</c:DisplayName>
    <c:TermsOfServiceUri i:nil="true"> http://ServerA/Service3Help.html
  </c:TermsOfServiceUri>
  <c:Uri>urn:schemas-microsoft-
com:sharepoint:service:cc5de64c76a54b129fa7e35c5124be49#authority=urn:uuid:e3f6969562ad47ea91
22638e9ab488b7&amp;authority=https%3A%2F%2FServerA%3A32844%2FTopology%2FTopology.svc</c:Uri>
  </c:SPSharedServiceApplicationInfo>
</b:list>
</EnumerateSharedServiceApplicationsResult>
</EnumerateSharedServiceApplicationsResponse>
</s:Body>
</s:Envelope>

```

The protocol client then uses the data in the **ApplicationClassId** and **ApplicationVersion** elements in the response message to identify the service application with which it can communicate. The protocol client extracts the **serviceId** element from the **Uri** element for a service application and uses that information to retrieve a list of endpoint (4) URIs for the service application as demonstrated in section [4.2](#).

If there were zero service application on the protocol server, the protocol server would have responded with the following message:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <EnumerateSharedServiceApplicationsResponse xmlns="http://tempuri.org/">
      <EnumerateSharedServiceApplicationsResult
xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list
xmlns:c="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"></b:lis
t>
          </EnumerateSharedServiceApplicationsResult>
        </EnumerateSharedServiceApplicationsResponse>
      </s:Body>
    </s:Envelope>

```

4.2 GetEndpoints Operation

In this example, a protocol client retrieves all the endpoint (4) URIs for a service application on a protocol server. It does so by extracting the value of the **topologyserviceid** parameter, which is "e3f69695-62ad-47ea-9122-638e9ab488b7", from the **Uri** element of a service application that was returned in response to an **EnumerateSharedServiceApplications** message.

To retrieve all the endpoint (4) URIs for the service application with serviceid e3f69695-62ad-47ea-9122-638e9ab488b7 that implements this protocol, the protocol client constructs the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndpoints xmlns="http://tempuri.org/">
      <serviceId>e3f69695-62ad-47ea-9122-638e9ab488b7</serviceId>
    </GetEndpoints>
  </s:Body>
</s:Envelope>
```

A service application with the specified GUID exists. Therefore, the protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndpointsResponse xmlns="http://tempuri.org/">
      <GetEndpointsResult
        xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
          <c:anyURI>https://ServerA:32844/Topology/Topology.svc</c:anyURI>
        </b:list>
      </GetEndpointsResult>
    </GetEndpointsResponse>
  </s:Body>
</s:Envelope>
```

As described in section 4.1, the **EnumerateSharedServiceApplicationsResponse** message contains **ApplicationClassId** and **ApplicationVersion** elements that provide information about the service application. The protocol client uses the values in the **ApplicationClassId** and **ApplicationVersion** elements to determine whether it can communicate with the service application. The protocol client also extracts the **serviceId** value from the **Uri** element in the response. The protocol client uses the **serviceId** value to construct the following WSDL message and retrieve a read-only collection of all the endpoint (4) URIs for the service application:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndpoints xmlns="http://tempuri.org/">
      <serviceId>cc5de64c-76a5-4b12-9fa7-e35c5124be49</serviceId>
    </GetEndpoints>
  </s:Body>
</s:Envelope>
```

The service application with the specified GUID exists. Therefore, the protocol server responds with the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <GetEndpointsResponse xmlns="http://tempuri.org/">
      <GetEndpointsResult
        xmlns:b="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:list xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
```

```
<c:anyURI>http://ServerA:32844/cc5de64c76a54b129fa7e35c5124be49/Service1.svc</c:anyURI>
<c:anyURI>https://ServerA:32844/cc5de64c76a54b129fa7e35c5124be49/Service1.svc</c:anyURI>
  </b:list>
  </GetEndpointsResult>
</GetEndpointsResponse>
</s:Body>
</s:Envelope>
```

If the service application with the specified GUID did not exist, the protocol server would send the following response instead:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Body>
    <s:Fault>
      <s:Code>
        <s:Value>s:Sender</s:Value>
      </s:Code>
      <s:Reason>
        <s:Text xml:lang="en-US">The creator of this fault did not specify a Reason.</s:Text>
      </s:Reason>
      <s:Detail>
        <SPTopologyWebServiceApplicationFault
          xmlns="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
          xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <FaultReason>The requested application could not be found.</FaultReason>
        </SPTopologyWebServiceApplicationFault>
      </s:Detail>
    </s:Fault>
  </s:Body>
</s:Envelope>
```


5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: Full WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:tns="http://tempuri.org/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsm="http://www.w3.org/2007/05/addressing/metadata"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SPTopologyWebServiceApplication"
targetNamespace="http://tempuri.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel" />
      <xs:element name="GetEndpoints">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/"
minOccurs="0" name="serviceId" type="q1:guid" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GetEndpointsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element
xmlns:q2="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
minOccurs="0" name="GetEndpointsResult" nillable="true" type="q2:ReadOnlyCollectionOfanyURI"
/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="EnumerateSharedServiceApplications">
        <xs:complexType>
          <xs:sequence />
        </xs:complexType>
      </xs:element>
      <xs:element name="EnumerateSharedServiceApplicationsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element
xmlns:q3="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
minOccurs="0" name="EnumerateSharedServiceApplicationsResult" nillable="true"
type="q3:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="anyType" nillable="true" type="xs:anyType" />
      <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
      <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
      <xs:element name="boolean" nillable="true" type="xs:boolean" />
      <xs:element name="byte" nillable="true" type="xs:byte" />
      <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
      <xs:element name="decimal" nillable="true" type="xs:decimal" />
      <xs:element name="double" nillable="true" type="xs:double" />

```

```

<xs:element name="float" nillable="true" type="xs:float" />
<xs:element name="int" nillable="true" type="xs:int" />
<xs:element name="long" nillable="true" type="xs:long" />
<xs:element name="QName" nillable="true" type="xs:QName" />
<xs:element name="short" nillable="true" type="xs:short" />
<xs:element name="string" nillable="true" type="xs:string" />
<xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
<xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
<xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
<xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
<xs:element name="char" nillable="true" type="tns:char" />
<xs:simpleType name="char">
  <xs:restriction base="xs:int" />
</xs:simpleType>
<xs:element name="duration" nillable="true" type="tns:duration" />
<xs:simpleType name="duration">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?" />
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="guid" nillable="true" type="tns:guid" />
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
  </xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName" />
<xs:attribute name="Id" type="xs:ID" />
<xs:attribute name="Ref" type="xs:IDREF" />
</xs:schema>
<xs:schema
xmlns:tns="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Collections.ObjectModel"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration" />
  <xs:complexType name="ReadOnlyCollectionOfanyURI">
    <xs:sequence>
      <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
name="list" nillable="true" type="q1:ArrayOfanyURI" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ReadOnlyCollectionOfanyURI" nillable="true"
type="tns:ReadOnlyCollectionOfanyURI" />
  <xs:complexType name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg">
    <xs:sequence>
      <xs:element
xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
name="list" nillable="true" type="q2:ArrayOfSPSharedServiceApplicationInfo" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg"
nillable="true" type="tns:ReadOnlyCollectionOfSPSharedServiceApplicationInfo1RC_ScQDg" />
</xs:schema>

```

```

    <xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    elementFormDefault="qualified"
    targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:complexType name="ArrayOfanyURI">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="anyURI" nillable="true"
type="xs:anyURI" />
        </xs:sequence>
      </xs:complexType>
      <xs:element name="ArrayOfanyURI" nillable="true" type="tns:ArrayOfanyURI" />
    </xs:schema>
  </xs:schema
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint.Administration"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System" />
  <xs:complexType name="ArrayOfSPSharedServiceApplicationInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="SPSharedServiceApplicationInfo" nillable="true"
type="tns:SPSharedServiceApplicationInfo" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfSPSharedServiceApplicationInfo" nillable="true"
type="tns:ArrayOfSPSharedServiceApplicationInfo" />
  <xs:complexType name="SPSharedServiceApplicationInfo">
    <xs:sequence>
      <xs:element minOccurs="0" name="ApplicationClassId" type="ser:guid" />
      <xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/System" minOccurs="0"
name="ApplicationVersion" nillable="true" type="q1:Version" />
      <xs:element minOccurs="0" name="Comments" nillable="true" type="xs:string" />
      <xs:element minOccurs="0" name="DisplayName" nillable="true" type="xs:string" />
      <xs:element minOccurs="0" name="TermsOfServiceUri" nillable="true" type="xs:anyURI"
/ >
    </xs:sequence>
  </xs:complexType>
  <xs:element minOccurs="0" name="Uri" nillable="true" type="xs:anyURI" />
</xs:sequence>
</xs:complexType>
<xs:element name="SPSharedServiceApplicationInfo" nillable="true"
type="tns:SPSharedServiceApplicationInfo" />
</xs:schema>
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/System"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Version">
    <xs:sequence>
      <xs:element name="_Build" type="xs:int" />
      <xs:element name="_Major" type="xs:int" />
      <xs:element name="_Minor" type="xs:int" />
      <xs:element name="_Revision" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Version" nillable="true" type="tns:Version" />
</xs:schema>
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
elementFormDefault="qualified"

```

```

targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SPTopologyWebServiceApplicationFault">
    <xs:sequence>
      <xs:element minOccurs="0" name="FaultReason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SPTopologyWebServiceApplicationFault" nillable="true"
type="tns:SPTopologyWebServiceApplicationFault" />
</xs:schema>
</wsdl:types>
<wsdl:message name="ITopologyWebServiceApplication_GetEndpoints_InputMessage">
  <wsdl:part name="parameters" element="tns:GetEndpoints" />
</wsdl:message>
<wsdl:message name="ITopologyWebServiceApplication_GetEndpoints_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetEndpointsResponse" />
</wsdl:message>
<wsdl:message
name="ITopologyWebServiceApplication_GetEndpoints_SPTopologyWebServiceApplicationFaultFault_F
aultMessage">
  <wsdl:part xmlns:q1="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
name="detail" element="q1:SPTopologyWebServiceApplicationFault" />
</wsdl:message>
<wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage">
  <wsdl:part name="parameters" element="tns:EnumerateSharedServiceApplications" />
</wsdl:message>
<wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage">
  <wsdl:part name="parameters" element="tns:EnumerateSharedServiceApplicationsResponse" />
</wsdl:message>
<wsdl:message
name="ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebServiceA
pplicationFaultFault_FaultMessage">
  <wsdl:part xmlns:q2="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
name="detail" element="q2:SPTopologyWebServiceApplicationFault" />
</wsdl:message>
<wsdl:portType name="ITopologyWebServiceApplication">
  <wsdl:operation name="GetEndpoints">
    <wsdl:input
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndpoints"
message="tns:ITopologyWebServiceApplication_GetEndpoints_InputMessage" />
    <wsdl:output
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndpointsResponse"
message="tns:ITopologyWebServiceApplication_GetEndpoints_OutputMessage" />
    <wsdl:fault
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/GetEndpointsSPTopologyWebServi
ceApplicationFaultFault" name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_GetEndpoints_SPTopologyWebServiceApplicationFault
Fault_FaultMessage" />
  </wsdl:operation>
  <wsdl:operation name="EnumerateSharedServiceApplications">
    <wsdl:input
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ions"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_InputMessage"
/>
    <wsdl:output
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsResponse"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_OutputMessage"
/>

```

```

        <wsdl:fault
wsam:Action="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicat
ionsSPTopologyWebServiceApplicationFaultFault"
name="SPTopologyWebServiceApplicationFaultFault"
message="tns:ITopologyWebServiceApplication_EnumerateSharedServiceApplications_SPTopologyWebS
erviceApplicationFaultFault_FaultMessage" />
    </wsdl:operation>
</wsdl:portType>
    <wsdl:binding name="CustomBinding_ITopologyWebServiceApplication"
type="tns:ITopologyWebServiceApplication">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="GetEndpoints">
            <soap:operation
soapAction="http://tempuri.org/ITopologyWebServiceApplication/GetEndpoints" style="document"
/>
                <wsdl:input>
                    <soap:body use="literal" />
                </wsdl:input>
                <wsdl:output>
                    <soap:body use="literal" />
                </wsdl:output>
                <wsdl:fault name="SPTopologyWebServiceApplicationFaultFault">
                    <soap:fault use="literal" name="SPTopologyWebServiceApplicationFaultFault"
namespace="" />
                </wsdl:fault>
            </wsdl:operation>
            <wsdl:operation name="EnumerateSharedServiceApplications">
                <soap:operation
soapAction="http://tempuri.org/ITopologyWebServiceApplication/EnumerateSharedServiceApplicati
ons" style="document" />
                    <wsdl:input>
                        <soap:body use="literal" />
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal" />
                    </wsdl:output>
                    <wsdl:fault name="SPTopologyWebServiceApplicationFaultFault">
                        <soap:fault use="literal" name="SPTopologyWebServiceApplicationFaultFault"
namespace="" />
                    </wsdl:fault>
                </wsdl:operation>
            </wsdl:binding>
        </wsdl:definitions>

```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SharePoint® Foundation 2010
- Microsoft® SharePoint® Foundation 15 Technical Preview

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

8 Change Tracking

This section identifies changes that were made to the [MS-SPTWS] protocol document between the June 2011 and January 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
Z Appendix B: Product Behavior	Updated the product list.	Y	Content updated.

9 Index

A

Abstract data model
[server](#) 12
[Applicability](#) 7
[Attribute groups](#) 11
[Attributes](#) 11

C

[Capability negotiation](#) 7
[Change tracking](#) 32
Client
[overview](#) 12
[Complex types](#) 10
[SPTopologyWebServiceApplicationFault](#) 10

D

Data model - abstract
[server](#) 12

E

[EnumerateSharedServiceApplications example](#) 21
Events
[local - server](#) 20
[timer - server](#) 20
Examples
[EnumerateSharedServiceApplications](#) 21
[GetEndPoints](#) 22

F

[Fields - vendor-extensible](#) 8
[Full WSDL](#) 26

G

[GetEndPoints example](#) 22
[Glossary](#) 5
[Groups](#) 11
[guid simple type](#) 11

I

[Implementer - security considerations](#) 25
[Index of security parameters](#) 25
[Informative references](#) 6
Initialization
[server](#) 13
[Introduction](#) 5

L

Local events
[server](#) 20

M

Message processing
[server](#) 13
Messages
[attribute groups](#) 11
[attributes](#) 11
[complex types](#) 10
[elements](#) 10
[enumerated](#) 10
[groups](#) 11
[guid simple type](#) 11
[namespaces](#) 9
[simple types](#) 10
[SPTopologyWebServiceApplicationFault complex type](#) 10
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9
[Normative references](#) 6

O

Operations
[EnumerateSharedServiceApplications](#) 13
[GetEndPoints](#) 17
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 25
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 31

R

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

S

Security
[implementer considerations](#) 25
[parameter index](#) 25
Sequencing rules
[server](#) 13
Server
[abstract data model](#) 12
[EnumerateSharedServiceApplications operation](#) 13
[GetEndPoints operation](#) 17

[initialization](#) 13
[local events](#) 20
[message processing](#) 13
[overview](#) 12
[sequencing rules](#) 13
[timer events](#) 20
[timers](#) 13
[Simple types](#) 10
 [guid](#) 11
[SPTopologyWebServiceApplicationFault complex type](#) 10
[Standards assignments](#) 8
Syntax
 [messages - overview](#) 9

T

Timer events
 [server](#) 20
Timers
 [server](#) 13
[Tracking changes](#) 32
[Transport](#) 9
Types
 [complex](#) 10
 [simple](#) 10

V

[Vendor-extensible fields](#) 8
[Versioning](#) 7

W

[WSDL](#) 26