[MS-ACMAUDWS]:

Audit and Control Management Server Audit Web Service Protocol

---

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release 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 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
7/16/2012	0.1	New	Released new document.
9/12/2012	0.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	0.2	Major	Significantly changed the technical content.
2/11/2013	1.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.0	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.1	Minor	Clarified the meaning of the technical content.
2/10/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.2	Minor	Clarified the meaning of the technical content.
8/24/2015	2.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Audit and Control Management Server Audit Web Service Protocol enables communication between a protocol server that handles audit control features and a protocol client that consumes the web service to implement audit control features of the document control system.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

session: A representation of application data in system memory. It is used to maintain state for application data that is being manipulated or monitored on a protocol server by a user.

session identifier: A key that enables an application to make reference to a session.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

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

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP envelope: A container for SOAP message information and the root element of a **SOAP** document. See [SOAP1.2-1/2007] section 5.1 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [SOAP1.2-1/2007] section 5.4 for more information.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint.

Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

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

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.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

[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] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

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

1.2.2 Informative References

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

[MS-SPTWS] Microsoft Corporation, "Service Platform Topology Web Service Protocol".

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

[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

1.3 Overview

This protocol is used to communicate between a protocol server that provides functionality for managing documents and a protocol client. The communication is always initiated by the protocol client through a variety of applications.

1.3.1 Session-based Protocol

The protocol is **session** based and uses a unique **session identifier**. This session identifier is generated when a new session is created and is used thereafter to refer to that particular session.

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 **HTTP**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

This relationship is shown in the following diagram:

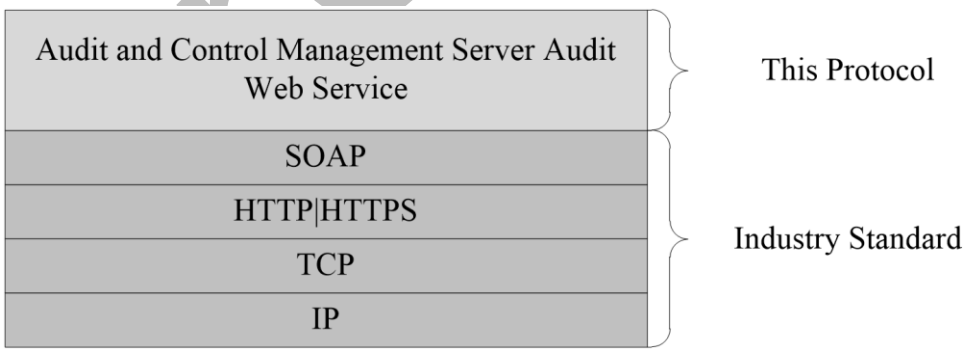

Figure 1: This protocol in relationship to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more **endpoint URIs** that are known by protocol clients. The endpoint 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 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 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 designed communicate between a protocol server and a protocol client to transfer metadata from the server to the 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 Simple Object Access Protocol (SOAP) messages, as described in section 2.1.
- **Protocol Versions**: This protocol is not versioned.

Capability Negotiation: This protocol does not support version negotiation.

An implementation does not conform to this specification if it fails to satisfy one or more of the MUST requirements that are defined herein. A SOAP node cannot use elements and attributes of the declared **XML namespace**, which is identified on the title page of this specification, within **SOAP envelopes**, unless it conforms to all MUST requirements of the specification. If a SOAP node does not conform to all of the MUST requirements that are defined in this specification, the behavior of the receiver of the SOAP node is undefined and causes unpredictable results.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

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**, **not null**, and **present**.

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.

2.2 Common Message Syntax

This section contains common structures used by this protocol. The syntax of the structures uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language (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
q1	http://schemas.datacontract.org/2004/07/System	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://prodiance.com/compliance	
tns1	http://schemas.microsoft.com/2003/10/Serialization/	
tns2	http://prodiance.com/compliance/Imports	
wsaw	http://www.w3.org/2006/05/addressing/wsdl	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

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 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.

Except where specified, protocol clients SHOULD interpret HTTP status codes returned by the protocol server as specified in [RFC2616], section 10.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults as specified previously in this section.

This protocol allows protocol servers to perform implementation-specific localization of text in various messages. Except where specified, the localization of this text is an implementation-specific behavior of the protocol server and not significant for interoperability.

The following high-level sequence diagram illustrates the operation of the protocol.

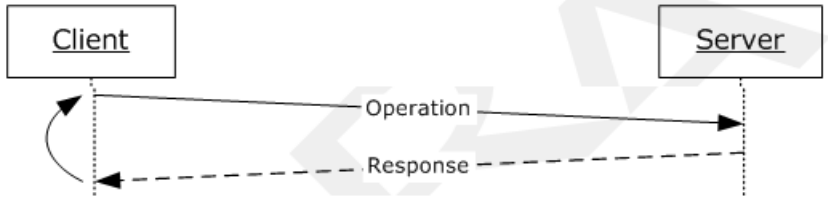

Figure 2: Communication sequence diagram - a typical client and server communication

3.1 Server Details

3.1.1 Abstract Data Model

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

The protocol server maintains auditing history records. Web service requests provide methods for getting old and new values for audit records.

Request: An entity which represents a running request on the protocol server.

3.1.2 Timers

3.1.2.1 Request Timeout

The **Request Timeout** timer measures the time it takes for a request to time out.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of operations as defined by this specification.

Operation	Description
GetAuditCompareValues	Specifies a web service call that retrieves old and new values for a particular audit trail row in the database. The call returns an instance of a **ComplianceAuditValueData** object containing the old and new values of the audit trail row.

3.1.4.1 GetAuditCompareValues

Specifies a web service call that retrieves old and new values for a particular audit trail row in the database. The call returns an instance of a **ComplianceAuditValueData** object containing the old and new values of the audit trail row.

The following is the WSDL port type specification of the **GetAuditCompareValues** **WSDL operation**.

```
<wsdl:operation name="GetAuditCompareValues" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValues"
message="tns:IComplianceAudit_GetAuditCompareValues_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValuesResponse"
message="tns:IComplianceAudit_GetAuditCompareValues_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValuesInvalidOpe
rationExceptionFault" name="InvalidOperationExceptionFault"
message="tns:IComplianceAudit_GetAuditCompareValues_InvalidOperationExceptionFault_FaultMessa
ge" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
IComplianceAudit_GetAuditCompareValues_InputMessage	The request WSDL message for the **GetAuditCompareValues** WSDL operation.
IComplianceAudit_GetAuditCompareValues_OutputMessage	The response WSDL message for the **GetAuditCompareValues** WSDL operation.
IComplianceAudit_GetAuditCompareValues_InvalidOperationExceptionFault_FaultMessage	The fault WSDL message for the

Message	Description
	GetAuditCompareValues WSDL operation.

3.1.4.1.1.1 IComplianceAudit_GetAuditCompareValues_InputMessage

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

The **SOAP action** value is:

```
http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValues
```

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

3.1.4.1.1.2 IComplianceAudit_GetAuditCompareValues_OutputMessage

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

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

3.1.4.1.1.3 IComplianceAudit_GetAuditCompareValues_InvalidOperationExceptionFault_FaultMessage

The fault WSDL message for the **GetAuditCompareValues** WSDL operation.

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

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetAuditCompareValues	The input data for the **GetAuditCompareValues** WSDL operation.
GetAuditCompareValuesResponse	The result data for the **GetAuditCompareValues** WSDL operation.
InvalidOperationException	The fault data for the **GetAuditCompareValues** WSDL operation.

3.1.4.1.2.1 GetAuditCompareValues

The **GetAuditCompareValues** element specifies the input data for the **GetAuditCompareValues** WSDL operation.

```
<xs:element name="GetAuditCompareValues" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="repositoryDocumentId" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" name="id" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

repositoryDocumentId: Unique identifier of the file in the content repository.

id: Identifier of an audit trail row in the database.

3.1.4.1.2.2 GetAuditCompareValuesResponse

The **GetAuditCompareValuesResponse** element specifies the result data for the **GetAuditCompareValues** WSDL operation.

```
<xs:element name="GetAuditCompareValuesResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetAuditCompareValuesResult" nillable="true"
type="tns:ComplianceAuditValueData"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetAuditCompareValuesResult: Returns an instance of a **ComplianceAuditValueData** object containing the old and new values of the audit trail row.

3.1.4.1.2.3 InvalidOperationException

The **InvalidOperationException** element specifies the fault data for the **GetAuditCompareValues** WSDL operation.

```
<xs:element name="InvalidOperationException" nillable="true"
type="q1:InvalidOperationException" xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.1.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
ComplianceAuditValueData	Specifies the old and new values available from a difference comparison.
Exception	Specifies any error that occurs during processing.
InvalidOperationException	Thrown when any error other than **SystemException** occurs during processing.
SystemException	Thrown when any error of the system type occurs. Those are predefined exceptions in the System namespace.

3.1.4.1.3.1 ComplianceAuditValueData

Namespace: http://prodiance.com/compliance

Specifies the old and new values available from a difference comparison.

```
<xs:complexType name="ComplianceAuditValueData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="NewValue" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="OldValue" nillable="true" type="xs:string"/>
  </xs:sequence>
```

```
</xs:complexType>
```

NewValue: Value from latest change.

OldValue: Value prior to latest change.

3.1.4.1.3.2 InvalidOperationException

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

Thrown when any error other than **SystemException** occurs during processing.

```
<xs:complexType name="InvalidOperationException" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q1:SystemException"/>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.1.3.3 SystemException

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

Thrown when any error of the system type occurs.

```
<xs:complexType name="SystemException" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="q1:Exception"/>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.1.3.4 Exception

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

Specifies any error that occurs during processing.

```
<xs:complexType name="Exception" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##local" processContents="skip"/>
  </xs:sequence>
  <xs:attribute ref="tns1:FactoryType"/>
</xs:complexType>
```

tns1:FactoryType: An anonymous complex type that contains an element that describes the factory object.

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.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Get the old and new values from the database

This example shows how this protocol is used to perform an approval. It demonstrates how this protocol is used to approve an item from the inbox or reject it. This example assumes that the repository document identifier is available.

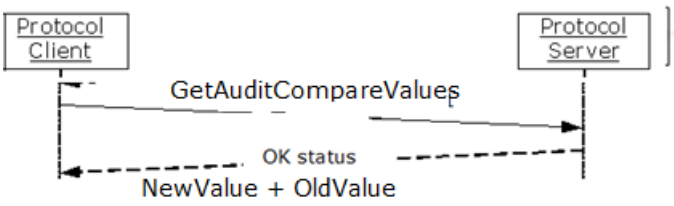

Figure 3: Communication sequence diagram for getting old and new values for comparison

Getting compare values typically involves the following steps, using **GetAuditCompareValues** (section 3.1.4.1):

1. Sending the **repositoryDocumentId**, the unique identifier of the file in the content repository and the **id**, the identifier of an audit trail row in the database.
2. Retrieving the old value and the new values of the audit trail row in the database.

4.1.1 GetAuditCompareValues

Request message:

```
<?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>


        <GetAuditCompareValues
xmlns="http://www.example.com/ProdianceERMAccount/ComplianceAudit.svc?xsd=xsd2"
namespace=http://schemas.datacontract.org/2004/07/System>
            <repositoryDocumentId>{2a9b20d9-cd9a-4c28-93e5-
9d14bda47029}</repositoryDocumentId>
            <id>2052</id>
         </GetAuditCompareValues>
    </soap:Body>
    </soap:Envelope>
```

Response message:

```
<?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>
```

```
    < GetAuditCompareValuesResponse
xmlns="http://www.example.com/ProdianceERMAccount/ComplianceAudit.svc?xsd=xsd2"
namespace="http://schemas.datacontract.org/2004/07/System">
      < GetAuditCompareValuesResult>
 <NewValue>100</NewValue><OldValue>150</OldValue>
</ GetAuditCompareValuesResult >
    </ GetAuditCompareValuesResponse >
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://prodiance.com/compliance"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
targetNamespace="http://prodiance.com/compliance"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns2="http://prodiance.com/compliance/Imports"
targetNamespace="http://prodiance.com/compliance/Imports">
      <xs:import namespace="http://prodiance.com/compliance"/>
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IComplianceAudit">
    <wsdl:operation name="GetAuditCompareValues">
      <wsdl:input
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValues"
message="tns:IComplianceAudit_GetAuditCompareValues_InputMessage"/>
      <wsdl:output
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValuesResponse"
message="tns:IComplianceAudit_GetAuditCompareValues_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValuesInvalidOpe
rationExceptionFault" name="InvalidOperationExceptionFault"
message="tns:IComplianceAudit_GetAuditCompareValues_InvalidOperationExceptionFault_FaultMessa
ge"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding IComplianceAudit" type="tns:IComplianceAudit">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetAuditCompareValues">
      <soap:operation
soapAction="http://prodiance.com/compliance/IComplianceAudit/GetAuditCompareValues"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="InvalidOperationExceptionFault">
        <soap:fault use="literal" name="InvalidOperationExceptionFault" namespace=""/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:message name="IComplianceAudit GetAuditCompareValues InputMessage">
    <wsdl:part name="parameters" element="tns:GetAuditCompareValues"/>
  </wsdl:message>
  <wsdl:message
name="IComplianceAudit_GetAuditCompareValues_InvalidOperationExceptionFault_FaultMessage">
    <wsdl:part xmlns:q1="http://schemas.datacontract.org/2004/07/System" name="detail"
element="q1:InvalidOperationException"/>
  </wsdl:message>
  <wsdl:message name="IComplianceAudit_GetAuditCompareValues_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetAuditCompareValuesResponse"/>
  </wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

Schema name	Prefix	Section
http://prodiance.com/compliance	tns	7.1
http://schemas.microsoft.com/2003/10/Serialization/	tns1	7.2
http://schemas.datacontract.org/2004/07/System	q1	7.3

For ease of implementation, the following sections provide the full XML schema for this protocol.

7.1 http://prodiance.com/compliance Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="http://prodiance.com/compliance" elementFormDefault="qualified"
targetNamespace="http://prodiance.com/compliance"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="GetAuditCompareValues">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="repositoryDocumentId" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" name="id" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetAuditCompareValuesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="GetAuditCompareValuesResult" nillable="true"
type="tns:ComplianceAuditValueData"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ComplianceAuditValueData">
    <xs:sequence>
      <xs:element minOccurs="0" name="NewValue" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="OldValue" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ComplianceAuditValueData" nillable="true"
type="tns:ComplianceAuditValueData"/>
</xs:schema>
```

7.2 http://schemas.microsoft.com/2003/10/Serialization/ Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns1="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="tns1:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns1: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="tns1: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>
```

7.3 http://schemas.datacontract.org/2004/07/System Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q1="http://schemas.datacontract.org/2004/07/System"
xmlns:tns1="http://schemas.microsoft.com/2003/10/Serialization/"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
  <xs:complexType name="InvalidOperationException">
    <xs:complexContent mixed="false">
      <xs:extension base="q1:SystemException"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="InvalidOperationException" nillable="true"
type="q1:InvalidOperationException"/>
  <xs:complexType name="SystemException">
    <xs:complexContent mixed="false">
      <xs:extension base="q1:Exception"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="SystemException" nillable="true" type="q1:SystemException"/>
  <xs:complexType name="Exception">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##local"
processContents="skip"/>
    </xs:sequence>
    <xs:attribute ref="tns1:FactoryType"/>
  </xs:complexType>
  <xs:element name="Exception" nillable="true" type="q1:Exception"/>
</xs:schema>
```

8 Appendix C: 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 Server 2013
- Microsoft SharePoint Server 2016 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.

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
8 Appendix C: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

10 Index

T

V

W

X