[MS-VIEWSS]:

Views Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation 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 might 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, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
7/13/2009	1.02	Major	Revised and edited the technical content
8/28/2009	1.03	Major	Updated and revised the technical content
11/6/2009	1.04	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	Major	Significantly changed the technical content.
12/17/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	Major	Significantly changed the technical content.
9/12/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	4.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	4.1	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
2/10/2014	4.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.2	Minor	Clarified the meaning of the technical content.
7/31/2014	4.3	Minor	Clarified the meaning of the technical content.
10/30/2014	4.3	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	5.0	Major	Significantly changed the technical content.
2/26/2016	6.0	Major	Significantly changed the technical content.
7/15/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Views Web Service Protocol enables a protocol client to manage a **list view**.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Collaborative Application Markup Language (CAML): An XML-based language that is used to describe various elements, such as queries and views, in sites that are based on SharePoint Products and Technologies.

default list view: The view of a SharePoint list that the owner of the list selected to appear when users browse to the list without specifying a view.

display name: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.

field: A container for metadata within a SharePoint list and associated list items.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

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].

list: A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

list item: An individual entry within a SharePoint list. Each list item has a schema that maps to fields in the list that contains the item, depending on the content type of the item.

list view: A named collection of settings for querying and displaying items in a SharePoint list. There are two types of views: Personal, which can be used only by the user who created the view; and Public, which can be used by all users who have permission to access to the site.

list view page: A Web Parts Page that displays a view of a SharePoint list.

server-relative URL: A relative URL that does not specify a scheme or host, and assumes a base URI of the root of the host, as described in [RFC3986].

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 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 Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

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 fragment: Lines of text that adhere to XML tag rules, as described in [XML], but do not have a Document Type Definition (DTD) or schema, processing instructions, or any other header information.

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

XML schema: 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.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring **XML schemas**.

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.

[MS-WSSCAML] Microsoft Corporation, "Collaborative Application Markup Language (CAML) Structure".

[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

[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., 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-LISTSWS] Microsoft Corporation, "Lists Web Service Protocol".

1.3 Protocol Overview (Synopsis)

The Views Web Service Protocol provides methods to create a list view, retrieve a specific list view or the collection of list views from a **list**, update a list view, and delete a list view. For more information about lists, see [MS-LISTSWS]. This protocol also provides methods to retrieve and update display properties of a list view by using **Collaborative Application Markup Language (CAML)** and **Hypertext Markup Language (HTML)**.

Each method in the protocol is a **WSDL operation** that accepts a set of parameters as a **SOAP** request and returns a set of values as a SOAP response.

The protocol client sends a request to the protocol server via a SOAP request message, and the protocol server sends return values to the protocol client via a SOAP response message, as shown in the following figure. All SOAP requests are made to one of several well-defined **Uniform Resource Locators (URLs)** on the protocol server, which protocol clients can discover. The protocol server never initiates any communication with the protocol client.

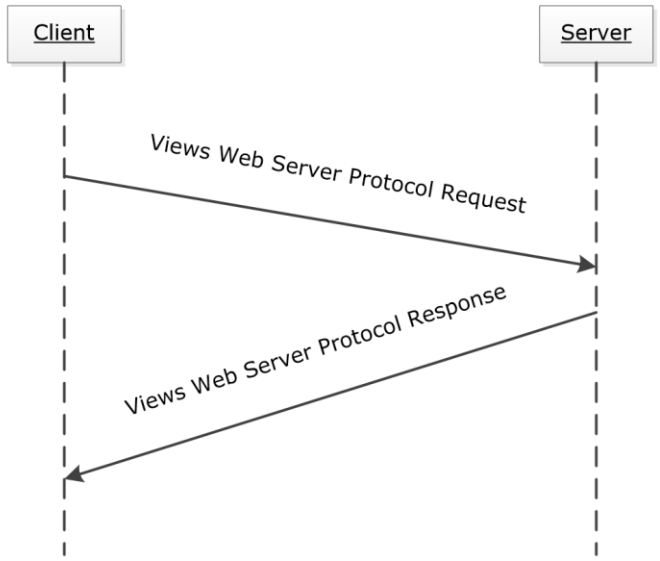

Figure 1: Views Web Service Protocol sequence diagram

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.

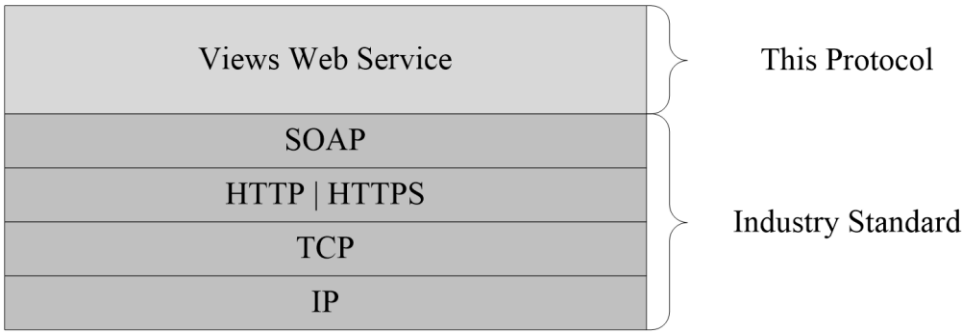

Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a URL that protocol clients recognize. The protocol server endpoint is formed by appending "/_vti_bin/views.asmx" to the URL of the site—for example, http://www.contoso.com/Repository/_vti_bin/views.asmx.

This protocol assumes that the underlying protocols have performed authentication.

1.6 Applicability Statement

The Views Web Service Protocol is applicable in the following scenarios:

- Creating and deleting a list view
- Retrieving the collection of list views of a list
- Retrieving and updating the definition of a list view
- Retrieving and updating the display properties of a list view by using CAML and HTML

1.7 Versioning and Capability Negotiation

This document covers versioning issues in one area. This protocol uses multiple transports with SOAP, as described in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS for enhancing the security of communication with protocol clients.

Protocol messages MUST be formatted as specified either in [SOAP1.1] section 4 (SOAP Envelope) or in [SOAP1.2/1] section 5 (SOAP Message Construct). Protocol server faults MUST be returned either via HTTP status codes, as specified in [RFC2616] section 10 (Status Code Definitions), or via **SOAP faults**, as specified either in [SOAP1.1] section 4.4 (SOAP Fault) or in [SOAP1.2/1] section 5.4 (SOAP Fault).

2.2 Common Message Syntax

This section contains common definitions that this protocol uses. 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** by using the mechanisms specified in [XMLNS]. Although this specification associates an XML namespace prefix for each XML namespace that is used, as shown in the following table, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
core	http://schemas.microsoft.com/sharepoint/soap/	[MS-WSSCAML]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
http	http://schemas.xmlsoap.org/wsdl/http/	[WSDL]

2.2.2 Messages

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

2.2.3 Elements

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

Element	Description
aggregations	Specifies the **fields** by which the list view is aggregated.
formats	Specifies the row and column formatting of a list view.
listName	The **display name** or the **GUID** of a list.
query	Specifies the query that a list view uses.
rowLimit	Specifies whether a list supports displaying items page by page, and the number of items that a list view displays per page.
viewFields	Specifies the fields included in a list view.
viewName	Specifies the GUID of a list view.
viewProperties	Specifies the attributes of a list view.

2.2.3.1 aggregations

The **aggregations** element specifies the fields used to aggregate a list view. The definition of the **aggregations** element is as follows.

```
<s:element name="aggregations">
  <s:complexType>
    <s:sequence>
      <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
```

Aggregations: An **XML fragment** that conforms to the schema of the **AggregationsDefinition** complex type, as specified in [MS-WSSCAML] section 2.3.2.1 (AggregationsDefinition Type).

2.2.3.2 formats

The **formats** element specifies the row and column formatting of a list view. The definition of the **formats** element is as follows.

```
<s:element name="formats">
  <s:complexType>
    <s:sequence>
      <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
```

Formats: An XML fragment that conforms to the schema of the **ViewFormatDefinitions** complex type, as specified in [MS-WSSCAML] section 2.3.2.21 (ViewFormatDefinitions Type).

2.2.3.3 listName

The **listName** element specifies a list on the protocol server. It MUST be the display name or the GUID of a list. The definition of the **listName** element is as follows.

```
<s:element name="listName" type="s:string" />
```

If the value of **listName** element is not the name or GUID of a list, the operation MUST return a SOAP fault message.

2.2.3.4 query

The **query** element includes the information that affects how a list view displays the data. The definition of the **query** element is as follows.

```
<s:element name="query">
  <s:complexType>
    <s:sequence>
      <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
```

Query: An XML fragment that conforms to the schema of the **CamlQueryRoot** complex type, as specified in [MS-WSSCAML] section 2.2.2.1 (CamlQueryRoot Type).

2.2.3.5 rowLimit

The **rowLimit** element specifies whether a list supports displaying **list items** page by page, and the number of list items that a list view displays per page. The definition of the **rowLimit** element is as follows.

```
<s:element name="rowLimit">
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
```

RowLimit: An XML fragment that conforms to the schema of the **RowLimitDefinition** complex type, as specified in [MS-WSSCAML] section 2.3.2.14 (RowLimitDefinition Type).

2.2.3.6 viewFields

The **viewFields** element specifies the fields included in a list view. The definition of the **viewFields** element is as follows.

```
<s:element name="viewFields">
  <s:complexType>
    <s:sequence>
       <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
```

```
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
```

ViewFields: An XML fragment that contains an XML element of **FieldRef**, which conforms to the schema of the **FieldRefDefinitionView** complex type, as specified in [MS-WSSCAML] section 2.3.2.19 (FieldRefDefinitionView Type).

2.2.3.7 viewName

The **viewName** element specifies a list view on the protocol server. It MUST be the GUID of a list view. The definition of the **viewName** element is as follows.

```
<s:element name="viewName" type="s:string"/>
```

If the value of **viewName** element is not the GUID of a list view, the operation MUST return a SOAP fault message.

When **viewName** is not present in the message or the value of **viewName** is empty, the protocol server MUST refer to the **default list view** of the list. If the default list view does not exist, the protocol server MUST return a SOAP fault message.

2.2.3.8 viewProperties

The **viewProperties** element specifies the properties of a list view. The definition of the **viewProperties** element is as follows.

```
<s:element name="viewProperties">
  <s:complexType>
    <s:sequence>
      <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

View: An XML fragment that conforms to the schema of the **UpdateViewPropertiesDefinition** complex type, as specified in section 2.2.4.3.

2.2.4 Complex Types

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

Complex type	Description
BriefViewDefinition	Specifies summary information about a list view.
SOAPFaultDetails	Specifies the details of a SOAP fault.
UpdateViewPropertiesDefinition	Specifies the attributes of a list view that are used in update methods.

2.2.4.1 BriefViewDefinition

The **BriefViewDefinition** complex type specifies summary information about a list view. The definition of the **BriefViewDefinition** element is as follows.

```
<s:complexType name="BriefViewDefinition" mixed="true">
  <s:sequence>
    <s:element name="Query" type="core:CamlQueryRoot" minOccurs="1" maxOccurs="1" />
    <s:element name="ViewFields" minOccurs="1" maxOccurs="1">
      <s:complexType>
        <s:sequence>
          <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
        </s:sequence>
       </s:complexType>
    </s:element>
    <s:element name="ViewData" minOccurs="0" maxOccurs="1">
      <s:complexType>
        <s:sequence>
          <s:element name="FieldRef" type="core:FieldRefDefinitionViewData" minOccurs="3"
maxOccurs="5" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="CalendarViewStyles" type="core:CalendarViewStylesDefinition"
minOccurs="0" maxOccurs="1" />
    <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1" />
    <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0" maxOccurs="1"
/>
    <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1" />
    <s:element name="ViewStyle" type="core:ViewStyleReference" minOccurs="0" maxOccurs="1" />
    <s:element name="OpenApplicationExtension" type="s:string" minOccurs="0" maxOccurs="1"
/>
  </s:sequence>
  <s:attributeGroup ref="tns:ViewAttributeGroup"/>
</s:complexType>
```

Query: For a definition, see [MS-WSSCAML] section 2.3.2.17.

ViewFields: For a definition, see [MS-WSSCAML] section 2.3.2.17.

ViewData: For a definition, see [MS-WSSCAML] section 2.3.2.17.

CalendarViewStyles: For a definition, see [MS-WSSCAML] section 2.3.2.17.

RowLimit: For a definition, see [MS-WSSCAML] section 2.3.2.17.

Formats: For a definition, see[MS-WSSCAML] section 2.3.2.17.

Aggregations: For a definition, see [MS-WSSCAML] section 2.3.2.17.

ViewStyle: For a definition, see [MS-WSSCAML] section 2.3.2.17.

OpenApplicationExtension: For a definition, see [MS-WSSCAML] section 2.3.2.17.

The attributes are specified in the attribute group **ViewAttributeGroup**; refer to section 2.2.8.1 for its definition.

2.2.4.2 SOAPFaultDetails

The **SOAPFaultDetails** complex type specifies the details of a SOAP fault. The definition of the **SOAPFaultDetails** element is 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: Human-readable text that explains the application-level fault.

errorcode: The hexadecimal representation of a 4-byte result code.

2.2.4.3 UpdateViewPropertiesDefinition

The **UpdateViewPropertiesDefinition** complex type specifies the attributes of the **View** element that are used for update methods. The definition of the **UpdateViewPropertiesDefinition** element is as follows.

```
<s:complexType name="UpdateViewPropertiesDefinition">
  <s:attributeGroup ref="tns:UpdateViewAttributeGroup"/>
</s:complexType>
```

The attribute group **UpdateViewAttributeGroup** is specified in section 2.2.8.2.

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

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

Attribute	Description
UpdateViewAttributeGroup	Contains the **XML schema definition (XSD)** attributes used in methods of updating a list view.
ViewAttributeGroup	Contains XSD attributes that specify a list view.

2.2.8.1 ViewAttributeGroup

The **ViewAttributeGroup** attribute group contains attributes that specify a list view. All attributes specified by **ViewAttributeGroup** are the same as those specified by the **ViewDefinition** complex type as specified in [MS-WSSCAML] section 2.3.2.17.

2.2.8.2 UpdateViewAttributeGroup

The **UpdateViewAttributeGroup** attribute group contains attributes that specify a list view. This group of attributes is used in update methods as the input parameter. The definition of the **UpdateViewAttributeGroup** element is as follows.

```
<s:attributeGroup name="UpdateViewAttributeGroup">
  <s:attribute name="DefaultView" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="DisplayName" type="s:string" />
  <s:attribute name="FPModified" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="Scope" type="core:ViewScope" default="" />
</s:attributeGroup>
```

All attributes are specified in [MS-WSSCAML] section 2.3.2.17 (ViewDefinition). The DefaultView attribute SHOULD be ignored by the protocol server if it is set to false.

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls that the higher-layer protocol or application makes are passed directly to the transport, and the results that the transport returns are passed directly back to the higher-layer protocol or application.

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

This protocol allows protocol servers to provide additional details for SOAP faults by including either a **detail** element as specified in [SOAP1.1] section 4.4 or a **Detail** element<1> as specified in [SOAP1.2/1] section 5.4.5, which conforms to the XML schema of the **SOAPFaultDetails** complex type specified in section 2.2.4.2. 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 by using HTTP status codes or by using SOAP faults, as specified previously in this section.

3.1 ViewsSoap 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. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol enables a protocol client to create a list view, retrieve a specific list view or the collection of list views from a list, update a list view, and delete a list view.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the WSDL operations specified in this specification.

Operation	Description
AddView	Adds a list view for the specified list.
DeleteView	Deletes a list view.
GetView	Retrieves information about a list view, without the display properties.
GetViewCollection	Retrieves a collection of list views for a specified list.
GetViewHtml	Retrieves information about a list view, including the display properties in CAML and HTML.

Operation	Description
UpdateView	Updates a list view, without the display properties.
UpdateViewHtml	Updates a list view, including the display properties in CAML and HTML. This method operates the same as **UpdateViewHtml2** with one less parameter.
UpdateViewHtml2	Updates a list view, including the display properties in CAML and HTML.

3.1.4.1 AddView

The **AddView** operation is used to create a list view for the specified list. The definition of the **AddView** operation is as follows.

```
<wsdl:operation name="AddView">
    <wsdl:input message="tns:AddViewSoapIn" />
    <wsdl:output message="tns:AddViewSoapOut" />
</wsdl:operation>
```

The protocol client sends an **AddViewSoapIn** request message (section 3.1.4.1.1.1), and the protocol server responds with an **AddViewSoapOut** response message (section 3.1.4.1.1.2).

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.1.1.1 AddViewSoapIn

The request WSDL message for an **AddView** WSDL operation (section 3.1.4.1).

The **SOAP action** value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/AddView
```

The **SOAP body** contains an **AddView** element (section 3.1.4.1.2.1).

3.1.4.1.1.2 AddViewSoapOut

The response WSDL message for an **AddView** WSDL operation (section 3.1.4.1).

The SOAP body contains an **AddViewResponse** element (section 3.1.4.1.2.3).

3.1.4.1.2 Elements

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

3.1.4.1.2.1 AddView

The **AddView** element defines the input parameters for the **AddView** operation (section 3.1.4.1). The definition of the **AddView** element is as follows.

```
<s:element name="AddView">
  <s:complexType>
```

```
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
<s:element minOccurs="0" maxOccurs="1" name="viewFields">
              <s:complexType>
                    <s:sequence>
                   <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
                     <s:complexType>
                      <s:sequence>
                       <s:element name="FieldRef" type="core:FieldRefDefinitionView"
minOccurs="0" maxOccurs="unbounded" />
                      </s:sequence>
                     </s:complexType>
                  </s:element>
                 </s:sequence>
                </s:complexType>
              </s:element>
             <s:element minOccurs="0" maxOccurs="1" name="query">
              <s:complexType>
               <s:sequence>
<s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
                </s:sequence>
              </s:complexType>
            </s:element>
<s:element minOccurs="0" maxOccurs="1" name="rowLimit">
             <s:complexType>
                <s:sequence>
<s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
                </s:sequence>
               </s:complexType>
             </s:element>
      <s:element name="type" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="makeViewDefault" type="s:boolean" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: For a definition, see section 2.2.3.3.

viewName: Display name of the list view. If the value of **viewName** element is empty or the **viewName** element is not present, the protocol server MUST return a SOAP fault message.

viewFields: For a definition, see section 2.2.3.6. When the value of the **viewFields** element is empty, the protocol server MUST create the list view with no fields included. This element MUST be present.

query: For a definition, see section 2.2.3.4. When the value of the **query** element is empty, the protocol server MUST create the list view without any additional restriction. This element MUST be present.

rowLimit: For a definition, see section 2.2.3.5. When the **rowLimit** element is not present or the value of the **rowLimit** element is empty, the protocol server MUST use the default value of "0x0064", and the list view MUST support page-by-page displaying of items.

type: For a definition, see section 3.1.4.1.2.2.

makeViewDefault: Specifies whether to make the list view the default list view for the specified list. The protocol server MUST create the list view as the default list view if "true" is specified.

3.1.4.1.2.2 type

The **type** element specifies the type of a list view. When this element is present, it MUST have one of the values in the following table.

Type	Description
"Calendar"	Represents a calendar list view.
"Grid"	Represents a datasheet list view.
"Html"	Represents a standard HTML list view.

When this element is not present or has an empty value, the protocol server MUST take it with a value of "Html".

When the value of the element is not empty and is not one of the values listed in the table, the protocol server MUST throw a SOAP fault message.

3.1.4.1.2.3 AddViewResponse

The **AddViewResponse** element defines the output parameters for the **AddView** operation (section 3.1.4.1). The definition of the **AddViewResponse** element is as follows.

```
<s:element name="AddViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

AddViewResult: Represents the result of the operation. It MUST create the list view, and include the resulting **View** element when the operation succeeds. The protocol server MAY<2> append ">" after the **View** element. The type of the **View** element is **BriefViewDefinition**, which is specified in section 2.2.4.1.

3.1.4.2 DeleteView

The **DeleteView** operation is used to delete the specified list view of the specified list. The definition of the **DeleteView** operation is as follows.

```
<wsdl:operation name="DeleteView">
    <wsdl:input message="tns:DeleteViewSoapIn" />
    <wsdl:output message="tns:DeleteViewSoapOut" />
</wsdl:operation>
```

The protocol client sends a **DeleteViewSoapIn** request message (section 3.1.4.2.1.1), and the protocol server responds with a **DeleteViewSoapOut** response message (section 3.1.4.2.1.2).

3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.1.1 DeleteViewSoapIn

The request WSDL message for a **DeleteView** WSDL operation (section 3.1.4.2).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/DeleteView
```

The SOAP body contains a **DeleteView** element (section 3.1.4.2.2.1).

3.1.4.2.1.2 DeleteViewSoapOut

The response WSDL message for a **DeleteView** WSDL operation (section 3.1.4.2).

The SOAP body contains a **DeleteViewResponse** element (section 3.1.4.2.2.2).

3.1.4.2.2 Elements

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

3.1.4.2.2.1 DeleteView

The **DeleteView** element defines the input parameters of the **DeleteView** operation (section 3.1.4.2). The definition of the **DeleteView** element is as follows.

```
<s:element name="DeleteView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: For a definition, see section 2.2.3.3.

viewName: For a definition, see section 2.2.3.7.

3.1.4.2.2.2 DeleteViewResponse

The **DeleteViewResponse** element defines the output of the **DeleteView** operation (section 3.1.4.2). The protocol server MUST delete the list view and respond with a **DeleteViewResponse** element if the operation succeeded.

The definition of the **DeleteViewResponse** element is as follows.

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

3.1.4.3 GetView

The **GetView** operation is used to obtain information about a specified list view of the specified list, without the display properties. The definition of the **GetView** operation is as follows.

```
<wsdl:operation name="GetView">
    <wsdl:input message="tns:GetViewSoapIn" />
    <wsdl:output message="tns:GetViewSoapOut" />
```

```
</wsdl:operation>
```

The protocol client sends a **GetViewSoapIn** request message (section 3.1.4.3.1.1), and the protocol server responds with a **GetViewSoapOut** response (section 3.1.4.3.1.2).

3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.1.1 GetViewSoapIn

The request WSDL message for a **GetView** WSDL operation (section 3.1.4.3).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/GetView
```

The SOAP body contains a **GetView** element (section 3.1.4.3.2.1).

3.1.4.3.1.2 GetViewSoapOut

The response WSDL message for a **GetView** WSDL operation (section 3.1.4.3).

The SOAP body contains a **GetViewResponse** element (section 3.1.4.3.2.2).

3.1.4.3.2 Elements

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

3.1.4.3.2.1 GetView

The **GetView** element specifies the input parameters of the **GetView** operation (section 3.1.4.3).

```
<s:element name="GetView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: Refer to section 2.2.3.3.

viewName: Refer to section 2.2.3.7.

3.1.4.3.2.2 GetViewResponse

The **GetViewResponse** element specifies the output of the **GetView** operation (section 3.1.4.3). The definition of the **GetViewResponse** element is as follows.

```
<s:element name="GetViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
```

```
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetViewResult: Represents the result of the operation. The protocol server MUST return a **View** element that contains the details of the specified list view when the operation succeeds. The type of the **View** element is **BriefViewDefinition**, which is specified in section 2.2.4.1.<3>

3.1.4.4 GetViewCollection

The **GetViewCollection** operation is used to retrieve the collection of list views of a specified list. The definition of the **GetViewCollection** operation is as follows.

```
<wsdl:operation name="GetViewCollection">
    <wsdl:input message="tns:GetViewCollectionSoapIn" />
    <wsdl:output message="tns:GetViewCollectionSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetViewCollectionSoapIn** request message (section 3.1.4.4.1.1), and the protocol server responds with a **GetViewCollectionSoapOut** response message (section 3.1.4.4.1.2).

3.1.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.1.1 GetViewCollectionSoapIn

The request WSDL message for a **GetViewCollection** WSDL operation (section 3.1.4.4).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/GetViewCollection
```

The SOAP body contains a **GetViewCollection** element (section 3.1.4.4.2.1).

3.1.4.4.1.2 GetViewCollectionSoapOut

The response WSDL message for a **GetViewCollection** WSDL operation (section 3.1.4.4).

The SOAP body contains a **GetViewCollectionResponse** element (section 3.1.4.4.2.2).

3.1.4.4.2 Elements

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

3.1.4.4.2.1 GetViewCollection

The **GetViewCollection** element defines the input parameter of the **GetViewCollection** operation (section 3.1.4.4). The definition of the **GetViewCollection** element is as follows.

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

listName: For a definition, see section 2.2.3.3.

3.1.4.4.2.2 GetViewCollectionResponse

The **GetViewCollectionResponse** element defines the output of the **GetViewCollection** operation (section 3.1.4.4). The definition of the **GetViewCollectionResponse** element is as follows.

```
<s:element name="GetViewCollectionResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewCollectionResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="Views" minOccurs="1" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:element name="View" minOccurs="0" maxOccurs="unbounded">
                    <s:complexType>
                      <s:attributeGroup ref="tns:ViewAttributeGroup"/>
                    </s:complexType>
                  </s:element>
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetViewCollectionResult: Represents the result of the operation. It MUST include the collection of **View** elements of the specified list, which includes an attribute group of type **ViewAttributeGroup**. The attribute group **ViewAttributeGroup** is specified in section 2.2.8.1, with the following exception: The **Url** attribute MUST be the **server-relative URL** of the list view.

3.1.4.5 GetViewHtml

The **GetViewHtml** operation is used to obtain details of a specified list view of the specified list, including display properties in CAML and HTML. The definition of the **GetViewHtml** operation is as follows.

```
<wsdl:operation name="GetViewHtml">
    <wsdl:input message="tns:GetViewHtmlSoapIn" />
    <wsdl:output message="tns:GetViewHtmlSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetViewHtmlSoapIn** request message (section 3.1.4.5.1.1), and the protocol server responds with a **GetViewHtmlSoapOut** response message (section 3.1.4.5.1.2).

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.1.1 GetViewHtmlSoapIn

The request WSDL message for a **GetViewHtml** WSDL operation (section 3.1.4.5).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/GetViewHtml
```

The SOAP body contains a **GetViewHtml** element (section 3.1.4.5.2.1).

3.1.4.5.1.2 GetViewHtmlSoapOut

The response WSDL message for a **GetViewHtml** WSDL operation (section 3.1.4.5).

The SOAP body contains a **GetViewHtmlResponse** element (section 3.1.4.5.2.2).

3.1.4.5.2 Elements

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

3.1.4.5.2.1 GetViewHtml

The **GetViewHtml** element defines the input parameters for the **GetViewHtml** operation (section 3.1.4.5). The definition of the **GetViewHtml** element is as follows.

```
<s:element name="GetViewHtml">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: For a definition, see section 2.2.3.3.

viewName: For a definition, see section 2.2.3.7.

3.1.4.5.2.2 GetViewHtmlResponse

The **GetViewHtmlResponse** element defines the output parameters for the **GetViewHtml** operation (section 3.1.4.5). The definition of the **GetViewHtmlResponse** element is as follows.

```
<s:element name="GetViewHtmlResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewHtmlResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetViewHtmlResult: Represents the result of the operation. It MUST include the corresponding **View** element when the operation succeeds. The type of the **View** element is specified in [MS-WSSCAML] section 2.3.2.17.

3.1.4.6 UpdateView

The **UpdateView** operation is used to update the specified list view, without the display properties. The definition of the **UpdateView** element is as follows.

```
<wsdl:operation name="UpdateView">
    <wsdl:input message="tns:UpdateViewSoapIn" />
    <wsdl:output message="tns:UpdateViewSoapOut" />
</wsdl:operation>
```

The protocol client sends an **UpdateViewSoapIn** request message (section 3.1.4.6.1.1), and the protocol server responds with an **UpdateViewSoapOut** response message (section 3.1.4.6.1.2).

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 UpdateViewSoapIn

The request WSDL message for an **UpdateView** WSDL operation (section 3.1.4.6).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/UpdateView
```

The SOAP body contains an **UpdateView** element (section 3.1.4.6.2.1).

3.1.4.6.1.2 UpdateViewSoapOut

The response WSDL message for an **UpdateView** WSDL operation (section 3.1.4.6).

The SOAP body contains an **UpdateViewResponse** element (section 3.1.4.6.2.2).

3.1.4.6.2 Elements

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

3.1.4.6.2.1 UpdateView

The **UpdateView** element defines the input parameters for the **UpdateView** operation (section 3.1.4.6). The definition of the **UpdateView** element is as follows.

```
<s:element name="UpdateView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1" />
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1" />
      <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
```

```
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="query">
        <s:complexType>
          <s:sequence>
            <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewFields">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="aggregations">
        <s:complexType>
          <s:sequence>
            <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="formats">
        <s:complexType>
          <s:sequence>
            <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
        <s:complexType>
          <s:sequence>
            <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: For a definition, see section 2.2.3.3.

viewName: For a definition, see section 2.2.3.7.

viewProperties: For a definition, see section 2.2.3.8.

query: For a definition, see section 2.2.3.4.

viewFields: For a definition, see section 2.2.3.6.

aggregations: For a definition, see section 2.2.3.1.

formats: For a definition, see section 2.2.3.2.

rowLimit: For a definition, see section 2.2.3.5.

3.1.4.6.2.2 UpdateViewResponse

The **UpdateViewResponse** element defines the output parameters for the **UpdateView** operation (section 3.1.4.6). The definition of the **UpdateViewResponse** element is as follows.

```
<s:element name="UpdateViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

UpdateViewResult: If the protocol server successfully updates the list view, it MUST return a **View** element that specifies the list view. The type of the **View** element is **BriefViewDefinition**, which is specified in section 2.2.4.1.

3.1.4.7 UpdateViewHtml

The **UpdateViewHtml** operation is used to update a list view for a specified list, including display properties in CAML and HTML. The definition of the **UpdateViewHtml** element is as follows.

```
<wsdl:operation name="UpdateViewHtml">
    <wsdl:input message="tns:UpdateViewHtmlSoapIn" />
    <wsdl:output message="tns:UpdateViewHtmlSoapOut" />
</wsdl:operation>
```

The protocol client sends an **UpdateViewHtmlSoapIn** request message (section 3.1.4.7.1.1), and the protocol server responds with an **UpdateViewHtmlSoapOut** response message (section 3.1.4.7.1.2).

When processing this call, the protocol server MUST perform the same actions as for the **UpdateViewHtml2** method (section 3.1.4.8), with the **openApplicationExtension** parameter as empty.

3.1.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.1.1 UpdateViewHtmlSoapIn

The request WSDL message for an **UpdateViewHtml** WSDL operation (section 3.1.4.7).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml
```

The SOAP body contains an **UpdateViewHtml** element (section 3.1.4.7.2.1).

3.1.4.7.1.2 UpdateViewHtmlSoapOut

The response WSDL message for an **UpdateViewHtml** WSDL operation (section 3.1.4.7).

The SOAP body contains an **UpdateViewHtmlResponse** element (section 3.1.4.7.2.2).

3.1.4.7.2 Elements

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

3.1.4.7.2.1 UpdateViewHtml

The **UpdateViewHtml** element defines the input parameters for the **UpdateViewHtml** operation (section 3.1.4.7). The definition of the **UpdateViewHtml** element is as follows.

```
<s:element name="UpdateViewHtml">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1" />
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1" />
      <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="toolbar">
        <s:complexType>
          <s:sequence>
            <s:element name="Toolbar" type="core:ToolbarDefinition" minOccurs="0"
maxOccurs="1" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewHeader">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewHeader" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewBody">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewBody" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewFooter">
        <s:complexType>
          <s:sequence>
```

```
            <s:element name="ViewFooter" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewEmpty">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewEmpty" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="rowLimitExceeded">
        <s:complexType>
          <s:sequence>
            <s:element name="RowLimitExceeded" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="query">
        <s:complexType>
          <s:sequence>
            <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewFields">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="aggregations">
        <s:complexType>
          <s:sequence>
            <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
```

```
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="formats">
        <s:complexType>
          <s:sequence>
            <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
        <s:complexType>
          <s:sequence>
            <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

All parameters are as specified by the **UpdateViewHtml2** element (section 3.1.4.8.2.1), with the following exception: This method does not use the **openApplicationExtension** parameter.

3.1.4.7.2.2 UpdateViewHtmlResponse

The **UpdateViewHtmlResponse** element defines the output parameters for the **UpdateViewHtml** operation (section 3.1.4.7). The definition of the **UpdateViewHtmlResponse** element is as follows.

```
<s:element name="UpdateViewHtmlResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewHtmlResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

UpdateViewHtmlResult: If the protocol server successfully updates the list view, it MUST return a **View** element that specifies the list view. The type of the **View** element is **ViewDefinition**, which is specified in [MS-WSSCAML] section 2.3.2.17 (ViewDefinition Type).

3.1.4.8 UpdateViewHtml2

The **UpdateViewHtml2** operation<4> is used to update details of a specified list view of the specified list, including display properties in CAML and HTML. The definition of the **UpdateViewHtml2** operation is as follows.

```
<wsdl:operation name="UpdateViewHtml2">
    <wsdl:input message="tns:UpdateViewHtml2SoapIn" />
    <wsdl:output message="tns:UpdateViewHtml2SoapOut" />
</wsdl:operation>
```

The protocol client sends an **UpdateViewHtml2SoapIn** request message (section 3.1.4.8.1.1), and the protocol server responds with an **UpdateViewHtml2SoapOut** response message (section 3.1.4.8.1.2).

3.1.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.8.1.1 UpdateViewHtml2SoapIn

The request WSDL message for an **UpdateViewHtml2** WSDL operation (section 3.1.4.8).

The SOAP action value of the message is specified as follows.

```
http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml2
```

The SOAP body contains an **UpdateViewHtml2** element (section 3.1.4.8.2.1).

3.1.4.8.1.2 UpdateViewHtml2SoapOut

The response WSDL message for an **UpdateViewHtml2** WSDL operation (section 3.1.4.8).

The SOAP body contains an **UpdateViewHtml2Response** element (section 3.1.4.8.2.8).

3.1.4.8.2 Elements

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

3.1.4.8.2.1 UpdateViewHtml2

The **UpdateViewHtml2** element defines the input parameters for the **UpdateViewHtml2** operation (section 3.1.4.8). The definition of the **UpdateViewHtml2** element is as follows.

```
<s:element name="UpdateViewHtml2">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1" />
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1" />
      <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="toolbar">
        <s:complexType>
          <s:sequence>
            <s:element name="Toolbar" type="core:ToolbarDefinition" minOccurs="0"
maxOccurs="1" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewHeader">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewHeader" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
```

```
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewBody">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewBody" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewFooter">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewFooter" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewEmpty">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewEmpty" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="rowLimitExceeded">
        <s:complexType>
          <s:sequence>
            <s:element name="RowLimitExceeded" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
                </s:sequence>
                <s:anyAttribute processContents="skip" />
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
```

```
      <s:element minOccurs="0" maxOccurs="1" name="query">
        <s:complexType>
          <s:sequence>
            <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="viewFields">
        <s:complexType>
          <s:sequence>
            <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
              <s:complexType>
                <s:sequence>
                  <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="aggregations">
        <s:complexType>
          <s:sequence>
            <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="formats">
        <s:complexType>
          <s:sequence>
            <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
        <s:complexType>
          <s:sequence>
            <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="openApplicationExtension" type="s:string" minOccurs="0" maxOccurs="1"
/>
    </s:sequence>
  </s:complexType>
</s:element>
```

listName: Refer to section 2.2.3.3 for a definition.

viewName: Refer to section 2.2.3.7 for a definition.

viewProperties: Refer to section 2.2.3.8 for a definition.

toolbar: Refer to section 3.1.4.8.2.2 for a definition.

viewHeader: Refer to section 3.1.4.8.2.3 for a definition.

viewBody: Refer to section 3.1.4.8.2.4 for a definition.

viewFooter: Refer to section 3.1.4.8.2.5 for a definition.

viewEmpty: Refer to section 3.1.4.8.2.6 for a definition.

rowLimitExceeded: Refer to section 3.1.4.8.2.7 for a definition.

query: Refer to section 2.2.3.4 for a definition.

viewFields: Refer to section 2.2.3.6 for a definition.

aggregations: Refer to section 2.2.3.1 for a definition.

formats: Refer to section 2.2.3.2 for a definition.

rowLimit: Refer to section 2.2.3.5 for a definition.

openApplicationExtension: Specifies what kind of application to use to edit the view. The protocol server can look up the application that is supposed to use that extension, and it can use that application to edit this view.

3.1.4.8.2.2 toolbar

The **toolbar** element specifies the rendering of the toolbar of a list. The definition of the **toolbar** element is as follows.

```
<s:element name="toolbar">
  <s:complexType>
    <s:sequence>
      <s:element name="Toolbar" type="core:ToolbarDefinition" minOccurs="0" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
```

The **Toolbar** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.3 viewHeader

The **viewHeader** element specifies the implementation-specific rendering of the header, or the top of a **list view page**. The definition of the **viewHeader** element is as follows.

```
<s:element name="viewHeader">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewHeader" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

The **ViewHeader** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.4 viewBody

The **viewBody** element specifies the implementation-specific rendering of the main, or the middle portion of a list view page. The definition of the **viewBody** element is as follows.

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

```
      <s:element name="ViewBody" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

The **viewBody** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.5 viewFooter

The **viewFooter** element specifies the implementation-specific rendering of the footer, or the bottom of a list view page. The definition of the **viewFooter** element is as follows.

```
<s:element name="viewFooter">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewFooter" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

The **viewFooter** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.6 viewEmpty

The **viewEmpty** element specifies the message to be displayed when no items are in a list view. The definition of the **viewEmpty** element is as follows.

```
<s:element minOccurs="0" maxOccurs="1" name="viewEmpty">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewEmpty" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

The **viewEmpty** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.7 rowLimitExceeded

The **rowLimitExceeded** element specifies the implementation-specific rendering of additional items when the number of items exceeds the value specified in the **rowLimit** element, but the list view is not set to support displaying items in multiple pages. The definition of the **rowLimitExceeded** element is as follows.

```
<s:element minOccurs="0" maxOccurs="1" name="rowLimitExceeded">
  <s:complexType>

    <s:sequence>
      <s:element name="RowLimitExceeded" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

The **rowLimitExceeded** element is specified in [MS-WSSCAML] section 2.3.2.17.3 (Child Elements).

3.1.4.8.2.8 UpdateViewHtml2Response

The **UpdateViewHtml2Response** element defines the output parameters for the **UpdateViewHtml2** operation (section 3.1.4.8). The definition of the **UpdateViewHtml2Response** element is as follows.

```
<s:element name="UpdateViewHtml2Response">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewHtml2Result" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

UpdateViewHtml2Result: If the protocol server successfully updates the list view, it MUST return a **View** element that specifies the list view. The type of the **View** element is **ViewDefinition**, which is specified in [MS-WSSCAML] section 2.3.2.17 (View Definition Type).

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following example demonstrates the creation, update, and deletion of a list view.

The user creates a list view by calling the **AddView** Web method (section 3.1.4.1) with the following information in the request message.

```
<AddView xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <listName>{C2294E28-05B6-4486-804A-80F0189B992B}</listName>
  <viewName>ViewA</viewName>
  <viewFields>
    <ViewFields>
      <FieldRef  Name='ID'/>
      <FieldRef  Name='Title'/>
    </ViewFields>
  </viewFields>
  <query>
    <Query>
      <OrderBy>
        <FieldRef Name='ID'/>
      </OrderBy>
    </Query>
  </query>
  <rowLimit>
    <RowLimit Paged='TRUE'>100</RowLimit>
  </rowLimit>
  <type>HTML</type>
  <makeViewDefault>false</makeViewDefault>
</AddView>
```

The protocol server returns the following **AddViewResponse** element (section 3.1.4.1.2.3) in the response message.

```
<AddViewResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <AddViewResult>
<View
Type='HTML'
Url='/Lists/ListA/ViewA.aspx'
Personal='FALSE'
DisplayName='ViewA'
DefaultView='FALSE'
Name='{4807AA00-190A-47CB-B2E0-8C13C04149BD}'>
      <Query>
        <OrderBy>
          <FieldRef Name='ID'/>
        </OrderBy>
      </Query>
      <ViewFields>
        <FieldRef Name='ID'/>
        <FieldRef Name='Title'/>
      </ViewFields>
        <RowLimit Paged='TRUE'>100</RowLimit>
    </View>
  </AddViewResult>
</AddViewResponse>
```

To verify that the previous list view was added to the list, the user retrieves the collection by calling the **GetViewCollection** Web method (section 3.1.4.4) with the following information in the request message.

```
<GetViewCollection xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <listName>{C2294E28-05B6-4486-804A-80F0189B992B}</listName>
```

```
</GetViewCollection>
```

The protocol server returns the following **GetViewCollectionResponse** element (section 3.1.4.4.2.2) in the response message.

```
<GetViewCollectionResponse xmlns='http://schemas.microsoft.com/sharepoint/soap/'>
          <GetViewCollectionResult >
            <Views >
              <View
   Name='{4807AA00-190A-47CB-B2E0-8C13C04149BD}'
        DefaultView='TRUE'
        Type='HTML'
        DisplayName='ViewA'
        Url='/Lists/ListA/ViewA.aspx'
        Level='1'
        BaseViewID='1'
        ContentTypeID='0x'
        ImageUrl='/_layouts/images/dlicon.png'>
              </View>
            </Views>
          </GetViewCollectionResult>
        </GetViewCollectionResponse>
```

The user updates the fields of the list view by calling the **UpdateView** Web method (section 3.1.4.6) with the following information in the request message.

```
        <UpdateView xmlns="http://schemas.microsoft.com/sharepoint/soap/">
          <listName>{C2294E28-05B6-4486-804A-80F0189B992B}</listName>
          <viewName>{4807AA00-190A-47CB-B2E0-8C13C04149BD}</viewName>
            <viewFields>
            <ViewFields >
              <FieldRef Name='ID'></FieldRef>
              <FieldRef Name='Title'></FieldRef>
              <FieldRef Name='Modified'></FieldRef>
             <FieldRef Name='Editor'></FieldRef>
            </ViewFields>
          </viewFields>               </UpdateView>
```

The protocol server returns the following **UpdateViewResponse** element (section 3.1.4.6.2.2) in the response message.

```
   <UpdateViewResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
     <UpdateViewResult>
       <View
Name='{4807AA00-190A-47CB-B2E0-8C13C04149BD}'
Type='HTML'
      DisplayName='ViewA'
Url='/Lists/ListA/ViewA.aspx'
Level='1'
      BaseViewID='1'
ContentTypeID='0x'
ImageUrl='/ layouts/images/dlicon.png'>
         <Query>
           <OrderBy>
             <FieldRef Name='ID'/>
           </OrderBy>
         </Query>
         <ViewFields>
           <FieldRef Name='ID'/>
           <FieldRef Name='Title'/>
           <FieldRef Name='Modified'/>
           <FieldRef Name='Editor'/>
```

```
        </ViewFields>
          <RowLimit Paged='TRUE'>100</RowLimit>
        <Formats>
          <FormatDef Type='RowHeight' Value='Auto'/>
          <Format Name='ID'>
            <FormatDef Type='ColWidth' Value='64'/>
          </Format>
          <Format Name='Title'>
            <FormatDef Type='WrapText' Value='1'/>
            <FormatDef Type='ColWidth' Value='52'/>
          </Format>
        </Formats>
        <Aggregations Value='Off'>
          <FieldRef Name='Title' Type='COUNT'/>
        </Aggregations>
      </View>
    </UpdateViewResult>
  </UpdateViewResponse>
```

The user deletes the list view by calling the **DeleteView** Web method (section 3.1.4.2) with the following information in the request message.

```
<DeleteView xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <listName>{C2294E28-05B6-4486-804A-80F0189B992B}</listName>
  <viewName>{4807AA00-190A-47CB-B2E0-8C13C04149BD}</viewName>
</DeleteView>
```

The protocol server returns the following **DeleteViewResponse** element (section 3.1.4.2.2.2) in the response message.

```
<DeleteViewResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/"/>
```

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. A link to wsswire.xsd denotes a reference to schemas as described in [MS-WSSCAML].

```
<?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/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:core="http://schemas.microsoft.com/sharepoint/soap/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
   <s:import namespace="http://schemas.microsoft.com/sharepoint/soap/"
schemaLocation="wsswire.xsd"/>
<s:element name="GetView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetViewHtml">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetViewHtmlResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewHtmlResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="DeleteView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0"/>
```

```
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="DeleteViewResponse">
  <s:complexType/>
</s:element>
<s:element name="AddView">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="viewName" type="s:string" minOccurs="0" maxOccurs="1"/>
<s:element minOccurs="0" maxOccurs="1" name="viewFields">
             <s:complexType>
                   <s:sequence>
                  <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
                    <s:complexType>
                     <s:sequence>
                      <s:element name="FieldRef" type="core:FieldRefDefinitionView"
minOccurs="0" maxOccurs="unbounded" />
                     </s:sequence>
                    </s:complexType>
                  </s:element>
                 </s:sequence>
               </s:complexType>
             </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="query">
             <s:complexType>
              <s:sequence>
<s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
               </s:sequence>
             </s:complexType>
           </s:element>
<s:element minOccurs="0" maxOccurs="1" name="rowLimit">
             <s:complexType>
                <s:sequence>
<s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
                </s:sequence>
               </s:complexType>
             </s:element>
      <s:element name="type" type="s:string" minOccurs="0" maxOccurs="1"/>
      <s:element name="makeViewDefault" type="s:boolean" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="AddViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="AddViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetViewCollection">
  <s:complexType>
    <s:sequence>
      <s:element name="listName" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetViewCollectionResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetViewCollectionResult" minOccurs="1" maxOccurs="1">
```

```
        <s:complexType>
          <s:sequence>
            <s:element name="Views" minOccurs="1" maxOccurs="1">
              <s:complexType>                <s:sequence>
                  <s:element name="View" minOccurs="0" maxOccurs="unbounded">
                    <s:complexType>
                      <s:attributeGroup ref="tns:ViewAttributeGroup"/>
                    </s:complexType>
                  </s:element>
                </s:sequence>
              </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="UpdateView">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="listName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
     <s:complexType>
    <s:sequence>
      <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
        </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="query">
  <s:complexType>
    <s:sequence>
      <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
             <s:element minOccurs="0" maxOccurs="1" name="viewFields">
  <s:complexType>
    <s:sequence>
       <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="aggregations">
  <s:complexType>
    <s:sequence>
      <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
            </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="formats">
  <s:complexType>
    <s:sequence>
      <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
            </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
```

```
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
            </s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
<s:element name="UpdateViewResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="tns:BriefViewDefinition" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="UpdateViewHtml">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="listName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
  <s:complexType>
    <s:sequence>
      <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
        </s:element>
<s:element minOccurs="0" maxOccurs="1" name="toolbar">
  <s:complexType>
    <s:sequence>
      <s:element name="Toolbar" type="core:ToolbarDefinition" minOccurs="0" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewHeader">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewHeader" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewBody">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewBody" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
```

```
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewFooter">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewFooter" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewEmpty">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewEmpty" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="rowLimitExceeded">
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimitExceeded" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
      </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="query">
  <s:complexType>
    <s:sequence>
      <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="viewFields">
  <s:complexType>
    <s:sequence>
       <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
```

```
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="aggregations">
  <s:complexType>
    <s:sequence>
      <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="formats">
  <s:complexType>
    <s:sequence>
      <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
          </s:sequence>
        </s:complexType>
      </s:element>
<s:element name="UpdateViewHtmlResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewHtmlResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="UpdateViewHtml2">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="listName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="viewProperties">
  <s:complexType>
    <s:sequence>
      <s:element name="View" type="tns:UpdateViewPropertiesDefinition" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
        </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="toolbar">
  <s:complexType>
    <s:sequence>
      <s:element name="Toolbar" type="core:ToolbarDefinition" minOccurs="0" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewHeader">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewHeader" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
```

```
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewBody">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewBody" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="viewFooter">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewFooter" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="viewEmpty">
  <s:complexType>
    <s:sequence>
      <s:element name="ViewEmpty" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element minOccurs="0" maxOccurs="1" name="rowLimitExceeded">
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimitExceeded" minOccurs="0" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:any minOccurs="0" maxOccurs="unbounded" namespace="##any"
processContents="skip" />
          </s:sequence>
          <s:anyAttribute processContents="skip" />
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
      </s:element>
            <s:element minOccurs="0" maxOccurs="1" name="query">
  <s:complexType>
    <s:sequence>
```

```
      <s:element name="Query" type="core:CamlQueryRoot" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="viewFields">
              <s:complexType>                <s:sequence>
                 <s:element name="ViewFields" minOccurs="0" maxOccurs="1">
                  <s:complexType>
                    <s:sequence>
                      <s:element name="FieldRef" type="core:FieldRefDefinitionView"
minOccurs="0" maxOccurs="unbounded" />
                    </s:sequence>
                  </s:complexType>
                 </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="aggregations">
  <s:complexType>
    <s:sequence>
      <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="formats">
  <s:complexType>
    <s:sequence>
      <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0"
maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="rowLimit">
  <s:complexType>
    <s:sequence>
      <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1"/>
</s:sequence>
  </s:complexType>
</s:element>
            <s:element minOccurs="0" maxOccurs="1" name="openApplicationExtension"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
<s:element name="UpdateViewHtml2Response">
  <s:complexType>
    <s:sequence>
      <s:element name="UpdateViewHtml2Result" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
            <s:element name="View" type="core:ViewDefinition" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="BriefViewDefinition" mixed="true">
  <s:sequence>
    <s:element name="Query" type="core:CamlQueryRoot" minOccurs="1" maxOccurs="1" />
    <s:element name="ViewFields" minOccurs="1" maxOccurs="1">
      <s:complexType>
        <s:sequence>
          <s:element name="FieldRef" type="core:FieldRefDefinitionView" minOccurs="0"
maxOccurs="unbounded" />
        </s:sequence>
       </s:complexType>
    </s:element>
```

```
    <s:element name="ViewData" minOccurs="0" maxOccurs="1">
      <s:complexType>
        <s:sequence>
          <s:element name="FieldRef" type="core:FieldRefDefinitionViewData" minOccurs="3"
maxOccurs="5" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="CalendarViewStyles" type="core:CalendarViewStylesDefinition"
minOccurs="0" maxOccurs="1" />
    <s:element name="RowLimit" type="core:RowLimitDefinition" minOccurs="0" maxOccurs="1" />
    <s:element name="Formats" type="core:ViewFormatDefinitions" minOccurs="0" maxOccurs="1"
/>
    <s:element name="Aggregations" type="core:AggregationsDefinition" minOccurs="0"
maxOccurs="1" form="unqualified" />
    <s:element name="ViewStyle" type="core:ViewStyleReference" minOccurs="0" maxOccurs="1" />
    <s:element name="OpenApplicationExtension" type="s:string" minOccurs="0" maxOccurs="1"
/>
  </s:sequence>
  <s:attributeGroup ref="tns:ViewAttributeGroup"/>
</s:complexType>
<s:complexType name="UpdateViewPropertiesDefinition">
  <s:attributeGroup ref="tns:UpdateViewAttributeGroup"/>
</s:complexType>
<s:attributeGroup name="ViewAttributeGroup">
  <s:attribute name="AggregateView" type="core:TRUEFALSE"  default="FALSE"/>
  <s:attribute name="BaseViewID" type="s:int" />
  <s:attribute name="CssStyleSheet" type="s:string" />
  <s:attribute name="DefaultView" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="DisplayName" type="s:string" />
  <s:attribute name="FailIfEmpty" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="FileDialog" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="FPModified" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="Hidden" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="List" type="core:UniqueIdentifierWithoutBraces" />
  <s:attribute name="Name" type="core:UniqueIdentifierWithBraces" />
  <s:attribute name="ContentTypeID" type="core:ContentTypeId" />
  <s:attribute name="OrderedView" type="core:TRUEFALSE" />
  <s:attribute name="DefaultViewForContentType" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="IncludeRootFolder" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="PageType" type="s:string" />
  <s:attribute name="Path" type="core:RelativeFilePath" />
  <s:attribute name="Personal" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="ReadOnly" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="RecurrenceRowset" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="RequiresClientIntegration" type="core:TRUEFALSE" default="FALSE"/>
  <s:attribute name="RowLimit" type="s:int" />
  <s:attribute name="ShowHeaderUI" type="core:TRUEFALSE"  default="FALSE" />
  <s:attribute name="Type" type="core:ViewType" default="HTML"/>
  <s:attribute name="Url" type="core:RelativeUrl"/>
  <s:attribute name="UseSchemaXmlToolbar" type="core:TRUEFALSE" default="FALSE"/>
  <s:attribute name="WebPartOrder" type="s:int" />
  <s:attribute name="WebPartZoneID" type="s:string" />
  <s:attribute name="FreeForm" type="core:TRUEFALSE" />
  <s:attribute name="ImageUrl" type="s:string" />
  <s:attribute name="SetupPath" type="core:RelativeFilePath" />
  <s:attribute name="ToolbarTemplate" type="s:string" />
  <s:attribute name="MobileView" type="core:TRUEFALSE" default="FALSE"/>
  <s:attribute name="MobileDefaultView" type="core:TRUEFALSE" />
  <s:attribute name="MobileUrl" type="core:RelativeUrl" />
  <s:attribute name="Level" type="core:ViewPageLevel" default="1" />
  <s:attribute name="FrameState" type="s:string" default="Normal" />
  <s:attribute name="IsIncluded" type="core:TRUEFALSE" default="TRUE" />
  <s:attribute name="IncludeVersions" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="HackLockWeb" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="ModerationType" type="core:ViewModerationType" default="" />
  <s:attribute name="Scope" type="core:ViewScope" default="" />
  <s:attribute name="Threaded" type="core:TRUEFALSE" default="FALSE" />
```

```
  <s:attribute name="TabularView" type="core:FALSE_Case_Insensitive_Else_Anything"
default="TRUE" />
</s:attributeGroup>
<s:attributeGroup name="UpdateViewAttributeGroup">
  <s:attribute name="DefaultView" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="DisplayName" type="s:string" />
  <s:attribute name="FPModified" type="core:TRUEFALSE" default="FALSE" />
  <s:attribute name="Scope" type="core:ViewScope" default="" />
</s:attributeGroup>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetViewSoapIn">
    <wsdl:part name="parameters" element="tns:GetView" />
  </wsdl:message>
  <wsdl:message name="GetViewSoapOut">
    <wsdl:part name="parameters" element="tns:GetViewResponse" />
  </wsdl:message>
  <wsdl:message name="GetViewHtmlSoapIn">
    <wsdl:part name="parameters" element="tns:GetViewHtml" />
  </wsdl:message>
  <wsdl:message name="GetViewHtmlSoapOut">
    <wsdl:part name="parameters" element="tns:GetViewHtmlResponse" />
  </wsdl:message>
  <wsdl:message name="DeleteViewSoapIn">
    <wsdl:part name="parameters" element="tns:DeleteView" />
  </wsdl:message>
  <wsdl:message name="DeleteViewSoapOut">
    <wsdl:part name="parameters" element="tns:DeleteViewResponse" />
  </wsdl:message>
  <wsdl:message name="AddViewSoapIn">
    <wsdl:part name="parameters" element="tns:AddView" />
  </wsdl:message>
  <wsdl:message name="AddViewSoapOut">
    <wsdl:part name="parameters" element="tns:AddViewResponse" />
  </wsdl:message>
  <wsdl:message name="GetViewCollectionSoapIn">
    <wsdl:part name="parameters" element="tns:GetViewCollection" />
  </wsdl:message>
  <wsdl:message name="GetViewCollectionSoapOut">
    <wsdl:part name="parameters" element="tns:GetViewCollectionResponse" />
  </wsdl:message>
  <wsdl:message name="UpdateViewSoapIn">
    <wsdl:part name="parameters" element="tns:UpdateView" />
  </wsdl:message>
  <wsdl:message name="UpdateViewSoapOut">
    <wsdl:part name="parameters" element="tns:UpdateViewResponse" />
  </wsdl:message>
  <wsdl:message name="UpdateViewHtmlSoapIn">
    <wsdl:part name="parameters" element="tns:UpdateViewHtml" />
  </wsdl:message>
  <wsdl:message name="UpdateViewHtmlSoapOut">
    <wsdl:part name="parameters" element="tns:UpdateViewHtmlResponse" />
  </wsdl:message>
  <wsdl:message name="UpdateViewHtml2SoapIn">
    <wsdl:part name="parameters" element="tns:UpdateViewHtml2" />
  </wsdl:message>
  <wsdl:message name="UpdateViewHtml2SoapOut">
    <wsdl:part name="parameters" element="tns:UpdateViewHtml2Response" />
  </wsdl:message>
  <wsdl:portType name="ViewsSoap">
    <wsdl:operation name="GetView">
      <wsdl:input message="tns:GetViewSoapIn" />
      <wsdl:output message="tns:GetViewSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetViewHtml">
      <wsdl:input message="tns:GetViewHtmlSoapIn" />
      <wsdl:output message="tns:GetViewHtmlSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="DeleteView">
```

```
      <wsdl:input message="tns:DeleteViewSoapIn" />
      <wsdl:output message="tns:DeleteViewSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="AddView">
      <wsdl:input message="tns:AddViewSoapIn" />
      <wsdl:output message="tns:AddViewSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetViewCollection">
      <wsdl:input message="tns:GetViewCollectionSoapIn" />
      <wsdl:output message="tns:GetViewCollectionSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="UpdateView">
      <wsdl:input message="tns:UpdateViewSoapIn" />
      <wsdl:output message="tns:UpdateViewSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml">
      <wsdl:input message="tns:UpdateViewHtmlSoapIn" />
      <wsdl:output message="tns:UpdateViewHtmlSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml2">
      <wsdl:input message="tns:UpdateViewHtml2SoapIn" />
      <wsdl:output message="tns:UpdateViewHtml2SoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ViewsSoap" type="tns:ViewsSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetView">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetView"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetViewHtml">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetViewHtml"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteView">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteView"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddView">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/AddView"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetViewCollection">
```

```
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetViewCollection" style="document"
/>
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateView">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateView"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml2">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml2" 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="ViewsSoap12" type="tns:ViewsSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetView">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetView"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetViewHtml">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetViewHtml"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteView">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteView"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
```

```
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="AddView">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/AddView"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetViewCollection">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetViewCollection" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateView">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateView"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateViewHtml2">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/UpdateViewHtml2" 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.

- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Windows SharePoint Services 2.0
- Windows SharePoint Services 3.0
- Microsoft SharePoint Foundation 2010
- Microsoft SharePoint Foundation 2013
- Microsoft Office 2016
- Microsoft SharePoint Server 2016

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.

<1> Section 3: Microsoft products use a **detail** element instead of the **Detail** element in SOAP 1.2.

<2> Section 3.1.4.1.2.3: SharePoint Foundation 2010 appends ">" after the **View** element.

<3> Section 3.1.4.3.2.2: In SharePoint Foundation 2010 and SharePoint Foundation 2013, when this method is called after **UpdateViewHtml2** (section 3.1.4.8) and the type of the view is HTML, the value of **OpenApplicationExtension** is returned as the value of the **View** element.

<4> Section 3.1.4.8: Windows SharePoint Services 2.0 does not support this operation.

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