

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
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	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
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
2.2.5 Simple Types	10
2.2.6 Attributes	10
2.2.7 Groups	10
2.2.8 Attribute Groups	10
3 Protocol Details	11
3.1 Server Details	11
3.1.1 Abstract Data Model	11
3.1.2 Timers	11
3.1.3 Initialization	12
3.1.4 Message Processing Events and Sequencing Rules	12
3.1.4.1 SendClientScriptErrorReport	12
3.1.4.1.1 Messages	12
3.1.4.1.1.1 SendClientScriptErrorReportSoapIn	12
3.1.4.1.1.2 SendClientScriptErrorReportSoapOut	12
3.1.4.1.2 Elements	13
3.1.4.1.2.1 SendClientScriptErrorReport	13
3.1.4.1.2.2 SendClientScriptErrorReportResponse	13
3.1.5 Timer Events	14
3.1.6 Other Local Events	14
4 Protocol Examples	15
5 Security	16
5.1 Security Considerations for Implementers	16
5.2 Index of Security Parameters	16
6 Appendix A: Full WSDL	17
7 Appendix B: Product Behavior	19

8 Change Tracking.....	20
9 Index	21

1 Introduction

This document specifies the SharePoint Diagnostics Web Service Protocol. This protocol enables a protocol client to submit diagnostic reports describing application errors that occur on the protocol client.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)

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

endpoint
SOAP (Simple Object Access Protocol)
SOAP action
SOAP body
SOAP fault
URI (Uniform Resource Identifier)
URL (Uniform Resource Locator)
WSDL message
WSDL operation
XML fragment
XML namespace
XML namespace prefix
XML Schema

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

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](#)", July 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

[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>

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

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, <http://www.w3.org/TR/REC-xml-names/>

[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/>

[XPath] Clark, J. and DeRose, S., "XML Path Language (XPath), Version 1.0", W3C Recommendation, November 1999, <http://www.w3.org/TR/xpath>

1.2.2 Informative References

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

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)", June 2008.

[MS-SPTWS] Microsoft Corporation, "[Service Platform Topology Web Service Protocol Specification](#)", July 2009.

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

1.3 Protocol Overview (Synopsis)

In many modern web pages, there is a large amount of code (for example, JavaScript) running in client web browser. To help diagnose common errors encountered with the web pages mentioned, it is desirable that the developers of the pages can get detailed information regarding these errors.

This protocol defines an operation that allows a protocol client to submit details about an error report (for example, call stack, error message, or operating environment). The developers can use the submitted error reports to discover and fix errors encountered by the users.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [\[SOAP1.2/1\]](#) and [\[SOAP1.2/2\]](#). It transmits those messages by using **HTTP**, as described in [\[RFC2616\]](#), or **HTTPS**, as described in [\[RFC2818\]](#).

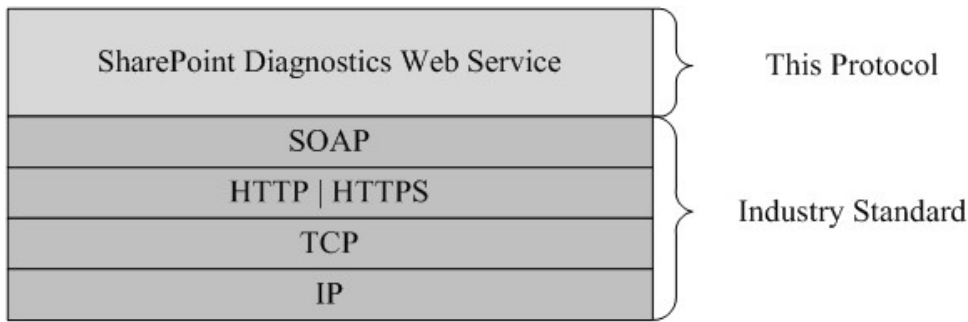


Figure 1: This protocol in relation to other protocols

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 endpoint (4) URI of the protocol server and the transport that is used by the protocol server are either known by the protocol client or obtained by using the discovery mechanism that is described in [\[MS-SPTWS\]](#).

The protocol client obtains the requisite **ApplicationClassId** and **ApplicationVersion** values and the endpoint (4) URI of the protocol server that provides the discovery mechanism, as described in [\[MS-SPTWS\]](#), by means that are independent of either protocol.

This protocol requires the protocol client to have appropriate permission to call the methods on the protocol server.

The protocol client implements the token-based security mechanisms that are required by the protocol server and related security protocols, as described in [\[MS-SPSTWS\]](#).

1.6 Applicability Statement

This protocol is intended to transfer small amounts of data (less than 6 kilobytes) from a protocol client to a protocol server. Therefore, the protocol client is expected to gather and format relevant information (such as the call stack) in an **XML fragment**.

This protocol is not intended to transfer large regions of memory or other comprehensive error data collection from a protocol client.

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 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.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP or HTTPS.

All protocol messages MUST be transported by using HTTP bindings at the transport level.

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.

If the HTTPS transport is used, a server certificate MUST be deployed.

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. In addition, the protocol client MUST know the format and transport that is used by the protocol server, for example, the SOAP format over an HTTP transport.

2.2 Common Message Syntax

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

2.2.1 Namespaces

This protocol specifies and references **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this document associates an **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
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/sharepoint/diagnostics/	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
(none)	http://schemas.microsoft.com/sharepoint/diagnostics/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
mime	http://schemas.xmlsoap.org/wsdl/mime/	[WSDL]
http	http://schemas.xmlsoap.org/wsdl/http/	[RFC2616]
soapenc	http://schemas.xmlsoap.org/soap/encoding/	[SOAP1.2/1] [SOAP1.2/2]

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

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

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.

Except where specified, protocol clients SHOULD interpret HTTP status codes returned by the protocol server as specified in [\[RFC2616\]](#) (Section 10, Status Code Definitions).

This protocol allows protocol servers to notify protocol clients of application-level faults using SOAP faults. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either using HTTP status codes or using SOAP faults as specified previously in this section.

3.1 Server Details

The following diagram describes the communication between the protocol client and the protocol server:

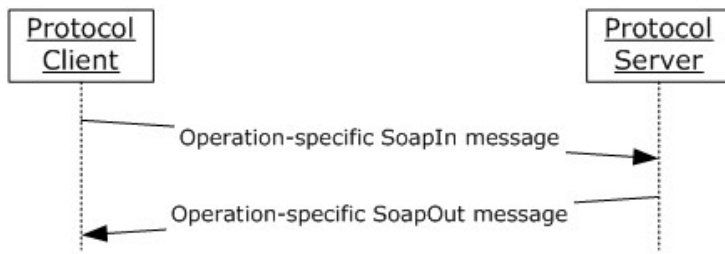


Figure 2: Message exchange between client and server

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.

This protocol does not dictate any specific information required in the error report. If available, the error report data SHOULD include information about the client operating environment (such as web browser name, browser version, and protocol client language). The error report data SHOULD include information about the error (message, **URL**, line number, and call stack). The error report SHOULD include information about the origination of the error (application name, file name). The error report is specified in section [3.1.4.1](#).

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
SendClientScriptErrorReport	This protocol is used to submit error reports originating from the protocol client to the protocol server.

3.1.4.1 SendClientScriptErrorReport

This protocol is used to submit error reports originating from the protocol client to the protocol server.

```
<wsdl:operation name="SendClientScriptErrorReport">
  <wsdl:input message="tns:SendClientScriptErrorReportSoapIn"/>
  <wsdl:output message="tns:SendClientScriptErrorReportSoapOut"/>
</wsdl:operation>
```

The protocol client sends a [SendClientScriptErrorReportSoapIn](#) request **WSDL message** and the protocol server responds with a [SendClientScriptErrorReportSoapOut](#) response WSDL message.

3.1.4.1.1 Messages

3.1.4.1.1.1 SendClientScriptErrorReportSoapIn

The requested WSDL message for a **SendClientScriptErrorReport** WSDL operation.

The **SOAP action** value is:

```
http://schemas.microsoft.com/sharepoint/diagnostics/SendClientScriptErrorReport
```

The **SOAP body** contains a **SendClientScriptErrorReport** element.

3.1.4.1.1.2 SendClientScriptErrorReportSoapOut

The response WSDL message for a **SendClientScriptErrorReport** method.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/diagnostics/SendClientScriptErrorReport
```

The SOAP body contains a **SendClientScriptErrorReportResponse** element.

3.1.4.1.2 Elements

3.1.4.1.2.1 SendClientScriptErrorReport

The input data for a **SendClientScriptErrorReport** WSDL operation.

```
<xs:element name="SendClientScriptErrorReport">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="file" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="line" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="client" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="stack" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="team" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="originalFile" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

message: The message associated with the current error.

file: The URL file name associated with the current error.

line: The line number associated with the current error.

client: A string argument representing the protocol client operating environment. The string MUST be a valid XML fragment. The protocol looks specifically for the following nodes (expressed using [\[XPath\]](#) notation): `client/browser/@name`, `client/browser/@version`, and `client/language`. Other nodes in the XML fragment MUST be ignored.

stack: A string argument representing the call stack of the error. The string MUST be a valid XML fragment. The protocol looks specifically for the following node (expressed using [\[XPath\]](#) notation): `stack/function[@depth="0"]/@signature`. Other nodes in the XML fragment MUST be ignored.

team: The application associated with the current error.

originalFile: The physical file name associated with the current error.

3.1.4.1.2.2 SendClientScriptErrorReportResponse

The result data for a **SendClientScriptErrorReport** WSDL operation.

```
<xs:element name="SendClientScriptErrorReportResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="SendClientScriptErrorReportResult"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

SendClientScriptErrorReportResult: Implementation specific result. The protocol client MUST NOT rely on this data to follow any particular format.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

To submit an error report to the server, the protocol client constructs the following SOAP message:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendClientScriptErrorReport
xmlns="http://schemas.microsoft.com/sharepoint/diagnostics/">
      <message>'null'%20is%20null%20or%20not%20an%20object</message>
      <file>init.debug.js</file>
      <line>407</line>

<stack>%3Cstack%3E%0A%3Cfunction%20depth%3D%22%22%20signature%3D%22CancelEvent (e) %22%3E%0A%3
C!%5BCDATA%5Bfunction%20CancelEvent (e) %0D%0A%7BULSxSy%3A%3B%0D%0A%20%20e.cancelBubble%3Dtrue%
3B%0D%0A%20%20if (e.preventDefault) %20e.preventDefault () %3B%0D%0A%20%20if (e.stopPropogation) %2
0e.stopPropogation () %3B%0D%0A%20%20e.returnValue%3Dfalse%3B%0D%0A%20%20return%20false%3B%0D%0
A%7D%5D%5D%3E%0A%0A%3Cargument%20name%3D%22e%22%20type%3D%22object%22%3E%7Bunde%7D%3C%2F
argument%3E%0A%3C%2Ffunction%3E%0A%3C%2Fstack%3E</stack>

<client>%3Cclient%3E%0A%3Cbrowser%20name%3D%22Microsoft%20Internet%20Explorer%22%20version%3D
%227%22%20%2F%3E%0A%3Cuseragent%3EMozilla%2F4.0%20 (compatible%3B%20MSIE%207.0%3B%20Windows%20
NT%206.0%3B%20Trident%2F4.0%3B%20SLCC1%3B%20.NET%20CLR%202.0.50727%3B%20.NET%20CLR%201.1.4322
%3B%20.NET%20CLR%203.5.30729%3B%20.NET%20CLR%203.0.30618%3B%20MS-
RTC%20LM%208%3B%20InfoPath.2) %3C%2Fuseragent%3E%0A%3Clanguage%3Een-
us%3C%2Flanguage%3E%0A%3Clocation%3Ehttp%3A%2F%2Fphillip-
a%2FSitePages%2FHome.aspx%3C%2Flocation%3E%0A%3C%2Fclient%3E</client>
    <team>Example</team>
    <originalFile>init.debug.js</originalFile>
  </SendClientScriptErrorReport>
</soap:Body>
</soap:Envelope>
```

The protocol server then responds with the following:

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><SendClientScriptErrorReportResponse
xmlns="http://schemas.microsoft.com/sharepoint/diagnostics/"><SendClientScriptErrorReportResu
lt>Example
14.0.4020
407
'null' is null or not an object
Microsoft Internet Explorer
7
init.debug.js
CancelEvent</SendClientScriptErrorReportResult></SendClientScriptErrorReportResponse></soap:B
ody></soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

This protocol introduces no additional security considerations beyond those applicable to its underlying protocols.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in the following:

```
<?xml version="1.0"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/diagnostics/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/diagnostics/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">SharePoint Diagnostics
  Web Service</wsdl:documentation>
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/diagnostics/">
      <xs:element name="SendClientScriptErrorReport">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="message" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="file" type="xs:string" />
            <xs:element minOccurs="1" maxOccurs="1" name="line" type="xs:int" />
            <xs:element minOccurs="0" maxOccurs="1" name="client" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="stack" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="team" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="originalFile" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SendClientScriptErrorReportResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="SendClientScriptErrorReportResult"
type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="SendClientScriptErrorReportSoapIn">
    <wsdl:part name="parameters" element="tns:SendClientScriptErrorReport" />
  </wsdl:message>
  <wsdl:message name="SendClientScriptErrorReportSoapOut">
    <wsdl:part name="parameters" element="tns:SendClientScriptErrorReportResponse" />
  </wsdl:message>
  <wsdl:portType name="SharePointDiagnosticsSoap">
    <wsdl:operation name="SendClientScriptErrorReport">
      <wsdl:input message="tns:SendClientScriptErrorReportSoapIn" />
      <wsdl:output message="tns:SendClientScriptErrorReportSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SharePointDiagnosticsSoap" type="tns:SharePointDiagnosticsSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="SendClientScriptErrorReport">
```

```
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/diagnostics/SendClientScriptErrorReport"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="SharePointDiagnosticsSoap12" type="tns:SharePointDiagnosticsSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="SendClientScriptErrorReport">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/diagnostics/SendClientScriptErrorReport"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </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

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

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

A

Abstract data model
 [server](#) 11
[Applicability](#) 7
[Attribute groups](#) 10
[Attributes](#) 10

C

[Capability negotiation](#) 7
[Change tracking](#) 20
Client
 [overview](#) 11
[Complex types](#) 10

D

Data model - abstract
 [server](#) 11

E

Events
 [local - server](#) 14
 [timer - server](#) 14
Examples
 [overview](#) 15

F

[Fields - vendor-extensible](#) 7
[Full WSDL](#) 17

G

[Glossary](#) 5
[Groups](#) 10

I

[Implementer - security considerations](#) 16
[Index of security parameters](#) 16
[Informative references](#) 6
Initialization
 [server](#) 12
[Introduction](#) 5

L

Local events
 [server](#) 14

M

Message processing
 [server](#) 12
Messages
 [attribute groups](#) 10

[attributes](#) 10
[complex types](#) 10
[elements](#) 10
[enumerated](#) 10
[groups](#) 10
[namespaces](#) 9
[simple types](#) 10
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9
[Normative references](#) 5

O

Operations
 [SendClientScriptErrorReport](#) 12
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 16
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 19

R

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

S

Security
 [implementer considerations](#) 16
 [parameter index](#) 16
Sequencing rules
 [server](#) 12
Server
 [abstract data model](#) 11
 [initialization](#) 12
 [local events](#) 14
 [message processing](#) 12
 [overview](#) 11
 [SendClientScriptErrorReport operation](#) 12
 [sequencing rules](#) 12
 [timer events](#) 14
 [timers](#) 11
 [Server details](#) 11
 [Simple types](#) 10
 [Standards assignments](#) 8
Syntax
 [messages - overview](#) 9

T

Timer events
 [server](#) 14
Timers
 [server](#) 11
 [Tracking changes](#) 20
 [Transport](#) 9
Types
 [complex](#) 10
 [simple](#) 10

V

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

W

[WSDL](#) 17