

# [MS-EAWF]: Group Approval 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](mailto: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
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.05	Editorial	Changed language and formatting in the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.6	Minor	Clarified the meaning of the technical content.
04/11/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	7
1.2.1 Normative References	7
1.2.2 Informative References	8
1.3 Protocol Overview (Synopsis)	8
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	9
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	9
<b>2 Messages</b>	<b>10</b>
2.1 Transport	10
2.2 Common Message Syntax	10
2.2.1 Namespaces	10
2.2.2 Messages	10
2.2.3 Elements	10
2.2.3.1 ApprovalLine element	11
2.2.3.2 Participant element	11
2.2.4 Complex Types	12
2.2.4.1 SOAPFaultDetails	12
2.2.5 Simple Types	12
2.2.6 Attributes	12
2.2.7 Groups	13
2.2.8 Attribute Groups	13
<b>3 Protocol Details</b>	<b>14</b>
3.1 EAWorkflowSoap Server Details	14
3.1.1 Abstract Data Model	14
3.1.2 Timers	15
3.1.3 Initialization	15
3.1.4 Message Processing Events and Sequencing Rules	15
3.1.4.1 GetCurrentUserFromWeb	15
3.1.4.1.1 Messages	16
3.1.4.1.1.1 GetCurrentUserFromWebSoapIn	16
3.1.4.1.1.2 GetCurrentUserFromWebSoapOut	16
3.1.4.1.2 Elements	16
3.1.4.1.2.1 GetCurrentUserFromWeb	16
3.1.4.1.2.2 GetCurrentUserFromWebResponse	16
3.1.4.1.3 Complex Types	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 GetDocumentLockStatus	17
3.1.4.2.1 Messages	18
3.1.4.2.1.1 GetDocumentLockStatusSoapIn	18
3.1.4.2.1.2 GetDocumentLockStatusSoapOut	18

3.1.4.2.2	Elements.....	19
3.1.4.2.2.1	GetDocumentLockStatus .....	19
3.1.4.2.2.2	GetDocumentLockStatusResponse .....	19
3.1.4.2.3	Complex Types .....	20
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
3.1.4.3	GetIsBackwardSigning .....	20
3.1.4.3.1	Messages .....	21
3.1.4.3.1.1	GetIsBackwardSigningSoapIn .....	21
3.1.4.3.1.2	GetIsBackwardSigningSoapOut .....	21
3.1.4.3.2	Elements.....	21
3.1.4.3.2.1	GetIsBackwardSigning .....	21
3.1.4.3.2.2	GetIsBackwardSigningResponse .....	21
3.1.4.3.3	Complex Types .....	22
3.1.4.3.4	Simple Types.....	22
3.1.4.3.5	Attributes.....	22
3.1.4.3.6	Groups.....	22
3.1.4.3.7	Attribute Groups .....	22
3.1.4.4	GetNewDocumentId .....	22
3.1.4.4.1	Messages .....	23
3.1.4.4.1.1	GetNewDocumentIdSoapIn .....	23
3.1.4.4.1.2	GetNewDocumentIdSoapOut .....	23
3.1.4.4.2	Elements.....	23
3.1.4.4.2.1	GetNewDocumentId.....	23
3.1.4.4.2.2	GetNewDocumentIdResponse .....	24
3.1.4.4.3	Complex Types .....	25
3.1.4.4.4	Simple Types.....	25
3.1.4.4.5	Attributes.....	25
3.1.4.4.6	Groups.....	25
3.1.4.4.7	Attribute Groups .....	25
3.1.4.5	GetPreferredApprovalLine .....	25
3.1.4.5.1	Messages .....	26
3.1.4.5.1.1	GetPreferredApprovalLineSoapIn .....	26
3.1.4.5.1.2	GetPreferredApprovalLineSoapOut .....	26
3.1.4.5.2	Elements.....	26
3.1.4.5.2.1	GetPreferredApprovalLine .....	26
3.1.4.5.2.2	GetPreferredApprovalLineResponse .....	27
3.1.4.5.3	Complex Types .....	27
3.1.4.5.4	Simple Types.....	27
3.1.4.5.5	Attributes.....	27
3.1.4.5.6	Groups.....	27
3.1.4.5.7	Attribute Groups .....	27
3.1.4.6	GetReceivingDepartment .....	27
3.1.4.6.1	Messages .....	28
3.1.4.6.1.1	GetReceivingDepartmentSoapIn .....	28
3.1.4.6.1.2	GetReceivingDepartmentSoapOut .....	28
3.1.4.6.2	Elements.....	28
3.1.4.6.2.1	GetReceivingDepartment .....	28
3.1.4.6.2.2	GetReceivingDepartmentResponse.....	29
3.1.4.6.3	Complex Types .....	29
3.1.4.6.4	Simple Types.....	29

3.1.4.6.5	Attributes.....	29
3.1.4.6.6	Groups.....	29
3.1.4.6.7	Attribute Groups .....	29
3.1.4.7	SetPreferredApprovalLine .....	30
3.1.4.7.1	Messages .....	30
3.1.4.7.1.1	SetPreferredApprovalLineSoapIn .....	30
3.1.4.7.1.2	SetPreferredApprovalLineSoapOut .....	30
3.1.4.7.2	Elements.....	31
3.1.4.7.2.1	SetPreferredApprovalLine .....	31
3.1.4.7.2.2	SetPreferredApprovalLineResponse .....	31
3.1.4.7.3	Complex Types .....	32
3.1.4.7.4	Simple Types.....	32
3.1.4.7.5	Attributes.....	32
3.1.4.7.6	Groups.....	32
3.1.4.7.7	Attribute Groups .....	32
3.1.5	Timer Events .....	32
3.1.6	Other Local Events .....	32
<b>4</b>	<b>Protocol Examples.....</b>	<b>33</b>
4.1	GetCurrentUserFromWeb .....	33
4.2	GetDocumentLockStatus.....	33
4.3	GetIsBackwardSigning.....	34
4.4	GetNewDocumentId .....	34
4.5	GetPreferredApprovalLine .....	35
4.6	GetReceivingDepartment .....	36
4.7	SetPreferredApprovalLine .....	37
<b>5</b>	<b>Security.....</b>	<b>39</b>
5.1	Security Considerations for Implementers.....	39
5.2	Index of Security Parameters .....	39
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>40</b>
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>48</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>49</b>
<b>9</b>	<b>Index .....</b>	<b>50</b>

# 1 Introduction

This document specifies the Group Approval Web Service Protocol, which enables the protocol client to specify and retrieve data that is required to submit or approve one or more documents that are part of Group Approval library and workflow policy.

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 RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

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

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

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

**check out**  
**checked out**  
**content type**  
**content type identifier**  
**current user**  
**document**  
**document identifier**  
**draft**  
**e-mail address**  
**folder**  
**item identifier**  
**labeling policy**  
**locked**  
**login name**  
**organization**  
**security group**  
**security group identifier**  
**security principal identifier**  
**Simple Object Access Protocol (SOAP)**  
**site**  
**SOAP action**  
**SOAP body**  
**SOAP fault**  
**SOAP message**  
**Uniform Resource Locator (URL)**  
**user identifier**  
**Web Services Description Language (WSDL)**  
**workflow**  
**WSDL operation**  
**XML namespace**  
**XML node**  
**zero-based index**

The following terms are specific to this document:

**approval line:** A sequence of approval participants in an approval workflow.

**backward signing:** A condition of a handwritten signature, in an image or .ink file, that specifies the direction of the characters in the signature, right-to-left or left-to-right.

**Group Approval document:** A document that is subject to the policies defined for a Group Approval workflow.

**Group Approval document identifier:** A string that uniquely identifies a document that is subject to the policies defined for a Group Approval workflow. The string is generated and assigned to a document automatically by a protocol server.

**organization identifier:** A positive 32-bit integer that uniquely identifies an organization.

**preferred approval line:** An approval line that is associated with a content type. Every time the content type is used, the associated approval line is also used.

**receiving organization:** An organization that receives a copy of a document and stores it in a specified folder when a workflow is complete.

**workflow participant:** A user or group that is involved in a workflow.

**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 Specifications documentation do not include a publishing year because links are to the latest version of the technical 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](mailto: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.

[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>

[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.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)

The Group Approval Web Service Protocol is used to specify and retrieve data that is required to submit or approve **Group Approval documents**. A typical usage for this protocol is in a custom Group Approval application that allows users to access Group Approval data remotely. Such an application could use this protocol to provide users with a way to maintain data on a central protocol server and access it through multiple protocol clients from different locations.

This protocol enables a protocol client to:

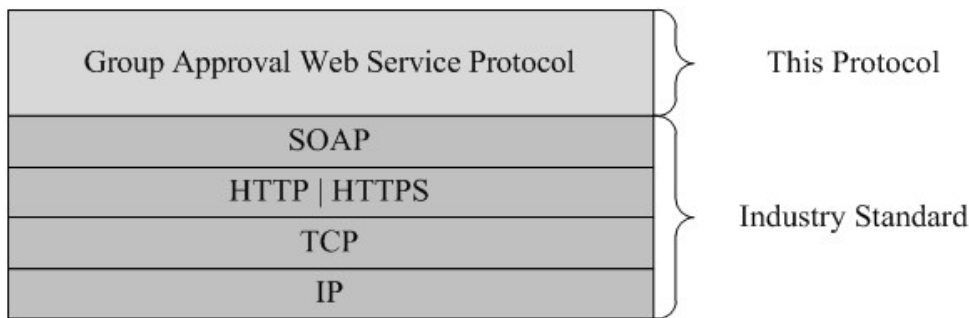
- Retrieve data about the current user of a **site** (2)
- Retrieve data about the document **check out** status on a site (2)
- Retrieve data about the **backward signing** of a **content type**
- Retrieve data about a **Group Approval document identifier** which is assigned to a specific **document**
- Add or update data about the **preferred approval line** of a content type
- Retrieve data about the preferred approval line of a content type
- Retrieve data about the **receiving organization** of a content type

### 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 [\[RFC2818\]](#).

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





**Figure 1: This protocol in relationship to other protocols**

## 1.5 Prerequisites/Preconditions

This protocol operates against a site (2) that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/eaworkflow.asmx"` to the URL of the site (2), for example:

`http://www.contoso.com/repository/_vti_bin/eaworkflow.asmx.`

This protocol assumes that authentication has been performed by the underlying protocols.

## 1.6 Applicability Statement

This protocol is used to retrieve information that is related to **current user**, document, and content type from a remote server.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports:** This protocol uses multiple transports with SOAP as specified in section [2.1](#).

## 1.8 Vendor-Extensible Fields

The message **GetNewDocumentIdResponse** referred to in section [3.1.4.4.2.2](#), contains the **LabelMetaProperty** element, which is vendor-extensible. **LabelMetaProperty** can be structured as **XML nodes** which are defined from a label specified in the **labeling policy**.

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

### 2.1 Transport

Protocol servers **MUST** support SOAP over HTTP. Protocol servers **SHOULD** additionally support SOAP over HTTPS for securing communication with clients.

Protocol messages **MUST** be formatted as specified either in [\[SOAP1.1\]](#) or in [\[SOAP1.2/1\]](#). Protocol server faults **MUST** be returned either using HTTP status codes as specified in [\[RFC2616\]](#) or using **SOAP faults** as specified either in [\[SOAP1.1\]](#) or in [\[SOAP1.2/1\]](#).

### 2.2 Common Message Syntax

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

#### 2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a prefix for each XML namespace that is referenced, the choice of any particular XML namespace prefix is implementation-specific and not required for interoperability.

Prefix	Namespace URI	Reference
soap	<a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a>	<a href="#">[SOAP1.1]</a>
tns	<a href="http://schemas.microsoft.com/sharepoint/soap/EADWorkflow/">http://schemas.microsoft.com/sharepoint/soap/EADWorkflow/</a>	
s	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a> <a href="#">[XMLSCHEMA2]</a>
soap12	<a href="http://schemas.xmlsoap.org/wsdl/soap12/">http://schemas.xmlsoap.org/wsdl/soap12/</a>	<a href="#">[SOAP1.2/1]</a> <a href="#">[SOAP1.2/2]</a>
(none)	<a href="http://schemas.microsoft.com/sharepoint/soap/EADWorkflow/">http://schemas.microsoft.com/sharepoint/soap/EADWorkflow/</a>	
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>

#### 2.2.2 Messages

None.

#### 2.2.3 Elements

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

Element	Description
<b>ApprovalLine</b>	An element which specifies the preferred approval line for a content type.
<b>Participant</b>	An element which specifies the <b>workflow participant</b> information for a <b>workflow</b> (2).

### 2.2.3.1 ApprovalLine element

The **ApprovalLine** element specifies the preferred approval line for a content type as follows:

```
<s:element name="ApprovalLine">
  <s:complexType>
    <s:sequence>
      <s:element name="Participant" minOccurs="1" maxOccurs="unbounded"/>
    </s:sequence>
    <s:attribute name="ContentTypeId" type="s:string" use="required"/>
  </s:complexType>
</s:element>
```

**Participant:** Specifies the **Participant** element as defined in [2.2.3.2](#). The number of **Participant** elements MUST be less than or equal to the number of signature placeholders of participants in a document.

**ContentTypeId:** A string which contains the valid **content type identifier** of a document.

### 2.2.3.2 Participant element

The **Participant** element specifies the workflow participant information for a workflow (2) as follows:

```
<s:element name="Participant">
  <s:complexType>
    <s:sequence>
      <s:element name="ID" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="Type" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="Order" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="ApprovalType" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="Department" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**ID:** A string which contains the valid **security principal identifier** for the workflow participant. This value MUST be a positive 32-bit **integer string** or an empty string.

**Type:** A string which contains the type of workflow participant. This value MUST be case-sensitive. This value MUST be one of the following:

- **Empty:** The type of workflow participant is not applicable.
- **Person:** The type of workflow participant is a user.
- **Department:** The type of workflow participant is an **organization**.
- **Role:** The type of workflow participant is a **security group**.

**Order:** A string which contains a **zero-based index** into **approval line**. The "0" index of the approval line **MUST** be the user who started workflow.

**ApprovalType:** A string which contains the type of the approval path where this workflow participant belongs. This value **MUST** be case-sensitive. This value **MUST** be one of the following:

- **Normal:** Specifies an approval path where one workflow participant approves after another.
- **Cooperative:** Specifies an approval path where multiple workflow participants can approve in parallel. If any one of the workflow participants approves the document, then the document is regarded as approved by one normal workflow participant.
- **Coapproval:** Specifies an approval path where multiple workflow participants can approve in parallel. If all the workflow participants approve the document, then the document is regarded as approved by one normal workflow participant.

**Department:** A string which contains the valid **organization identifier** for the workflow participant. This value **MUST** be either a positive 32-bit integer string or an empty string.

## 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
SOAPFaultDetails	Specifies the details of a SOAP fault.

### 2.2.4.1 SOAPFaultDetails

The details of a SOAP fault defined as follows:

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema" targetNamespace="
http://schemas.microsoft.com/sharepoint/soap">
  <s:complexType name="SOAPFaultDetails">
    <s:sequence>
      <s:element name="errorstring" type="s:string"/>
      <s:element name="errorcode" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:schema>
```

**errorString:** A string which contains text describing the application-level fault.

**errorCode:** A string which contains the hexadecimal representation of a 4-byte result code.

## 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

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.

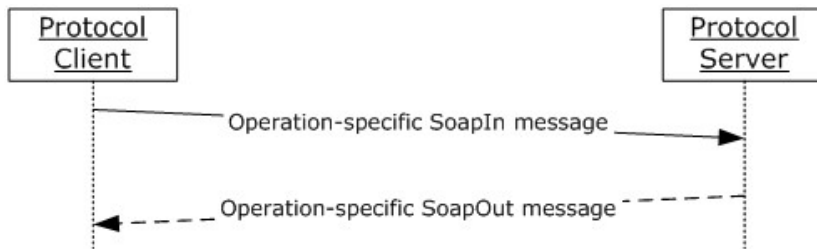
Except where specified, protocol clients SHOULD interpret HTTP Status Codes returned by the protocol server as specified in [RFC2616](#).

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 EAWorkflowSoap Server Details

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



**Figure 2: Communication between protocol client and protocol 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.

The protocol server maintains a list of content types and their associated properties. The properties are the backward signing and the receiving organization.

The protocol server maintains a list of users and security groups that are authorized to submit and approve documents.

The protocol server maintains a list of files and their associated properties. The properties include the Group Approval document identifier and the check out status.

### 3.1.2 Timers

None.

### 3.1.3 Initialization

None.

### 3.1.4 Message Processing Events and Sequencing Rules

The following table describes the methods used to gather and update information contained on the protocol server:

Operation	Description
<b>GetCurrentUserFromWeb</b>	Get the current user information of a site (2).
<b>GetDocumentLockStatus</b>	Get the document check out status.
<b>GetIsBackwardSigning</b>	Get the backward signing flag which specifies whether input content type requires backward signing or not.
<b>GetNewDocumentId</b>	Get the label information from the labeling policy.
<b>GetPreferredApprovalLine</b>	Get the preferred approval line information for an input content type.
<b>GetReceivingDepartment</b>	Get the receiving organization for an input content type.
<b>SetPreferredApprovalLine</b>	Set the preferred approval line information for an input content type.

#### 3.1.4.1 GetCurrentUserFromWeb

This operation is used to get information about the current user of a site (2) and is defined as follows:

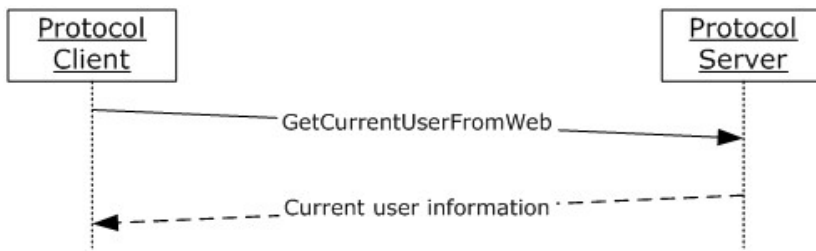
```
<wsdl:operation name="GetCurrentUserFromWeb">
  <wsdl:input message="GetCurrentUserFromWebSoapIn" />
  <wsdl:output message="GetCurrentUserFromWebSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetCurrentUserFromWebSoapIn** request message, and the protocol server responds with a **GetCurrentUserFromWebSoapOut** response message.

The protocol client can request information about the current user of the protocol server's site (2).

1. The protocol client asks the protocol server for the information about the current user using the **GetCurrentUserFromWeb** web service method.
2. The protocol server responds with user information which is comprised of a **user identifier**, a **login name** and an **e-mail address**.

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 3: Retrieving information about the current user**

### 3.1.4.1.1 Messages

#### 3.1.4.1.1.1 GetCurrentUserFromWebSoapIn

A **SOAP message** that contains information required by the **GetCurrentUserFromWeb WSDL operation** that is used to call the **GetCurrentUserFromWeb** SOAP method.

The **SOAP action** value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetCurrentUserFromWeb
```

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

#### 3.1.4.1.1.2 GetCurrentUserFromWebSoapOut

A SOAP message that contains results returned by the **GetCurrentUserFromWeb** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetCurrentUserFromWeb
```

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

### 3.1.4.1.2 Elements

#### 3.1.4.1.2.1 GetCurrentUserFromWeb

The input data of a **GetCurrentUserFromWeb** WSDL operation defined as follows:

```
<s:element name="GetCurrentUserFromWeb">
  <s:complexType/>
</s:element>
```

#### 3.1.4.1.2.2 GetCurrentUserFromWebResponse

The following is the result data of a **GetCurrentUserFromWeb** WSDL operation:

```
<s:element name="GetCurrentUserFromWebResponse">
  <s:complexType>
    <s:sequence>
```



```

<s:element name="GetCurrentUserFromWebResult" minOccurs="1" maxOccurs="1">
  <s:complexType>
    <s:sequence>
      <s:element name="CurrentUserFromWeb" minOccurs="1" maxOccurs="1"/>
      <s:complexType>
        <s:sequence>
          <s:element name="ID" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="LoginName" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="Email" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>

```

**GetCurrentUserFromWebResult:** Contains a **CurrentUserFromWeb** element which specifies current user information of a site (2).

**ID:** A string that contains the valid user identifier for the current user. This value **MUST** be a positive 32-bit **integer string**.

**LoginName:** A string that contains the login name of the current user.

**Email:** A string that contains the e-mail address of the current user.

#### 3.1.4.1.3 Complex Types

None.

#### 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 GetDocumentLockStatus

This operation is used to get the check out status of a given document at a given URL and is defined as follows:

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

```

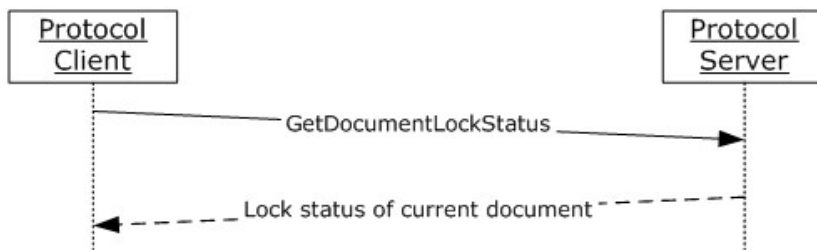
    <wsdl:input message="GetDocumentLockStatusSoapIn" />
    <wsdl:output message="GetDocumentLockStatusSoapOut" />
</wsdl:operation>

```

The protocol client sends a **GetDocumentLockStatusSoapIn** request message, and the protocol server responds with a **GetDocumentLockStatusSoapOut** response message, as follows:

1. The protocol client asks the protocol server for the information of the current document to see if it is opened for editing or viewing using the **GetDocumentLockStatus** web service method.
2. The protocol server responds with document check out status information. The status information **MUST** be one of the following values: "none", "shortterm", "longterm", "longtermoffline", or "error".

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 4: Retrieving information about the check out status for the current document**

### 3.1.4.2.1 Messages

#### 3.1.4.2.1.1 GetDocumentLockStatusSoapIn

A SOAP message that contains information required by the **GetDocumentLockStatus** WSDL operation that is used to call the **GetDocumentLockStatus** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetDocumentLockStatus
```

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

#### 3.1.4.2.1.2 GetDocumentLockStatusSoapOut

A SOAP message that contains results returned by the **GetDocumentLockStatus** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetDocumentLockStatus
```

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

### 3.1.4.2.2 Elements

#### 3.1.4.2.2.1 GetDocumentLockStatus

The input data of a **GetDocumentLockStatus** WSDL operation defined as follows:

```
<s:element name="GetDocumentLockStatus">
  <s:complexType>
    <s:sequence>
      <s:element name="documentUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**documentUrl:** A URL which identifies a document.

#### 3.1.4.2.2.2 GetDocumentLockStatusResponse

The following is the result data of a **GetDocumentLockStatus** WSDL operation.

```
<s:element name="GetDocumentLockStatusResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetDocumentLockStatusResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="LockStatus" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetDocumentLockStatusResult:** Contains a **LockStatus** element that specifies the document check out status.

**Type:** A string that contains the document check out status. This value is case-sensitive. This value MUST be one of the following:

- **none:** Document is not **locked**.
- **shortterm:** Document is opened in edit mode.
- **longterm:** Document is **checked out**.
- **longtermoffline:** Document is checked out to local **draft folder**.
- **error:** Error occurred.

**User:** A string that contains the valid user identifier of the owner of the lock. This value **MUST** be a positive 32-bit **integer string** when the value of **Type** element is "shortterm", "longterm", or "longtermoffline". Otherwise this value **MUST** be empty.

### 3.1.4.2.3 Complex Types

None.

### 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.

### 3.1.4.3 GetIsBackwardSigning

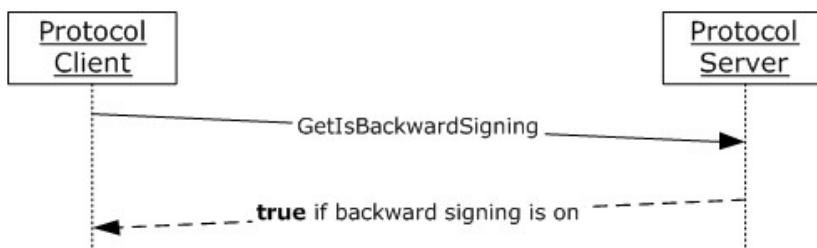
This operation is used to get the backward signing flag which specifies whether the input content type requires backward signing and is defined as follows:

```
<wsdl:operation name="GetIsBackwardSigning">
  <wsdl:input message="GetIsBackwardSigningSoapIn" />
  <wsdl:output message="GetIsBackwardSigningSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetIsBackwardSigningSoapIn** request message, and the protocol server responds with a **GetIsBackwardSigningSoapOut** response message, as follows:

1. The protocol client asks the protocol server for the information of content type to see if it is designed for backward signing or not using the **GetIsBackwardSigning** web service method.
2. The protocol server responds with **true** if backward signing is enabled. Otherwise, it returns **false**.

The following diagram shows the exchange between the protocol client and the protocol server.



## Figure 5: Retrieving information about whether backward signing is enabled

### 3.1.4.3.1 Messages

#### 3.1.4.3.1.1 GetIsBackwardSigningSoapIn

A SOAP message that contains information required by the **GetIsBackwardSigning** WSDL operation that is used to call the **GetIsBackwardSigning** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetIsBackwardSigning
```

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

#### 3.1.4.3.1.2 GetIsBackwardSigningSoapOut

A SOAP message that contains results returned by the **GetIsBackwardSigning** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetIsBackwardSigning
```

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

### 3.1.4.3.2 Elements

#### 3.1.4.3.2.1 GetIsBackwardSigning

The input data of a **GetIsBackwardSigning** WSDL operation defined as follows:

```
<s:element name="GetIsBackwardSigning">
  <s:complexType>
    <s:sequence>
      <s:element name="contentTypeid" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**contentTypeid**: A string that contains the valid content type identifier of a document.

#### 3.1.4.3.2.2 GetIsBackwardSigningResponse

The following is the result data of a **GetIsBackwardSigning** WSDL operation.

```
<s:element name="GetIsBackwardSigningResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetIsBackwardSigningResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="IsBackwardSigning" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

```

        <s:sequence>
            <s:element name="Value" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>

```

**GetIsBackwardSigningResult:** Contains a **IsBackwardSigning** element that specifies whether input content type requires backward signing or not.

**Value:** A string specifying if the input content type requires backward signing. The value **MUST** be either "True" or "False". "True" indicates that backward signing is required. "False" indicates that it is not required. This string is case-sensitive.

### 3.1.4.3.3 Complex Types

None.

### 3.1.4.3.4 Simple Types

None.

### 3.1.4.3.5 Attributes

None.

### 3.1.4.3.6 Groups

None.

### 3.1.4.3.7 Attribute Groups

None.

## 3.1.4.4 GetNewDocumentId

This operation is used to get the label information from the labeling policy and is defined as follows:

```

<wsdl:operation name="GetNewDocumentId">
    <wsdl:input message="GetNewDocumentIdSoapIn" />
    <wsdl:output message="GetNewDocumentIdSoapOut" />
</wsdl:operation>

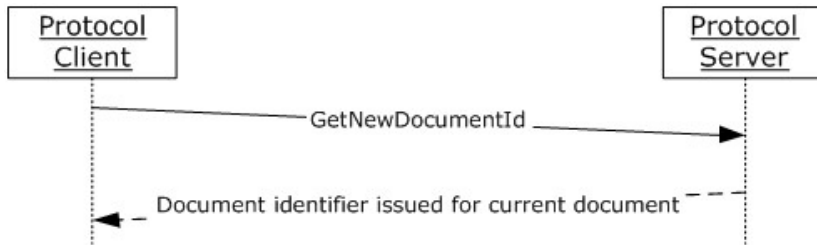
```

The protocol client sends a **GetNewDocumentIdSoapIn** request message, and the protocol server responds with a **GetNewDocumentIdSoapOut** response message, as follows:

1. The protocol client asks the protocol server for the issued Group Approval document identifier of current document using the **GetNewDocumentId** web service method.

2. The protocol server responds with **LabelMetaProperty** element which includes Group Approval document identifier.

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 6: Retrieving a LabelMetaProperty element that includes a Group Approval document identifier**

### 3.1.4.4.1 Messages

#### 3.1.4.4.1.1 GetNewDocumentIdSoapIn

A SOAP message that contains information required by the **GetNewDocumentId** WSDL operation that is used to call the **GetNewDocumentId** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetNewDocumentId
```

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

#### 3.1.4.4.1.2 GetNewDocumentIdSoapOut

A SOAP message that contains results returned by the **GetNewDocumentId** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetNewDocumentId
```

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

### 3.1.4.4.2 Elements

#### 3.1.4.4.2.1 GetNewDocumentId

The input data of a GetNewDocumentId WSDL operation defined as follows:

```
<s:element name="GetNewDocumentId">
  <s:complexType>
    <s:sequence>
      <s:element name="fileUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="orgId" type="s:int" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

</s:element>

**fileUrl:** A URL that identifies a document on this site (2).

**orgId:** A positive 32-bit organization identifier.

### 3.1.4.4.2.2 GetNewDocumentIdResponse

The following is the result data of a **GetNewDocumentId** WSDL operation.

```
<s:element name="GetNewDocumentIdResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetNewDocumentIdResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="LabelMetaProperty" minOccurs="1" maxOccurs="1"/>
            <s:complexType>
              <s:sequence>
                <s:element name="EawfCompanyName" type="s:string" minOccurs="0"
maxOccurs="1"/>
                <s:element name="EawfDepartment" type="s:string" minOccurs="0"
maxOccurs="1"/>
                <s:element name="EawfDepartmentCode" type="s:string" minOccurs="0"
maxOccurs="1"/>
                <s:element name="EawfSubmittedYear" type="s:string" minOccurs="0"
maxOccurs="1"/>
                <s:element name="EawfSequenceNumber" type="s:string" minOccurs="0"
maxOccurs="1"/>
                <s:any minOccurs="0" maxOccurs="unbounded" namespace="##local"/>
                <s:element name="EawfDocumentId" type="s:string" minOccurs="1"
maxOccurs="1"/>
              </s:sequence>
            </s:complexType>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetNewDocumentIdResult:** Contains a **LabelMetaProperty** element that is vendor-extensible and can be structured as XML nodes, which are defined from a label specified in the labeling policy.

**EawfCompanyName:** A string that contains the root name of organizations. This value MUST NOT be empty.

**EawfDepartment:** A string that contains the name of an organization to which the current user belongs. This value MUST NOT be empty.

**EawfDepartmentCode:** A string that contains a unique **string** value that represents the organization to which the current user belongs. This value MUST NOT be empty.

**EawfSubmittedYear:** A string that contains the submitted year of a document for approval. This value MUST be a positive 32-bit **integer string**. This value MUST NOT be empty.



**EawfSequenceNumber:** A string that contains the valid **item identifier** of a document. This value MUST be a positive 32-bit **integer string**. If the **integer string** is less than five characters, this **string** MUST be padded with zeros so that it is five characters in length. This value MUST NOT be empty.

**EawfDocumentId:** A string that contains the Group Approval document identifier. This value MUST NOT be empty.

### 3.1.4.4.3 Complex Types

None.

### 3.1.4.4.4 Simple Types

None.

### 3.1.4.4.5 Attributes

None.

### 3.1.4.4.6 Groups

None.

### 3.1.4.4.7 Attribute Groups

None.

### 3.1.4.5 GetPreferredApprovalLine

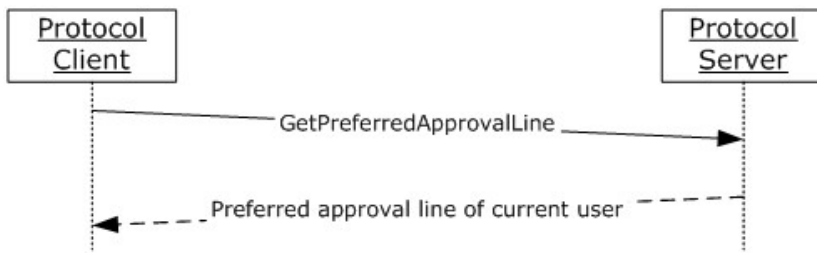
This operation is used to get the preferred approval line for a content type. A user has one associated property that specifies their preferred approval lines. The number of preferred approval lines MUST be equal to the number of content types. Preferred approval lines and content types have a one-to-one relationship. This operation is defined as follows:

```
<wsdl:operation name="GetPreferredApprovalLine">
  <wsdl:input message="GetPreferredApprovalLineSoapIn" />
  <wsdl:output message="GetPreferredApprovalLineSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetPreferredApprovalLineSoapIn** request message, and the protocol server responds with a **GetPreferredApprovalLineSoapOut** response message, as follows:

1. The protocol client asks the protocol server for the information about the preferred approval line of current content type using the **GetPreferredApprovalLine** web service method.
2. The protocol server responds with the **XML** of the preferred approval line corresponding to current content type and current user.

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 7: Retrieving the preferred approval line of the current user**

### 3.1.4.5.1 Messages

#### 3.1.4.5.1.1 GetPreferredApprovalLineSoapIn

A SOAP message that contains information required by the **GetPreferredApprovalLine** WSDL operation that is used to call the **GetPreferredApprovalLine** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetPreferredApprovalLine
```

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

#### 3.1.4.5.1.2 GetPreferredApprovalLineSoapOut

A SOAP message that contains results returned by the **GetPreferredApprovalLine** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetPreferredApprovalLine
```

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

### 3.1.4.5.2 Elements

#### 3.1.4.5.2.1 GetPreferredApprovalLine

The input data of a **GetPreferredApprovalLine** WSDL operation defined as follows:

```
<s:element name="GetPreferredApprovalLine">
  <s:complexType>
    <s:sequence>
      <s:element name="userId" type="s:int" minOccurs="1" maxOccurs="1"/>
      <s:element name="contentTypeId" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**userId:** A positive 32-bit integer that contains the valid user identifier identifying the user for the associated preferred approval line.

**contentTypeid:** A string that contains the valid content type identifier of a document.

### 3.1.4.5.2.2 GetPreferredApprovalLineResponse

The following is the result data of a **GetPreferredApprovalLine** WSDL operation.

```
<s:element name="GetPreferredApprovalLineResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetPreferredApprovalLineResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="ApprovalLine" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

**GetPreferredApprovalLineResult:** Contains an **ApprovalLine** element, found in [2.2.3.1](#), which specifies the preferred approval line for a content type.

### 3.1.4.5.3 Complex Types

None.

### 3.1.4.5.4 Simple Types

None.

### 3.1.4.5.5 Attributes

None.

### 3.1.4.5.6 Groups

None.

### 3.1.4.5.7 Attribute Groups

None.

### 3.1.4.6 GetReceivingDepartment

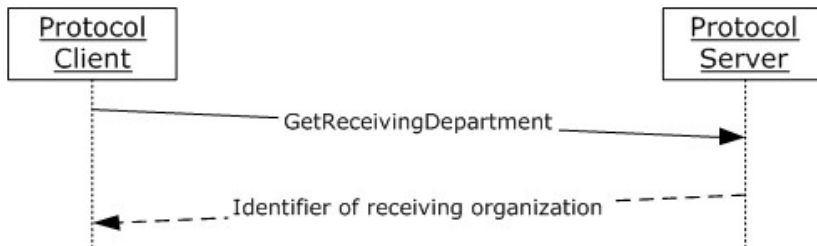
This operation is used to get the receiving organization for input content type and is defined as follows:

```
<wsdl:operation name="GetReceivingDepartment">
  <wsdl:input message="GetReceivingDepartmentSoapIn" />
  <wsdl:output message="GetReceivingDepartmentSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetReceivingDepartmentSoapIn** request message, and the protocol server responds with a **GetReceivingDepartmentSoapOut** response message, as follows:

1. The protocol client asks the protocol server for information about the receiving organization of the current content type using the **GetReceivingDepartment** web service method.
2. The protocol server responds with the **security group identifier** of receiving organization.

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 8: Retrieving the security group identifier of the receiving organization**

### 3.1.4.6.1 Messages

#### 3.1.4.6.1.1 GetReceivingDepartmentSoapIn

A SOAP message that contains information required by the **GetReceivingDepartment** WSDL operation that is used to call the **GetReceivingDepartment** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetReceivingDepartment
```

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

#### 3.1.4.6.1.2 GetReceivingDepartmentSoapOut

A SOAP message that contains results returned by the **GetReceivingDepartment** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetReceivingDepartment
```

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

### 3.1.4.6.2 Elements

#### 3.1.4.6.2.1 GetReceivingDepartment

The input data of a **GetReceivingDepartment** WSDL operation defined as follows:

```
<s:element name="GetReceivingDepartment">
  <s:complexType>
    <s:sequence>
```

```

        <s:element name="contentTypeid" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
</s:complexType>
</s:element>

```

**contentTypeid:** A string which contains the valid content type identifier of a document.

### 3.1.4.6.2.2 GetReceivingDepartmentResponse

The following is the result data of a GetReceivingDepartment WSDL operation.

```

<s:element name="GetReceivingDepartmentResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetReceivingDepartmentResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="ReceivingDepartment" minOccurs="1" maxOccurs="1"/>
            <s:complexType>
              <s:sequence>
                <s:element name="ID" type="s:string" minOccurs="1" maxOccurs="1"/>
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:sequence>
</s:complexType>
</s:element>

```

**GetReceivingDepartmentResult:** Contains a **ReceivingDepartment** element that specifies a receiving organization for an input content type.

**ID:** An integer string that contains the security group identifier of the receiving organization for an input content type.

### 3.1.4.6.3 Complex Types

None.

### 3.1.4.6.4 Simple Types

None.

### 3.1.4.6.5 Attributes

None.

### 3.1.4.6.6 Groups

None.

### 3.1.4.6.7 Attribute Groups

None.

### 3.1.4.7 SetPreferredApprovalLine

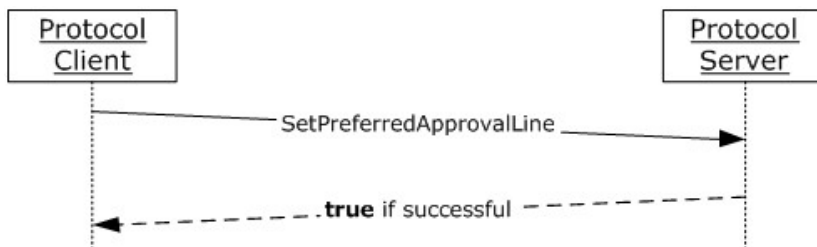
This operation is used to set the preferred approval line for a content type and is defined as follows:

```
<wsdl:operation name="SetPreferredApprovalLine">
  <wsdl:input message="SetPreferredApprovalLineSoapIn" />
  <wsdl:output message="SetPreferredApprovalLineSoapOut" />
</wsdl:operation>
```

The protocol client sends a **SetPreferredApprovalLineSoapIn** request message, and the protocol server responds with a **SetPreferredApprovalLineSoapOut** response message, as follows:

1. The protocol client asks the protocol server to specify the preferred approval line of the current content type using the **SetPreferredApprovalLine** web service method.
2. The protocol server responds with **true** if successful. Otherwise, it returns **false**.

The following diagram shows the exchange between the protocol client and the protocol server.



**Figure 9: Updating the preferred approval line of the current content type**

#### 3.1.4.7.1 Messages

##### 3.1.4.7.1.1 SetPreferredApprovalLineSoapIn

A SOAP message that contains information required by the **SetPreferredApprovalLine** WSDL operation that is used to call the **SetPreferredApprovalLine** SOAP method.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/SetPreferredApprovalLine
```

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

##### 3.1.4.7.1.2 SetPreferredApprovalLineSoapOut

A SOAP message that contains results returned by the **GetPreferredApprovalLine** WSDL operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/SetPreferredApprovalLine
```

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

### 3.1.4.7.2 Elements

#### 3.1.4.7.2.1 SetPreferredApprovalLine

The input data of a **SetPreferredApprovalLine** WSDL operation defined as follows:

```
<s:element name="SetPreferredApprovalLine">
  <s:complexType>
    <s:sequence>
      <s:element name="userId" type="s:int" minOccurs="1" maxOccurs="1"/>
      <s:element name="contentTypeId" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="approvalLineXml" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**userId:** A positive 32-bit integer which contains the valid user identifier identifying the user for the associated preferred approval line.

**contentTypeId:** A string which contains the valid content type identifier of a document.

**approvalLineXml:** A string representing the **ApprovalLine** element, defined in [2.2.3.1](#), which specifies the preferred approval line. All of the following characters within this value **MUST** be escaped. All other characters **MUST NOT** be escaped.

- The ampersand (&) (ASCII hexadecimal 0x26) **MUST** be represented using the string "&amp;" (ASCII hexadecimal 0x26 0x61 0x6D 0x70 0x3B).
- The left angle bracket (<) (ASCII hexadecimal 0x3C) **MUST** be represented using the string "&lt;" (ASCII hexadecimal 0x26 0x6C 0x74 0x3B).
- The right angle bracket (>) (ASCII hexadecimal 0x3E) **MUST** be represented using the string "&gt;" (ASCII hexadecimal 0x26 0x67 0x74 0x3B).
- The apostrophe or single-quote character (') (ASCII hexadecimal 0x27) **MUST** be represented using the string "&apos;" (ASCII hexadecimal 0x26 0x61 0x70 0x6F 0x73 0x3B).
- The double-quote character (") (ASCII hexadecimal 0x22) **MUST** be represented using the string "&quot;" (ASCII hexadecimal 0x26 0x71 0x75 0x6F 0x74 0x3B).

#### 3.1.4.7.2.2 SetPreferredApprovalLineResponse

The following is the result data of a **SetPreferredApprovalLine** WSDL operation.

```
<s:element name="SetPreferredApprovalLineResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="SetPreferredApprovalLineResult" type="s:boolean" minOccurs="1"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

**SetPreferredApprovalLineResult:** A Boolean that contains whether the preferred approval line is updated successfully.

### **3.1.4.7.3 Complex Types**

None.

### **3.1.4.7.4 Simple Types**

None.

### **3.1.4.7.5 Attributes**

None.

### **3.1.4.7.6 Groups**

None.

### **3.1.4.7.7 Attribute Groups**

None.

### **3.1.5 Timer Events**

None.

### **3.1.6 Other Local Events**

None.



## 4 Protocol Examples

### 4.1 GetCurrentUserFromWeb

To retrieve the current user information of a site (2), the protocol client sends the following SOAP message:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrentUserFromWeb xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      </GetCurrentUserFromWeb>
    </soap:Body>
  </soap:Envelope>
```

The protocol server returns the current user information of a site (2) in the form of the following SOAP 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>
    <GetCurrentUserFromWebResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      <GetCurrentUserFromWebResult>
        <CurrentUserFromWeb>
          <ID>17</ID>
          <LoginName>mydomain\user1</LoginName>
          <Email>user1@mydomain.com</Email>
        </CurrentUserFromWeb>
      </GetCurrentUserFromWebResult>
    </GetCurrentUserFromWebResponse>
  </soap:Body>
</soap:Envelope>
```

### 4.2 GetDocumentLockStatus

To retrieve the document check out status, the protocol client sends the following SOAP message:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDocumentLockStatus
xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      <documentUrl>http://www.contoso.com/my_approval/sample.docx</documentUrl>
    </GetDocumentLockStatus>
  </soap:Body>
</soap:Envelope>
```

The protocol server returns the document check out status in the form of the following SOAP 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>
```

```

    <GetDocumentLockStatusResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
    <GetDocumentLockStatusResult>
        <LockStatus>
            <Type>shortterm</Type>
            <User>17</User>
        </LockStatus>
    </GetDocumentLockStatusResult>
</GetDocumentLockStatusResponse>
</soap:Body>
</soap:Envelope>

```

### 4.3 GetIsBackwardSigning

To retrieve the backward signing flag, the protocol client sends the following SOAP message:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetIsBackwardSigning xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <contentTypeId>0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0FB004D9949F22610BE48B0047E011A2F60F7</contentTypeId>
      </GetIsBackwardSigning>
    </soap:Body>
  </soap:Envelope>

```

The protocol server returns the backward signing flag in the form of the following SOAP 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>
    <GetIsBackwardSigningResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <GetIsBackwardSigningResult>
        <IsBackwardSigning>
          <Value>True</Value>
        </IsBackwardSigning>
      </GetIsBackwardSigningResult>
    </GetIsBackwardSigningResponse>
  </soap:Body>
</soap:Envelope>

```

### 4.4 GetNewDocumentId

To retrieve the label information, the protocol client sends the following SOAP message:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNewDocumentId xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <fileUrl>http://www.contoso.com/my approval/sample.docx</fileUrl>
      <orgId>34</orgId>
    </GetNewDocumentId>
  </soap:Body>
</soap:Envelope>

```

```
</soap:Envelope>
```

The protocol server returns the label information in the form of the following SOAP 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>
    <GetNewDocumentIdResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <GetNewDocumentIdResult>
        <LabelMetaProperty>
          <EawfCompanyName>Contoso</EawfCompanyName>
          <EawfDepartmentCode>RND</EawfDepartmentCode>
          <EawfSubmittedYear>2008</EawfSubmittedYear>
          <EawfSequenceNumber>00039</EawfSequenceNumber>
          <EawfDocumentId> Contoso-RND-2008-00039</EawfDocumentId>
        </LabelMetaProperty>
      </GetNewDocumentIdResult>
    </GetNewDocumentIdResponse>
  </soap:Body>
</soap:Envelope>
```

#### 4.5 GetPreferredApprovalLine

To retrieve the preferred approval line information for an input content type, the protocol client sends the following SOAP message:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetPreferredApprovalLine
xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <userId>22</userId>

      <contentTypeId>0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0FB004D
9949F22610BE48B0047E011A2F60F7</contentTypeId>
    </GetPreferredApprovalLine>
  </soap:Body>
</soap:Envelope>
```

The protocol server returns the preferred approval line information for an input content type in the form of the following SOAP 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>
    <GetPreferredApprovalLineResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <GetPreferredApprovalLineResult>
        <ApprovalLine
ContentTypeId="0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0FB004D
9949F22610BE48B0047E011A2F60F7">
          <Participant>
```

```

        <ID>22</ID>
        <Type>Person</Type>
        <Order>0</Order>
        <ApprovalType>Normal</ApprovalType>
        <Department>16</Department>
    </Participant>
    <Participant>
        <ID>18</ID>
        <Type>Person</Type>
        <Order>1</Order>
        <ApprovalType>Normal</ApprovalType>
        <Department>16</Department>
    </Participant>
    <Participant>
        <ID>19</ID>
        <Type>Person</Type>
        <Order>2</Order>
        <ApprovalType>Normal</ApprovalType>
        <Department>16</Department>
    </Participant>
    <Participant>
        <ID>
</ID>
        <Type>Empty</Type>
        <Order>3</Order>
        <ApprovalType>Normal</ApprovalType>
        <Department>
</Department>
    </Participant>
</ApprovalLine>
</GetPreferredApprovalLineResult>
</GetPreferredApprovalLineResponse>
</soap:Body>
</soap:Envelope>

```

## 4.6 GetReceivingDepartment

To retrieve the receiving organization for an input content type, the protocol client sends the following SOAP message:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetReceivingDepartment xmlns="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/">
      <contentTypeId>0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0FB004D9949F22610BE48B0047E011A2F60F7</contentTypeId>
      </GetReceivingDepartment>
    </soap:Body>
  </soap:Envelope>

```

The protocol server returns the receiving organization for an input content type in the form of the following SOAP 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>
    <GetReceivingDepartmentResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      <GetReceivingDepartmentResult>
        <ReceivingDepartment>
          <ID>16</ID>
        </ReceivingDepartment>
      </GetReceivingDepartmentResult>
    </GetReceivingDepartmentResponse>
  </soap:Body>
</soap:Envelope>

```

## 4.7 SetPreferredApprovalLine

To specify the preferred approval line information for an input content type, the protocol client sends the following SOAP message:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetPreferredApprovalLine
xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      <userId>22</userId>

      <contentTypeId>0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0FB004D
9949F22610BE48B0047E011A2F60F7</contentTypeId>
      <approvalLineXml>&lt;ApprovalLine
ContentTypeId=&quot;0x010100BE38E8CD20424EBBA7CC1546E8BF49B7007EABF8A777E841499F4FDE601F16D0F
B004D9949F22610BE48B0047E011A2F60F7&quot;&gt;&lt;Participant&gt;&lt;ID&gt;22&lt;/ID&gt;&lt;Type&gt;Person&lt;/Type&gt;&lt;Order&gt;0&lt;/Order&gt;&lt;ApprovalType&gt;Normal&lt;/ApprovalT
ype&gt;&lt;Department&gt;34&lt;/Department&gt;&lt;Participant&gt;&lt;ID&gt;18&lt;/ID&gt;&lt;Type&gt;Person&lt;/Type&gt;&lt;Order&gt;1&lt;/Order&gt;&lt;ApprovalType&gt;
Normal&lt;/ApprovalType&gt;&lt;Department&gt;16&lt;/Department&gt;&lt;Participant&gt;&lt;Pa
rticipant&gt;&lt;ID&gt;&lt;/ID&gt;&lt;Type&gt;Empty&lt;/Type&gt;&lt;Order&gt;2&lt;/Order&gt;&
lt;ApprovalType&gt;Normal&lt;/ApprovalType&gt;&lt;Department&gt;&lt;/Department&gt;&lt;Parti
cipant&gt;&lt;Participant&gt;&lt;ID&gt;&lt;/ID&gt;&lt;Type&gt;Empty&lt;/Type&gt;&lt;Order&gt;
3&lt;/Order&gt;&lt;ApprovalType&gt;Normal&lt;/ApprovalType&gt;&lt;Department&gt;&lt;/Departme
nt&gt;&lt;/Participant&gt;&lt;/ApprovalLine&gt;</approvalLineXml>
      </SetPreferredApprovalLine>
    </soap:Body>
  </soap:Envelope>

```

The protocol server returns Boolean value specifying whether the preferred approval line is updated in the form of the following SOAP 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>
    <SetPreferredApprovalLineResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/">
      <SetPreferredApprovalLineResult>true</SetPreferredApprovalLineResult>
    </SetPreferredApprovalLineResponse>
  </soap:Body>

```

</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, as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/"
  <s:import namespace="http://www.w3.org/2001/XMLSchema" />
    <s:element name="GetNewDocumentId">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="fileUrl" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="orgId" type="s:int" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetNewDocumentIdResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="GetNewDocumentIdResult">
            <s:complexType>
              <s:sequence>
                <s:element minOccurs="1" maxOccurs="1" name="LabelMetaProperty">
                  <s:complexType>
                    <s:sequence>
                      <s:element name="EawfCompanyName" type="s:string" minOccurs="0"
maxOccurs="1" />
                      <s:element name="EawfDepartment" type="s:string" minOccurs="0"
maxOccurs="1" />
                      <s:element name="EawfDepartmentCode" type="s:string" minOccurs="0"
maxOccurs="1" />
                      <s:element name="EawfSubmittedYear" type="s:string" minOccurs="0"
maxOccurs="1" />
                      <s:element name="EawfSequenceNumber" type="s:string" minOccurs="0"
maxOccurs="1" />
                      <s:any minOccurs="0" maxOccurs="unbounded" namespace="##local" />
                      <s:element name="EawfDocumentId" type="s:string" minOccurs="1"
maxOccurs="1" />
                    </s:sequence>
                  </s:complexType>
                </s:element>
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetCurrentUserFromWeb">
      <s:complexType />
    </s:element>
    <s:element name="GetCurrentUserFromWebResponse">
      <s:complexType>

```



```

    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetCurrentUserFromWebResult">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="CurrentUserFromWeb">
              <s:complexType>
                <s:sequence>
                  <s:element name="ID" type="s:string" minOccurs="1" maxOccurs="1" />
                  <s:element name="LoginName" type="s:string" minOccurs="1"
maxOccurs="1" />
                  <s:element name="Email" type="s:string" minOccurs="1" maxOccurs="1"
/>
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetPreferredApprovalLine">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="userId" type="s:int" />
            <s:element minOccurs="1" maxOccurs="1" name="contentTypeId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetPreferredApprovalLineResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetPreferredApprovalLineResult">
              <s:complexType>
                <s:sequence>
                  <s:element name="ApprovalLine" minOccurs="0" maxOccurs="1">
                    <s:complexType>
                      <s:sequence>
                        <s:element name="Participant" minOccurs="1" maxOccurs="unbounded">
                          <s:complexType>
                            <s:sequence>
                              <s:element name="ID" type="s:string" minOccurs="1"
maxOccurs="1" />
                              <s:element name="Type" type="s:string" minOccurs="1"
maxOccurs="1" />
                              <s:element name="Order" type="s:string" minOccurs="1"
maxOccurs="1" />
                              <s:element name="ApprovalType" type="s:string" minOccurs="1"
maxOccurs="1" />
                              <s:element name="Department" type="s:string" minOccurs="1"
maxOccurs="1" />
                            </s:sequence>
                          </s:complexType>
                        </s:element>
                      </s:sequence>
                    </s:complexType>
                  </s:element>
                </s:sequence>
              </s:complexType>
            </s:element>
            <s:attribute name="ContentTypeId" type="s:string" use="required" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>

```

```

    </s:complexType>
  </s:element>
  <s:element name="SetPreferredApprovalLine">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="userId" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1" name="contentTypeId" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="approvalLineXml" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="SetPreferredApprovalLineResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="SetPreferredApprovalLineResult"
type="s:boolean" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetReceivingDepartment">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="contentTypeId" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetReceivingDepartmentResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetReceivingDepartmentResult">
          <s:complexType>
            <s:sequence>
              <s:element name="ReceivingDepartment" minOccurs="1" maxOccurs="1">
                <s:complexType>
                  <s:sequence>
                    <s:element name="ID" type="s:string" minOccurs="1" maxOccurs="1" />
                  </s:sequence>
                </s:complexType>
              </s:element>
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetIsBackwardSigning">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="contentTypeId" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetIsBackwardSigningResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetIsBackwardSigningResult">
          <s:complexType>
            <s:sequence>
              <s:element name="IsBackwardSigning" minOccurs="1" maxOccurs="1">
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>

```

```

        <s:complexType>
            <s:sequence>
                <s:element name="Value" type="s:string" minOccurs="1" maxOccurs="1" />
            </s:sequence>
        </s:complexType>
    </s:element>
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="GetDocumentLockStatus">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="documentUrl" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetDocumentLockStatusResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetDocumentLockStatusResult">
                <s:complexType>
                    <s:sequence>
                        <s:element name="LockStatus" minOccurs="1" maxOccurs="1">
                            <s:complexType>
                                <s:sequence>
                                    <s:element name="Type" type="s:string" minOccurs="1" maxOccurs="1" />
                                    <s:element name="User" type="s:string" minOccurs="1" maxOccurs="1" />
                                </s:sequence>
                            </s:complexType>
                        </s:element>
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:sequence>
    </s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="GetNewDocumentIdSoapIn">
    <wsdl:part name="parameters" element="tns:GetNewDocumentId" />
</wsdl:message>
<wsdl:message name="GetNewDocumentIdSoapOut">
    <wsdl:part name="parameters" element="tns:GetNewDocumentIdResponse" />
</wsdl:message>
<wsdl:message name="GetCurrentUserFromWebSoapIn">
    <wsdl:part name="parameters" element="tns:GetCurrentUserFromWeb" />
</wsdl:message>
<wsdl:message name="GetCurrentUserFromWebSoapOut">
    <wsdl:part name="parameters" element="tns:GetCurrentUserFromWebResponse" />
</wsdl:message>
<wsdl:message name="GetPreferredApprovalLineSoapIn">
    <wsdl:part name="parameters" element="tns:GetPreferredApprovalLine" />
</wsdl:message>
<wsdl:message name="GetPreferredApprovalLineSoapOut">
    <wsdl:part name="parameters" element="tns:GetPreferredApprovalLineResponse" />
</wsdl:message>
<wsdl:message name="SetPreferredApprovalLineSoapIn">
    <wsdl:part name="parameters" element="tns:SetPreferredApprovalLine" />

```

```

</wsdl:message>
<wsdl:message name="SetPreferredApprovalLineSoapOut">
  <wsdl:part name="parameters" element="tns:SetPreferredApprovalLineResponse" />
</wsdl:message>
<wsdl:message name="GetReceivingDepartmentSoapIn">
  <wsdl:part name="parameters" element="tns:GetReceivingDepartment" />
</wsdl:message>
<wsdl:message name="GetReceivingDepartmentSoapOut">
  <wsdl:part name="parameters" element="tns:GetReceivingDepartmentResponse" />
</wsdl:message>
<wsdl:message name="GetIsBackwardSigningSoapIn">
  <wsdl:part name="parameters" element="tns:GetIsBackwardSigning" />
</wsdl:message>
<wsdl:message name="GetIsBackwardSigningSoapOut">
  <wsdl:part name="parameters" element="tns:GetIsBackwardSigningResponse" />
</wsdl:message>
<wsdl:message name="GetDocumentLockStatusSoapIn">
  <wsdl:part name="parameters" element="tns:GetDocumentLockStatus" />
</wsdl:message>
<wsdl:message name="GetDocumentLockStatusSoapOut">
  <wsdl:part name="parameters" element="tns:GetDocumentLockStatusResponse" />
</wsdl:message>
<wsdl:portType name="EAWorkflowSoap">
  <wsdl:operation name="GetNewDocumentId">
    <wsdl:input message="tns:GetNewDocumentIdSoapIn" />
    <wsdl:output message="tns:GetNewDocumentIdSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetCurrentUserFromWeb">
    <wsdl:input message="tns:GetCurrentUserFromWebSoapIn" />
    <wsdl:output message="tns:GetCurrentUserFromWebSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetPreferredApprovalLine">
    <wsdl:input message="tns:GetPreferredApprovalLineSoapIn" />
    <wsdl:output message="tns:GetPreferredApprovalLineSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="SetPreferredApprovalLine">
    <wsdl:input message="tns:SetPreferredApprovalLineSoapIn" />
    <wsdl:output message="tns:SetPreferredApprovalLineSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetReceivingDepartment">
    <wsdl:input message="tns:GetReceivingDepartmentSoapIn" />
    <wsdl:output message="tns:GetReceivingDepartmentSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetIsBackwardSigning">
    <wsdl:input message="tns:GetIsBackwardSigningSoapIn" />
    <wsdl:output message="tns:GetIsBackwardSigningSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetDocumentLockStatus">
    <wsdl:input message="tns:GetDocumentLockStatusSoapIn" />
    <wsdl:output message="tns:GetDocumentLockStatusSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="EAWorkflowSoap" type="tns:EAWorkflowSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="GetNewDocumentId">
    <soap:operation
      soapAction="http://schemas.microsoft.com/sharepoint/soap/EAWorkflow/GetNewDocumentId"
      style="document" />
    <wsdl:input>

```

```

        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetCurrentUserFromWeb">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetCurrentUserFromWeb"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPreferredApprovalLine">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetPreferredApprovalLine"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetPreferredApprovalLine">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/SetPreferredApprovalLine"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetReceivingDepartment">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetReceivingDepartment"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetIsBackwardSigning">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetIsBackwardSigning"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>

```

```

    </wsdl:operation>
    <wsdl:operation name="GetDocumentLockStatus">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetDocumentLockStatus"
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="EWorkflowSoap12" type="tns:EWorkflowSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetNewDocumentId">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetNewDocumentId"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCurrentUserFromWeb">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetCurrentUserFromWeb"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetPreferredApprovalLine">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetPreferredApprovalLine"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetPreferredApprovalLine">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/SetPreferredApprovalLine"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:operation name="GetReceivingDepartment">

```

```

    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetReceivingDepartment"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetIsBackwardSigning">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetIsBackwardSigning"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDocumentLockStatus">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/EWorkflow/GetDocumentLockStatus"
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® Office SharePoint® Server 2007 CHT, CHS, KOR, JPN LANGPAK
- Microsoft® SharePoint® Server 2010 JPN, KOR, CHT, CHS LANGPAK

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](#) 14  
[Applicability](#) 9  
[ApprovalLine element](#) 11  
[Attribute groups](#) 13  
[Attributes](#) 12

### C

[Capability negotiation](#) 9  
[Change tracking](#) 49  
Client  
    [overview](#) 14  
[Complex types](#) 12  
    [SOAPFaultDetails](#) 12

### D

Data model - abstract  
    [server](#) 14

### E

[EAWorkflowSoap port type](#) 14  
Elements  
    [ApprovalLine](#) 11  
    [Participant](#) 11  
Events  
    [local - server](#) 32  
    [timer - server](#) 32  
Examples  
    [GetCurrentUserFromWeb](#) 33  
    [GetDocumentLockStatus](#) 33  
    [GetIsBackwardSigning](#) 34  
    [GetNewDocumentId](#) 34  
    [GetPreferredApprovalLine](#) 35  
    [GetReceivingDepartment](#) 36  
    [SetPreferredApprovalLine](#) 37

### F

[Fields - vendor-extensible](#) 9  
[Full WSDL](#) 40

### G

[GetCurrentUserFromWeb example](#) 33  
[GetDocumentLockStatus example](#) 33  
[GetIsBackwardSigning example](#) 34  
[GetNewDocumentId example](#) 34  
[GetPreferredApprovalLine example](#) 35  
[GetReceivingDepartment example](#) 36  
[Glossary](#) 6  
[Groups](#) 13

### I

[Implementer - security considerations](#) 39  
[Index of security parameters](#) 39  
[Informative references](#) 8  
Initialization  
    [server](#) 15  
[Introduction](#) 6

### L

Local events  
    [server](#) 32

### M

Message processing  
    [server](#) 15  
Messages  
    [ApprovalLine element](#) 11  
    [attribute groups](#) 13  
    [attributes](#) 12  
    [complex types](#) 12  
    [elements](#) 10  
    [enumerated](#) 10  
    [groups](#) 13  
    [namespaces](#) 10  
    [Participant element](#) 11  
    [simple types](#) 12  
    [SOAPFaultDetails complex type](#) 12  
    [syntax](#) 10  
    [transport](#) 10

### N

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

### O

Operations  
    [GetCurrentUserFromWeb](#) 15  
    [GetDocumentLockStatus](#) 17  
    [GetIsBackwardSigning](#) 20  
    [GetNewDocumentId](#) 22  
    [GetPreferredApprovalLine](#) 25  
    [GetReceivingDepartment](#) 27  
    [SetPreferredApprovalLine](#) 30  
[Overview \(synopsis\)](#) 8

### P

[Parameters - security index](#) 39  
[Participant element](#) 11  
Port types  
    [EAWorkflowSoap](#) 14  
[Preconditions](#) 9  
[Prerequisites](#) 9  
[Product behavior](#) 48

### R

[References](#) 7  
    [informative](#) 8  
    [normative](#) 7  
[Relationship to other protocols](#) 8

## S

Security  
    [implementer considerations](#) 39  
    [parameter index](#) 39  
Sequencing rules  
    [server](#) 15  
Server  
    [abstract data model](#) 14  
    [EAWorkflowSoap port type](#) 14  
    [GetCurrentUserFromWeb operation](#) 15  
    [GetDocumentLockStatus operation](#) 17  
    [GetIsBackwardSigning operation](#) 20  
    [GetNewDocumentId operation](#) 22  
    [GetPreferredApprovalLine operation](#) 25  
    [GetReceivingDepartment operation](#) 27  
    [initialization](#) 15  
    [local events](#) 32  
    [message processing](#) 15  
    [overview](#) 14  
    [sequencing rules](#) 15  
    [SetPreferredApprovalLine operation](#) 30  
    [timer events](#) 32  
    [timers](#) 15  
    [SetPreferredApprovalLine example](#) 37  
    [Simple types](#) 12  
    [SOAPFaultDetails complex type](#) 12  
    [Standards assignments](#) 9  
Syntax  
    [messages - overview](#) 10

## T

Timer events  
    [server](#) 32  
Timers  
    [server](#) 15  
[Tracking changes](#) 49  
[Transport](#) 10  
Types  
    [complex](#) 12  
    [simple](#) 12

## V

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

## W

[WSDL](#) 40