[MS-VGSP]:

Visio Graphics Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Editorial	Revised and edited the technical content
6/7/2010	1.03	Editorial	Revised and edited the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.04	No Change	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.04	No Change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No Change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No Change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.04	No Change	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.5	Minor	Clarified the meaning of the technical content.
1/20/2012	2.0	Major	Significantly changed the technical content.
4/11/2012	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.0	Major	Significantly changed the technical content.
2/11/2013	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	3.1	Minor	Clarified the meaning of the technical content.
8/24/2015	4.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Visio Graphics Service Protocol enables a protocol client to communicate with a protocol server to retrieve information about a web drawing that is stored in a document repository.

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

1.1 Glossary

The following terms are specific to this document:

Coordinated Universal Time (UTC): A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

data connection: A collection of information, such as the type and location, that defines how to connect to an external data source, such as a database, web service, SharePoint list, or XML file.

data provider: A known data source that is specific to a target type and that provides data to a collector type.

data source: A database, web service, disk, file, or other collection of information from which data is queried or submitted. Supported data sources vary based on application and data provider.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the Active Directory service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

drawing: A collection of drawing objects, such as shapes, curves, or WordArt, that are viewed together as a single image.

embedded object: An object that is created by using one application and is hosted in a document that was created by using another application. Embedding an object, rather than inserting or pasting it, ensures that the object retains its original format. Users can double-click an embedded object and edit it with the toolbars and menus from the application that was used to create it. See also Object Linking and Embedding (OLE).

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

Extensible Application Markup Language (XAML): A declarative XML-based language that is used to represent a tree of objects. XAML markup is stored in .xaml files or, for workflow schemas, .xoml files.

font: An object that defines the graphic design, or formatting, of a collection of numbers, symbols, and letters. A font specifies the style (such as bold and strikeout), size, family (a typeface such as Times New Roman), and other qualities to describe how the collection is drawn.

fully qualified class name: A class name that includes namespace information. Use of a fully qualified class name ensures that the class name is treated as unique.

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).

hyperlink: A relationship between two anchors, as described in [RFC1866].

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].

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC4627]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

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.

Office data connection (ODC) file: A file that stores information about a connection to a data source, such as an Access database, worksheet, or text file. This file facilitates data source administration.

pixel: A discrete unit of display on a computer display device.

Portable Network Graphics (PNG): A bitmap graphics file format that uses lossless data compression and supports variable transparency of images (alpha channels) and control of image brightness on different computers (gamma correction). PNG-format files have a .png file name extension.

primary key: A field or set of fields that uniquely identifies each record in a table. A primary key cannot contain a null value.

row: A single set of data that is displayed horizontally in a worksheet or a table.

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.

Status-Code: A 3-digit integer result code in an HTTP response message, as described in [RFC2616].

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

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

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].

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or View (Microsoft Business Connectivity Services).

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

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

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

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

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

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

zoom level: The degree to which a portion of an image, document, or other screen object is made to appear closer or farther away relative to its default appearance. This value is usually expressed as a percentage of the default appearance.

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-DSDIFFGRAM] Microsoft Corporation, "SharePoint Web Services: DataSet DiffGram Structure".

[MS-SLXV] Microsoft Corporation, "Silverlight XAML Vocabulary Specification 2008".

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

[MS-VGSFF] Microsoft Corporation, "Visio Graphics Service File Format (.vdw) Specification".

[MS-VSDX] Microsoft Corporation, "Visio Graphics Service VSDX File Format".

[MS-WSSTS] Microsoft Corporation, "Windows SharePoint Services".

[RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., Eds., "Uniform Resource Locators (URL)", RFC 1738, December 1994, http://www.ietf.org/rfc/rfc1738.txt

[RFC1980] Seidman, J., "A Proposed Extension to HTML : Client-Side Image Maps", RFC 1980, August 1996, http://www.rfc-editor.org/rfc/rfc1980.txt

[RFC2083] Boutell, T., et al., "PNG (Portable Network Graphics) Specification Version 1.0", RFC 2083, March 1997, http://www.ietf.org/rfc/rfc2083.txt

[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

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006, http://www.ietf.org/rfc/rfc4627.txt

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

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

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

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

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

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

1.2.2 Informative References

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

[RFC1952] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, May 1996, http://www.rfc-editor.org/rfc/rfc1952.txt

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

[RFC3629] Yergeau, F., "UTF-8, A Transformation Format of ISO 10646", STD 63, RFC 3629, November 2003, http://www.ietf.org/rfc/rfc3629.txt

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

1.3   Overview

The protocol enables a protocol client to request and receive a drawing page (section 1.3.2) of a web drawing from the protocol server in either raster or vector format. The supported raster format is the **Portable Network Graphics (PNG)** format, as described in [RFC2083]. The supported vector format is **Extensible Application Markup Language (XAML)**, as described in [MS-SLXV].
Additionally, the protocol enables a protocol client to add, edit, or delete comments associated with a drawing page or its shapes.
A typical scenario for using this protocol is an application that retrieves and displays web drawings, and collects collaborative content feedback in comments. Typically a web drawing is contained in a file stored in a document repository.

1.3.1   Web Drawing

A **web drawing** is a collection of drawings page (section 1.3.2), shapes, resources, comments, and information that can be rendered as a **drawing** in a web browser. Based on the format of the file that contains the web drawing, the protocol makes the distinction between the following types of web drawings.

Type	Description
VGSFF	A web drawing contained in a file that conforms to the format specified in [MS-VGSFF] section 2.2.1
VSDX	A web drawing contained in a file that conforms to the format specified in [MS-VSDX] section 2.2.1

1.3.2   Drawing Page

A **drawing page** is a collection of shapes (section 1.3.3) that are viewed together.

Drawing pages of **VGSFF** web drawings (section 1.3.1) are specified in [MS-VGSFF] section 2.2.2.

Drawing pages of **VSDX** web drawings are specified in [MS-VSDX] section 2.2.2.

1.3.2.1   Foreground Page

Foreground pages of **VSDX** web drawings are specified in [MS-VSDX] section 2.2.2.4.

1.3.3 Shape

A **shape** is a collection of geometry, formatting, text, images, hyperlinks, and shape data in a drawing page (section 1.3.2).

Shapes contained in **VGSFF** web drawings are specified in [MS-VGSFF] section 2.2.3.

Shapes contained in **VSDX** web drawings are specified in [MS-VSDX] section 2.2.3.

1.3.4 Shape Data

A shape (section 1.3.3) can have data associated with it that provides information about its meaning. Shape data (section 1.3.4) can come from information stored in the file refreshed from **data providers** or from recalculations of an evaluated formula. The shape data is also formatted for display.

Shape data items contained in **VGSFF** web drawings (section 1.3.1) are specified in [MS-VGSFF] section 2.2.3.5

Shape data items contained in **VSDX** web drawings are specified in [MS-VSDX] section 2.2.3.6.

1.3.5 Comment

A **comment** is a plain text annotation in a web drawing. Each comment specifies an author, and a relationship with a drawing page (section 1.3.2) or a specific shape on the drawing page.

Comments contained in **VSDX** web drawings are specified in [MS-VSDX] section 2.2.9.

Comments are not contained in **VGSFF** web drawings.

1.4 Relationship to Other Protocols

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

The following diagram shows the underlying messaging and transport stack used by the protocol:
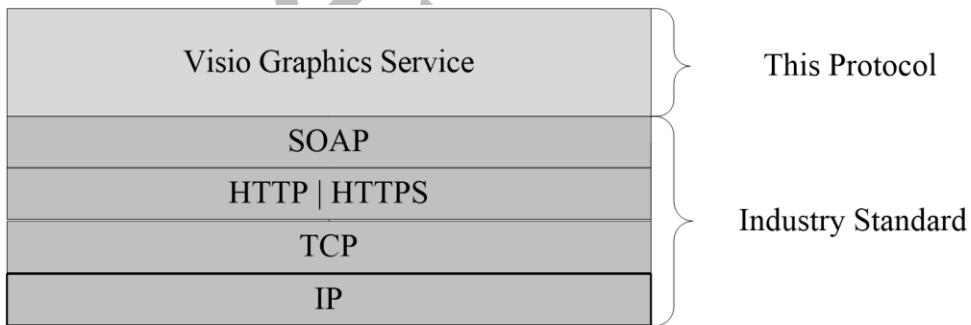
Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more **endpoint URIs** that are known by protocol clients. The endpoint URI of the protocol server and the transport that is used by

the protocol server are either known by the protocol client or obtained by using the discovery mechanism that is described in [MS-SPTWS].

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

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

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

1.6   Applicability Statement

This protocol is designed to support retrieval of web drawings, and collecting collaborative content feedback in comments.

1.7   Versioning and Capability Negotiation

This specification covers versioning issues in the following areas:

- **Supported Transports**: This protocol can be implemented by using transports that support sending Simple Object Access Protocol (SOAP) messages, as specified in section 2.1.

- **Protocol Versions**: This protocol is not versioned.

Capability Negotiation: This protocol does not support version negotiation.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2 Messages

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

2.1 Transport

Protocol servers MUST support Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), or **TCP**.

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

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

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

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

This protocol does not define any means for activating a protocol server or protocol client. The protocol server MUST be configured and begin listening in an implementation-specific way. In addition, the protocol client MUST know the format and transport that is used by the server, for example, the SOAP format over an HTTP transport.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
q1	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer	
tns1	http://schemas.microsoft.com/2003/10/Serialization/	
tns2	http://tempuri.org/	

Prefix	Namespace URI	Reference
tns3	http://tempuri.org/Imports	
wsaw	http://www.w3.org/2006/05/addressing/wsdl	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

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

2.2.3 Elements

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

Element	Description
VisioGraphicsServiceFault	The **VisioGraphicsServiceFault** element, with complex type **VisioGraphicsServiceFault** (section 2.2.4.5), contains information about errors that occur during protocol server operations.

2.2.3.1 VisioGraphicsServiceFault

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **VisioGraphicsServiceFault** element, with complex type **VisioGraphicsServiceFault** (section 2.2.4.5), contains information about errors that occur during protocol server operations.

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

2.2.4 Complex Types

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

Complex type	Description
AddonDataSource	The **AddonDataSource** complex type contains information about a data provider implemented by a protocol client and used in a web drawing.
ArrayOfAddonDataSource	The **ArrayOfAddonDataSource** complex type contains an array of

Complex type	Description
	AddonDataSource elements (section 2.2.4.1).
BaseRequestContract	The **BaseRequestContract** complex type contains information used to specify the request parameters for a protocol server operation to retrieve a **VGSFF** web drawing.
DiagramBase	The **DiagramBase** complex type contains information used to specify response data for a protocol server operation to retrieve a **VGSFF** web drawing.
VisioGraphicsServiceFault	The **VisioGraphicsServiceFault** complex type contains information about errors that occur during protocol server operations.

2.2.4.1 AddonDataSource

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **AddonDataSource** complex type contains information about a data provider implemented by a protocol client and used in a web drawing.

```
<xs:complexType name="AddonDataSource" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="ConnectionString" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Data" nillable="true">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="xs:schema"/>
          <xs:any/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element minOccurs="0" name="Id" type="xs:int"/>
    <xs:element minOccurs="0" name="QueryString" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

ConnectionString: A string that specifies arguments that define the location of the data provider and how to connect to it. There MUST be exactly one instance of this element, and it MUST NOT be null. It MUST contain the **fully qualified class name** of the data provider.

Data: Specifies the data retrieved from the data provider. It MUST be encoded in the DiffGram [MS-DSDIFFGRAM] format, and it MUST NOT be null.

xs:schema: Specifies the schema for the **Data** element.

Id: An integer that specifies the identifier of this **AddonDataSource**. It MUST be greater than or equal to zero, and it MUST be unique across all the **AddonDataSource** elements in the containing **ArrayOfAddonDataSource** array (section 2.2.4.2). There MUST be exactly one instance of this element.

QueryString: A string that specifies information used to call and retrieve data from the data provider. There MUST be exactly one instance of this element, and it MUST NOT be null.

2.2.4.2 ArrayOfAddonDataSource

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **ArrayOfAddonDataSource** complex type contains an array of **AddonDataSource** elements (section 2.2.4.1).

```
<xs:complexType name="ArrayOfAddonDataSource" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="AddonDataSource" nillable="true"
type="tns:AddonDataSource"/>
  </xs:sequence>
</xs:complexType>
```

AddonDataSource: Specifies an **AddonDataSource** element. This element MUST NOT be null.

2.2.4.3 BaseRequestContract

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **BaseRequestContract** complex type contains information used to specify the request parameters for a protocol server operation to retrieve a **VGSFF** web drawing.

```
<xs:complexType name="BaseRequestContract" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
    <xs:element minOccurs="0" name="DisableRefresh" type="xs:boolean"/>
    <xs:element minOccurs="0" name="FileUri" nillable="true" type="xs:anyURI"/>
    <xs:element minOccurs="0" name="PageNumber" type="xs:int"/>
    <xs:element minOccurs="0" name="TimeLastModified" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
```

DataSources: An **ArrayOfAddonDataSource** element (section 2.2.4.2) that specifies the data providers implemented by a protocol client, and used in the web drawing. If there are no data providers, **DataSources** MUST have zero child elements. If **DataSources** has child elements, the **Data** field of each **AddonDataSource** element (section 2.2.4.1) MUST NOT be null.

DisableRefresh: A Boolean that specifies whether the requested drawing page (section 1.3.2) is refreshed by the protocol server. A value of "true" indicates that the returned drawing page is not refreshed; a value of "false" indicates that the returned drawing page is refreshed. There MUST be exactly one instance of this element.

FileUri: An **anyURI** element that specifies the **URL** of the file in a document repository that contains the **VGSFF** web drawing being requested. There MUST be exactly one instance of this element, and it MUST NOT be null.

PageNumber: An integer that specifies the one-based index of a drawing page in the web drawing. It MUST be greater than zero and less than or equal to the total number of drawing pages (section 1.3.2) in the requested web drawing. There MUST be exactly one instance of this element.

TimeLastModified: An optional **dateTime** element that specifies the date and time, in **Coordinated Universal Time (UTC)** format, when the file in a document repository that contains the **VGSFF** web drawing, was last modified. If the attribute is present, there MUST be exactly one instance of this element.

2.2.4.4 DiagramBase

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **DiagramBase** complex type contains information used to specify response data for a protocol server operation to retrieve a **VGSFF** web drawing.

```xml
<xs:complexType name="DiagramBase" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="CacheAge" type="xs:int"/>
    <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
  </xs:sequence>
</xs:complexType>
```

CacheAge: An integer that specifies the duration in minutes for which the processed drawing page (section 1.3.2) is cached in memory by the protocol server. It MUST be greater than or equal to zero. There MUST be exactly one instance of this element.

DataSources: An **ArrayOfAddonDataSource** element (section 2.2.4.2) that specifies the data providers implemented by a protocol client, and used in the web drawing. If there are no data providers, **DataSources** MUST have zero child elements. If **DataSources** has child elements, the **Data** field of each **AddonDataSource** element (section 2.2.4.1) MUST NOT be empty.

2.2.4.5 VisioGraphicsServiceFault

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **VisioGraphicsServiceFault** complex type contains information about errors that occur during protocol server operations.

```xml
<xs:complexType name="VisioGraphicsServiceFault" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Details" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Error" type="tns:FaultCode"/>
    <xs:element minOccurs="0" name="Flags" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

Details: A string that specifies a detailed description of an error. There MUST be exactly one instance of this element.

Error: A **FaultCode** simple type (section 2.2.5.1) that specifies the type of error. There MUST be exactly one instance of this element.

Flags: An integer that specifies additional information related to the **FaultCode**. A nonzero value indicates an error occurred during mapping **rows** of data to shapes when processing the request for a drawing page (section 1.3.2) of a **VGSFF** web drawing. There MUST be exactly one instance of this element. If **FaultCode** is not equal to **DataBindingConflict**, the value MUST be "0x00000000". If **FaultCode** is equal to **DataBindingConflict**, then the value MUST be a bitwise OR combination of one or more of the values in the following table, or the value "0x40000000", which indicates a generic data binding failure.

Value	Description
0x00000001	A column was deleted in the **data source**.
0x00000002	A column's data type was changed in the data source.
0x00000004	A row was deleted in the data source.
0x00000008	A duplicate row was found in the data source.

Value	Description
0x00000010	A **primary key** column was deleted in the data source.

2.2.5 Simple Types

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

Simple type	Description
FaultCode	The **FaultCode** simple type contains information about a specific error that occurs during a protocol operation. It MUST be a valid value as defined in the table in this section.

2.2.5.1 FaultCode

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **FaultCode** simple type contains information about a specific error that occurs during a protocol operation.

It MUST be a valid value as defined in the table in this section.

```
<xs:simpleType name="FaultCode" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DataConnectivityError"/>
    <xs:enumeration value="DataSourceConnectionError"/>
    <xs:enumeration value="DataSourcePermissionError"/>
    <xs:enumeration value="EmptyRasterDiagramRequest"/>
    <xs:enumeration value="EmptyVectorDiagramRequest"/>
    <xs:enumeration value="OfficeDataConnectionFileNotFound"/>
    <xs:enumeration value="OfficeDataConnectionFileRetrievalError"/>
    <xs:enumeration value="OfficeDataConnectionFileAccessDenied"/>
    <xs:enumeration value="IncompatibleDomain"/>
    <xs:enumeration value="WssViewAccessDenied"/>
    <xs:enumeration value="WssListAccessDenied"/>
    <xs:enumeration value="InvalidPageNumber"/>
    <xs:enumeration value="InvalidResolution"/>
    <xs:enumeration value="InvalidFileUrl"/>
    <xs:enumeration value="RasterizationError"/>
    <xs:enumeration value="InvalidFileExtension"/>
    <xs:enumeration value="EmptyServerData"/>
    <xs:enumeration value="OpenServerFileFailed"/>
    <xs:enumeration value="NoReadPermissionToFile"/>
    <xs:enumeration value="KeyNotFoundFromCollection"/>
    <xs:enumeration value="ExceededMaxDiagramSize"/>
    <xs:enumeration value="IncompatibleVersion"/>
    <xs:enumeration value="DataProviderNotTrusted"/>
    <xs:enumeration value="DataBindingConflict"/>
    <xs:enumeration value="UnableToGetExternalDataSourceCredentials"/>
    <xs:enumeration value="UnattendedAccountTypeMismatch"/>
    <xs:enumeration value="RecalcTimedOut"/>
    <xs:enumeration value="LicenseExpired"/>
    <xs:enumeration value="InvalidZoomLevel"/>
    <xs:enumeration value="VSDXFileLoadGenericError"/>
```

```
<xs:enumeration value="VSDXFileLoadPermissionError"/>
<xs:enumeration value="VSDXFileSizeError"/>
<xs:enumeration value="WebPartConfigError"/>
<xs:enumeration value="EmptyRasterPageRequest"/>
<xs:enumeration value="EmptyRasterPageItemRequest"/>
<xs:enumeration value="VDWNotSupportExternalList"/>
<xs:enumeration value="ExternalListConfigurationNotSupported"/>
<xs:enumeration value="IncompatibleDomain_15"/>
<xs:enumeration value="FileNotLockedOnServer"/>
<xs:enumeration value="FileAlreadyCheckedOutOnServer"/>
<xs:enumeration value="FileAlreadyLockedOnServer"/>
<xs:enumeration value="DocumentCheckoutRequired"/>
<xs:enumeration value="ExcelFileNotInSharePoint"/>
<xs:enumeration value="PageDrawingAreaSizeUnsupported"/>
<xs:enumeration value="PageImageSizeUnsupported"/>
<xs:enumeration value="PageFailToRaster"/>
<xs:enumeration value="CannotRetrieveExternalListData"/>
</xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **FaultCode** simple type.

Value	Meaning
DataConnectivityError	An error occurred while trying to connect to data sources specified in the document.
DataSourceConnectionError	An error occurred while trying to connect to data sources specified in the **VSDX Web drawing** (section 1.3.1).
DataSourcePermissionError	The user does not have permission to connect to data sources specified in the **VSDX Web drawing**.
EmptyRasterDiagramRequest	A required argument to the **GetRasterDiagram** (section 3.1.4.2) **WSDL operation** is empty.
EmptyVectorDiagramRequest	A required argument to the **GetVectorDiagram** WSDL operation is empty.
OfficeDataConnectionFileNotFound	The **Office data connection (ODC) file** specified in the document could not be found.
OfficeDataConnectionFileRetrievalError	The Office data connection (ODC) file URL specified in the document is not a well formed URL as specified in [RFC1738].
OfficeDataConnectionFileAccessDenied	Access to the Office data connection (ODC) file specified in the document was denied.
IncompatibleDomain	The specified **VGSFF Web drawing** is referring to a **list** data source that is not in a compatible **domain**.
WssViewAccessDenied	The user does not have permissions to the **view** used as external data source by the web drawing.
WssListAccessDenied	The user does not have permissions to the list used as external data source by the web drawing.
InvalidPageNumber	The page index is less than one.
InvalidResolution	The **Resolution** specified in **RasterDiagramRequestContract** (section 3.1.4.1.3.1) is invalid.
InvalidFileUrl	Unused and MUST be ignored.

Value	Meaning
RasterizationError	An error occurred during rasterization of the document.
InvalidFileExtension	The file reference does not refer to a file with a valid extension.
EmptyServerData	The data in the document is empty or invalid.
OpenServerFileFailed	The document could not be opened.
NoReadPermissionToFile	Access to the document was denied.
KeyNotFoundFromCollection	The names of drawing pages (section 1.3.2) cannot be retrieved from the document.
ExceededMaxDiagramSize	The document size exceeded the limit configured by the service administrator.
IncompatibleVersion	The document version is incompatible with the server.
DataProviderNotTrusted	The document requires a connection to data provider(s) that are not allowed by the service administrator.
DataBindingConflict	Unable to update the graphical elements that are mapped to data while processing a request for a drawing page (section 1.3.2) of a **VGSFF Web drawing**.
UnableToGetExternalDataSourceCredentials	Credentials required to connect to data sources (1) specified in the document could not be retrieved.
UnattendedAccountTypeMismatch	The default credentials used for connecting to data sources are of invalid type.
RecalcTimedOut	The document recalculation duration exceeded the limit configured by the service administrator.
LicenseExpired	Visio Graphics Service license has expired.
InvalidZoomLevel	Unused and MUST be ignored.
VSDXFileLoadGenericError	An error occurred opening the **VSDX Web drawing**.
VSDXFileLoadPermissionError	The user does not have permission to read the **VSDX** Web drawing.
VSDXFileSizeError	The **VSDX** document size exceeded the limit configured by the service administrator.
WebPartConfigError	An invalid request occurred for a **VSDX Web drawing**.
EmptyRasterPageRequest	The **rasterPageRequestContract** element of the **GetRasterPage** element (section 3.1.4.3.2.1) is null.
EmptyRasterPageItemRequest	The **rasterPageItemRequestContract** element of the **GetRasterPageItem** element (section 3.1.4.4.2.1) is null.
VDWNotSupportExternalList	A **VGSFF Web drawing** does not support an external list ([MS-WSSTS] section 2.1.2.7) data source.
ExternalListConfigurationNotSupported	The configuration of an external list ([MS-WSSTS] section 2.1.2.7) data source is not supported.
IncompatibleDomain_15	The specified **VSDX** web drawing (section 1.3.1) is referring to a list ([MS-WSSTS] section 2.1.2.7) data source that is not in a compatible domain.

Value	Meaning
FileNotLockedOnServer	The **SaveComments** operation (section 3.1.4.5) failed to acquire a lock on the target document.
FileAlreadyCheckedOutOnServer	The **SaveComments** operation (section 3.1.4.5) failed to acquire a lock on the target document because it was checked-out ([MS-WSSTS] section 2.1.2.11.1.3) by a different client.
FileAlreadyLockedOnServer	The **SaveComments** operation (section 3.1.4.5) failed because there is an already existing exclusive lock on the target document or a schema lock with a different schema lock identifier.
DocumentCheckoutRequired	The **SaveComments** operation (section 3.1.4.5) failed because the target document was not checked-out ([MS-WSSTS] section 2.1.2.11.1.3).
ExcelFileNotInSharePoint	An Excel file used as data source is not located in a document library ([MS-WSSTS] section 2.1.2.7.1)
PageDrawingAreaSizeUnsupported	The protocol server was unable to render the drawing page because it exceeds the supported size limit.
PageImageSizeUnsupported	The protocol server was unable to create an image item (section 3.1.1) because the image exceeds the supported size limit.
PageFailToRaster	An error has occurred while the protocol server was rendering the **VSDX Web drawing**.
CannotRetrieveExternalListData	The protocol server failed to retrieve data from an external list ([MS-WSSTS] section 2.1.2.7) data source.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

2.2.9 Common Data Structures

The following table summarizes the set of data structures specific to messages that are associated with **VSDX** web drawings.

Data Structure	Description
ShapeInfo	A string that is an XML fragment containing information about shapes, drawing pages (section 1.3.2), and web drawing.
Comments	A string that is an XML fragment containing comments.
CommentData	A string that is an XML fragment containing the comment action input data.

Data Structure	Description
CommentResult	A string that is an XML fragment containing the result of a comment action.

2.2.9.1 ShapeInfo

The **ShapeInfo** element is a string that specifies information about the shapes (section 1.3.3), drawing pages (section 1.3.2), and web drawing (section 1.3.1) of a **VSDX** web drawing.<1> The syntax of this data structure is an XML fragment. Elements of this fragment are listed in the following sections.

The set of elements returned is determined by the contents of the **web drawing**, as described in the following table.

Element	Description
Page	The information about the requested drawing page, and the information about additional pages in the web drawing.
Pages	A list of published foreground pages (section 1.3.2.1) in the web drawing.
PageInfo	The information about the identity of a drawing page in the web drawing.
ShapeInfo	The information about the relevant shape, including shape data (section 1.3.4), hyperlinks, text and identifying information. A relevant shape is selectable, as specified in [MS-VSDX] section 2.2.3.4, and contains hyperlinks, shape data items, or comments (section 1.3.5).
Hyperlinks	The list of each visible **hyperlink** element associated with a shape.
Hyperlink	The hyperlink associated with a shape.
ShapeDataItems	The list of visible shape data items associated with a shape.
ShapeData	The shape data item associated with a shape.
Shapes	The list of geometric outline information for each relevant shape in the web drawing. A relevant shape is selectable, as specified in [MS-VSDX] section 2.2.3.4, and contains hyperlinks, shape data items, or comments.
Shape	The geometric outline information for a shape.
Path	The geometric outline information in the form of a path.

2.2.9.1.1 Page

The **Page** complex type specifies information about the shapes (section 1.3.3) on a drawing page (section 1.3.2) in a web drawing (section 1.3.1). One element is returned for the web drawing.

```
<xs:complexType name="Page" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Pages" minOccurs="1" maxOccurs="1" type="Pages"/>
    <xs:element name="ShapeInfo" minOccurs="0" maxOccurs="unbounded" type="ShapeInfo"/>
    <xs:element name="Shapes" minOccurs="0" maxOccurs="1" type="Shapes"/>
  </xs:sequence>
  <xs:attribute name="ContentBounds" type="xs:string" use="required"/>
  <xs:attribute name="DataBindingConflict" type="xs:boolean" use="required"/>
  <xs:attribute name="DocumentID" type="xs:string" use="required"/>
```

```
            <xs:attribute name="DocumentTimestamp" type="xs:string" use="required"/>
            <xs:attribute name="Name" type="xs:string" use="optional"/>
            <xs:attribute name="ReadOnly" type="xs:boolean" use="required"/>
            <xs:attribute name="RefreshWarning" type="xs:boolean" use="required"/>
            <xs:attribute name="RequiresCheckout" type="xs:boolean" use="optional"/>
            <xs:attribute name="UnsupportedFeatures" type="xs:int" use="optional"/>
            <xs:attribute name="ViewOnly" type="xs:boolean" use="required"/>
      </xs:complexType>
```

ContentBounds: A string that specifies the bounding rectangle of the drawing page as specified in the following table.

Name	Description
x	An integer that specifies the upper left x-coordinate of the bounding rectangle.
y	An integer that specifies the upper left y-coordinate of the bounding rectangle.
width	A positive integer that specifies the width of the bounding rectangle.
height	A positive integer that specifies the height of the bounding rectangle.

The coordinates of a bounding rectangle MUST be relative to the top, left corner of the window that contains the drawing page. All the values MUST be specified in **pixels**.

The string MUST be formatted using the **JavaScript Object Notation (JSON)**, as specified in [RFC4627], as follows:

```
{"x":x-val,"y":y-val,"width":width-val,"height":height-val}
```

where **x-val**, **y-val**, **width-val**, and **height-val** are integers specifying the x, y, width, and height fields, respectively.

DataBindingConflict: A Boolean that specifies the status of retrieving data from data providers for the drawing page. The value MUST be one of the following:

Value	Description
0	The drawing page does not have any data binding conflicts.
1	The drawing page does have data binding conflicts.

DocumentID: A string that specifies a **GUID** that identifies the file in a document repository that contains the web drawing being requested.

DocumentTimestamp: A string that specifies an opaque identifier for the version of the web drawing. The value MUST NOT be an empty string.

Name: A string that specifies the name of the drawing page. The value MUST NOT be an empty string. It MUST match the **Name** or **NameU** attribute of the **Page_Type** elements in the **Pages XML Part**, as specified in [MS-VSDX] section 2.2.2.1, of the web drawing. It MUST match a name in the **Name** attribute of one of the **PageInfo** child elements of the **Pages** element (section 2.2.9.1.2).

ReadOnly: A Boolean that specifies the read-only state of the file in the document repository. A value of one indicates that the file is read-only; a value of zero indicates the file is not read-only.

RefreshWarning: An optional Boolean that specifies whether the drawing page was refreshed by the protocol server. A nonzero value indicates that data connections exist for the drawing page but that the drawing page was not refreshed from the data providers associated with these data connections.

RequiresCheckout: A Boolean that specifies whether or not the file requires check-out from the document repository. A value of one indicates that the file requires check-out; a value of zero indicates that the file does not require check-out.

UnsupportedFeatures: An integer that specifies features in the file that are not supported by the server. The value MUST be zero, or a bitwise or combination of one or more of the following values.

Value	Description
0x01	The file contains legacy ink.
0x02	The file contains one or more reviewer pages.
0x04	The file contains macros.
0x08	The file contains one or more **embedded objects**.
0x10	The file contains unsupported cell recalculations.

ViewOnly: A Boolean that specifies whether or not the file can be viewed but not opened from the document repository. A value of one indicates that the file is view only and cannot be opened; a value of zero indicates the file can be viewed and opened.

2.2.9.1.2 Pages

The **Pages** complex type specifies all the foreground pages (section 1.3.2.1) in a web drawing (section 1.3.1). One element is returned in the **Page** element of the web drawing.

```
<xs:complexType name="Pages" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="PageInfo" minOccurs="1" maxOccurs="unbounded" type="PageInfo"/>
  </xs:sequence>
</xs:complexType>
```

2.2.9.1.3 PageInfo

The **Page Info** complex type specifies the identity of a drawing page (section 1.3.2) in the web drawing (section 1.3.1). One element is returned for each foreground page (section 1.3.2.1) within the **Pages** element (section 2.2.9.1.2).

```
<xs:complexType name="PageInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="ID" type="xs:unsignedLong" use="required"/>
  <xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
```

ID: An unsigned long that specifies the identifier of the drawing page. The value MUST match the **ID** attribute of one of the **Page_Type** elements in the **Pages XML Part**, as specified in [MS-VSDX] section 2.2.2.1, of the web drawing.

Name: A string that specifies the name of the drawing page. The value MUST NOT be an empty string. It MUST match the **Name** or **NameU** attribute of the **Page_Type** elements in the **Pages XML Part**, as specified in [MS-VSDX] section 2.2.2.1, of the web drawing referenced by the **ID** attribute.

2.2.9.1.4 ShapeInfo

The **ShapeInfo** complex type specifies information about a shape. One element is returned for each relevant shape in the **Page** element of the web drawing.

```xml
<xs:complexType name="ShapeInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="ShapeDataItems" minOccurs="0" maxOccurs="1" type="ShapeDataItems"/>
    <xs:element name="Hyperlinks" minOccurs="0" maxOccurs="1" type="Hyperlinks"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required"/>
  <xs:attribute name="DisplayName" type="xs:string" use="optional"/>
  <xs:attribute name="Guid" type="xs:string" use="optional"/>
  <xs:attribute name="Layout" type="xs:string" use="required"/>
  <xs:attribute name="ShapeText" type="xs:string" use="optional"/>
  <xs:attribute name="TabOrderString" type="xs:string" use="required"/>
</xs:complexType>
```

Name: A string that specifies the identifier of the shape. It MUST be the **ID** attribute of a shape, as specified in the ShapeSheet_Type, as specified in [MS-VSDX] section 2.3.4.2.88, of the web drawing. It MUST NOT be an empty string, and it MUST be unique amongst all the identifiers specified by the **Name** attribute of the **ShapeInfo** elements in the containing **Page** element.

DisplayName: A string that specifies the display name for the shape. It MUST be the **Name** attribute of a shape, as specified in the ShapeSheet_Type, as specified in [MS-VSDX] section 2.3.4.2.88, of the web drawing, of the shape specified by the **Name** attribute of this **ShapeInfo** element.

Guid : An optional string that specifies a GUID for the shape. If this attribute does not exist, the shape does not have a GUID. If this attribute exists, it MUST be the 32 letters and digits separated by hyphens contained in the **UniqueID** attribute of a shape, as specified in the ShapeSheet_Type, as specified in [MS-VSDX] section 2.3.4.2.88, of the web drawing, of the shape specified by the **Name** attribute of this **ShapeInfo** element.

Layout: A string that specifies the geometric location of the shape. If the shape is 2D, the string specifies the bounding rectangle of the shape on the drawing page as specified in the following table.

Name	Description
x	An integer that specifies the upper left x-coordinate of the bounding rectangle.
y	An integer that specifies the upper left y-coordinate of the bounding rectangle.
width	A positive integer that specifies the width of the bounding rectangle.
height	A positive integer that specifies the height of the bounding rectangle.

The string MUST be formatted using the JavaScript Object Notation (JSON) as specified in [RFC4627] as follows:

```
{"type":"2D","bounds":{"x":x-val,"y":y-val,"width":width-val,"height":height-val}}
```

where **x-val**, **y-val**, **width-val**, and **height-val** are integers specifying the x, y, width, and height fields, respectively.

If the shape is 1D, the string specifies the begin and end coordinates of the shape as specified in the following table.

Name	Description
beginX	A decimal that specifies the x-coordinate of the beginning of the 1D shape.

Name	Description
beginY	A decimal that specifies the y-coordinate of the beginning of the 1D shape.
endX	A decimal that specifies the x-coordinate of the end of the 1D shape.
endY	A decimal that specifies the y-coordinate of the end of the 1D shape.

The string MUST be formatted using the JavaScript Object Notation as specified in [RFC4627] as follows:

```
{"type":"1D","points":{"beginX":beginx-val,"beginY":beginy-val, ":{"endX":endx-
val,"endY":endy-val}}
```

where **beginx-val**, **beginy-val**, **endx-val**, and **endy-val** specify the beginX, beginY, endX, and endY fields, respectively.

The coordinates MUST be relative to the top left corner of the window that contains the drawing page (section 1.3.2). All the values MUST be specified in pixels.

ShapeText: An optional string that specifies a text element. If this attribute exists, it is composed of information from the document, shape data items, recalculations on shape data items and formatting information, for a shape.

TabOrderString: A string that specifies the tab order of this shape on the drawing page. The format of the string MUST be "rowIndex_columnIndex", where rowIndex and columnIndex are unsigned integer values separated by an underscore "_".

2.2.9.1.5 Hyperlinks

The **Hyperlinks** complex type specifies each **Hyperlink** element (section 2.2.9.1.6) associated with a shape.

```
<xs:complexType name="Hyperlinks" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Hyperlink" minOccurs="1" maxOccurs="unbounded" type="Hyperlink"/>
  </xs:sequence>
</xs:complexType>
```

2.2.9.1.6 Hyperlink

The **Hyperlink** complex type specifies a hyperlink associated with a shape (section 1.3.3). One element is returned for each visible hyperlink associated with the shape within a **ShapeInfo** element (section 2.2.9.1).

```
<xs:complexType name="Hyperlink" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Name" type="xs:string" use="required"/>
  <xs:attribute name="Value" type="xs:string" use="optional"/>
  <xs:attribute name="Description" type="xs:string" use="optional"/>
  <xs:attribute name="SubAddress" type="xs:string" use="optional"/>
  <xs:attribute name="SubAddressShape" type="xs:string" use="optional"/>
  <xs:attribute name="Zoom" use="optional">
    <xs:simpleType>
      <xs:union>
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minExclusive value="0"/>
          </xs:restriction>
        </xs:simpleType>
```

```
            <xs:simpleType>
              <xs:restriction base="xs:int">
                <xs:enumeration value="-2"/>
                <xs:enumeration value="-1"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="Default" type="xs:boolean" use="optional"/>
    </xs:complexType>
```

Name: A string attribute that specifies an identifier for a hyperlink. It MUST NOT be an empty string, and it MUST be unique among the identifiers specified by the **Name** attribute of the hyperlink elements in this XML part.

Value: An optional string attribute that specifies the URI of an external resource referenced by the hyperlink. If this attribute exists, it MUST NOT be an empty string, and the **SubAddress**, **SubAddressShape**, and **Zoom** elements MUST be ignored. If this attribute does not exist, **SubAddress** MUST exist.

Description: An optional string attribute that specifies the description of the hyperlink. If this attribute does not exist, the hyperlink does not have a description.

SubAddress: An optional string attribute that specifies the identifier of the drawing page (section 1.3.2) referenced by this hyperlink within the web drawing (section 1.3.1). If this attribute exists, it MUST NOT be an empty string. If this attribute does not exist, the **SubAddressShape** and **Zoom** elements MUST be ignored.

SubAddressShape: An optional string attribute that specifies the identifier of the shape referenced by this hyperlink within the drawing page specified by **SubAddress**. If this attribute exists, it MUST NOT be an empty string. If this attribute does not exist, the hyperlink references the entire drawing page.

Zoom: An optional integer attribute that specifies the **zoom level** for the drawing page specified by **SubAddress**. If this attribute does not exist, the zoom level is 100%. If this attribute does exist, it MUST be a value from the following table.

Value	Description
-2	The drawing page is zoomed to fit the width of the view.
-1	The drawing page is zoomed to fit in the view.
n > 0	The drawing page is zoomed to n%. n MUST be greater than zero.

Default: An optional **Boolean** attribute that specifies the default hyperlink associated with the shape. It MUST be true if and only if this hyperlink is the default. Each shape MUST NOT have more than one default hyperlink.

2.2.9.1.7 ShapeDataItems

The **ShapeDataItems** complex type specifies a list of shape data items, as specified in [MS-VSDX] section 2.2.3.6. One element is returned for visible shape data items associated with a shape within a **ShapeInfo** element (section 2.2.9.1).

```
    <xs:complexType name="ShapeDataItems" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:sequence>
        <xs:element name="ShapeData" minOccurs="1" maxOccurs="unbounded" type="ShapeData"/>
      </xs:sequence>
```

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

2.2.9.1.8 ShapeData

The **ShapeData** complex type specifies a shape data item, as specified in [MS-VSDX] section 2.2.3.6. One element is returned for each visible shape data item associated with the shape with a **ShapeInfo** element (section 2.2.9.1).

```
<xs:complexType name="ShapeData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Name" type="xs:string" use="required"/>
  <xs:attribute name="FormattedValue" type="xs:string" use="required"/>
  <xs:attribute name="Value" type="xs:string" use="optional"/>
  <xs:attribute name="Format" type="xs:string" use="optional"/>
</xs:complexType>
```

Name: A string that is determined by a combination of shape data row properties that specifies the label of the shape data item.

FormattedValue: A string representation of the value of the data formatted for display. It MUST NOT be an empty string.

Value: An optional string that specifies the value of the shape data item, as specified in [MS-VSDX] section 2.2.3.6.

Format: An optional string that specifies a vFormatString custom structure, as specified in [MS-VSDX] section 2.5.9.6. If this attribute exists, it is used to format the value for display.

2.2.9.1.9 Shapes

The **Shapes** complex type specifies the geometric outline information for multiple shapes. One element is returned for the web drawing.

```
<xs:complexType name="Shapes" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Shape" minOccurs="1" maxOccurs="unbounded" type="Shape"/>
  </xs:sequence>
</xs:complexType>
```

2.2.9.1.10   Shape

The **Shape** complex type specifies the geometric outline information for a shape. One element is returned for each relevant shape.

```
<xs:complexType name="Shape" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Path" minOccurs="1" maxOccurs="1" type="Path"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
```

Name: A string that specifies the identifier of the **ShapeInfo** element (section 2.2.9.1) in the **Page** structure (section 2.2.9.1.1) that this element represents. It MUST match exactly the **Name** of a **ShapeInfo** element in the **Pages** element of this **Page** structure.

2.2.9.1.11   Path

The **Path** complex type specifies the geometric outline information for a [shape](). One element is returned for a **Shape** element.

```
<xs:complexType name="Path" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="Shape" type="xs:string" fixed="poly"/>
  <xs:attribute name="Data" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="\d*(,\d*){3,}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Shape: A string that specifies the type of outline. It MUST be equal to "poly".

Data: A string that specifies the Cartesian coordinates of the outline. It MUST have the following format:

```
x1,y1,x2,y2...xn,yn
```

The fields xn and yn MUST be integer values, and they MUST occur in pairs. Each xn,yn pair specifies the coordinates of a point that defines one vertex of the outline. The outline is formed by connecting the points in the order they appear in this string attribute. There MUST be at least 4 points. The first point MUST equal the last point. The coordinates MUST be measured in pixels relative to the top, left corner of the drawing page (section [1.3.2]()) that contains the shape. All values MUST be separated by commas.

2.2.9.2 Comments

The **Comments** element is a string that specifies information about the [comments]() in a drawing page (section [1.3.2]()).[<2>]() The syntax of this data structure is an XML fragment. Elements of this fragment are listed in the following sections.

2.2.9.2.1 Comments

The **Comments** complex type specifies all [comment]() information for a drawing page (section [1.3.2]()). There MUST be zero or exactly one instance of this element.

```
<xs:complexType name="comments" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="CommentEntry"/>
  </xs:sequence>
</xs:complexType>
```

The **Comments** element MUST contain one or more **CommentEntry** elements as specified in section [2.2.9.2.2]().

2.2.9.2.2 CommentEntry

The **CommentEntry** complex type specifies a comment (section [1.3.5]()) on a drawing page (section [1.3.2]()).

```
<xs:complexType name="CommentEntry" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleContent>
      <xs:extension base="xs:string">
```

```
            <xs:attribute name="CommentID" type="xs:string" use="required"/>
            <xs:attribute name="Author" type="xs:string" use="required"/>
            <xs:attribute name="ShapeID" type="xs:string" use="required"/>
            <xs:attribute name="CreationDate" type="xs:string" use="required"/>
            <xs:attribute name="ModifiedDate" type="xs:string" use="required"/>
            <xs:attribute name="Done" type="xs:string" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

CommentID: An integer attribute that specifies the identifier of this comment. It MUST be greater than or equal to zero. It MUST be unique across all the **CommentEntry** elements in the containing **Comments** element (section 2.2.9.2). It MUST match the value of a **CommentID** attribute in a **CommentEntry_Type** element, as specified in [MS-VSDX] section 2.3.4.2.9, in the containing drawing page.

Author: A string attribute that specifies the display name of the author of this comment. The value MUST match the value of the **Name** attribute of an **AuthorEntry_Type**, as specified [MS-VSDX] section 2.3.4.2.2.

ShapeID: An integer attribute that specifies the identifier of the shape (section 1.3.3) this comment is associated with. If no **ShapeID** is specified or if **ShapeID** is zero, the comment refers to the drawing page. If the attribute exists and is not zero, it MUST match the value of the **ID** attribute of a shape, as specified in the **ShapeSheet_Type**, as specified in [MS-VSDX] section 2.3.4.2.88, of the web drawing (section 1.3.1).

CreationDate: A **dateTime** attribute that specifies the date and time, in Coordinated Universal Time (UTC) format, of when the comment was created.

ModifiedDate: A **dateTime** attribute that specifies the date and time, in Coordinated Universal Time (UTC) format, of when the comment was last modified. This attribute MUST be greater than or equal to the value of **CreationDate**.

Done: A **Boolean** attribute that specifies the current state of the comment. A value of "true" indicates that the comment is completed; a value of "false" indicates that the comment is not completed.

2.2.9.3 CommentData

The **CommentData** attribute is a string that specifies a save comment action and the comment (section 1.3.5) data for a shape (section 1.3.3) or drawing page (section 1.3.2) in the **VSDX** web drawing (section 1.3.1).<3> The syntax of this data structure is an XML fragment. Elements of this fragment are listed in the following sections.

2.2.9.3.1 Comment

The **Comment** complex type specifies a comment (section 1.3.5) for a shape or drawing page (section 1.3.2) in a **VSDX** web drawing (section 1.3.1). There MUST be exactly one instance of this element, and it MUST NOT be null.

The value of the **Comment** element is a string with the comment text. Additional attributes define properties and actions for the comment.

```
<xs:complexType name="Comment" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="Action" type="xs:string" use="required"/>
      <xs:attribute name="PageID" type="xs:integer" use="required"/>
      <xs:attribute name="ShapeID" type="xs:integer"/>
```

```
          <xs:attribute name="CommentID" type="xs:integer"/>
          <xs:attribute name="Author" type="xs:string"/>
          <xs:attribute name="ModifiedDate" type="xs:dateTime"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
```

Action: A string attribute that specifies the action to perform. This attribute MUST exist, and it MUST be one of the following values.

Value	Description
"Add"	Write a new comment into the file using the contents and attributes of this element.
"Edit"	Edit an existing comment by modifying the file using the contents and attributes of this element.
"Delete"	Remove a comment from the file.
"Done"	Mark an existing comment as complete in the file.

PageID: An integer attribute that identifies the drawing page in the **VSDX** web drawing specified by **fileUrl** that the comment is on.

ShapeID: An integer attribute that specifies the identifier of the shape (section 1.3.3), within the drawing page specified by **PageID**, this comment is associated with. If no **ShapeID** is specified or if **ShapeID** is zero, the comment refers to the drawing page specified by **PageID**.

CommentID: An integer attribute that specifies the identifier of the comment. If the **Action** value is "Edit", "Delete" or "Done", this attribute MUST exist, and the value MUST match the value of the **CommentID** attribute of a **CommentEntry_Type**, as specified in [MS-VSDX] section 2.3.4.2.9, in the **VSDX** web drawing specified by **fileUrl**.

Author: A string attribute that specifies the display name of the author of this comment. If the **Action** value is "Add" or "Edit", this attribute MUST exist, and it MUST NOT be empty.

ModifiedDate: A **dateTime** attribute that specifies the date and time, in Coordinated Universal Time (UTC) format, of when the comment was last modified. If the **Action** value is "Delete" or "Edit", this attribute MUST exist, and it MUST NOT be null.

2.2.9.4 CommentResult

The **CommentResult** element is a string that specifies the result of the **SaveComments** operation (section 3.1.4.5).<4> The syntax of this data structure is an XML fragment. Elements of this fragment are listed in the following sections.

2.2.9.4.1 Comment

The **Comment** complex type specifies the result of a comment action (section 1.3.5) in the **VSDX** web drawing (section 1.3.1). The value of the **Comment** element is a string with comment text. Additional attributes define the properties and actions for the comment response.

```
    <xs:complexType name="Comment" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="Action" type="xs:string" use="required"/>
          <xs:attribute name="ShapeID" type="xs:integer"/>
          <xs:attribute name="CommentID" type="xs:integer"/>
          <xs:attribute name="Author" type="xs:string"/>
```

```
      <xs:attribute name="CreationDate" type="xs:dateTime"/>
      <xs:attribute name="ModifiedDate" type="xs:dateTime"/>
      <xs:attribute name="Done" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Action: A string attribute that specifies the action performed. It MUST exist, and it MUST be one of the following values:

Value	Description
"Add"	Add a new comment.
"Edit"	Edit an existing comment.
"Delete"	Delete an existing comment.
"Done"	Mark an existing comment as complete.

ShapeID: An integer attribute that specifies the identifier of a shape (section 1.3.3) that this comment is associated with. If no **ShapeID** is specified or if **ShapeID** is zero, the comment refers to the drawing page (section 1.3.2).

CommentID: An integer attribute that specifies the identifier of the comment. If the **Action** value is "Add" or "Edit", this attribute MUST exist, and the value MUST match the value of the **CommentID** attribute of a **CommentEntry_Type**, as specified in [MS-VSDX] section 2.3.4.2.9, in the **VSDX** web drawing specified by **fileUrl**.

Author: A string attribute that specifies the display name of the author of this comment. If the **Action** value is "Add", this attribute MUST exist, and it MUST NOT be empty.

CreationDate: A **dateTime** attribute that specifies the date and time, in Coordinated Universal Time (UTC) format, of when the comment was created. If the **Action** value is "Add" or "Edit", this attribute MUST exist, and it MUST NOT be null.

ModifiedDate: A **dateTime** attribute that specifies the date and time, in Coordinated Universal Time (UTC) format, of when the comment was last modified. If the **Action** value is "Delete" or "Edit", this attribute MUST exist, and it MUST NOT be null.

Done: A **Boolean** attribute that specifies the current state of the comment. If the **Action** value is "Done", this attribute MUST exist. A value of "true" indicates that the comment is completed; a value of "false" indicates that the comment has not been completed.

3   Protocol Details

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

Except where specified, protocol clients MUST interpret HTTP **Status-Codes** returned by the protocol server as specified in [RFC2616] (section 10, Status Code Definitions).

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

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

3.1   Common Details

The **GetRasterDiagram** (section 3.1.4.1) and **GetVectorDiagram** (section 3.1.4.2) WSDL operations are independent and stateless. The protocol server returns the requested drawing page (section 1.3.2) of the specified **VGSFF** web drawing in raster and vector format, respectively. There are no interaction dependencies between instances of client and server.

The **GetRasterPage** (section 3.1.4.3) and **GetRasterPageItem** (section 3.1.4.4) WSDL operations are related and require client state. A drawing page (section 1.3.2) of a **VSDX** web drawing in raster format is retrieved in full as a result of a message sequence, as illustrated in the following high-level sequence diagram:
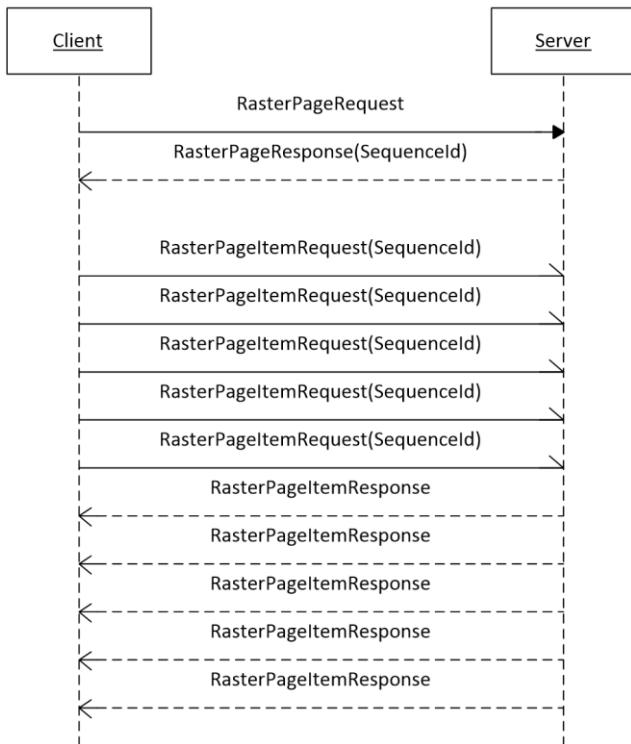
Figure 2: Communication message sequence for retrieving a drawing page of a VSDX web drawing in raster format

First, the protocol client sends an **IVisioGraphicsService_GetRasterPage_InputMessage** request (section 3.1.4.3.1.1) WSDL message that MUST be unconditional, as specified in section 3.1.1. The protocol server initiates the process of retrieving or generating the drawing page. The client waits for the server to respond with the **IVisioGraphicsService_GetRasterPage_OutputMessage** (section 3.1.4.3.1.2) response WSDL message that contains drawing page information and includes a message sequence identifier. This message sequence identifier is set in the **SequenceId** element of the **RasterPageTag** complex type (section 3.1.4.3.3.3) and is contained in the **EntityTag** element of the **RasterPageResponse** complex type (section 3.1.4.3.3.2).

Then, the protocol client sends a sequence of **IVisioGraphicsService_GetRasterPageItem_InputMessage** request (section 3.1.4.4.1.1) WSDL messages, one for each item, as specified in section 3.1.1, of the drawing page. The client MUST set the value of the **SequenceId** element of the **RasterPageItemRequest** complex type (section 3.1.4.4.3.1), to the message sequence identifier obtained in the preceding **IVisioGraphicsService_GetRasterPage_OutputMessage** response WSDL message. These requests are independent and can be sent asynchronously in any order. The protocol server uses the message sequence identifier to match the requests with the drawing page retrieved or generated as result of the **IVisioGraphicsService_GetRasterPage_InputMessage** request WSDL message that initiated the conversation. The server responds to each of these requests with a corresponding **IVisioGraphicsService_GetRasterPageItem_OutputMessage** response (section 3.1.4.4.1.2) WSDL message that contains information about the requested item, as specified in section 3.1.1, of the drawing page.

The **SaveComments** WSDL operation (section 3.1.4.5) is independent and stateless. The protocol server makes the requested comment updates and saves them to the specified web drawing. This web drawing file MUST conform to the format specified in [MS-VSDX] section 2.2.9.

3.1.1 Abstract Data Model

There is no abstract data model for a **VGSFF** web drawing.

A drawing page (section 1.3.2) of a **VSDX** web drawing in raster format is composed of 6 **items**: 4 images in Portable Network Graphics (PNG) format, as specified in [RFC2083], the shape information in XML format, as specified in section 2.2.9.1, and comments in XML format, as specified in section 2.2.9.2.<5> Each image represents the drawing page in different sizes, as specified by the **ItemName** element of the **RasterPageItemRequest** complex type (section 3.1.4.4.3.1).

An **IVisioGraphicsService_GetRasterPage_InputMessage** request (section 3.1.4.3.1.1) or an **IVisioGraphicsService_GetRasterPageItem_InputMessage** request (section 3.1.4.4.1.1) WSDL message is unconditional if the value of the **EntityTag** element of the corresponding complex types **RasterPageRequest** (section 3.1.4.3.3.1) and **RasterPageItemRequest** (section 3.1.4.4.3.1), respectively, is null or empty, as specified by the **RasterPageTag** complex type (section 3.1.4.3.3.3) and the **RasterPageItemTag** complex type (section 3.1.4.4.3.2) respectively. Otherwise, the requesting WSDL message is conditional**. Message sequencing rules for conditional messages are specified in 3.1.4.

After the initial request for a drawing page of a **VSDX** web drawing is processed and before returning the result to the protocol client, the protocol server caches it for future use. Subsequent requests for the same drawing page are served directly from this cache. The protocol server assigns a unique identifier, **OutputETag** element, to each version of a drawing page in the cache. The protocol server also assigns a unique identifier, **FileETag** element, for the current version of the web drawing that contains this drawing page in the cache. The **OutputETag** and **FileETag** are included in the **EntityTag** element of the **RasterPageResponse** complex type (section 3.1.4.3.3.2). The **EntityTag** received by a client can be used on subsequent conditional requests to determine if the version of the drawing page cached on the client needs to be updated. The protocol server compares the **FileETag** and **OutputETag** from the **EntityTag** of the conditional **RasterPageTag** with the current **FileETag** and **OutputETag,** respectively, of the cached drawing page. If both elements match, the server returns a **RasterPageResponse** having the response code "NotModified"; otherwise, it generates a new drawing page.

3.1.2 Timers

The **Recalculation Timeout** timer measures the time it takes for the protocol server to recalculate the drawing page (section 1.3.2) of a web drawing. This timeout is configured on the protocol server.

The **Message Sequence Timeout** timer measures the time it takes for the protocol server to receive an **IVisioGraphicsService_GetRasterPageItem_InputMessage** request (section 3.1.4.4.1.1) WSDL message beginning when the **IVisioGraphicsService_GetRasterPage_OutputMessage** response (section 3.1.4.3.1.2) WSDL message is sent, in the sequence of messages to retrieve a drawing page of a **VSDX** web drawing in raster format, as specified in section 3.1.<6> This timeout is set to one minute, and it is not configurable on the protocol server.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The protocol operations can be executed in any order with the following restrictions:

- To fully retrieve a drawing page (section 1.3.2) of a **VSDX** web drawing in raster format, the protocol client MUST follow the message sequence specified and illustrated in section 3.1.

- The protocol client can cache a fully retrieved drawing page of a **VSDX** web drawing in raster format. Furthermore, it can use the information set by the protocol server in the **EntityTag** element of the **RasterPageResponse** complex type (section 3.1.4.3.3.2) and send it back to the server in the **EntityTag** element of a subsequent **RasterPageRequest** complex type (section 3.1.4.3.3.1) in a conditional message. Based on that information, if the server decides that the drawing page has not changed, it responds with a **RasterPageResponse** having the response code "NotModified". Similarly the client can use the information set by the server in the **EntityTag** element of a **RasterPageItemResponse** complex type (section 3.1.4.4.3.3) and send it back to the server in the **EntityTag** element of a subsequent **RasterPageItemRequest** complex type (section 3.1.4.4.3.1) in a conditional message. If the server decides that the requested item, as specified in section 3.1.1, of the drawing page has not changed, it responds with a **RasterPageItemResponse** having the response code "NotModified". In these cases, the server expects the client will use the drawing page data that it cached previously. This mechanism reduces network traffic.

- If a **GetRasterPageItem** operation (section 3.1.4.4) is not preceded by a **GetRasterPage** operation (section 3.1.4.3) or the protocol server cannot match the value of the **SequenceId** element of the **RasterPageItemRequest**, as specified in section 3.1, it MUST return a **RasterPageItemResponse** with the response code "NotModified".

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

Operation	Description
GetRasterDiagram	The **GetRasterDiagram** operation retrieves a drawing page (section 1.3.2) from the **VGSFF** web drawing (section 1.3.1) rendered in a raster format.
GetRasterPage	The **GetRasterPage** operation retrieves information about a drawing page (section 1.3.2) of a **VSDX** web drawing rendered in raster format.<7>
GetRasterPageItem	The **GetRasterPageItem** operation retrieves an item, as specified in section 3.1.1, of a drawing page (section 1.3.2) of a **VSDX** web drawing rendered in raster format.<8>
GetVectorDiagram	The **GetVectorDiagram** operation retrieves a drawing page (section 1.3.2) from the **VGSFF** web drawing rendered in a vector format.
SaveComments	The **SaveComments** operation saves a comment to a **VSDX** web drawing in a document repository, in the form of a **CommentEntry_Type** element as specified [MS-VSDX] section 2.3.4.2.9.<9>

3.1.4.1 GetRasterDiagram

The **GetRasterDiagram** operation retrieves a drawing page (section 1.3.2) from the **VGSFF** web drawing (section 1.3.1) rendered in a raster format.

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

```
<wsdl:operation name="GetRasterDiagram" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagram"
message="tns2:IVisioGraphicsService_GetRasterDiagram_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagramResponse"
message="tns2:IVisioGraphicsService_GetRasterDiagram_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
```

```
    <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagramVisioGraphicsServiceFau
ltFault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetRasterDiagram_VisioGraphicsServiceFaultFault_FaultMess
age" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:operation>
```

The protocol client sends an **IVisioGraphicsService_GetRasterDiagram_InputMessage** request (section 3.1.4.1.1.1) WSDL message, and the protocol server MUST respond with an **IVisioGraphicsService_GetRasterDiagram_OutputMessage** response (section 3.1.4.1.1.2) WSDL message as follows:

1. If the protocol server determines that an error occurred during the protocol server operation, the protocol server MUST respond with a **VisioGraphicsServiceFault** complex type (section 2.2.4.5) SOAP fault with the **Error** field set to the correct **FaultCode** value (section 2.2.5.1).

2. Otherwise, the protocol server MUST return a **RasterDiagram** complex type (section 3.1.4.1.3.2).

If the requested drawing page contains data providers to be implemented by the protocol client, the following sequence of messages MUST be followed:

1. The protocol server MUST respond with an **IVisioGraphicsService_GetRasterDiagram_OutputMessage** response WSDL message. The data providers MUST be specified in the **DataSources** element of the **RasterDiagram** contained within this response. The **Image** element of this **RasterDiagram** MUST be null.

2. The protocol client MUST retrieve the data from the data providers and send another **IVisioGraphicsService_GetRasterDiagram_InputMessage** request WSDL message. This request MUST be identical to the original request except that the **DataSources** element of the **RasterDiagramRequestContract** complex type (section 3.1.4.1.3.1) MUST NOT be null.

3. The protocol server MUST respond with an **IVisioGraphicsService_GetRasterDiagram_OutputMessage** response WSDL message. If no errors occurred, the response MUST contain a **RasterDiagram**, and the **Image** element of the **RasterDiagram** MUST NOT be null.

3.1.4.1.1 Messages

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

Message	Description
IVisioGraphicsService_GetRasterDiagram_InputMessage	The request WSDL message for the **GetRasterDiagram** WSDL operation.
IVisioGraphicsService_GetRasterDiagram_OutputMessage	The response WSDL message for the **GetRasterDiagram** WSDL operation.
IVisioGraphicsService_GetRasterDiagram_VisioGraphicsServiceFaultFault_FaultMessage	The fault WSDL message for the **GetRasterDiagram** WSDL

Message	Description
	operation.

3.1.4.1.1.1 IVisioGraphicsService_GetRasterDiagram_InputMessage

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

The **SOAP action** value is:

```
http://tempuri.org/IVisioGraphicsService/GetRasterDiagram
```

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

3.1.4.1.1.2 IVisioGraphicsService_GetRasterDiagram_OutputMessage

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

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

3.1.4.1.1.3 IVisioGraphicsService_GetRasterDiagram_VisioGraphicsServiceFaultFault_FaultMessage

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

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

3.1.4.1.2 Elements

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

Element	Description
GetRasterDiagram	The input data for the **GetRasterDiagram** WSDL operation.
GetRasterDiagramResponse	The result data for the **GetRasterDiagram** WSDL operation.

3.1.4.1.2.1 GetRasterDiagram

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

```
<xs:element name="GetRasterDiagram" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="rasterDiagramRequestContract" nillable="true"
type="tns:RasterDiagramRequestContract"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

rasterDiagramRequestContract: The request parameters for a **GetRasterDiagram** WSDL operation (section 3.1.4.1). There MUST be exactly one instance of this element.

3.1.4.1.2.2 GetRasterDiagramResponse

The **GetRasterDiagramResponse** element specifies the result data for the **GetRasterDiagram** WSDL operation.

```
<xs:element name="GetRasterDiagramResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterDiagramResult" nillable="true" type="tns:RasterDiagram"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRasterDiagramResult: The results for a **GetRasterDiagram** WSDL operation (section 3.1.4.1). There MUST be exactly one instance of this element.

3.1.4.1.3 Complex Types

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

Complex type	Description
RasterDiagram	The response data from the **GetRasterDiagram** WSDL operation.
RasterDiagramRequestContract	The input data for the **GetRasterDiagram** WSDL operation.

3.1.4.1.3.1 RasterDiagramRequestContract

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterDiagramRequestContract** complex type contains information used to specify the request parameters for a **GetRasterDiagram** WSDL operation (section 3.1.4.1). This complex type inherits from the **BaseRequestContract** complex type, as specified in section 2.2.4.3, which specifies additional parameters.

```
<xs:complexType name="RasterDiagramRequestContract"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestContract">
      <xs:sequence>
        <xs:element minOccurs="0" name="Resolution" type="xs:int"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Resolution: An integer that is unused, and MUST be equal to 1 or 2.

3.1.4.1.3.2 RasterDiagram

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterDiagram** complex type contains information used to specify the response for a **GetRasterDiagram** WSDL operation (section 3.1.4.1). This complex type inherits from the **DiagramBase** complex type, as specified in section 2.2.4.4, which specifies additional parameters.

```
<xs:complexType name="RasterDiagram" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:DiagramBase">
      <xs:sequence>
        <xs:element minOccurs="0" name="AreaMaps" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="Image" nillable="true" type="xs:base64Binary"/>
        <xs:element minOccurs="0" name="ShapeInfo" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

AreaMaps: A string that specifies the geometric outline information of the shapes, as specified in section 1.3.3, in the raster image. The format of this data is specified in [RFC1980]. There MUST be exactly one instance of this element.

Image: A base64Binary that specifies the drawing page, as specified in section 1.3.2, as a raster image in PNG format, as specified in [RFC2083]. There MUST be exactly one instance of this element.

ShapeInfo: A string that specifies information about the shapes, as specified in section 1.3.3, in the drawing page, as specified in section 1.3.2. There MUST be exactly one instance of this element.

The syntax of this element is specified in [MS-VGSFF] section 2.4.8. The syntax of this element is also specified by additional attributes of the **CT_Page** element, as specified in [MS-VGSFF] section 2.4.8.2.8, that are listed in the following table.

Attribute of CT_Page element	Description
Conflict	An optional integer that specifies the status of retrieving data from data providers for the drawing page (section 1.3.2).
DocumentID	A string that specifies a GUID that identifies the file in a document repository that contains the **VGSFF** web drawing that is being requested.
DocumentTimestamp	A long value that specifies the number of ticks (10,000 per millisecond) that elapsed between 12:00:00 midnight, January 1, 0001 and when the file in a document repository that contains the **VGSFF** web drawing was last modified in the document repository.
RefreshWarning	An optional integer that specifies whether the drawing page was refreshed by the protocol server. If this attribute is present, a value of one indicates that **data connections** exist for the drawing page, but that the drawing page was not refreshed from the data providers associated with these data connections.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 GetVectorDiagram

The **GetVectorDiagram** operation retrieves a drawing page (section 1.3.2) from the **VGSFF** web drawing rendered in a vector format.

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

```
<wsdl:operation name="GetVectorDiagram" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagram"
message="tns2:IVisioGraphicsService_GetVectorDiagram_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagramResponse"
message="tns2:IVisioGraphicsService_GetVectorDiagram_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagramVisioGraphicsServiceFau
ltFault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetVectorDiagram_VisioGraphicsServiceFaultFault_FaultMess
age" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IVisioGraphicsService_GetVectorDiagram_InputMessage** request (section 3.1.4.2.1.1) WSDL message, and the protocol server MUST respond with an **IVisioGraphicsService_GetVectorDiagram_OutputMessage** response (section 3.1.4.2.1.2) WSDL message as follows:

1. If the protocol server determines that an error occurred during the protocol server operation, the protocol server MUST respond with a **VisioGraphicsServiceFault** complex type (section 2.2.4.5) SOAP fault with the **Error** field set to the correct **FaultCode** value (section 2.2.5.1).

2. Otherwise, the protocol server MUST return a **VectorDiagram** complex type (section 3.1.4.2.3.2).

If the requested drawing page contains data providers to be implemented by the protocol client, the following sequence of messages MUST be followed:

1. The protocol server MUST respond with an **IVisioGraphicsService_GetVectorDiagram_OutputMessage** response WSDL message. The data providers MUST be specified in the **DataSources** element of the **VectorDiagram** contained within this response. The **OutputPage** element of this **VectorDiagram** MUST be null.

2. The protocol client MUST retrieve the data from the data providers and send another **IVisioGraphicsService_GetVectorDiagram_InputMessage** request WSDL message. This request MUST be identical to the original request except that the **DataSources** element of the **VectorDiagramRequestContract** complex type (section 3.1.4.2.3.1) MUST NOT be null.

3. The protocol server MUST respond with an **IVisioGraphicsService_GetVectorDiagram_OutputMessage** response WSDL message. If no errors occurred, the response MUST contain a **VectorDiagram** and the **OutputPage** element of the **VectorDiagram** MUST NOT be null.

3.1.4.2.1 Messages

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

Message	Description
IVisioGraphicsService_GetVectorDiagram_InputMessage	The request WSDL message for the **GetVectorDiagram** WSDL operation.
IVisioGraphicsService_GetVectorDiagram_OutputMessage	The response WSDL message for the **GetVectorDiagram** WSDL operation.
IVisioGraphicsService_GetVectorDiagram_VisioGraphicsServiceFaultFault_FaultMessage	The fault WSDL message for the **GetVectorDiagram** WSDL operation.

3.1.4.2.1.1   IVisioGraphicsService_GetVectorDiagram_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IVisioGraphicsService/GetVectorDiagram
```

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

3.1.4.2.1.2   IVisioGraphicsService_GetVectorDiagram_OutputMessage

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

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

3.1.4.2.1.3   IVisioGraphicsService_GetVectorDiagram_VisioGraphicsServiceFaultFault_FaultMessage

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

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

3.1.4.2.2 Elements

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

Element	Description
GetVectorDiagram	The input data for the **GetVectorDiagram** WSDL operation.
GetVectorDiagramResponse	The result data for the **GetVectorDiagram** WSDL operation.

3.1.4.2.2.1 GetVectorDiagram

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

```
<xs:element name="GetVectorDiagram" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer" minOccurs="0" name="vectorDiagramRequest" nillable="true"
type="tns:VectorDiagramRequestContract"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

vectorDiagramRequest: The request parameters for a **GetVectorDiagram** (section 3.1.4.2) WSDL operation. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.2.2.2 GetVectorDiagramResponse

The **GetVectorDiagramResponse** element specifies the result data for the **GetVectorDiagram** WSDL operation.

```
<xs:element name="GetVectorDiagramResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer" minOccurs="0" name="GetVectorDiagramResult" nillable="true" type="tns:VectorDiagram"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetVectorDiagramResult: The results for a **GetVectorDiagram** (section 3.1.4.2) WSDL operation. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.2.3 Complex Types

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

Complex type	Description
ArrayOfDiagramResource	An array of resources information contained in the **RenderedPage** complex type (section 3.1.4.2.3.3).
DiagramResource	A font or an image required by a drawing page.
RenderedPage	The drawing page (section 1.3.2) information contained in the **VectorDiagram** complex type (section 3.1.4.2.3.2).
VectorDiagram	The response data from the **GetVectorDiagram** WSDL operation.
VectorDiagramRequestContract	The input data for the **GetVectorDiagram** (section 3.1.4.2) WSDL operation.

3.1.4.2.3.1   VectorDiagramRequestContract

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **VectorDiagramRequestContract** complex type contains information used to specify the request parameters for a **GetVectorDiagram** (section 3.1.4.2) WSDL operation. This complex type inherits from the **BaseRequestContract** complex type (section 2.2.4.3), which specifies additional parameters.

```
<xs:complexType name="VectorDiagramRequestContract"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestContract">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.2.3.2   VectorDiagram

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **VectorDiagram** complex type contains information used to specify the response for a **GetVectorDiagram** (section 3.1.4.2) WSDL operation. This complex type inherits from **DiagramBase** complex type (section 2.2.4.4), which specifies additional parameters.

```
<xs:complexType name="VectorDiagram" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:DiagramBase">
      <xs:sequence>
        <xs:element minOccurs="0" name="OutputPage" nillable="true" type="tns:RenderedPage"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

OutputPage: A **RenderedPage** complex type (section 3.1.4.2.3.3), that specifies the drawing page (section 1.3.2) in vector format.

3.1.4.2.3.3   RenderedPage

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RenderedPage** complex type contains information that specifies a drawing page, as specified in section 1.3.2, in a vector format.

```
<xs:complexType name="RenderedPage" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Resources" nillable="true"
type="tns:ArrayOfDiagramResource"/>
    <xs:element minOccurs="0" name="ShapeInfo" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Xaml" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Resources: An **ArrayOfDiagramResource**, as specified in section 3.1.4.2.3.4, that specifies all the resources used by the **Xaml** field. There MUST be exactly one instance of this element, and it MUST NOT be null.

ShapeInfo: A string that specifies information about the shapes, as specified in section 1.3.3, in the drawing page, as specified in section 1.3.2. There MUST be exactly one instance of this element, and it MUST NOT be null.

The syntax of this element is specified in [MS-VGSFF] section 2.4.8. The syntax of this element is also specified by additional attributes of the **CT_Page** element, as specified in [MS-VGSFF] section 2.4.8.2.8, that are listed in the following table.

Attribute of CT_Page element	Description
Conflict	An integer that specifies the status of retrieving data from data providers for the drawing page (section 1.3.2).
DocumentID	A string that specifies a GUID that identifies the file in a document repository that contains the **VGSFF** web drawing being requested.
DocumentTimestamp	A long value that specifies the number of ticks (10,000 per millisecond) that elapsed between 12:00:00 midnight, January 1, 0001 and when the file in a document repository that contains the **VGSFF** web drawing was last modified in the document repository
RefreshWarning	An optional integer that specifies whether the drawing page was refreshed by the protocol server. If this attribute is present, a value of one indicates that data connections exist for the drawing page but that the drawing page was not refreshed from the data providers associated with these data connections.

Xaml: A string that specifies the vector drawing page, as specified in section 1.3.2, in XAML format, as specified in [MS-SLXV]. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.2.3.4   ArrayOfDiagramResource

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **ArrayOfDiagramResource** complex type contains an array of **DiagramResource** complex types (section 3.1.4.2.3.5).

```
<xs:complexType name="ArrayOfDiagramResource" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="DiagramResource" nillable="true"
type="tns:DiagramResource"/>
  </xs:sequence>
</xs:complexType>
```

DiagramResource: A **DiagramResource** element that specifies the **fonts** and images required by a drawing page (section 1.3.2). It MUST NOT be null.

3.1.4.2.3.5   DiagramResource

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **DiagramResource** complex type contains information that specifies either a font or an image required by a drawing page (section 1.3.2).

```
<xs:complexType name="DiagramResource" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="ByteData" nillable="true" type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

ByteData: A base64Binary that specifies the resource in the format of a stream of bytes. There MUST be exactly one instance of this element.

Name: A string that specifies the name for the resource as referenced by the **Xaml** field of the **RenderedPage** complex type (section 3.1.4.2.3.3). It MUST refer to either a font as specified in [MS-VGSFF] section 2.1.6.2 or an image as a specified in [MS-VGSFF] section 2.1.6.3. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 GetRasterPage

The **GetRasterPage** operation retrieves information about a drawing page (section 1.3.2) of a **VSDX web drawing** rendered in raster format.<10>

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

```
<wsdl:operation name="GetRasterPage" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPage"
message="tns2:IVisioGraphicsService_GetRasterPage_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageResponse"
message="tns2:IVisioGraphicsService_GetRasterPage_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageVisioGraphicsServiceFaultF
ault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetRasterPage_VisioGraphicsServiceFaultFault_FaultMessage
" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IVisioGraphicsService_GetRasterPage_InputMessage** request (section 3.1.4.3.1.1) WSDL message, and the protocol server MUST respond with an **IVisioGraphicsService_GetRasterPage_OutputMessage** response (section 3.1.4.3.1.2) WSDL message as follows:

1. If the protocol server determines that an error occurred during the protocol server operation, the protocol server MUST respond with a **VisioGraphicsServiceFault** complex type (section 2.2.4.5) SOAP fault having the **Error** field set to the corresponding **FaultCode** value (section 2.2.5.1).

2. If the protocol server determines that the drawing page requested by the protocol client has not changed, based on the information set in the **EntityTag** element of the **RasterPageRequest** complex type (section 3.1.4.3.3.1), as described in the Messages and Event Sequence Rules in section 3.1.4, the protocol server MUST return a **RasterPageResponse** (section 3.1.4.3.3.2) with the response code "NotModified".

3. If the requested drawing page contains data providers to be implemented by the protocol client, the protocol server MUST return a **RasterPageResponse** with the response code "DataSources" and the following sequence of messages MUST be followed:

   1. The protocol server MUST respond with an **IVisioGraphicsService_GetRasterPage_OutputMessage** response (section 3.1.4.3.1.2) WSDL message. The data providers MUST be specified in the **DataSources** element of the **RasterPageResponse** complex type (section 3.1.4.3.3.2) contained within this response.

   2. The protocol client MUST retrieve the data from the data providers and MUST send another **IVisioGraphicsService_GetRasterPage_InputMessage** request (section 3.1.4.3.1.1) WSDL message. This request MUST be identical to the original request except that the **DataSources** element of the **RasterPageRequest** complex type (section 3.1.4.3.3.1) MUST NOT be null.

   3. The protocol server MUST respond with an **IVisioGraphicsService_GetRasterPage_OutputMessage** response WSDL message. If no errors occurred, the response MUST contain a **RasterPageResponse** with the response code "OK".

4. Otherwise, the protocol server MUST return a **RasterPageResponse** with the response code **OK.**

3.1.4.3.1 Messages

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

Message	Description
IVisioGraphicsService_GetRasterPage_InputMessage	The request WSDL message for the **GetRasterPage** WSDL operation.
IVisioGraphicsService_GetRasterPage_OutputMessage	The response WSDL message for the **GetRasterPage** WSDL operation.
IVisioGraphicsService_GetRasterPage_VisioGraphicsServiceFaultFault_FaultMessage	The fault WSDL message for the **GetRasterPage** WSDL operation.

3.1.4.3.1.1   IVisioGraphicsService_GetRasterPage_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IVisioGraphicsService/GetRasterPage
```

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

3.1.4.3.1.2   IVisioGraphicsService_GetRasterPage_OutputMessage

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

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

3.1.4.3.1.3   IVisioGraphicsService_GetRasterPage_VisioGraphicsServiceFaultFault_FaultMessage

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

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

3.1.4.3.2 Elements

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

Element	Description
GetRasterPage	The input data for the **GetRasterPage** WSDL operation.
GetRasterPageResponse	The result data for the **GetRasterPage** WSDL operation.

3.1.4.3.2.1   GetRasterPage

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

```
<xs:element name="GetRasterPage" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer" minOccurs="0" name="rasterPageRequestContract" nillable="true"
type="tns:RasterPageRequest"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

rasterPageRequestContract: A **RasterPageRequest** complex type (section 3.1.4.3.3.1) that specifies the input data. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.3.2.2   GetRasterPageResponse

The **GetRasterPageResponse** element specifies the result data for the **GetRasterPage** WSDL operation.

```
<xs:element name="GetRasterPageResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterPageResult" nillable="true" type="tns:RasterPageResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRasterPageResult: A **RasterPageResponse** complex type (section 3.1.4.3.3.2), that specifies the result data. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.3.3 Complex Types

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

Complex type	Description
RasterPageInfo	The information about a drawing page of a **VSDX** web drawing.
RasterPageRequest	The set of elements that specifies the input data for a **GetRasterPage** (section 3.1.4.3) WSDL operation.
RasterPageResponse	A set of elements that specifies the result data for a **GetRasterPage** WSDL operation.
RasterPageTag	The set of elements that specifies opaque information assigned by the protocol server to a drawing page (section 1.3.2) of a **VSDX** web drawing when it is requested for the first time. It is used on subsequent requests to determine whether the same drawing page has changed since the initial request.

3.1.4.3.3.1 RasterPageRequest

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageRequest** complex type contains the set of elements that specifies the input data for a **GetRasterPage** (section 3.1.4.3.2.1) WSDL operation.

```
<xs:complexType name="RasterPageRequest" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="DataModeName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
    <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageTag"/>
    <xs:element minOccurs="0" name="FileUrl" nillable="true" type="xs:anyURI"/>
    <xs:element minOccurs="0" name="PageId" type="xs:long"/>
    <xs:element minOccurs="0" name="RefreshCookie" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

DataModeName: A string that specifies whether the protocol server refreshes the requested drawing page (section 1.3.2). There MUST be exactly one instance of this element, and it MUST be one of the following values.

Value	Description
Static	The drawing page MUST NOT be updated, and contains only data embedded in the corresponding web drawing.
DynamicSafe	The protocol server will decide whether the refresh queries, specified by the **Command** attribute described in [MS-VSDX] section 2.3.4.2.37, are safe to execute in order to retrieve data from data providers. If all the queries are considered safe, or if **RefreshCookie** matches the current **FileETag** (section 3.1.1), as specified in the description of the **RefreshCookie** element, the drawing page MUST be updated with data retrieved from data providers, as specified in [MS-VSDX] section 2.2.10, and from the system, as specified in [MS-VSDX] section 2.2.11. Otherwise, the drawing page MUST NOT be refreshed, as specified by **Static**, and the **Page** element of the shape information XML, specified in section 3.1.1, MUST have the **RefreshWarning** attribute set to one, as specified in section 2.2.9.1.1. This allows the protocol client to implement a mechanism that requests permission from the user before executing queries that are not considered safe.

DataSources: An **ArrayOfAddonDataSource** complex type (section 2.2.4.2) that specifies the data providers implemented by a protocol client and used in the requested drawing page. There MUST be exactly one instance of this element. It MUST have zero child elements if there are no data providers. If **DataSources** has child elements, the **Data** field of each **AddonDataSource** complex type (section 2.2.4.1) MUST NOT be null.

EntityTag: A **RasterPageTag** complex type (section 3.1.4.3.3.3) that specifies opaque information about the requested drawing page, and used by the protocol server to determine if the drawing page has changed since it was initially requested. There MUST be exactly one instance of this element. A null or empty value indicates an unconditional message as specified in section 3.1.1.

FileUrl: An **anyURI** that specifies the URL of the file in a document repository that contains the web drawing being requested. There MUST be exactly one instance of this element, and it MUST NOT be null.

PageId: A long that specifies the requested drawing page. If greater than or equal to zero it specifies the identifier of the drawing page. If less than zero it specifies the bitwise complement of the zero-based index of the drawing page. There MUST be exactly one instance of this element.

RefreshCookie: A string that specifies a version of the web drawing that contains the requested drawing page. There MUST be exactly one instance of this element. It MUST match the current **FileETag** (section 3.1.1), in order for the protocol server to refresh the drawing page, as specified in the description of the **DynamicSafe** value of the **DataModeName** element. The matching is performed through string comparison using ordinal sort rules, with the exception of the value "*", which matches any **FileETag**. It MUST be ignored if **DataModeName** is not **DynamicSafe**.

3.1.4.3.3.2   RasterPageResponse

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageResponse** complex type contains the set of elements that specifies the result data for a **GetRasterPage** (section 3.1.4.3) WSDL operation.

```
<xs:complexType name="RasterPageResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
    <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageTag"/>
    <xs:element minOccurs="0" name="PageInfo" nillable="true" type="tns:RasterPageInfo"/>
    <xs:element minOccurs="0" name="ResponseCode" type="tns:RasterPageResponseCode"/>
  </xs:sequence>
```

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

DataSources: An **ArrayOfAddonDataSource** complex type (section 2.2.4.2) that specifies the data providers implemented by a protocol client and used in the requested drawing page. There MUST be exactly one instance of this element. It MUST have zero child elements, if there are no data providers. If **DataSources** has child elements, the **Data** field of each contained **AddonDataSource** complex type (section 2.2.4.1) MUST NOT be empty. If the value of **ResponseCode** is "DataSources", it MUST NOT be null or empty. Otherwise, the **DataSources** element MUST be null.

EntityTag: A **RasterPageTag** complex type (section 3.1.4.3.3.3), that specifies opaque information about the requested drawing page. There MUST be exactly one instance of this element. If the value of **ResponseCode** is "OK" or "NotModified", it MUST NOT be null. If the value of **ResponseCode** is "NotModified", it MUST be equal to the value of the **EntityTag** element of the corresponding **RasterPageRequest** complex type (section 3.1.4.3.3.1). Otherwise, the **EntityTag** element MUST be null.

PageInfo: A **RasterPageInfo** complex type (section 3.1.4.3.3.4) that specifies information about the requested drawing page. There MUST be exactly one instance of this element. If the value of **ResponseCode** is "OK", it MUST NOT be null. Otherwise, the **PageInfo** element MUST be null.

ResponseCode: A **RasterPageResponseCode** value (section 3.1.4.3.4.1) that specifies the type of the response. There MUST be exactly one instance of this element.

3.1.4.3.3.3   RasterPageTag

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageTag** complex type contains the set of elements that specifies opaque information assigned by the protocol server to a drawing page of a **VSDX** web drawing when it is requested for the first time, and used on subsequent requests to determine whether the same drawing page has changed since the initial request. A **RasterPageTag** is empty if all its elements have default values.

```
    <xs:complexType name="RasterPageTag" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:sequence>
        <xs:element minOccurs="0" name="DataModeName" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="FileETag" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="OutputETag" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="SequenceId" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
```

DataModeName: A string, that specifies whether the drawing page was refreshed by the protocol server. There MUST be exactly one instance of this element, and it MUST be one of the following values.

Value	Description
Static	The drawing page was not updated, and contains only data embedded in the corresponding web drawing. This is the default value.
Dynamic	The drawing page was updated with data retrieved from data providers, as specified in [MS-VSDX] section 2.2.10, or from the system, as specified in [MS-VSDX] section 2.2.11.

FileETag: A string that specifies an opaque identifier for the version of the web drawing. There MUST be exactly one instance of this element. It MUST NOT be null if **OutputETag** or **SequenceId** is not null. The default value is null.

OutputETag: A string that specifies an opaque identifier for the entity that stores the drawing page in the protocol server's output cache, as described in the Messages and Event Sequence Rules in section 3.1.4,. There MUST be exactly one instance of this element. It MUST NOT be null if **FileETag** or **SequenceId** is not null. The default value is null.

SequenceId: A string that specifies an identifier for the sequence of subsequent messages required to retrieve the items of the drawing page, as specified in section 3.1. There MUST be exactly one instance of this element. It MUST NOT be null if **FileETag** or **OutputETag** is not null. The default value is null.

3.1.4.3.3.4   RasterPageInfo

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageInfo** complex type contains information about a drawing page (section 1.3.2) of a **VSDX** web drawing rendered in raster format.

```
<xs:complexType name="RasterPageInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="ZoomValues" nillable="true" type="q1:ArrayOfint"/>
  </xs:sequence>
</xs:complexType>
```

ZoomValues: An **ArrayOfInt** that contains the zoom values for the items of a drawing page, as specified in section 3.1.1, which correspond to images. There MUST be exactly one instance of this element. The length of the array MUST be equal to 4. Each zoom value in the array MUST be an **integer** greater than or equal to 10, and less than or equal to 200.

3.1.4.3.4 Simple Types

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

Simple type	Description
RasterPageResponseCode	The possible types of response codes returned as an element of a **GetRasterPage** (section 3.1.4.3) WSDL operation.

3.1.4.3.4.1   RasterPageResponseCode

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageResponseCode** simple type specifies the possible types of response codes returned as an element of a **GetRasterPage** (section 3.1.4.3) WSDL operation.

```
<xs:simpleType name="RasterPageResponseCode" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotModified"/>
    <xs:enumeration value="OK"/>
    <xs:enumeration value="DataSources"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **RasterPageResponseCode** simple type.

Value	Meaning
NotModified	Specifies that the requested drawing page (section 1.3.2) has not changed since it was initially requested by the protocol client.
OK	Specifies that the response includes full information about the requested drawing page.
DataSources	Specifies that the response includes information about the data providers implemented by the protocol client and used by the requested drawing page.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 GetRasterPageItem

The **GetRasterPageItem** operation retrieves an item, as specified in section 3.1.1, of a drawing page (section 1.3.2) of a **VSDX** web drawing rendered in raster format.<11>

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

```
<wsdl:operation name="GetRasterPageItem" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItem"
message="tns2:IVisioGraphicsService_GetRasterPageItem_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItemResponse"
message="tns2:IVisioGraphicsService_GetRasterPageItem_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItemVisioGraphicsServiceFa
ultFault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetRasterPageItem_VisioGraphicsServiceFaultFault_FaultMes
sage" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IVisioGraphicsService_GetRasterPageItem_InputMessage** request (section 3.1.4.4.1.1) WSDL message, and the protocol server MUST respond with an **IVisioGraphicsService_GetRasterPageItem_OutputMessage** response (section 3.1.4.4.1.2) WSDL message as follows:

1. If the protocol server determines that an error occurred during the protocol server operation, then the protocol server SHOULD<12> respond with a **VisioGraphicsServiceFault** complex type (section 2.2.4.5) SOAP fault with the **Error** field set to the correct **FaultCode** value (section 2.2.5.1).

2. If the protocol server determines that the item specified by the protocol client has not changed based on the information set in the **EntityTag** element of the **RasterPageItemRequest** complex type (section 3.1.4.4.3.1), as specified in section 3.1.4, the protocol server MUST return a

RasterPageItemResponse complex type (section 3.1.4.4.3.3) with the response code "NotModified".

3. Otherwise, the protocol server MUST return a **RasterPageItemResponse** with the response code "OK".

3.1.4.4.1 Messages

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

Message	Description
IVisioGraphicsService_GetRasterPageItem_InputMessage	The request WSDL message for the **GetRasterPageItem** WSDL operation.
IVisioGraphicsService_GetRasterPageItem_OutputMessage	The response WSDL message for the **GetRasterPageItem** WSDL operation.
IVisioGraphicsService_GetRasterPageItem_VisioGraphicsServiceFaultFault_Fault Message	The fault WSDL message for the **GetRasterPageItem** WSDL operation.

3.1.4.4.1.1 IVisioGraphicsService_GetRasterPageItem_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IVisioGraphicsService/GetRasterPageItem
```

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

3.1.4.4.1.2 IVisioGraphicsService_GetRasterPageItem_OutputMessage

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

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

3.1.4.4.1.3 IVisioGraphicsService_GetRasterPageItem_VisioGraphicsServiceFaultFault_FaultMessage

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

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

3.1.4.4.2 Elements

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

Element	Description
GetRasterPageItem	The input data for the **GetRasterPageItem** WSDL operation.
GetRasterPageItemResponse	The result data for the **GetRasterPageItem** WSDL operation.

3.1.4.4.2.1 GetRasterPageItem

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

```
<xs:element name="GetRasterPageItem" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="rasterPageItemRequestContract" nillable="true"
type="tns:RasterPageItemRequest"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

rasterPageItemRequestContract: A **RasterPageItemRequest** complex type (section 3.1.4.4.3.1), that specifies the input data. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.4.2.2 GetRasterPageItemResponse

The **GetRasterPageItemResponse** element specifies the result data for the **GetRasterPageItem** WSDL operation.

```
<xs:element name="GetRasterPageItemResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterPageItemResult" nillable="true"
type="tns:RasterPageItemResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRasterPageItemResult: A **RasterPageItemReponse** complex type (section 3.1.4.4.3.3), that specifies the result data. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.4.3 Complex Types

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

Complex type	Description
RasterPageItemRequest	The set of elements that specifies the input data for the **GetRasterPageItem** (section 3.1.4.4) WSDL operation.

Complex type	Description
RasterPageItemResponse	The set of elements that specifies the result data for the **GetRasterPageItem** WSDL operation.
RasterPageItemTag	The set of elements that specifies opaque information assigned by the protocol server to an item of a drawing page (section 1.3.2), as specified in section 3.1.1, of a **VSDX** web drawing, when it is requested for the first time, and used on subsequent requests to determine whether the same item has changed since the initial request.

3.1.4.4.3.1 RasterPageItemRequest

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageItemRequest** complex type contains a set of elements that specifies the input data for the **GetRasterPageItem** (section 3.1.4.4) WSDL operation.

```
<xs:complexType name="RasterPageItemRequest" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageItemTag"/>
    <xs:element minOccurs="0" name="FileUrl" nillable="true" type="xs:anyURI"/>
    <xs:element minOccurs="0" name="ItemName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="SequenceId" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

EntityTag: A **RasterPageItemTag** complex type (section 3.1.4.4.3.2) that specifies opaque information about the requested item of the drawing page (section 1.3.2), as specified in section 3.1.1, and used by the protocol server to determine if it has changed since it was initially requested. There MUST be exactly one instance of this element. A null or empty value indicates an unconditional message as specified in section 3.1.1.

FileUrl: An **anyURI** that specifies the URL of the file in a document repository that contains the web drawing being requested. There MUST be exactly one instance of this element, and it MUST NOT be null.

ItemName: A string that specifies the name of the requested item in a drawing page, as specified in section 3.1.1. There MUST be exactly one instance of this element, and it MUST be one of the following values.

Value	Description
Image0	Specifies the requested item is a small image, in Portable Network Graphics (PNG) format as specified in [RFC2083]
Image1	Specifies the requested item is a medium-small image, in Portable Network Graphics (PNG) format as specified in [RFC2083]
Image2	Specifies the requested item is a medium-large image, in Portable Network Graphics (PNG) format as specified in [RFC2083]
Image3	Specifies the requested item is a large image, in Portable Network Graphics (PNG) format as specified in [RFC2083]
ShapeInfo	Specifies the requested item is shape information in XML format, as specified in section 2.2.9.1

Value	Description
Comments	Specifies the requested item is comments in XML format, as specified in section 2.2.9.2

SequenceId: A string that specifies the message sequence identifier obtained in the **IVisioGraphicsService_GetRasterPage_InputMessage** request (section 3.1.4.3.1.1) WSDL message that initiated the retrieval of a drawing page, as specified in section 3.1.

3.1.4.4.3.2   RasterPageItemTag

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageItemTag** complex type contains the set of elements that specifies opaque information assigned by the protocol server to an item of a drawing page (section 1.3.2), as specified in section 3.1.1, of a **VSDX** web drawing, when it is requested for the first time. The **RasterPageItemTag** is used on subsequent requests to determine whether the same item has changed since the initial request. A **RasterPageItemTag** is equal to the same element in **RasterPageItemRequest** (section 3.1.4.4.3.1), if all its elements have default values.

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

DataModeName: A string that specifies whether the drawing page was refreshed by the protocol server. There MUST be exactly one instance of this element, and it MUST be one of the following values.

Value	Description
Static	The drawing page was not updated and contains only data embedded in the corresponding web drawing. This is the default value.
Dynamic	The drawing page was updated with data retrieved from data providers, as specified in [MS-VSDX] section 2.2.10, or from the system, as specified in [MS-VSDX] section 2.2.11.

FileETag: A string that specifies an opaque identifier for the version of the web drawing. There MUST be exactly one instance of this element. The default value is null.

3.1.4.4.3.3   RasterPageItemResponse

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageItemResponse** contains the set of elements that specifies the response data for the **GetRasterPageItem** (section 3.1.4.4) WSDL operation.

```
<xs:complexType name="RasterPageItemResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageItemTag"/>
    <xs:element minOccurs="0" name="ItemData" nillable="true" type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="ResponseCode" type="tns:RasterPageItemResponseCode"/>
  </xs:sequence>
```

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

EntityTag: A **RasterPageItemTag** complex type (section 3.1.4.4.3.2) that specifies opaque information about the requested drawing page (section 1.3.2) item, as specified in section 3.1.1. There MUST be exactly one instance of this element, and it MUST NOT be null. If the value of **ResponseCode** is "NotModified", it MUST be equal to the value of the **EntityTag** element of the corresponding **RasterPageItemRequest** complex type (section 3.1.4.4.3.1).

ItemData: A base64Binary that specifies the drawing page item data. There MUST be exactly one instance of this element. The value of **ItemData** depends on the value of the **ItemName** element of the corresponding **RasterPageItemRequest**, according to the following table.

Value of ItemName in RasterPageItemRequest	Value of ItemData
Image0	A small image of the rendered drawing page, in Portable Network Graphics (PNG) format as specified in [RFC2083].
Image1	A medium-small image of the rendered drawing page, in Portable Network Graphics (PNG) format as specified in [RFC2083].
Image2	A medium-large image of the rendered drawing page, in Portable Network Graphics (PNG) format as specified in [RFC2083].
Image3	A large image of the rendered drawing page, in Portable Network Graphics (PNG) format as specified in [RFC2083].
ShapeInfo	The shape information of the drawing page, in XML format, as specified in section 2.2.9.1, and compressed using the GZIP format, as specified in [RFC1952].
Comments	The comments of the drawing page, in XML format, as specified in section 2.2.9.2, and encoded using UTF-8, as specified in [RFC3629].

ResponseCode: A **RasterPageItemResponseCode** value (section 3.1.4.4.4.1), that specifies the type of the response. There MUST be exactly one instance of this element.

3.1.4.4.4 Simple Types

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

Simple type	Description
RasterPageItemResponseCode	The possible types of response codes returned as an element of a **GetRasterPageItem** (section 3.1.4.4) WSDL operation.

3.1.4.4.4.1   RasterPageItemResponseCode

Namespace: http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer

The **RasterPageItemResponseCode** simple type specifies the possible types of response codes returned as an element of a **GetRasterPageItem** (section 3.1.4.4) WSDL operation.

```
<xs:simpleType name="RasterPageItemResponseCode" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotModified"/>
    <xs:enumeration value="OK"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **RasterPageItemResponseCode** simple type.

Value	Meaning
NotModified	Specifies that the requested drawing page (section 1.3.2) item, as specified in section 3.1.1, has not changed since it was initially requested by the protocol client.
OK	Specifies that the response includes drawing page item data.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 SaveComments

The **SaveComments** operation saves a comment to a **VSDX** web drawing in a document repository, in the form of a **CommentEntry_Type** element as specified [MS-VSDX] section 2.3.4.2.9.<13>

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

```
<wsdl:operation name="SaveComments" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveComments"
message="tns2:IVisioGraphicsService SaveComments InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveCommentsResponse"
message="tns2:IVisioGraphicsService_SaveComments_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveCommentsVisioGraphicsServiceFaultFa
ult" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService SaveComments VisioGraphicsServiceFaultFault FaultMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IVisioGraphicsService_SaveComments_InputMessage** request (section 3.1.4.5.1.1) WSDL message, and the protocol server MUST respond with an **IVisioGraphicsService_SaveComments_OutputMessage** response (section 3.1.4.5.1.2) WSDL message as follows:

1. If the protocol server determines that an error occurred during the protocol server operation, then the protocol server MUST respond with a **VisioGraphicsServiceFault** complex type (section 2.2.4.5) SOAP fault with the **Error** field set to the correct **FaultCode** value (section 2.2.5.1).

2. Otherwise, the protocol server MUST return a **SaveCommentsResponse** element (section 3.1.4.5.2.2).

3.1.4.5.1 Messages

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

Message	Description
IVisioGraphicsService_SaveComments_InputMessage	The request WSDL message for the **SaveComments** WSDL operation.
IVisioGraphicsService_SaveComments_OutputMessage	The response WSDL message for the **SaveComments** WSDL operation.
IVisioGraphicsService_SaveComments_VisioGraphicsServiceFaultFault_FaultMessage	The fault WSDL message for the **SaveComments** WSDL operation.

3.1.4.5.1.1 IVisioGraphicsService_SaveComments_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IVisioGraphicsService/SaveComments
```

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

3.1.4.5.1.2 IVisioGraphicsService_SaveComments_OutputMessage

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

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

3.1.4.5.1.3 IVisioGraphicsService_SaveComments_VisioGraphicsServiceFaultFault_FaultMessage

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

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

3.1.4.5.2 Elements

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

Element	Description
SaveComments	The input data for the **SaveComments** WSDL operation.
SaveCommentsResponse	The result data for the **SaveComments** WSDL operation.

3.1.4.5.2.1   SaveComments

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

```
<xs:element name="SaveComments" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="fileUrl" nillable="true" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="commentData" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

fileUrl: An **anyURI** that specifies the URL of the **VSDX** web drawing file in a document repository. There MUST be exactly one instance of this element, and it MUST NOT be null.

commentData: A string attribute that specifies data about a comment, in XML format as specified in section 2.2.9.3, for a shape in the **VSDX** web drawing. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.5.2.2   SaveCommentsResponse

The **SaveCommentsResponse** element specifies the result data for the **SaveComments** WSDL operation.

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

SaveCommentsResult: A string that specifies the result of the **SaveComments** operation (section 3.1.4.5), in XML format as specified in section 2.2.9.4. There MUST be exactly one instance of this element, and it MUST NOT be null.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.5 Timer Events

The **Recalculation Timeout** timer event cancels the request for the drawing page, as specified in section 1.3.2, with a **VisioGraphicsServiceFault**, as specified in section 2.2.4.5. The **Error** field of **VisioGraphicsServiceFault** MUST be set to the "RecalcTimedOut" **FaultCode** value, as specified in section 2.2.5.1.

The **Message Sequence Timeout** timer event expires the entity cached on the protocol server that is used to match the sequence identifier of the message sequence initiated by an **IVisioGraphicsService_GetRasterPageItem_InputMessage** request (section 3.1.4.4.1.1) WSDL message to retrieve a drawing page (section 1.3.2) of a **VSDX** web drawing in raster format, as specified in section 3.1.<14>

3.1.6 Other Local Events

None.

4 Protocol Examples

The following sections contain examples of this protocol's usage.

4.1 GetRasterDiagram

In this scenario, the protocol client calls the **GetRasterDiagram** method (section 3.1.4.1) on a protocol server named "ExampleServer". The protocol server responds with a **GetRasterDiagramResponse** element (section 3.1.4.1.2.2). In this case, the drawing page (section 1.3.2) does not contain data providers to be implemented by the protocol client.

To request the raster formatted drawing page, the protocol client constructs the following WSDL message:

```
<s:Envelope>
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IVisioGraphicsService/GetRasterDiagram</a:Action>
    <a:MessageID>urn:uuid:891be72c-aaaf-473a-a4f8-82c3da217414</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">https://ExampleServer:
32844/69e08a7b0e904d1a95de6dbf31c51529/VisioGraphicsService.svc/secure</a:To>
    <o:Security>……</o:Security>
  </s:Header>
  <s:Body>
    <GetRasterDiagram>
      <rasterDiagramRequestContract>
        <b:DataSources i:nil="true"/>
        <b:DisableRefresh>true</b:DisableRefresh>
        <b:FileUri>http://ExampleServer/BasicDiagram.vdw</b:FileUri>
        <b:PageNumber>1</b:PageNumber>
        <b:TimeLastModified>2009-05-20T17:42:20</b:TimeLastModified>
        <b:Resolution>1</b:Resolution>
      </rasterDiagramRequestContract>
    </GetRasterDiagram>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following:

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IVisioGraphicsService/GetRasterDiagramResponse</a:Act
ion>
  </s:Header>
  <s:Body>
    <GetRasterDiagramResponse xmlns="http://tempuri.org/">
      <GetRasterDiagramResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:CacheAge>60</d4p1:CacheAge>
        <d4p1:DataSources i:nil="true"></d4p1:DataSources>
        <d4p1:AreaMaps>
          <Page>
            <Shapes>
              <Shape Name="_1"
Layout="\{&#34;type&#34;:&#34;2D&#34;,&#34;bounds&#34;:\{&#34;x&#34;:192,&#34;y&#34;:192,&#34
;width&#34;:285,&#34;height&#34;:285\}\}">
                <Path Shape="poly" Data="192,477,477,477,477,192,192,192,192,477"/>
```

```
        </Shape>
      </Shapes>
    </Page>
  </d4p1:AreaMaps>
```

<d4p1:Image>iVBORw0KGgoAAAANSUhEUgAAAzAAAAQgCAYAAAA5ebkOAABTr0lEQVR4Xu3a0W1tRwgF0KQyl+ZSXcrLc
yRXsAWYYVmK8sVhZnGFZiv598/fv3//8ESBgAABAgIECBAYIPAd4DxR4AAAQIECBAgIIPAd4DxR4AAAQIAAgQ0C/2w4pDMSIECAAAAECBA
gQIEDg//97DAMBAgQIECBAgAAgS2CAgwWyblnAQIECBAgAABgQI+C8wfgMECBQAgIAAAIECOwR8F9g9szKSQkQIEC
AAAECBAicFxBgzv8EABAgQIAAAQIIECBADYIYDA7JmVkxIgIAAAECBA4LYDAnP8JACBwR0C2TMrrJyVJvAAAQIECBA
gAIADg//97DAMBAgQIAAAAGgS2CAgwWylnAQIECBAgAABgQI+C8wfgMECBQAgIAAAECOwR0CA2TMrJyVAV
gAABAgQIECBwXkCAOf8TAECAAAAECBAgj4AAs2dWTkqAAAECBAgIECBADgvIAAc/4nAIAIECBAgAIDHgEBZs+sn
JQAAAQIECBAgIDAeQEB5vxPAAAEBAgQIECBAgACBPQICzJ5ZOSkBAgQIECAgMC8gwE9wICzPmfAAAAECAAAAEcewQEmD
2zclICBAAQIECAAEC5wUEmPM/AQAECBAgIAAAgS2CAgwe2blpAAAECBAAAAEC5wUEmPM/AQAECBAgIAAAAAgIAAAAQIE9ggIMHtm5aQEC
BAgQIAAAAQIzgsIMOd/AgAIEChAgIECBAAAQYPBGQYPbMykkJECBAgQICAAAACCAgi7BEQYPbMykkJECBAgACBA4LyDA2Cmgw0yZlMSIEC
BAgACB8wICzPmfAAAAECBAgIAAAAgwe2blpAQIECBAgIAAAAgcFBBgzv8EABAgQIAAAQIIECAAAAgIE9ggIMHtm5aQ
ECBAAQIAAgQICAAEcFBGg9IMDsmZWTEiBAgIAAAQIAAAOsAgEABAgQIECBAgMAeAQFmz6yclAABAgQIECBAgMB5AQHm/E8AAAAECAAIE9
AgLMnlk5KQECBAAAAAAAAO5BAQIAAAATOCwgw538CAAgQIECBAAAgQG9ggIMHtm5aQECBAgQICAAAAcFBZgzv8EABAgQIAAAAgQICBADYIYDA7J
mVkxKgIAAAECBA4LSDAnP8JACBwR0C2TMrrJyVJvAAAQIECBAgAIADg//97DAMBAgQIAAAAGgS2CAgwWylnAQIECBAgAABgQI+C8wfgMEC
BQAgIAAAECOwR0CA2TMrJyVAVgAABAgQIECBwXkCAOf8TAECAAAAECBAgj4AAs2dWTkqAAAECBAgIECBADgvIAAc/4nAIAIECBAgAIDHgE
BZs+snJQAAAQIECBAgIDAeQEB5vxPAAAEBAgQIECBAgACBPQICzJ5ZOSkBAgQIECAgMC8gwE9wIABBQ/gxbQ/4eXj46Pt4/Az/7Av/7Zv
HadHQCBAj8CgEB5leMwSSEmBH4ekxO99SRA4J6AnXNv5m5MgECNgABT4+qrCwQ8JhyMyEREJCRg5zw0TFchQBUQIAZ5dd8UsBjYYlJfbwL3BOycz3YYIEagQEmBpqPrqqAgGPiVMTYm59Aya9Atn2M55aJiuQoDAj5M22nlk5KQEEagBFmMKql/mkvIEE5P6ehO4J2Dn3Ju5GxMgUCMgwNS4+uoSLBDwmFzgE8Jk9PPPRMVyFfwh4TCwvkyDAjEjAr0krPingJGgalx9GAk8qERBgBnl13xSwGMhidNTXm55bwkmMYEjAznlomK5CgGKL/dGcr4oNTZUbRgBgbn13xZzFrTAsTwN4+e+k5N23Jju5GxMgwCNQ7C7z0DBdhQCUQEEzi/JpPCnhMTOr+4fGUF+/zSAf4zrk3K5Cdw4CcsmP5pIDH5qrvpAwWAWNIbVVfh4TCwvjlwhCsEHHhKcnIG6SoECIwKDCqX XVBQIeeuG5IgEHhKwcx4apqsQoDAqIMCo8Mms4/k8KAxMamvN4F7AnbOvZm5MgECNQICTI2qri66BGwGP iEHhKwcx4apqsQoDAqIMCo8Mns4/k8KAxMamvN4F7AnbOvZm5MgECNQICTI2qri66BGwGP iEHhKwcx4apqsQoDAqIMCo8Mns4/k8KAxMamvN4F7AnbOvZm5MgECNQICTI2qri66BGwGP
iEHhKwcx4apqsQIDAqIMCM8ms+KeAxMamvN4F7AnbOvZm7MQECNQICTI2rry4Q8JhYMCRHJPCQgJ3z0DB
dhQCBUQEBZpr80kBj4lJfb0J3BOwc+w+7N7BQwc+7N3I3I0JEKgREGRBBqQXH11gyDHxIIhOSKnIeG6SoEICwKDCj/JpPCnhMTOr
r

TeCegJ1zb+ZuTIBAjYAAU+PqqwsEPCYWDMkRCTwkYOc8NExXIUBgVECAGeXXfFLAY2JSX28C9wTsnHszd2MCBGoEBJgaV
19dIOAxsWBIjkjgIQE756FhugoBAqMCAswov+aTAh4Tk/p6E7gnYOfcm7kbEyBQIyDA1Lj66gIBj4kFQ3JEAg8J2DkPDd
NVCBAYFRBgRvk1nxTwmJjU15vAPQE7597M3ZgAgRoBaabG1VcXCHhMLBiSIxJ4SMDOeWiYrkKAwKiAADPKr/mkgMfEpL7
eBO4J2Dn3Zu7GBAjUCAgwNa6+ukDAY2LBkByRwEMCds5Dw3QVAgRGBQSYUX7NJwU8Jib19SZwT8DOuTdzNyZAoEZAgKlx
9dUFAh4TC4bkiAQeErBzHhqmqxAgMCogwIzyaz4p4DExqaz4p4DExqa83gXsCds69mbsxAQI1AgJMjMJauvLhDwmFgwJEck8JCAnfPQM
F2FAIFRAQFmlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk8KeExM6u
tN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMCRHJPCQgJ3z0PFchQGBUQIAZ5dd8UsBjYlJfbwL3BOyce3NzJ4gYECNQICCBp
XX10g4DGxYEiOSOAhATvnoWG6CgECowICzCi/5pMCHhOT+noTuCdg59ybuRsTIFAjIMDUuprQAgGPiQVDckQCDwnYOQ8N
01UIEBgVEBG+TWffPCYmNTXm8A9ATvn3szdmACBGgEBpsbVVxcIeEwsGJIjEnhIwM55aJiuQoDAqIAAM8qv+aSAx8Skvt
4E7gnYOfdm7sYECNQICDA1rr66QMBjYsGQHJHAQwJ2zkPDdBUCBEYFBJhRfs0nBTwmJvX1JnBPwM65N3M3JkCgRkCAqaX
H11QUCHhMLhuSIBB4SsHMeGqarECAwKiDAjPJrPinPilgMTEprzeBewJ2zr2ZuzEBAjUCAkyNq68uEPCYWDAkRyTwkICd89A
wXYUAgVEBAWuUX/NJAY+JSX29CdwTsHPuzdyNCRCoERBgalx9dYGAx8SCITkigYcE7JyHhukqBAiMCggwo/yaTwp4TEzq
603gnoCdc2/mbkyAQI2AAFPj6qsLBDwmFgwJEck8JGDnPDRMVyFAYFRAgBnl13xSwGNiUl9vAvcE7Jx7M3djAgRqBASYG
ldfXSDgMbFgSI5I4CEBO+ehYboKAQKjAgLMKL/mkwIeE5P6ehO4J2Dn3Ju5GxMgUCMgwNS4+uoCAY+JBUNyRAIPCdg5Dw
3TVQgQGBUQYEb5NZ8U8JiY1NebwD0BO+fezN2YAIEaAQGmxtVXFwh4TCwYkiMSeEjAznlomK5CgMCogAAzyq/5pIDHxKS+
3gTuCdg592buxgQI1AgIMDWuvrpAwGNiwZAckcBDAnbOQ8N0FQIERgUEmFF+zScFPCYm9fUmcE/Azrk3czcmQKBGQIC Cp
cfXVBQIeEwuG5IgEHhKwcx4bpqsQIDAqIMCM8ms+KeAxMam vN4F7AnbOvZm7MQECNQICTI2ry4Q8JhYMCRHJPCQgJ3z0
DBdhQCBUQEBZpRf80kBj4lJfb0J3BOwc+7N3I0JEKgRKREGBqXH11gYDHxIIhOSsnIeG6SoECIwKCDCj/JpPCnhMTO
rrTeCegJ1zb+ZuTIBAjYAAU+PqqwsEPCYWDMkRCTwkYOc8NExXIUBgVECAGeXXfFLAY2JSX28C9wTsnHszd2MCBGoEBJg
aV19dIOAxsWBIjkjgIQE756FhugoBAqMCAswov+aTAh4Tk/p6E7gnYOfcm7kbEyBQIyDA1Lj66gIBj4kFQ3JEAg8J2DkP
DdNVCBAYFRBgRvk1nxTwmJjU15vAPQE7597M3ZgAgRoBaabG1VcXCHhMLBiSIxJ4SMDOeWiYrkKAwKiAADPKr/mkgMfEp
L7eBO4J2Dn3Zu7GBAjUCAgwNa6+ukDAY2LBkByRwEMCds5Dw3QVAgRGBQSYUX7NJwU8Jib19SZwT8DOuTdzNyZAoEZAgK
lx9dUFAh4TC4bkiAQeErBzHhqmqxAgMCogwIzyaz4p4DExqaz4p4DExqa83gXsCds69mbsxAQI1AgJMjMJauvLhDwmFgwJEck8JCAnfP
QMF2FAIFRAQFmlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk8KeEx M
6utN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMCRHJPCQgJ3z0PFchQGBUQIAZ5dd8UsBjYlJfbwL3BOyce3NzJ4gYEagQEm
BpXX10g4DGxYEiOSOAhATvnoWG6CgECowICzCi/5pMCHhOT+noTuCdg59ybuRsTIFAjIMDUuprQAgGPiQVDckQCDwnY YQ
08N01UIEBgVEBG+TWffPCYmNTXm8A9ATvn3szdmACBGgEBpsbVVxcIeEwsGJIjEnhIwM55aJiuQoDAqIAAM8qv+aSAx8
kvt4E7gnYOfdm7sYECNQICDA1rr66QMBjYsGQHJHAQwJ2zkPDdBUCBEYFBJhRfs0nBTwmJvX1JnBPwM65N3M3JkCgRkCA
qAXH11QUCHhMLhuSIBB4SsHMeGqarECAwKiDAjPJrPingMTEprzeBewJ2zr2ZuzEBAjUCAkyNq68uEPCYWDAkRyTwkICd8
9AwXYUAgVEBAWuUX/NJAY+JSX29CdwTsHPuzdyNCRCoERBgalx9dYGAx8SCITkigYcE7JyHhukqBAiMCggwo/yaTwp4TE
zq603gnoCdc2/mbkyAQI2AAFPj6qsLBDwmFgwJEck8JGDnPDRMVyFAYFRAgBnl13xSwGNiUl9vAvcE7Jx7M3djAgRqBAS
YGldfXSDgMbFgSI5I4CEBO+ehYboKAQKjAgLMKL/mkwIeE5P6ehO4J2Dn3Ju5GxMgUCMgwNS4+uoCAY+JBUNyRAIPCdg5
Dw3TVQgQGBUQYEb5NZ8U8JiY1NebwD0BO+fezN2YAIEaAQGmxtVXFwh4TCwYkiMSeEjAznlomK5CgMCogAAzyq/5pIDHx
KS+3gTuCdg592buxgQI1AgIMDWuvrpAwGNiwZAckcBDAnbOQ8N0FQIERgUEmFF+zScFPCYm9fUmcE/Azrk3czcmQKBGBG
QICpcfXVBQIeEwuG5IgEHhKwcx4apqsQIDAqIMCM8ms+KeAxMamvN4F7AnbOvZm7MQECNQICTI2ry4Q8JhYMCRHJPCQgJ3
z0DBdhQCBUQEBZpRf80kBj4lJfb0J3BOwc+7N3I0JEKgREGBqXH11gYDHxIIhOSsnIeG6SoECIwKCDCj/JpPCnhM
TOrrTeCegJ1zb+ZuTIBAjYAAU+PqqwsEPCYWDMkRCTwkYOc8NExXIUBgVECAGeXXfFLAY2JSX28C9wTsnHszd2MCBGoEB
JgaV19dIOAxsWBIjkjgIQE756FhugoBAqMCAswov+aTAh4Tk/p6E7gnYOfcm7kbEyBQIyDA1Lj66gIBj4kFQ3JEAg8J2D
kPDdNVCBAYFRBgRvk1nxTwmJjU15vAPQE7597M3ZgAgRoBaabG1VcXCHhMLBiSIxJ4SMDOeWiYrkKAwKiAADPKr/mkgMf
EpL7eBO4J2Dn3Zu7GBAjUCAgwNa6+ukDAY2LBkByRwEMCds5Dw3QVAgRGBQSYUX7NJwU8Jib19SZwT8DOuTdzNyZAoEZA
gKlx9dUFAh4TC4bkiAQeErBzHhqmqxAgMCogwIzyaz4p4DExqa83gXsCds69mbsxAQI1AgJMjMJauvLhDwmFgwJEck8JCAn
fPQMF2FAIFRAQFmlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk8KeE
xM6utN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMCRHJPCQgJ3z0PFchQGBUQIAZ5dd8UsBjYlJfbwL3BOyce3NzJ4gYEagQ
EmBpXX10g4DGxYEiOSOAhATvnoWG6CgECowICzCi/5pMCHhOT+noTuCdg59ybuRsTIFAjIMDUuprQAgGPiQVDckQCDwnY
OQ8N01UIEBgVEBG+TWffPCYmNTXm8A9ATvn3szdmACBGgEBpsbVVxcIeEwsGJIjEnhIwM55aJiuQoDAqIAAM8qv+aSAx
8Skvt4E7gnYOfdm7sYECNQICDA1rr66QMBjYsGQHJHAQwJ2zkPDdBUCBEYFBJhRfs0nBTwmJvX1JnBPwM65N3M3JkCgRkCA
qAXH11QUCHhMLhuSIBB4SsHMeGqarECAwKiDAjPJrPingMTEprzeBewJ2zr2ZuzEBAjUCAkyNq68uEPCYWDAkRyTwkIC d
89AwXYUAgVEBAWuUX/NJAY+JSX29CdwTsHPuzdyNCRCoERBgalx9dYGAx8SCITkigYcE7JyHhukqBAiMCggwo/yaTwp4
TEzq603gnoCdc2/mbkyAQI2AAFPj6qsLBDwmFgwJEck8JGDnPDRMVyFAYFRAgBnl13xSwGNiUl9vAvcE7Jx7M3djAgRqB
ASYGldfXSDgMbFgSI5I4CEBO+ehYboKAQKjAgLMKL/mkwIeE5P6ehO4J2Dn3Ju5GxMgUCMgwNS4+uoCAY+JBUNyRAIPCd
g5Dw3TVQgQGBUQYEb5NZ8U8JiY1NebwD0BO+fezN2YAIEaAQGmxtVXFwh4TCwYkiMSeEjAznlomK5CgMCogAAzyq/5pID
HxKS+3gTuCdg592buxgQI1AgIMDWuvrpAwGNiwZAckcBDAnbOQ8N0FQIERgUEmFF+zScFPCYm9fUmcE/Azrk3czcmQKBG
QICpcfXVBQIeEwuG5IgEHhKwcx4apqsQIDAqIMCM8ms+KeAxMamvN4F7AnbOvZm7MQECNQICTI2ry4Q8JhYMCRHJPCQg
J3z0DBdhQCBUQEBZpRf80kBj4lJfb0J3BOwc+7N3I0JEKgREGBqXH11gYDHxIIhIIhOSsnIeG6SoECIwKCDCj/JpPCn
hMTOrrTeCegJ1zb+ZuTIBAjYAAU+PqqwsEPCYWDMkRCTwkYOc8NExXIUBgVECAGeXXfFLAY2JSX28C9wTsnHszd2MCBGo
EBJgaV19dIOAxsWBIjkjgIQE756FhugoBAqMCAswov+aTAh4Tk/p6E7gnYOfcm7kbEyBQIyDA1Lj66gIBj4kFQ3JEAg8J
2DkPDdNVCBAYFRBgRvk1nxTwmJjU15vAPQE7597M3ZgAgRoBaabG1VcXCHhMLBiSIxJ4SMDOeWiYrkKAwKiAADPKr/mkg
MfEpL7eBO4J2Dn3Zu7GBAjUCAgwNa6+ukDAY2LBkByRwEMCds5Dw3QVAgRGBQSYUX7NJwU8Jib19SZwT8DOuTdzNyZAoE
ZAgKlx9dUFAh4TC4bkiAQeErBzHhqmqxAgMCogwIzyaz4p4DExqa83gXsCds69mbsxAQI1AgJMjMJauvLhDwmFgwJEck8JC
AnfPQMF2FAIFRAQFmlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk8K
eExM6utN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMCRHJPCQgJ3z0PFchQGBUQIAZ5dd8UsBjYlJfbwL3BOyce3NzJ4gYIEa
gQEmBpXX10g4DGxYEiOSOAhATvnoWG6CgECowICzCi/5pMCHhOT+noTuCdg59ybuRsTIFAjIMDUuprQAgGPiQVDckQCDw
nYOQ8N01UIEBgVEBG+TWffPCYmNTXm8A9ATvn3szdmACBGgEBpsbVVxcIeEwsGJIjEnhIwM55aJiuQoDAqIAAM8qv+aS
Ax8Skvt4E7gnYOfdm7sYECNQICDA1rr66QMBjYsGQHJHAQwJ2zkPDdBUCBEYFBJhRfs0nBTwmJvX1JnBPwM65N3M3JkCg
RkCAqAXH11QUCHhMLhuSIBB4SsHMeGqarECAwKiDAjPJrPingMTEprzeBewJ2zr2ZuzEBAjUCAkyNq68uEPCYWDAkRyTwk
ICd89AwXYUAgVEBAWuUX/NJAY+JSX29CdwTsHPuzdyNCRCoERBgalx9dYGAx8SCITkigYcE7JyHhukqBAiMCggwo/yaTw
p4TEzq603gnoCdc2/mbkyAQI2AAFPj6qsLBDwmFgwJEck8JGDnPDRMVyFAYFRAgBnl13xSwGNiUl9vAvcE7Jx7M3djAgR
qBASYGldfXSDgMbFgSI5I4CEBO+ehYboKAQKjAgLMKL/mkwIeE5P6ehO4J2Dn3Ju5GxMgUCMgwNS4+uoCAY+JBUNyRAIPCd
Cdg5Dw3TVQgQGBUQYEb5NZ8U8JiY1NebwD0BO+fezN2YAIEaAQGmxtVXFwh4TCwYkiMSeEjAznlomK5CgMCogAAzyq/5p
IDHxKS+3gTuCdg592buxgQI1AgIMDWuvrpAwGNiwZAckcBDAnbOQ8N0FQIERgUEmFF+zScFPCYm9fUmcE/Azrk3czcmQK

BGQICpcfXVBQIeEwuG5IgEHhKwcx4apqsQIDAqIMCM8ms+KeAxMamvN4F7AnbOvZm7MQECNQICTI2rry4Q8JhYMCRHJPC
QgJ3z0DBdhQCBUQEBZpRf80kBj4lJfb0J3BOwc+7N3I0JEKgREGBqXH11gYDHxIIhOSKBhwTsnIeG6SoECIwKCDCj/JpP
CnhMTOrrTeCegJ1zb+ZuTIBAjYAAU+PqqwsEPCYWDMkRCTwkYOc8NExxXIUBgVECAGeXXfFLAY2JSX28C9wTsnHszd2MCB
GoEBJgaV19dIOAxsWBIjkjgIQE756FhugoBAqMCAswov+aTAh4Tk/p6E7gnYOfcm7kbEyBQIyDA1Lj66gIBj4kFQ3jJEAg
8J2DkPDdNVCBAYFRBgRvk1nxTwmJjU15vAPQE7597M3ZgAgRoBAabG1VcXCHhMLBiSIxJ4SMDOeWiYrkAwKiAADPKr/m
kgMfEpL7eBO4J2Dn3Zu7GBAjUCAgwNa6+ukDAY2LBkByRwEMCds5Dw3QVAgWjAgLDKr9dUFAh4TC4bkiAQeErBzHhqmqAgM
CogwIzyaz4p4DExqa83gXsCds69mbsxAQI1AgKMjauvLhDwmFgwJEck8JCAnfPQMF2FAMFoGkD8wmFgwJEck8
JCAnfPQMF2FAIFRAQFmlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk
8KeAxM6utN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMyREJPCRg5zw0TFchQDAqIMCs8mv4IXWg
EagQEmBpXX10g4DGxYEiOSOAhATvnoWG6CgECowIICTi/5pMCHhOT+noTuCdg59ybuRsTIFAjIMDUuPrqAgGPiVQVDckQC
DwnY08Aw8XYUUAgVEBAWaaUX/NJAY+JSX29CdwTsHPuzdyNCRCoERBgaly9dYGAx8SCITkHPuzdyNCRC
wpp4TEzq603gnoCdc2/mbkyAQI2AAFPj6qsLBDwmFgzJEQk8JGDnPDQMF2FAMFoGkD8wmFgwJEck8JCAnfPQMF2FAIFRAQF
mlF/zSQGPiUl9vQncE7Bz7s3cjQkQqBEQYGpcfXWBgMfEgiE5IoGHBOych4bpKgQIjAoIMKP8mk
8KeExM6utN4J6AnXNv5m5MgECNgABT4+qrCwQ8JhYMyREJPCRg5zw0TFchQDAqIMCs8mv4IX
ZAoEZAgKlx9dUFAt+PiY+Pjz8/jwr//mTx9zfhd8Cg6jfws28WrEdHJECAwK8WEGB+9Xgcrl1Lg6+vLY9W+Y9Y9WD3W/Ab6D1N/C
9d/wRIECAQCYgwGR+qgkQIECAAAECBAgQaBQQYBqxtSJAgABAgQICBAIBIBMQYDI/1QQIECBAABAgQINAoIMI3IY3WhEg
QIAAAQIECBAgkAkIMJJmfagIECBAgQIAAAQIEGgUEmEZsrQgQIECAAAECBAgQyAQEmEmMxPNQECBAgQIECjQICTCO2V
gQIECBAgAABAgIZAICTOanmgABAgQIECBAgACBQgEBphFbKwIECBAgQIAAAQIIOAgEBJgaV19dIOAxsWBIjkjgIQE756FhugoBAqMC
ANGJrRYAAAQIECBAgQIBAjiDAZH6qcRAgIAAQIECBBoFBBgGrG1IkCAAAECBAgQIEAgExBgJjMTz81AgQIECBAg0
CgwjdhaESBAgAABAgQIEECgQwmwZ9qAgQIECBAgAABAgQaBQSYRmytCBAgAABAgQIECDBDIBASYzE81AgECBAAKNNAgJMI7
ZWBAgQIECBAAhkAgJM5qeaAAECBAgQIECAAIFGAQGmEVsrAgQIECBAgQyAQEmEm81NNgAABAgQIEm81NNgAABAgQIEC
BAgECjgADTiK0VAQIECBAgQIAAAgZKBQpxNaKAAECBAgQIECAAIFFMQIDJ/FQTIEAAAAECBAgQINAoIMA0YmtFg
CBAgINAoIMA0YmtFgICBAgECBAgAECAEAoFBJgGbK0IECBAgQIECBAgAECjgADTiK0VAQIEGggCBKoIECBAECBAgQIECBAg0
CggwjdhaESBAAABAgQIAAABAhkAgJM5qeaAAECBAgQIECBBoFDBgGrG1IkCAAAECBAgQIAAgOUZCAgJM5qea
ZWBAgQIECBAAhkAgJM5qeaAAECBAgQIIAAgOUZAABCgg8qEB5qea
8VBMgQIAAAQIECBAg0CggwDRia0WAAABAgQIECBAIBIMYDI/1QQIECBAgQIECBAgAAgZKBRmytCBAgAACBAgQIEEgEBJg0
YDI/1QQIECBAgQIECBBoFDBgGrG1IkCAAAECBAgQINAoIMI3IY3WhEgQIAAAQIECBBoFBBgGrG1IkCAAAECBAgINAo4AA04
itFQECBAgQIECAAAEGCYAk/mpJkCAAAECBAgQINAoIMI0YmtFgAABBAgQIECBBIBMQYDI/1QQIECBAgQIECBBoFDBgGrG1IkCAAAECBAgI
IAAgOUZAAABCgg8qEB5qeaAAECBAgQIAAAgIAgOUZAAABCggwjdhaESBAAABAgQIAAABAhkAgJM5qeaAAECBAgQIAAgU
YBAaYRWysCBAgQIECBDIBASbzU02AAABAgQIAAAAQoFAABKECBAgQCgUEmAZsrQgQIECBAABAgIAQYMkGBBgGbG1IkCAAAECBAg0
CgwDRia0WAAAEECBAgQIEEECggwjdhaESBAgQIAAABAhkAgJM5qeaAAECBAgQIAAgOUZAAAECBBoFFAgGnE1ooAAABBAgQIAAgU
AAAECBAgQIECAQCYgwGR+qgkQIECAAABAgQaBQYYBqxtSJAgABAgQIAAABIBMQYDI/1QQIECBAgAABAg0CgwjdhaESBAgQIAAABAgkAkIMJJmfagIECBAgQIE
CBAgAABAgZKBRayCAgIAAAAECjQICTCO2VgQIECBAgAABAgQIAAgOUZAAABCggwjdhaESBAgAABAgQIECBBoFBBgGrG1IkCAAAgOUZAAABCgg8qEB5qea
AAAECBAgQIECAoFFAgGnE1ooAAAABBAgQIECBAECBAgAAgZKBQpxNaKAAECBAgQIAAgOUZAAABCgg8qEB5qeaAAAECBAgQIECAAAECBAgINAoIMA0YmtFg
AA04itFQECBAgQIECAAAECBBoFFAgGnE1ooAAAABBAgQIECBAECBAgAAAgOUZAAAECBDIBASbzU02AAAECBAgQIAAgU
YBAaYRWysCBAgQIECBDIBASbzU02AAAECBAgQIECBBoFDBgGrG1IkCAAAECBAgINAoIMA0YmtFg
IECBAgECjgADTiK0VAQIECBAgQIAAAgUZAAABCgg8qEB5qeaAAECBAgQIECAAIFFMQIDJ/FQTIECA

[MS-VGSP] - v20150824
Visio Graphics Service Protocol
Copyright © 2015 Microsoft Corporation
Release: August 24, 2015

AAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9UEC
BAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTz
UBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgQIAAgUYBAaYRWysCBAgQIECAAAECBDIBASb
zU02AAAECBAgQIECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgnE1ooAAQIECBAgAAAgUxA
gMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQIECAQCYgwGR+qgkQIECAAAECBAgQaBQQYBqxtSJAgAABAgQIECBAI
BMQYDI/1QQIECBAgAABAgQINAoIMI3YWhEgQIAAAQIECBAgkAkIMJmfagIECBAgAAAQIEGgUEmEZsrQgQIECAAAECBA
gQyAQEmMxPNQECBAgQIECAAAECjQICTCO2VgQIECBAgAABAgQIZAICTCOanmgABAgQIECBAgACRRgEBphFbKwIECBAgQIA
AAQIEMgEBJvNTTYAAAAQIECBAgQIBAo4AA04itFQECBAgQIECAAAAECmYAAk/mpJkCAAAECBAgQIECgUUCAacTWigABAgQI
ECBAgACBTECAyfxUEyBAgAABAgQIECDQKCDANGJrRYAAAQIECBAgQIBAJiDAZH6qCRAgQIAAAQIECBBoFBBgGrG1IkCAA
AAECBAgQIEAgExBgMj/VBAgQIAAAQIECBAg0CggwjdhaESBAgAABAgQIECCQCQgwmZ9qAgQIECBAgAABAgQaBQSYRmytCB
AgQIAAAQIECBBDIBASbzE81AQIECBAgAAAQKNAgMJI7ZWBAgQIAAAECBAhkAgJM5qeaAAECBAgAAIFGAQGmEVsrAgQIEC
BAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECBAgAIAAAQKZgACT+akmQIAAAQIECBQKBRQIBpxNaKAAECBAgAAIFMQIDJ/FQTIECAAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9
UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgQIAAgUYBAaYRWysCBAgQIECAAAECBDIBASbzU02AAAECBAgQ
IECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgnE1ooAAQIECBAgAAAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQIECAQCYgwGR+qgkQIECAAAECBAgQaBQQYBqxtSJAgAABAgQIECBAIBMQYDI/1QQIECBAgAABAgQINAoIMI3YWhEg
QIAAAQIECBAgkAkIMJmfagIECBAgAAAQIEGgUEmEZsrQgQIECAAAECBAgQyAQEmMxPNQECBAgQIECAAAECjQKCTCO2VgQIECBAgAABAgQIZAICTCOanmgABAgQIECBAgACRRgEBphFbKwIECBAgAAAQIEMgEBJvNTTYAAAAQIECBAgQIBAo4AA04itFQECBAgQIECAAAAECmYAAk/mp
JkCAAAECBAgQIECgUUCAacTWigABAgQIECBAgACBTECAyfxUEyBAgAABAgQIECDQKCDANGJrRYAAAQIECBAgQIBAJiDAZH6qCRAgQIAAAQIECBBoFBBgGrG1IkCAAAECBAgQIEAgExBgMj/VBAgQIAAAQIECBAg0CggwjdhaESBAgAABAgQIECCQCQgwmZ9qAgQIECBAgAABAgQaB
QSYRmytCBAgQIAAAQIECBDIBASbzE81AQIECBAgAAAQKNAgMJI7ZWBAgQIAAAECBAhkAgJM5qeaAAECBAgAAIFGAQGmEVsrAgQIECBAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECBAgAIAAAQKZgACT+akmQIAAAQIECBQKBRQIBpxNaKAAECBAgAAIFMQIDJ/FQT
ICAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9
UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJj
MTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgQIAAgUYBAaYRWysCBAgQIECAAAECBDIB
ASbzU02AAAECBAgQIECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgnE1ooAAQIECBAgAAAgUIAAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQIECAQCYgwGR+qgkQIECAAAECBAgQaBQSYRmyt
CBAgQIAAAQIECBDIBASbzE81AQIECBAgAAAQKNAgMJI7ZWBAgQIAAAECBAhkAgJM5qeaAAECBAgAAIFGAQGmEVsrAgQIEC
BAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECBAgAIAAAQKZgACT+akmQIAAAQIECBQKBRQIBpxNaKAAECBAgQIECAAIFMQIDJ/FQTICAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEG
gUEGAasbUiQIAAAQIECBAgQCATEGAyP9UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCDCZn2oCBAgQIECAAAE
CBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAg
QIAAgUYBAaYRWysCBAgQIECAAAECBDIBASbzU02AAAECBAgQIECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgnE1ooAAQIECBAgAAAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQIECAQCYgwGR+qgkQI
ECAAAECBAgQaBQYYBqxtSJAgAABAgQIECBAIBMQYDI/1QQIECBAgAABAgQINAoIMI3YWhEgQIAAAQIECBAgkAkIMJmfagI
ECBAgAAAQIEGgUEmEZsrQgQIECAAAECBAgQyAQEmMxPNQECBAgQIECAAAECjQICTCO2VgQIECBAgAABAgQIZAICTCOan
mgABAgQIECBAgACRRgEBphFbKwIECBAgAAAQIEMgEBJvNTTYAAAAQIECBAgQIBAo4AA04itFQECBAgQIECAAAAECmYAAk
/mpJkCAAAECBAgQIECgUUCAacTWigABAgQIECBAgACBTECAyfxUEyBAgAABAgQIECDQKCDANGJrRYAAAQIECBAgQIBAJiBAJi
DAZH6qCRAgQIAAAQIECBBoFBBgGrG1IkCAAAECBAgQIEAgExBgMj/VBAgQIAAAQIECBAg0CggwjdhaESBAgAABAgQIECC
QCQgwmZ9qAgQIECBAgAABAgQaBQSYRmytCBAgQIAAAQIECBDIBASYzE81AQIECBAgAAAQKNAgMJI7ZWBAgQIAAAECBA
hkAgJM5qeaAAECBAgAAIFGAQGmEVsrAgQIECBAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECBAgAIAAAQ
KZgACT+akmQIAAAQIECBQKBRQIBpxNaKAAECBAgAAIFMQIDJ/FQTIECAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAm
IMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQI
JAJCDCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgQIAAgUYBAaYRWysCBAgQIECAAAECBDIBASbzU02AAAECBAgQIECAQKOAANOIrRUB
AgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgnE1ooAAQIECBAgAAAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRi
a0WAAAECBAgQIECAQCYgwGR+qgkQIECAAAECBAgQaBQYYBqxtSJAgAABAgQIECBAIBMQYDI/1QQIECBAgAABAgQINAo
IMI3YWhEgQIAAAQIECBAgkAkIMJmfagIECBAgAAAQIEGgUEmEZsrQgQIECAAAECBAgQyAQEmMxPNQECBAgQIECAAAECjQICTCO2VgQIECBAgAABAgQIZAICTCOanmgABAgQIECBAgACRRgEBphFbKwIECBAgAAAQIEMgEBJvNTTYAAAAQIECBAgQ
IBAo4AA04itFQECBAgQIECAAAAECmYAAk/mpJkCAAAECBAgQIECgUUCAacTWigABAgQIECBAgACBTECAyfxUEyBAgAABAg

QIECDQKCDANGJrRYAAAQIECBAgQIBAJiDAZH6qCRAgQIAAAQIECBBoFBBgGrG1IkCAAAECBAgQIEAgExBgMj/VBAgQIEC
AAAECBAg0CggwjdhaESBAgAABAgQIECCQCQgwmZ9qAgQIECBAgAABAgQaBQSYRmytCBAgQIAAAQIECBDIBASYzE81AQIE
CBAgQIAAAQKNAgJMI7ZWBAgQIECAAAECBAhkAgJM5qeaAAECBAgQIECAAIFGAQGmEVsrAgQIECBAgAABAgQyAQEm81NNg
AABAgQIECBAgECjgADTiK0VAQIECBAgAAAQKZgACT+akmQIAAAQIECBAgQKBRQIBpxNaKAAECBAgQIECAAIFMQIDJ/F
QTIECAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGA
yP9UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgE
BJjMTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgAAAgUYBAaaRWysCBAgQIAAAQIECAAAECB
DIBASbzU02AAAECBAgQIECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAAABAgQIEBAoFFAgGnE1ooAAQIECBAgQI
AAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQICAQCYgwGR+qgkQIECAAAECBBoFBQYYBqxtSJAgAABAgQ
IECBAIBMQYDI/1QQIECBAgAABAgQINAoIMI3YWhEgQIAAAQIEGgkAkAkIMJmfagIECBAgQIAABAgQyAQEmEZsrQgQIECA
AAECBAgQyAQEmMxPNQECBAgQIECAAAEECjICTCO2VgQIECBAgABARgEBEphFbKwIECBAgAABAgQIEMgEBJvNTTYAAQIEC
BAgQIAABAgQACBTECAyfxUEyBAgAABAgQIEDQKCDCN2FoRIECAAAECBAgQIBAJiDAZH6qCRAgQIAAAQIECBBoFBBgGrG
1IkCAAAECBAgQIEAgExBgMj/VBAgQIECAAAECBAg0CggwjdhaESBAgAABAgQIECCQCQgwmZ9qAgQIECBAgAABAgQaBQSY
RmytCBAgQIAAAQIECBDIBASYzE81AQIECBAgQIAAAQKNAgJMI7ZWBAgQIECAAAECBAhkAgJM5qeaAAECBAgQIECAAIFGA
QGmEVsrAgQIECBAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECBAgAIAAAQKZgACT+akmQIAAAQIECBAgQK
BRQIBpxNaKAAECBAgQIECAAIFMQIDJ/FQTIECAAAECBAgQINAoIMA0YmtFgAABAgQIECBAgEAmIMBkfqoJECBAgAABAgQ
IEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9UECBAgQIAAAQIECDQKCDCN2FoRIECAAAECBAgQIJAJCDCZn2oCBAgQIECA
AAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTzUBAgQIECBAgAABAo0CAkwjtlYECBAgQIAAAQIECGQCAkzmp5oAAQIEC
BAgQIAAgUYBAaaRWysCBAgQIAAAQIECBDIBASbzU02AAAECBAgQIECAQKOAANOIrRUBAgQIECBAgAABApmAAJP5qSZAgA
ABAgQIECBAoFFAgGnE1ooAAQIECBAgQIAAgUxAgMn8VBMgQIAAAQIECBAg0CggwDRia0WAAAECBAgQICAQCYgwGR+qgk
QIECAAAECBAgQaBQQYBqxtSJAgAABAgQIECBAIBMQYDI/1QQIECBAgAABAgQINAoIMI3YWhEgQIAAAQIECBAgkAkIMJmf
agIECBAgQIAAAQIEGgUEmEZsrQgQIECAAAECBAgQyAQEmMxPNQECBAgQIAAAECjICTCO2VgQIECBAgAABAgQIZAICTO
anmgABAgQIECBAgACBRgEBphFbKwIECBAgAAAQIEMgEBJvNTTYAAAQIECBAgQIBAo4AA04itFQECBAgQIECAAAECmY
AAk/mpJkCAAAECBAgQIECgUUCAacTWigABAgQIECBAgACBTECAyfxUEyBAgAABAgQIECDQKCDCN2FoRIECAAAECBAgQIB
AJiDAZH6qCRAgQIAAAQIECBBoFBBgGrG1IkCAAAECBAgQIEAgExBgMj/VBAgQIECAAAECBAg0CggwjdhaESBAgAABAgQI
ECCQCQgwmZ9qAgQIECBAgAABAgQaBQSYRmytCBAgQIAAAQIECBDIBASYzE81AQIECBAgQIAAAQKNAgJMI7ZWBAgQIECAA
AECBAhkAgJM5qeaAAECBAgQIECAAIFGAQGmEVsrAgQIECBAgAABAgQyAQEm81NNgAABAgQIECBAgECjgADTiK0VAQIECB
AgQIAAAQKZgACT+akmQIAAAQIECBAgQKBRQIBpxNaKAAECBAgQIECAAIFMQIDJ/FQTIECAAAECBAgQINAoIMA0YmtFgAA
BAgQIECBAgEAmIMBkfqoJECBAgAABAgQIEGgUEGAasbUiQIAAAQIECBAgQCATEGAyP9UECBAgQIAAAQIECDQKCDCN2FoR
IECAAAECBAgQIJAJCDCZn2oCBAgQIECAAAECBBoFBJhGbK0IECBAgAABAgQIEMgEBJjMTzUBAgQIECBAgAABAo0CAkwjt
lYECBAgQIAAAQIECGQCAkzmp5oAAQIECBAgQIAAgUYBAaaRWysCBAgQIAAAQIECBDIBASbzU02AAAECBAgQIECAQKOAAN
OIrRUBAgQIECBAgAABApmAAJP5qSZAgAABAgQIECBAoFFAgGnE1ooAAQIECBAgQIAAgUxAgMn8VBMgQIAAAQIECBAg0Cg
gwDRia0WAAAECBAgQICAQCYgwGR+qgkQIECAAAECBAgQaBQQYBqxtSJAgAABAgQIECBAIBP4D72IbrKC1aEiAAAAAElF
TkSuQmCC</d4p1:Image>
            <d4p1:ShapeInfo>
            <Page Name="Page-1" Zoom="-1.000000" OffsetX="407.142857" OffsetY="528.857143"
DefaultUnitsText="50" DefaultUnitsAngle="81" DefaultUnitsDuration="44" DefaultUnitsPage="73"
DocumentID="1360559653">
                <Pages>
                  <Page Name="Page-1" />
                </Pages>
                <ShapeInfo Name="_1" DisplayName="Sheet.1"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:191,&quot;y&quot
;:191,&quot;width&quot;:287,&quot;height&quot;:287\}\}" />
              </Page>
          </d4p1:ShapeInfo>
      </GetRasterDiagramResult>
    </GetRasterDiagramResponse>
  </s:Body>
</s:Envelope>

4.2 GetVectorDiagram

In this scenario, the protocol client calls the **GetVectorDiagram** method (section 3.1.4.2) on a protocol server named "ExampleServer". The protocol server responds with a **GetVectorDiagramResponse** element (section 3.1.4.2.2.2). In this case the drawing page (section 1.3.2) does not contain data providers to be implemented by the protocol client.

To request the vector formatted drawing page, the protocol client constructs the following WSDL message:

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

```
    <a:Action s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">http://tempuri.org/IVisioGraphicsService/GetVe
ctorDiagram</a:Action>
    <a:MessageID xmlns:a="http://www.w3.org/2005/08/addressing">urn:uuid:d6266798-5744-42ed-
9753-cc263fda28ae</a:MessageID>
    <a:ReplyTo xmlns:a="http://www.w3.org/2005/08/addressing">
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">https://ExampleServer:32844/69e08a7b0e904d1a95
de6dbf31c51529/VisioGraphicsService.svc/secure</a:To>
    <o:Security xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">……</o:Security>
  </s:Header>
  <s:Body>
    <GetVectorDiagram xmlns="http://tempuri.org/">
      <vectorDiagramRequestContract
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:DataSources i:nil="true"></b:DataSources>
        <b:DisableRefresh>true</b:DisableRefresh>
        <b:FileUri>http://ExampleServer/BasicDiagram.vdw</b:FileUri>
        <b:PageNumber>1</b:PageNumber>
        <b:TimeLastModified>2009-05-20T17:42:20</b:TimeLastModified>
      </vectorDiagramRequestContract>
    </GetVectorDiagram>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following:

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IVisioGraphicsService/GetVectorDiagramResponse</a:Act
ion>
  </s:Header>
  <s:Body>
    <GetVectorDiagramResponse xmlns="http://tempuri.org/">
      <GetVectorDiagramResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:CacheAge>60</d4p1:CacheAge>
        <d4p1:DataSources i:nil="true"></d4p1:DataSources>
        <d4p1:OutputPage>
          <d4p1:Resources></d4p1:Resources>
          <d4p1:ShapeInfo>
            <Page Name="Page-1" Zoom="-1.000000" OffsetX="407.142857" OffsetY="528.857143"
DefaultUnitsText="50" DefaultUnitsAngle="81" DefaultUnitsDuration="44" DefaultUnitsPage="73"
DocumentID="1360559653">
              <Pages>
                <Page Name="Page-1" />
              </Pages>
              <ShapeInfo Name=" 1" DisplayName="Sheet.1"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:191,&quot;y&quot
;:191,&quot;width&quot;:287,&quot;height&quot;:287\}\}" />
            </Page>
          </d4p1:ShapeInfo>
          <d4p1:Xaml>
            <Canvas Height="1056" Name="Workspace" Width="816"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
              <!-- Page  -->
              <Canvas Width="816" Height="1056" Name="D">
                <Canvas.Background>
                  <SolidColorBrush Color="#FFFFFFFF"/>
                </Canvas.Background>
```

```
                <Canvas
Tag="\{&#34;type&#34;:&#34;2D&#34;,&#34;bounds&#34;:\{&#34;x&#34;:192,&#34;y&#34;:192,&#34;wi
dth&#34;:285,&#34;height&#34;:285\}\}"
                           Name="_1" RenderTransform="1, 0, 0, 1, 192, -579">
                  <Rectangle Canvas.Left="-0.5" Canvas.Top="770.5" Height="286"
Name="_1 G0 P0" Width="286" StrokeThickness="1"
                             StrokeDashCap="Round" StrokeEndLineCap="Round"
StrokeLineJoin="Round" StrokeStartLineCap="Round">
                    <Rectangle.Stroke>
                      <SolidColorBrush Color="#FF000000"/>
                    </Rectangle.Stroke>
                    <Rectangle.Fill>
                      <SolidColorBrush Color="#FFFFFFFF"/>
                    </Rectangle.Fill>
                  </Rectangle>
                </Canvas>
              </Canvas>
            </Canvas>
          </d4p1:Xaml>
        </d4p1:OutputPage>
      </GetVectorDiagramResult>
    </GetVectorDiagramResponse>
  </s:Body>
</s:Envelope>
```

4.3   GetVectorDiagram with AddonDataSource

In this scenario, the protocol client calls the **GetVectorDiagram** method (section 3.1.4.2) on a protocol server named "ExampleServer". In this case the protocol client is requesting a drawing page (section 1.3.2) in a web drawing that is linked to data with a data provider to be implemented by the protocol client.

Initially the protocol client constructs the following WSDL message:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">http://tempuri.org/IVisioGraphicsService/GetVe
ctorDiagram</a:Action>
    <a:MessageID xmlns:a="http://www.w3.org/2005/08/addressing">urn:uuid:cd490d30-778d-447c-
91f3-c0b0ea0e5de4</a:MessageID>
    <a:ReplyTo xmlns:a="http://www.w3.org/2005/08/addressing">
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">https://ExampleServer:32844/69e08a7b0e904d1a95
de6dbf31c51529/VisioGraphicsService.svc/secure</a:To>
    <o:Security xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">……</o:Security>
  </s:Header>
  <s:Body>
    <GetVectorDiagram xmlns="http://tempuri.org/">
      <vectorDiagramRequestContract
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:DataSources i:nil="true"></b:DataSources>
        <b:DisableRefresh>false</b:DisableRefresh>
        <b:FileUri>http://ExampleServer/Data_Connected.vdw</b:FileUri>
        <b:PageNumber>1</b:PageNumber>
        <b:TimeLastModified>2009-05-22T16:24:27</b:TimeLastModified>
      </vectorDiagramRequestContract>
    </GetVectorDiagram>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following **GetVectorDiagramResponse** element (section 3.1.4.2.2.2):

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IVisioGraphicsService/GetVectorDiagramResponse</a:Act
ion>
  </s:Header>
  <s:Body>
    <GetVectorDiagramResponse xmlns="http://tempuri.org/">
      <GetVectorDiagramResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:CacheAge>0</d4p1:CacheAge>
        <d4p1:DataSources>
          <d4p1:AddonDataSource>
            <d4p1:ConnectionString>datamodule=Microsoft.Office.Visio.Server.EcsDataHandler,
Microsoft.Office.Visio.Server, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c;provider=Microsoft.ACE.OLEDB.12.0;user id=Admin;data
source=http://ExampleServer/sampledata.xlsx;mode=Read;extended
properties='HDR=YES;imex=1;maxscanrows=0;excel 12.0;';jet oledb:system database='';jet
oledb:registry path='';jet oledb:engine type=37;jet oledb:database locking mode=0;jet
oledb:global partial bulk ops=2;jet oledb:global bulk transactions=1;jet oledb:new database
password='';jet oledb:create system database=False;jet oledb:encrypt database=False;jet
oledb:don't copy locale on compact=False;jet oledb:compact without replica repair=False;jet
oledb:sfp=False;jet oledb:support complex data=False</d4p1:ConnectionString>
            <d4p1:Data>
              <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
                <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="">
                  <xs:complexType>
                    <xs:choice minOccurs="0" maxOccurs="unbounded">
                      <xs:element name="Table1" msdata:Locale="">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="ID" type="xs:string"
minOccurs="0"></xs:element>
                            <xs:element name="FN" type="xs:string"
minOccurs="0"></xs:element>
                            <xs:element name="LN" type="xs:string"
minOccurs="0"></xs:element>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:choice>
                  </xs:complexType>
                </xs:element>
              </xs:schema>
              <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1"></diffgr:diffgram>
            </d4p1:Data>
            <d4p1:Id>1</d4p1:Id>
            <d4p1:QueryString>SheetName=Sheet1; RangeName=A1:C5;</d4p1:QueryString>
          </d4p1:AddonDataSource>
        </d4p1:DataSources>
        <d4p1:OutputPage i:nil="true"></d4p1:OutputPage>
      </GetVectorDiagramResult>
    </GetVectorDiagramResponse>
  </s:Body>
</s:Envelope>
```

To request the vector formatted drawing page (section 1.3.2) with an **AddonDataSource** complex type (section 2.2.4.1), the protocol client now constructs the following WSDL message:

```xml
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">http://tempuri.org/IVisioGraphicsService/GetVe
ctorDiagram</a:Action>
    <a:MessageID xmlns:a="http://www.w3.org/2005/08/addressing">urn:uuid:c22e255b-6923-496d-
a3f5-a2f3c45f4bbd</a:MessageID>
    <a:ReplyTo xmlns:a="http://www.w3.org/2005/08/addressing">
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1"
xmlns:a="http://www.w3.org/2005/08/addressing">https://ExampleServer:32844/69e08a7b0e904d1a95
de6dbf31c51529/VisioGraphicsService.svc/secure</a:To>
    <o:Security xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">……</o:Security>
  </s:Header>
  <s:Body>
    <GetVectorDiagram xmlns="http://tempuri.org/">
      <vectorDiagramRequest
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:DataSources>
          <b:AddonDataSource>
            <b:ConnectionString>datamodule=Microsoft.Office.Visio.Server.EcsDataHandler,
Microsoft.Office.Visio.Server, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c;provider=Microsoft.ACE.OLEDB.12.0;user id=Admin;data
source=http://ExampleServer/sampledata.xlsx;mode=Read;extended
properties='HDR=YES;imex=1;maxscanrows=0;excel 12.0;';jet oledb:system database='';jet
oledb:registry path='';jet oledb:engine type=37;jet oledb:database locking mode=0;jet
oledb:global partial bulk ops=2;jet oledb:global bulk transactions=1;jet oledb:new database
password='';jet oledb:create system database=False;jet oledb:encrypt database=False;jet
oledb:don't copy locale on compact=False;jet oledb:compact without replica repair=False;jet
oledb:sfp=False;jet oledb:support complex data=False</b:ConnectionString>
            <b:Data>
              <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=""
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
                <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="">
                  <xs:complexType>
                    <xs:choice minOccurs="0" maxOccurs="unbounded">
                      <xs:element name="Table1">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="ID" type="xs:double"
minOccurs="0"></xs:element>
                            <xs:element name="FN" type="xs:string"
minOccurs="0"></xs:element>
                            <xs:element name="LN" type="xs:string"
minOccurs="0"></xs:element>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:choice>
                  </xs:complexType>
                </xs:element>
              </xs:schema>
              <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
                <NewDataSet xmlns="">
                  <Table1 diffgr:id="Table11" msdata:rowOrder="0"
diffgr:hasChanges="inserted">
                    <ID>1</ID>
                    <FN>Happy</FN>
                    <LN>kitty</LN>
                  </Table1>
                  <Table1 diffgr:id="Table12" msdata:rowOrder="1"
diffgr:hasChanges="inserted">
                    <ID>2</ID>
                    <FN>Silly</FN>
                    <LN>dog</LN>
```

```
                    </Table1>
                    <Table1 diffgr:id="Table13" msdata:rowOrder="2"
diffgr:hasChanges="inserted">
                        <ID>3</ID>
                        <FN>Order</FN>
                        <LN>Id</LN>
                    </Table1>
                    <Table1 diffgr:id="Table14" msdata:rowOrder="3"
diffgr:hasChanges="inserted">
                        <ID>4</ID>
                        <FN>green</FN>
                        <LN>turtle</LN>
                    </Table1>
                </NewDataSet>
            </diffgr:diffgram>
        </b:Data>
        <b:Id>1</b:Id>
        <b:QueryString>SheetName=Sheet1; RangeName=A1:C5;</b:QueryString>
      </b:AddonDataSource>
    </b:DataSources>
    <b:DisableRefresh>false</b:DisableRefresh>
    <b:FileUri>http://ExampleServer/Data_Connected.vdw</b:FileUri>
    <b:PageNumber>1</b:PageNumber>
    <b:TimeLastModified>2009-05-22T16:24:27</b:TimeLastModified>
      </vectorDiagramRequest>
    </GetVectorDiagram>
  </s:Body>
</s:Envelope>
```

Finally, the protocol server responds with the following **GetVectorDiagramResponse** element (section 3.1.4.2.2.2):

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://tempuri.org/IVisioGraphicsService/GetVectorDiagramResponse</a:Act
ion>
  </s:Header>
  <s:Body>
    <GetVectorDiagramResponse xmlns="http://tempuri.org/">
      <GetVectorDiagramResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:CacheAge>5</d4p1:CacheAge>
        <d4p1:DataSources i:nil="true"></d4p1:DataSources>

        <d4p1:OutputPage>
          <d4p1:Resources></d4p1:Resources>
          <d4p1:ShapeInfo>
            <Page Name="Page-1" Zoom="1.192093" OffsetX="346.714286" OffsetY="769.714286"
DefaultUnitsText="50" DefaultUnitsAngle="81" DefaultUnitsDuration="44" DefaultUnitsPage="73"
Conflict="0" DocumentID="-1092095043">
              <Pages>
                <Page Name="Page-1" />
              </Pages>
              <ShapeInfo Name="_1" DisplayName="Tagged process"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:132,&quot;y&quot
;:276,&quot;width&quot;:222,&quot;height&quot;:75\}\}">
                <ShapeDataItems>
                <ShapeData Name="ID" FormattedValue="1" Value="1" Format="{0:0.####} {1}"
Type="2" LangID="1033" UnitLabel="0" Unit="32" BindingID="1" />
                <ShapeData Name="FN" FormattedValue="Happy" Value="Happy" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="1" />
                <ShapeData Name="LN" FormattedValue="kitty" Value="kitty" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="1" />
                </ShapeDataItems>
```

```xml
                </ShapeInfo>
                <ShapeInfo Name="_9" DisplayName="Tagged process.9"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;::96,&quot;y&quot;
::456,&quot;width&quot;::222,&quot;height&quot;::75\}\}">
                  <ShapeDataItems>
                    <ShapeData Name="ID" FormattedValue="2" Value="2" Format="{0:0.####} {1}"
Type="2" LangID="1033" UnitLabel="0" Unit="32" BindingID="2" />
                    <ShapeData Name="FN" FormattedValue="Silly" Value="Silly" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="2" />
                    <ShapeData Name="LN" FormattedValue="dog" Value="dog" Format="{0}" Type="0"
LangID="1033" UnitLabel="0" Unit="105" BindingID="2" />
                  </ShapeDataItems>
                </ShapeInfo>
                <ShapeInfo Name="_17" DisplayName="Tagged process.17"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;::120,&quot;y&quot
;:648,&quot;width&quot;::222,&quot;height&quot;::75\}\}">
                  <ShapeDataItems>
                    <ShapeData Name="ID" FormattedValue="3" Value="3" Format="{0:0.####} {1}"
Type="2" LangID="1033" UnitLabel="0" Unit="32" BindingID="3" />
                    <ShapeData Name="FN" FormattedValue="Order " Value="Order" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="3" />
                    <ShapeData Name="LN" FormattedValue="Id" Value="Id" Format="{0}" Type="0"
LangID="1033" UnitLabel="0" Unit="105" BindingID="3" />
                  </ShapeDataItems>
                </ShapeInfo>
                <ShapeInfo Name="_25" DisplayName="Tagged process.25"
Layout="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;::480,&quot;y&quot
;:492,&quot;width&quot;::222,&quot;height&quot;::75\}\}">
                  <ShapeDataItems>
                    <ShapeData Name="ID" FormattedValue="4" Value="4" Format="{0:0.####} {1}"
Type="2" LangID="1033" UnitLabel="0" Unit="32" BindingID="4" />
                    <ShapeData Name="FN" FormattedValue="green" Value="green" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="4" />
                    <ShapeData Name="LN" FormattedValue="turtle" Value="turtle" Format="{0}"
Type="0" LangID="1033" UnitLabel="0" Unit="105" BindingID="4" />
                  </ShapeDataItems>
                </ShapeInfo>
              </Page>
          </d4p1:ShapeInfo>
          <d4p1:Xaml>
            <Canvas Height="1056" Name="Workspace" Width="816"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
              <!-- Page  -->
              <Canvas Width="816" Height="1056" Name="D">
                <Canvas.Background>
                  <SolidColorBrush Color="#FFFFFFFF" />
                </Canvas.Background>
                <Canvas
Tag="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;::132,&quot;y&quot;::2
76,&quot;width&quot;::96,&quot;height&quot;::72\}\}" Name="_1" RenderTransform="1, 0, 0, 1,
132, -708">
                  <Canvas Name=" 2">
                    <Canvas Tag="Shdw" RenderTransform="1, 0, 0, 1, 1.92, 1.92">
                      <Path Name="S 2 G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
                        <Path.Stroke>
                          <SolidColorBrush Color="#7FDEDEDE" />
                        </Path.Stroke>
                        <Path.Fill>
                          <SolidColorBrush Color="#7FDEDEDE" />
                        </Path.Fill>
                      </Path>
                    </Canvas>
                    <Path Name=" 2 G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
                      <Path.Stroke>
                        <SolidColorBrush Color="#00404040" />
```

```xml
          </Path.Stroke>
          <Path.Fill>
            <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
              <LinearGradientBrush.GradientStops>
                <GradientStop Color="#00FF00FF" />
                <GradientStop Color="#00FF00FF" Offset="1" />
              </LinearGradientBrush.GradientStops>
            </LinearGradientBrush>
          </Path.Fill>
        </Path>
      </Canvas>
      <Canvas Name="_3" RenderTransform="1, 0, 0, 1, 72, 0">
        <Path Name="_3_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L24 1056 L24 1032 L0 1056 Z">
          <Path.Stroke>
            <SolidColorBrush Color="#FF404040" />
          </Path.Stroke>
          <Path.Fill>
            <SolidColorBrush Color="#FF000000" />
          </Path.Fill>
        </Path>
      </Canvas>
      <Canvas Name="_5">
        <Canvas.RenderTransform>
          <MatrixTransform Matrix="1,0,0,1,102,-16.65076" />
        </Canvas.RenderTransform>
        <Path Name="_5_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.57080078125 L1.92 1038.57080078125 L1.92 1054.08 Z"></Path>
        <Path Name="_5_G1_P1" Data="M1.92 1054.08 L118.08 1054.08"></Path>
        <Path Name="_5_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.57080078125 L1.92 1038.57080078125 L1.92 1054.08 Z">
          <Path.Stroke>
            <SolidColorBrush Opacity="0" Color="#FF404040" />
          </Path.Stroke>
        </Path>
        <Path Name="_5_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
          <Path.Stroke>
            <SolidColorBrush Color="#FF404040" />
          </Path.Stroke>
        </Path>
        <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.201553942620754" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0,
96, 1.92, 1038.608582" TextWrapping="Wrap">
          <Run Name="txt_5_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">LN </Run>
        </TextBlock>
        <Canvas>
        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="LN">
          <Glyphs.Fill>
            <SolidColorBrush Color="#FF000000" />
          </Glyphs.Fill>
        </Glyphs>
        </Canvas>
      <Canvas Name="_6">
        <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.201553942620754" Visibility="Collapsed" Width="1.0386175337222"
RenderTransform="96, 0, 0, 96, 18.372717, 1038.608582" TextWrapping="Wrap">
          <Run Name="txt_6_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">not kitty </Run>
        </TextBlock>
        <Canvas>
          <Canvas.RenderTransform>
```

```
                    <MatrixTransform Matrix="96,0,0,96,18.3744,1038.571" />
                  </Canvas.RenderTransform>
                  <Glyphs UnicodeString="kitty" OriginX="0.825711078964174"
OriginY="0.111110001802444" FontUri="CommonFonts/4C1AABE5-BA71-4F2C-8263-66731F995C4B.odttf"
FontRenderingEmSize="0.111110001802444" Fill="#FF000000" Visibility="Visible" />
                </Canvas>
              </Canvas>
            </Canvas>
            <Canvas Name="_7">
              <Canvas.RenderTransform>
                <MatrixTransform Matrix="1,0,0,1,102,-36" />
              </Canvas.RenderTransform>
              <Path Name="_7_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.57080078125 L1.92 1038.57080078125 L1.92 1054.08 Z"></Path>
              <Path Name="_7_G1_P1" Data="M1.92 1054.08 L118.08 1054.08"></Path>
              <Path Name="_7_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.57080078125 L1.92 1038.57080078125 L1.92 1054.08 Z">
                <Path.Stroke>
                  <SolidColorBrush Opacity="0" Color="#FF404040" />
                </Path.Stroke>
              </Path>
              <Path Name="_7_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                <Path.Stroke>
                  <SolidColorBrush Color="#FF404040" />
                </Path.Stroke>
              </Path>
              <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.201553942620754" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0,
96, 1.92, 1038.608582" TextWrapping="Wrap">
                <Run Name="txt_7_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">FN </Run>
              </TextBlock>
              <Canvas>
                <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="FN">
                  <Glyphs.Fill>
                    <SolidColorBrush Color="#FF000000" />
                  </Glyphs.Fill>
                </Glyphs>
              </Canvas>
            </Canvas>
            <Canvas Name="_8">
              <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.201553942620754" Visibility="Collapsed" Width="1.0342764082339"
RenderTransform="96, 0, 0, 96, 18.789465, 1038.608582" TextWrapping="Wrap">
                <Run Name="txt_8_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">not Happy </Run>
              </TextBlock>
              <Canvas>
                <Canvas.RenderTransform>
                  <MatrixTransform Matrix="96,0,0,96,18.7872,1038.571" />
                </Canvas.RenderTransform>
                <Glyphs UnicodeString="Happy" OriginX="0.787411080680788"
OriginY="0.111110001802444" FontUri="CommonFonts/4C1AABE5-BA71-4F2C-8263-66731F995C4B.odttf"
FontRenderingEmSize="0.111110001802444" Fill="#FF000000" Visibility="Visible" />
              </Canvas>
            </Canvas>
            <Canvas Name="_4">
              <Path Name="_4_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L72 1056 L96 1032 L96 984 L0 984 L0 1056 Z">
                <Path.Stroke>
                  <SolidColorBrush Color="#FF404040" />
                </Path.Stroke>
```

```xml
                              <Path.Fill>
                                <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
                                  <LinearGradientBrush.GradientStops>
                                    <GradientStop Color="#FFFAFAFA" />
                                    <GradientStop Color="#FFFFFFFF" Offset="1" />
                                  </LinearGradientBrush.GradientStops>
                                </LinearGradientBrush>
                              </Path.Fill>
                            </Path>
                          </Canvas>
                        </Canvas>
                        <Canvas
Tag="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:96,&quot;y&quot;:45
6,&quot;width&quot;:96,&quot;height&quot;:72\}\}" Name="_9" RenderTransform="1, 0, 0, 1, 96,
-528">
                          <Canvas Name="_10">
                            <Canvas Tag="Shdw" RenderTransform="1, 0, 0, 1, 1.92, 1.92">
                              <Path Name="S_10_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
                                <Path.Stroke>
                                  <SolidColorBrush Color="#7FDEDEDE" />
                                </Path.Stroke>
                                <Path.Fill>
                                  <SolidColorBrush Color="#7FDEDEDE" />
                                </Path.Fill>
                              </Path>
                            </Canvas>
                            <Path Name="_10_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
                              <Path.Stroke>
                                <SolidColorBrush Color="#00404040" />
                              </Path.Stroke>
                              <Path.Fill>
                                <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
                                  <LinearGradientBrush.GradientStops>
                                    <GradientStop Color="#00FF00FF" />
                                    <GradientStop Color="#00FF00FF" Offset="1" />
                                  </LinearGradientBrush.GradientStops>
                                </LinearGradientBrush>
                              </Path.Fill>
                            </Path>
                          </Canvas>
                          <Canvas Name="_11" RenderTransform="1, 0, 0, 1, 72, 0">
                            <Path Name="_11_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L24 1056 L24 1032 L0 1056 Z">
                              <Path.Stroke>
                                <SolidColorBrush Color="#FF404040" />
                              </Path.Stroke>
                              <Path.Fill>
                                <SolidColorBrush Color="#FF000000" />
                              </Path.Fill>
                            </Path>
                          </Canvas>
                          <Canvas Name="_13">
                            <Canvas.RenderTransform>
                              <MatrixTransform Matrix="1, 0, 0, 1, 102, -16.688582" />
                            </Canvas.RenderTransform>
                            <Path Name="_13_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
                            <Path Name="_13_G1_P1" Data="M1.92 1054.08 L118.08 1054.08" />
                            <Path Name="_13_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
                              <Path.Stroke>
                                <SolidColorBrush Opacity="0" Color="#FF404040" />
                              </Path.Stroke>
```

```
                    </Path>
                    <Path Name="_13_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Color="#FF404040" />
                        </Path.Stroke>
                    </Path>
                    <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
                        <Run Name="txt_13_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">LN </Run>
                    </TextBlock>
                    <Canvas>
                        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="LN">
                          <Glyphs.Fill>
                            <SolidColorBrush Color="#FF000000" />
                          </Glyphs.Fill>
                        </Glyphs>
                    </Canvas>
                  </Canvas>
                  <Canvas Name="_14">
                    <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0386" RenderTransform="96, 0, 0, 96,
18.372717, 1038.608582" TextWrapping="Wrap">
                        <Run Name="txt 14 fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">dog </Run>
                    </TextBlock>
                    <Canvas>
                        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="100.5" OriginY="1049.5" UnicodeString="dog">
                          <Glyphs.Fill>
                            <SolidColorBrush Color="#FF000000" />
                          </Glyphs.Fill>
                        </Glyphs>
                    </Canvas>
                  </Canvas>
                </Canvas>
                <Canvas Name=" 15">
                  <Canvas.RenderTransform>
                    <MatrixTransform Matrix="1, 0, 0, 1, 102, -36" />
                  </Canvas.RenderTransform>
                  <Path Name=" 15 G0 P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
                  <Path Name=" 15 G1 P1" Data="M1.92 1054.08 L118.08 1054.08" />
                  <Path Name="_15_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Opacity="0" Color="#FF404040" />
                        </Path.Stroke>
                    </Path>
                    <Path Name="_15_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Color="#FF404040" />
                        </Path.Stroke>
                    </Path>
                    <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
                        <Run Name="txt_15_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">FN </Run>
                    </TextBlock>
```

```xml
<Canvas>
  <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="FN">
    <Glyphs.Fill>
      <SolidColorBrush Color="#FF000000" />
    </Glyphs.Fill>
  </Glyphs>
</Canvas>
<Canvas Name="_16">
  <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0343" RenderTransform="96, 0, 0, 96,
18.789465, 1038.608582" TextWrapping="Wrap">
    <Run Name="txt_16_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">Silly </Run>
  </TextBlock>
  <Canvas>
    <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="84.559" OriginY="1049.5" UnicodeString="Silly">
      <Glyphs.Fill>
        <SolidColorBrush Color="#FF000000" />
      </Glyphs.Fill>
    </Glyphs>
  </Canvas>
</Canvas>
</Canvas>
<Canvas Name="_12">
  <Path Name="_12_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L72 1056 L96 1032 L96 984 L0 984 L0 1056 Z">
    <Path.Stroke>
      <SolidColorBrush Color="#FF404040" />
    </Path.Stroke>
    <Path.Fill>
      <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
        <LinearGradientBrush.GradientStops>
          <GradientStop Color="#FFFAFAFA" />
          <GradientStop Color="#FFFFFFFF" Offset="1" />
        </LinearGradientBrush.GradientStops>
      </LinearGradientBrush>
    </Path.Fill>
  </Path>
</Canvas>
</Canvas>
<Canvas
Tag="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:120,&quot;y&quot;:6
48,&quot;width&quot;:96,&quot;height&quot;:72\}\}" Name="_17" RenderTransform="1, 0, 0, 1,
120, -336">
  <Canvas Name="_18">
    <Canvas Tag="Shdw" RenderTransform="1, 0, 0, 1, 1.92, 1.92">
      <Path Name="S_18_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
        <Path.Stroke>
          <SolidColorBrush Color="#7FDEDEDE" />
        </Path.Stroke>
        <Path.Fill>
          <SolidColorBrush Color="#7FDEDEDE" />
        </Path.Fill>
      </Path>
    </Canvas>
    <Path Name="_18_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
      <Path.Stroke>
        <SolidColorBrush Color="#00404040" />
      </Path.Stroke>
      <Path.Fill>
        <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
```

```xml
            <LinearGradientBrush.GradientStops>
              <GradientStop Color="#00FF00FF" />
              <GradientStop Color="#00FF00FF" Offset="1" />
            </LinearGradientBrush.GradientStops>
          </LinearGradientBrush>
        </Path.Fill>
      </Path>
    </Canvas>
    <Canvas Name="_19" RenderTransform="1, 0, 0, 1, 72, 0">
      <Path Name="_19_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L24 1056 L24 1032 L0 1056 Z">
        <Path.Stroke>
          <SolidColorBrush Color="#FF404040" />
        </Path.Stroke>
        <Path.Fill>
          <SolidColorBrush Color="#FF000000" />
        </Path.Fill>
      </Path>
    </Canvas>
    <Canvas Name="_21">
      <Canvas.RenderTransform>
        <MatrixTransform Matrix="1, 0, 0, 1, 102, -16.688582" />
      </Canvas.RenderTransform>
      <Path Name="_21_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
      <Path Name="_21_G1_P1" Data="M1.92 1054.08 L118.08 1054.08" />
      <Path Name="_21_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
        <Path.Stroke>
          <SolidColorBrush Opacity="0" Color="#FF404040" />
        </Path.Stroke>
      </Path>
      <Path Name="_21_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
        <Path.Stroke>
          <SolidColorBrush Color="#FF404040" />
        </Path.Stroke>
      </Path>
      <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
        <Run Name="txt 21 fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">LN </Run>
      </TextBlock>
      <Canvas>
        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="LN">
          <Glyphs.Fill>
            <SolidColorBrush Color="#FF000000" />
          </Glyphs.Fill>
        </Glyphs>
      </Canvas>
    </Canvas>
    <Canvas Name=" 22">
      <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0386" RenderTransform="96, 0, 0, 96,
18.372717, 1038.608582" TextWrapping="Wrap">
        <Run Name="txt_22_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">Id </Run>
      </TextBlock>
      <Canvas>
        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="103.67" OriginY="1049.5" UnicodeString="Id">
          <Glyphs.Fill>
            <SolidColorBrush Color="#FF000000" />
```

```
                    </Glyphs.Fill>
                  </Glyphs>
                </Canvas>
              </Canvas>
            </Canvas>
            <Canvas Name=" 23">
              <Canvas.RenderTransform>
                <MatrixTransform Matrix="1, 0, 0, 1, 102, -36" />
              </Canvas.RenderTransform>
              <Path Name="_23_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
              <Path Name="_23_G1_P1" Data="M1.92 1054.08 L118.08 1054.08" />
              <Path Name="_23_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
                <Path.Stroke>
                  <SolidColorBrush Opacity="0" Color="#FF404040" />
                </Path.Stroke>
              </Path>
              <Path Name="_23_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                <Path.Stroke>
                  <SolidColorBrush Color="#FF404040" />
                </Path.Stroke>
              </Path>
              <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
                <Run Name="txt 23 fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">FN </Run>
              </TextBlock>
              <Canvas>
                <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="FN">
                  <Glyphs.Fill>
                    <SolidColorBrush Color="#FF000000" />
                  </Glyphs.Fill>
                </Glyphs>
              </Canvas>
            </Canvas>
            <Canvas Name=" 24">
              <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0343" RenderTransform="96, 0, 0, 96,
18.789465, 1038.608582" TextWrapping="Wrap">
                <Run Name="txt_24_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">Order </Run>
              </TextBlock>
              <Canvas>
                <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="93.242" OriginY="1049.5" UnicodeString="Order">
                  <Glyphs.Fill>
                    <SolidColorBrush Color="#FF000000" />
                  </Glyphs.Fill>
                </Glyphs>
              </Canvas>
            </Canvas>
          </Canvas>
          <Canvas Name="_20">
            <Path Name="_20_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L72 1056 L96 1032 L96 984 L0 984 L0 1056 Z">
              <Path.Stroke>
                <SolidColorBrush Color="#FF404040" />
              </Path.Stroke>
              <Path.Fill>
                <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
                  <LinearGradientBrush.GradientStops>
```

```xml
              <GradientStop Color="#FFFAFAFA" />
              <GradientStop Color="#FFFFFFFF" Offset="1" />
            </LinearGradientBrush.GradientStops>
          </LinearGradientBrush>
        </Path.Fill>
      </Path>
    </Canvas>
  </Canvas>
  <Canvas
Tag="\{&quot;type&quot;:&quot;2D&quot;,&quot;bounds&quot;:\{&quot;x&quot;:480,&quot;y&quot;:4
92,&quot;width&quot;:96,&quot;height&quot;:72\}\}" Name="_25" RenderTransform="1, 0, 0, 1,
480, -492">
    <Canvas Name="_26">
      <Canvas Tag="Shdw" RenderTransform="1, 0, 0, 1, 1.92, 1.92">
        <Path Name="S_26_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
          <Path.Stroke>
            <SolidColorBrush Color="#7FDEDEDE" />
          </Path.Stroke>
          <Path.Fill>
            <SolidColorBrush Color="#7FDEDEDE" />
          </Path.Fill>
        </Path>
      </Canvas>
      <Path Name="_26_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M72 1056
L96 1056 L96 1032 L96 984 L0 984 L0 1056 L72 1056 Z">
        <Path.Stroke>
          <SolidColorBrush Color="#00404040" />
        </Path.Stroke>
        <Path.Fill>
          <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
            <LinearGradientBrush.GradientStops>
              <GradientStop Color="#00FF00FF" />
              <GradientStop Color="#00FF00FF" Offset="1" />
            </LinearGradientBrush.GradientStops>
          </LinearGradientBrush>
        </Path.Fill>
      </Path>
    </Canvas>
    <Canvas Name="_27" RenderTransform="1, 0, 0, 1, 72, 0">
      <Path Name="_27_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L24 1056 L24 1032 L0 1056 Z">
        <Path.Stroke>
          <SolidColorBrush Color="#FF404040" />
        </Path.Stroke>
        <Path.Fill>
          <SolidColorBrush Color="#FF000000" />
        </Path.Fill>
      </Path>
    </Canvas>
    <Canvas Name="_29">
      <Canvas.RenderTransform>
        <MatrixTransform Matrix="1, 0, 0, 1, 102, -16.688582" />
      </Canvas.RenderTransform>
      <Path Name="_29_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
      <Path Name="_29_G1_P1" Data="M1.92 1054.08 L118.08 1054.08" />
      <Path Name="_29_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
        <Path.Stroke>
          <SolidColorBrush Opacity="0" Color="#FF404040" />
        </Path.Stroke>
      </Path>
```

```xml
<Path Name="_29_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Color="#FF404040" />
                        </Path.Stroke>
                      </Path>
                      <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
                        <Run Name="txt_29_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">LN </Run>
                      </TextBlock>
                      <Canvas>
                        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="LN">
                          <Glyphs.Fill>
                            <SolidColorBrush Color="#FF000000" />
                          </Glyphs.Fill>
                        </Glyphs>
                      </Canvas>
                      <Canvas Name=" 30">
                        <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0386" RenderTransform="96, 0, 0, 96,
18.372717, 1038.608582" TextWrapping="Wrap">
                          <Run Name="txt_30_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">turtle </Run>
                        </TextBlock>
                        <Canvas>
                          <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="92.523" OriginY="1049.5" UnicodeString="turtle">
                            <Glyphs.Fill>
                              <SolidColorBrush Color="#FF000000" />
                            </Glyphs.Fill>
                          </Glyphs>
                        </Canvas>
                      </Canvas>
                    </Canvas>
                    <Canvas Name="_31">
                      <Canvas.RenderTransform>
                        <MatrixTransform Matrix="1, 0, 0, 1, 102, -36" />
                      </Canvas.RenderTransform>
                      <Path Name="_31_G0_P0" Data="M1.92 1054.08 L118.08 1054.08 L118.08
1038.61 L1.92 1038.61 L1.92 1054.08 Z" />
                      <Path Name="_31_G1_P1" Data="M1.92 1054.08 L118.08 1054.08" />
                      <Path Name="_31_G0_P0_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08 L118.08 1038.61 L1.92 1038.61 L1.92 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Opacity="0" Color="#FF404040" />
                        </Path.Stroke>
                      </Path>
                      <Path Name="_31_G1_P1_line" StrokeThickness="0.5" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="M1.92
1054.08 L118.08 1054.08">
                        <Path.Stroke>
                          <SolidColorBrush Color="#FF404040" />
                        </Path.Stroke>
                      </Path>
                      <TextBlock
Tag="Left;Top;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.013888888888889"
Height="0.16116" Visibility="Collapsed" Width="0.75625" RenderTransform="96, 0, 0, 96, 1.92,
1038.608582" TextWrapping="Wrap">
                        <Run Name="txt_31_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">FN </Run>
                      </TextBlock>
                      <Canvas>
```

```
                    <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="3.2534" OriginY="1049.5" UnicodeString="FN">
                        <Glyphs.Fill>
                          <SolidColorBrush Color="#FF000000" />
                        </Glyphs.Fill>
                    </Glyphs>
                  </Canvas>
                  <Canvas Name="_32">
                    <TextBlock
Tag="Right;Center;;1.2;0.013888888888889;0.013888888888889;0.013888888888889;0.01388888888888
9" Height="0.16116" Visibility="Collapsed" Width="1.0343" RenderTransform="96, 0, 0, 96,
18.789465, 1038.608582" TextWrapping="Wrap">
                        <Run Name="txt_32_fld0" FontFamily="Calibri" FontSize="0.11111"
Foreground="#FF000000">green </Run>
                    </TextBlock>
                    <Canvas>
                        <Glyphs FontRenderingEmSize="10.667" FontUri="CommonFonts/4C1AABE5-
BA71-4F2C-8263-66731F995C4B.odttf" OriginX="91.789" OriginY="1049.5" UnicodeString="green">
                          <Glyphs.Fill>
                            <SolidColorBrush Color="#FF000000" />
                          </Glyphs.Fill>
                        </Glyphs>
                    </Canvas>
                  </Canvas>
                </Canvas>
                <Canvas Name="_28">
                  <Path Name="_28_G0" StrokeThickness="0.33333" StrokeDashCap="Round"
StrokeEndLineCap="Round" StrokeLineJoin="Round" StrokeStartLineCap="Round" Data="F0 M0 1056
L72 1056 L96 1032 L96 984 L0 984 L0 1056 Z">
                    <Path.Stroke>
                      <SolidColorBrush Color="#FF404040" />
                    </Path.Stroke>
                    <Path.Fill>
                      <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
                        <LinearGradientBrush.GradientStops>
                          <GradientStop Color="#FFFAFAFA" />
                          <GradientStop Color="#FFFFFFFF" Offset="1" />
                        </LinearGradientBrush.GradientStops>
                      </LinearGradientBrush>
                    </Path.Fill>
                  </Path>
                </Canvas>
              </Canvas>
            </Canvas>
          </Canvas>
        </d4p1:Xaml>
      </d4p1:OutputPage>

    </GetVectorDiagramResult>
  </GetVectorDiagramResponse>
  </s:Body>
</s:Envelope>
```

4.4 GetRasterPage

In this scenario, the protocol client calls the **GetRasterPage** method (section 3.1.4.3) on a protocol server named "ExampleServer". The protocol server responds with a **GetRasterPageResponse** element (section 3.1.4.3.2.2). To request the raster formatted drawing page (section 1.3.2) of a VSDX web drawing (section 1.3.1) , the protocol client constructs the following WSDL message.

```
<s:Body>
  <GetRasterPage xmlns="http://tempuri.org/">
    <rasterPageRequestContract
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```
        <b:DataModeName>DynamicSafe</b:DataModeName>
        <b:DataSources i:nil="true"></b:DataSources>
        <b:EntityTag>
          <b:DataModeName>Static</b:DataModeName>
          <b:FileETag i:nil="true"></b:FileETag>
          <b:OutputETag i:nil="true"></b:OutputETag>
          <b:SequenceId i:nil="true"></b:SequenceId>
        </b:EntityTag>
        <b:FileUrl> http://ExampleServer/BasicDiagram.vsdx</b:FileUrl>
        <b:PageId>-1</b:PageId>
        <b:RefreshCookie i:nil="true"></b:RefreshCookie>
      </rasterPageRequestContract>
    </GetRasterPage>
  </s:Body>
```

The protocol server responds with the following:

```
  <s:Body>
    <GetRasterPageItemResponse xmlns="http://tempuri.org/">
      <GetRasterPageItemResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:EntityTag>
          <d4p1:DataModeName>Static</d4p1:DataModeName>
          <d4p1:FileETag>{BFC23F51-DFCB-4C68-A3C0-E2422785F256},1</d4p1:FileETag>
        </d4p1:EntityTag>
        <d4p1:ItemData></d4p1:ItemData>
        <d4p1:ResponseCode>OK</d4p1:ResponseCode>
      </GetRasterPageItemResult>
    </GetRasterPageItemResponse>
  </s:Body>
```

4.5   GetRasterPageItem

In this scenario, the protocol client calls the **GetRasterPageItem** method (section 3.1.4.4) on a protocol server named "ExampleServer". The protocol server responds with a **GetRasterPageItemResponse** element (section 3.1.4.4.2.2). The **GetRasterPageItem** operation retrieves an item, as specified in section 3.1.1, of a drawing page (section 1.3.2) of a **VSDX** web drawing (section 1.3.1) rendered in raster format.

To request the this item, the protocol client constructs the following WSDL message:

```
  <s:Envelope>
    <s:Body>
      <GetRasterPageItem xmlns="http://tempuri.org/">
        <rasterPageItemRequestContract
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <b:EntityTag>
            <b:DataModeName>Static</b:DataModeName>
            <b:FileETag i:nil="true"></b:FileETag>
          </b:EntityTag>
          <b:FileUrl> http://ExampleServer/BasicDiagram.vsdx</b:FileUrl>
          <b:ItemName>ShapeInfo</b:ItemName>
          <b:SequenceId>363696e759d24d3aa58818e65c33b6dd</b:SequenceId>
        </rasterPageItemRequestContract>
      </GetRasterPageItem>
    </s:Body>
  </s:Envelope>
```

The protocol server responds with the following for Drawing with Static **DataModeName** (section 3.1.4.3.3.1):

```
<s:Envelope>
  <s:Body>
    <GetRasterPageItemResponse xmlns="http://tempuri.org/">
      <GetRasterPageItemResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:EntityTag>
          <d4p1:DataModeName>Static</d4p1:DataModeName>
          <d4p1:FileETag>{BFC23F51-DFCB-4C68-A3C0-E2422785F256},1</d4p1:FileETag>
        </d4p1:EntityTag>
        <d4p1:ItemData></d4p1:ItemData>
        <d4p1:ResponseCode>OK</d4p1:ResponseCode>
      </GetRasterPageItemResult>
    </GetRasterPageItemResponse>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following for Drawing with Dynamic **DataModeName** (section 3.1.4.3.3.1):

```
<s:Envelope>
  <s:Body>
    <GetRasterPageItemResponse xmlns="http://tempuri.org/">
      <GetRasterPageItemResult
xmlns:d4p1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsSer
ver" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <d4p1:EntityTag>
          <d4p1:DataModeName>Dynamic</d4p1:DataModeName>
          <d4p1:FileETag>{E15A7F56-A0BC-44FB-9EE1-5B3DE24E3176},1</d4p1:FileETag>
        </d4p1:EntityTag>
        <d4p1:ItemData></d4p1:ItemData>
        <d4p1:ResponseCode>OK</d4p1:ResponseCode>
      </GetRasterPageItemResult>
    </GetRasterPageItemResponse>
  </s:Body>
</s:Envelope>
```

In addition, for both Static & Dynamic DataModeName (for details on DataModeName see section 3.1.4.3.3.1), the protocol server responds with the following for returned image:

```
<s:Envelope>
  <s:Body>
    <GetRasterPageItemResponse xmlns="http://tempuri.org/">
      <GetRasterPageItemResult
xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServer
" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <b:EntityTag>
          <b:DataModeName>Dynamic</b:DataModeName>
          <b:FileETag>{E15A7F56-A0BC-44FB-9EE1-5B3DE24E3176},1</b:FileETag>
        </b:EntityTag>
```

```
<b:ItemData>iVBORw0KGgoAAAANSUhEUgAAAnkAAAHpCAYAAAGJgBrpAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8Y
QUAAAAJcEhZcwAADsMAAA7DAcdvqGQAACBxSURBVHhe7d3tbxVXegDwgBaSQEBWAiUgsoSEYpIQkpAFId7syKmREisB6i
pkq4gVQu6qzW5T0ipbuhASJVuR126OaVowGJZ2s0rlVJHSF6kfqNWu2mAbw4da6vf+ARvbGOxg4+4+l5kkO/KcbBJDPff9/6
u9GjuvNyZc34znDw+M3yyyy0+BAgQIECAAgECBAgQIECBAgQIECBAAAIGyF2hqavp8oqipqbm77AFu
tIIjIyPZRNHHa2vrWje6/7H8/NDSUTTRamxra5stUGp9/98MMPPX1y7du1flT3SZBW8dOlSNlEEEYIYILKrhhcVDjdjX15dNF
AG4ZcuWlnnnz5r22bNmy964FtWjRomurxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsrxjUnp6ebKayunsQKL7AZKoqFb90JV6CyUZUK/GiF794k42Mln
p0thWL1dXV/+g+DUoYgkmG0ktjaD2v0ZRm+TkTDaDaKWvxs7ty5f7NmzZp9x6V1ddUU8x6V56MlGUCvvNEisVAQIECBAgQIAA
AQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBC4jkA2xU9ra+tbcAnckMDw8HA2lchdhfASXLmxJBdg7b7b968N2
```

bNmvX7e/bsqU7zWUdHx+r333+/Yf369UsbGhreuaFC+nF5CgwODmZTidzFt3Xr1p91d3ffUZ46apVXgf7+/mwq4T+7eT0
tlbHz3t7ebCoRF1/634R+Ul9f/9mNRmVIq+VXBNL/IvWVKcZf4CRAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAg
AABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECiUwM6dOweampo+n0oUqoy
OU4YCJ0+ezKb6KUMOVSqUQFx4IyMjU4pCldFxylDgxIkT2fDw8JRilCObP3/+6fr6+perq6t/vHr16sNpecPatPatWtfT9M5
r7766sbsHH3/8J/+3t7XeXXIZ8qTVWgpaUlGxoamlKMPWZnZ2/ft1yzDnpkZZZz/Z2dHSsTtttntbW1cYH6VJrA8ePHs8HBwSlFW
K1fv37+hg0b/rrS3NNT3BgWOHTuWXXbp0aUoRnh04tVrRaNxxQ1NTXfu8FA+pU0E2hubs76+/unFNOtrspbQgjHJjx7+n+jvr6ph
RRjcbGGxhUPPPfTQ32/fvv35qqqf9+0adPfhxu4aUocfM12+5SJw5v6qw3TUF4sLr+eMdZUnSNQ9t1UADR60vwn45aNb1RvqqqmodT
r6emZUnzNQ9iMwFcFzpw5s7SSrq+uVqQRPAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIEC
BAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAg
AABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgTKS+DkyZNZYT0kbGh5r0dVlZei2hAgMG0EotEbGBGRkpay3loaAEyk/qxIkT2fDw8JTilGObP3/+6fr6+perq6t/vHr16sNpecPatPatWtfT9M5
pyYaGhgoaxu98+fPV729/PV6w729/eGNb5d3zd/z5szS+F1s29jYuCLUUB70vwn6w{6mLbcUB70vwn45aNb1RvqqqmdEBSQwgLR6F2+fLmgUeg/b6NxTA3Aots7Oztpz5Oztpz5849m75ylju9U/KUgiXUBQECFSBv9wOjR7MKFKFwWWNsY3e4swLWLv71ly5
Y/SNS7Ulztf5s7d+5zaXooxbaYzp/3wvXWsJcJwgqvvV5tf+XaXooxbaYzp/3w+Phz3z4gGfl49T8uKLLz6Sj/3aJwcECJSgvQjV5f19X9XBxzHw1qoCV4yhWJJQGULHdlyJOvt7S1oFoFDdlyJOvt7S1oGegaAABAgQIECBA
gAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIE
CBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgAA
QIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgTKS+DkyZPZM888M6Uoc/c0NTU9n0oUqoy
mZucupT+/1mI8/cRsaGlbFg8WjDWC+Xhu7eqxinQPHJZUCgQAQAAIl0ODd0tHRsTyVKFQgWNsY3e4swLWLv71ly5
wWHCPe3CgQucMQIFDdwdTK2M65RoSQYdHKifSKfWMFlb51QECFSBvzdwOjR7MKHGIFYUZDwwGJ4y0Qj7/+fvX5sePI43lEV9vJ9wLGPQfuSo0wc0OEUTRsTqekUv9Zz9vjbjbYlo+GIaz8blnpmLB4Pz9jeeaa+vjXTpAG07Pxq+qJCgAA1AQ9j8k9DDfF6IFKbe4PwpPzeMIYkqB4tdS8Pjz7zOvvr4Pdx3WrFfLlvF2WbT4E4NJsk9KxfXXPWWprqvTw29MIYm{rsaGlbFg8WjDWC+Xhu7eqxinQPNzeXB8XtwGIFFMIFAsgWjWnt7s0LG2Zyrq/uP9x8nEECBAgQIECBA
gAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgAA
QIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgTKS+DkyZPZ
QIAAAQAAAgTKS+BkyZPZM888U1zbzz33XDY0NLTYg8ViWWtra9ba2pq1tbVl7e3tWUdHR9bZ2Zl1dXVl3d3dWU9P
gQIAAAQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIE
CBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQI
QIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAAB
AgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgQIECBAgAABAgTKS+DkyZPZ

CBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAA
QIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQIAAAQIECBAgQI
AAAQIECBAgQIAAAQIE8iNQU1NTWwGxKj969kqAAAECBAgQKEGBlNwdOnnyZFbun507d37x8ccf//T8+fP3luBpUCQCBAg
QIECAwM0VqKQkr7W19a329vZHb66gvVREgQIAAAQIESlAgl+SNjIxk5k5RzRkkyfJK8ELUJEIECBAgACB/AhI8vLjaq8ECBAg
QIAAgaIKRJJJ34sSJ7MqVK2K2UdevKKepk5OAECBAgQIFBoAUneLbtWrly5fPPXq1f/4zDPP7Ev+WYdpdo+dhz+j8J7Fs7ty5z
43OHxpdvy1Nf50i5netX79+6TvvvNP0xBNP/MmqqVaueHrOv2E9/ioUpNo88u/2Wa/l+KP47fpe3jmLGvN1J8mqLrzjvv+/G
GsS88R3l3o68LxCBAgQIAgWkukEvyhoeHs3KKOSrydjU2Nq546qmnTtTX10filT322GP/0NXXV9VxVVVXXXzjBkzLm/fvv3
5urq6N+bNm/dbcVE/J3r3r+dPXt2wcaNG//3ruAOG/c3NDQkQSRWsldv8RCeL8+fNP19bWvh/rvCwyzf+RKKT4BAgQIECiiQCCOR5
LS9t2dQUFnjjj07Jnzxoq2t7rFdP/bcklDTHPvmVm61eevv7Vu53rdP/bckdTHPvmVm61eenfFr4+uQ09epV/p6k+AAAECBCpMIJK8
8o0ePhcxuXICjruN4zeem0H7rOqY88x8mIw44j4GMWTJn6eT4Um6UsyZRpdpSvhySG6x5GhVbUQkQIECAAIEJBSR5V8t9eSy6qrq3/+/
ePHiU/H9lltv/ZfRcfNi4OLV96at+351E2Ti0zuJ2eE1Um6UsyZRpdSvHySG6x5GhVbUQkQIEAAAAAgBgBMSRAgBgCrutAkece
LW31urLn4Pp0uPm8UT6ezpawECBAgQOA6ApHkHTlyJOvt7S3S3ruFaFl+r+aIrayeKFF17YeeDAgX379+/ckvnh6y73vZl
sr7KPxwCBAgQIEBgmgtUcpIXp87btdd+s99jbtdP8H7biEyBAgACBXJLX09OTlXN4u9a1ToAAAQIECFSUgCSvok63hyIgQ
IAAAQKVIpCSvO/V1tZm5R51dXXDn3766U/i5YhKObfqSYYAAAQIECBC45cyZM0tzb1eW6zTq6FQTIEAAAECBAgQIECAAA
ECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQ
IECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECB
AgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAA
ECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIEC
AAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQ
IECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECB
AgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAA
ECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIEC
AAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQ
IECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECB
AgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAA
ECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIEC
AAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQIECAAAECBAgQ
IECAAAECBAgQIECAAAECBAgUSOD/ATdbUn0Nne9uAAAAAElFTkSuQmCC</b:ItemData>
        <b:ResponseCode>OK</b:ResponseCode>
      </GetRasterPageItemResult>
    </GetRasterPageItemResponse>
  </s:Body>
</s:Envelope>
```

## 4.6 SaveComments

In this example, the protocol client calls the **SaveComments** method (section 3.1.4.5) on a protocol server named "ExampleServer". The protocol server responds with a **SaveCommentsResponse** element (section 3.1.4.5.2.2). To request the raster-formatted drawing page (section 1.3.2) of a **VSDX** web drawing (section 1.3.1), the protocol client constructs the following WSDL message:

```
<s:Envelope>
  <s:Body>
    <SaveComments xmlns="http://tempuri.org/">
      <fileUrl> http://ExampleServer/BasicDiagram.vsdx</fileUrl>
      <commentData>
        <Comment Author='Name of Author' ResolutionID='&lt;resolutionId
provider=&quot;AD&quot;&gt;&lt;localId sid=&quot;s-1-5-21-2127521184-1604012920-1887927527-
2413478&quot;/&gt;&lt;/resolutionId&gt;' PageID='0' ShapeID='-1'
Action='Add'>SampleCommentText</Comment>
      </commentData>
    </SaveComments>
  </s:Body>
```

```
        </s:Envelope>
```

The protocol server responds with the following:

```
        <s:Envelope>
          <s:Body>
            <SaveCommentsResponse xmlns="http://tempuri.org/">
              <SaveCommentsResult>
                <Comment Action="Add" CommentID="0" Author="Name of Author" ShapeID="-1"
        CreationDate="2012-08-28T22:21:53" ModifiedDate="2012-08-28T22:21:53"
        Done="false">SampleCommentText</Comment>
              </SaveCommentsResult>
            </SaveCommentsResponse>
          </s:Body>
        </s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns2="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns3="http://tempuri.org/Imports"
targetNamespace="http://tempuri.org/Imports">
      <xs:import namespace="http://tempuri.org/"/>
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
      <xs:import/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IVisioGraphicsService">
    <wsdl:operation name="GetRasterDiagram">
      <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagram"
message="tns2:IVisioGraphicsService_GetRasterDiagram_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagramResponse"
message="tns2:IVisioGraphicsService_GetRasterDiagram_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterDiagramVisioGraphicsServiceFau
ltFault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetRasterDiagram_VisioGraphicsServiceFaultFault_FaultMess
age"/>
    </wsdl:operation>
    <wsdl:operation name="GetVectorDiagram">
      <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagram"
message="tns2:IVisioGraphicsService_GetVectorDiagram_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagramResponse"
message="tns2:IVisioGraphicsService_GetVectorDiagram_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetVectorDiagramVisioGraphicsServiceFau
ltFault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetVectorDiagram_VisioGraphicsServiceFaultFault_FaultMess
age"/>
    </wsdl:operation>
    <wsdl:operation name="GetRasterPage">
      <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPage"
message="tns2:IVisioGraphicsService_GetRasterPage_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageResponse"
message="tns2:IVisioGraphicsService_GetRasterPage_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageVisioGraphicsServiceFaultF
ault" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_GetRasterPage_VisioGraphicsServiceFaultFault_FaultMessage
"/>
    </wsdl:operation>
    <wsdl:operation name="GetRasterPageItem">
      <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItem"
message="tns2:IVisioGraphicsService_GetRasterPageItem_InputMessage"/>
      <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItemResponse"
message="tns2:IVisioGraphicsService_GetRasterPageItem_OutputMessage"/>
      <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/GetRasterPageItemVisioGraphicsServiceFa
ultFault" name="VisioGraphicsServiceFaultFault"
```

```
message="tns2:IVisioGraphicsService_GetRasterPageItem_VisioGraphicsServiceFaultFault_FaultMes
sage"/>
      </wsdl:operation>
      <wsdl:operation name="SaveComments">
        <wsdl:input wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveComments"
message="tns2:IVisioGraphicsService_SaveComments_InputMessage"/>
        <wsdl:output
wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveCommentsResponse"
message="tns2:IVisioGraphicsService_SaveComments_OutputMessage"/>
        <wsdl:fault
wsaw:Action="http://tempuri.org/IVisioGraphicsService/SaveCommentsVisioGraphicsServiceFaultFa
ult" name="VisioGraphicsServiceFaultFault"
message="tns2:IVisioGraphicsService_SaveComments_VisioGraphicsServiceFaultFault_FaultMessage"
/>
      </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding_IVisioGraphicsService"
type="tns2:IVisioGraphicsService">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="GetRasterDiagram">
        <soap:operation soapAction="http://tempuri.org/IVisioGraphicsService/GetRasterDiagram"
style="document"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="VisioGraphicsServiceFaultFault">
          <soap:fault use="literal" name="VisioGraphicsServiceFaultFault" namespace=""/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="GetVectorDiagram">
        <soap:operation soapAction="http://tempuri.org/IVisioGraphicsService/GetVectorDiagram"
style="document"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="VisioGraphicsServiceFaultFault">
          <soap:fault use="literal" name="VisioGraphicsServiceFaultFault" namespace=""/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="GetRasterPage">
        <soap:operation soapAction="http://tempuri.org/IVisioGraphicsService/GetRasterPage"
style="document"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="VisioGraphicsServiceFaultFault">
          <soap:fault use="literal" name="VisioGraphicsServiceFaultFault" namespace=""/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="GetRasterPageItem">
        <soap:operation soapAction="http://tempuri.org/IVisioGraphicsService/GetRasterPageItem"
style="document"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="VisioGraphicsServiceFaultFault">
          <soap:fault use="literal" name="VisioGraphicsServiceFaultFault" namespace=""/>
        </wsdl:fault>
```

```
      </wsdl:operation>
      <wsdl:operation name="SaveComments">
        <soap:operation soapAction="http://tempuri.org/IVisioGraphicsService/SaveComments"
style="document"/>
        <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="VisioGraphicsServiceFaultFault">
          <soap:fault use="literal" name="VisioGraphicsServiceFaultFault" namespace=""/>
        </wsdl:fault>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:message name="IVisioGraphicsService_GetRasterDiagram_InputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterDiagram"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetRasterDiagram_OutputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterDiagramResponse"/>
    </wsdl:message>
    <wsdl:message
name="IVisioGraphicsService_GetRasterDiagram_VisioGraphicsServiceFaultFault_FaultMessage">
      <wsdl:part
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" name="detail" element="tns:VisioGraphicsServiceFault"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetRasterPage_InputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterPage"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetRasterPage_OutputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterPageResponse"/>
    </wsdl:message>
    <wsdl:message
name="IVisioGraphicsService_GetRasterPage_VisioGraphicsServiceFaultFault_FaultMessage">
      <wsdl:part
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" name="detail" element="tns:VisioGraphicsServiceFault"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetRasterPageItem_InputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterPageItem"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetRasterPageItem_OutputMessage">
      <wsdl:part name="parameters" element="tns2:GetRasterPageItemResponse"/>
    </wsdl:message>
    <wsdl:message
name="IVisioGraphicsService_GetRasterPageItem_VisioGraphicsServiceFaultFault_FaultMessage">
      <wsdl:part
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" name="detail" element="tns:VisioGraphicsServiceFault"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetVectorDiagram_InputMessage">
      <wsdl:part name="parameters" element="tns2:GetVectorDiagram"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_GetVectorDiagram_OutputMessage">
      <wsdl:part name="parameters" element="tns2:GetVectorDiagramResponse"/>
    </wsdl:message>
    <wsdl:message
name="IVisioGraphicsService_GetVectorDiagram_VisioGraphicsServiceFaultFault_FaultMessage">
      <wsdl:part
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" name="detail" element="tns:VisioGraphicsServiceFault"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_SaveComments_InputMessage">
      <wsdl:part name="parameters" element="tns2:SaveComments"/>
    </wsdl:message>
    <wsdl:message name="IVisioGraphicsService_SaveComments_OutputMessage">
      <wsdl:part name="parameters" element="tns2:SaveCommentsResponse"/>
    </wsdl:message>
```

```
    <wsdl:message
name="IVisioGraphicsService_SaveComments_VisioGraphicsServiceFaultFault_FaultMessage">
    <wsdl:part
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" name="detail" element="tns:VisioGraphicsServiceFault"/>
    </wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

Schema name	Prefix	Section
http://schemas.datacontract.org/2004/07 /Microsoft.Office.Visio.Server.GraphicsServer	tns	7.1
http://schemas.microsoft.com/2003/10/Serialization/Arrays	q1	7.2
http://schemas.microsoft.com/2003/10/Serialization/	tns1	7.3
http://tempuri.org/	tns2	7.4

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

7.1 http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.Graphic sServer Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.Graphi
csServer" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="RasterDiagramRequestContract">
    <xs:complexContent mixed="false">
      <xs:extension base="tns:BaseRequestContract">
        <xs:sequence>
          <xs:element minOccurs="0" name="Resolution" type="xs:int"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="RasterDiagramRequestContract" nillable="true"
type="tns:RasterDiagramRequestContract"/>
  <xs:complexType name="BaseRequestContract">
    <xs:sequence>
      <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
      <xs:element minOccurs="0" name="DisableRefresh" type="xs:boolean"/>
      <xs:element minOccurs="0" name="FileUri" nillable="true" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="PageNumber" type="xs:int"/>
      <xs:element minOccurs="0" name="TimeLastModified" type="xs:dateTime"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="BaseRequestContract" nillable="true" type="tns:BaseRequestContract"/>
  <xs:complexType name="ArrayOfAddonDataSource">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="AddonDataSource" nillable="true"
type="tns:AddonDataSource"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfAddonDataSource" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
  <xs:complexType name="AddonDataSource">
    <xs:sequence>
      <xs:element minOccurs="0" name="ConnectionString" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Data" nillable="true">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="xs:schema"/>
            <xs:any/>
          </xs:sequence>
```

```
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="Id" type="xs:int"/>
        <xs:element minOccurs="0" name="QueryString" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="AddonDataSource" nillable="true" type="tns:AddonDataSource"/>
    <xs:complexType name="RasterDiagram">
      <xs:complexContent mixed="false">
        <xs:extension base="tns:DiagramBase">
          <xs:sequence>
            <xs:element minOccurs="0" name="AreaMaps" nillable="true" type="xs:string"/>
            <xs:element minOccurs="0" name="Image" nillable="true" type="xs:base64Binary"/>
            <xs:element minOccurs="0" name="ShapeInfo" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="RasterDiagram" nillable="true" type="tns:RasterDiagram"/>
    <xs:complexType name="DiagramBase">
      <xs:sequence>
        <xs:element minOccurs="0" name="CacheAge" type="xs:int"/>
        <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="DiagramBase" nillable="true" type="tns:DiagramBase"/>
    <xs:complexType name="VisioGraphicsServiceFault">
      <xs:sequence>
        <xs:element minOccurs="0" name="Details" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="Error" type="tns:FaultCode"/>
        <xs:element minOccurs="0" name="Flags" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="VisioGraphicsServiceFault" nillable="true"
type="tns:VisioGraphicsServiceFault"/>
    <xs:simpleType name="FaultCode">
      <xs:restriction base="xs:string">
        <xs:enumeration value="DataConnectivityError"/>
        <xs:enumeration value="DataSourceConnectionError"/>
        <xs:enumeration value="DataSourcePermissionError"/>
        <xs:enumeration value="EmptyRasterDiagramRequest"/>
        <xs:enumeration value="EmptyVectorDiagramRequest"/>
        <xs:enumeration value="OfficeDataConnectionFileNotFound"/>
        <xs:enumeration value="OfficeDataConnectionFileRetrievalError"/>
        <xs:enumeration value="OfficeDataConnectionFileAccessDenied"/>
        <xs:enumeration value="IncompatibleDomain"/>
        <xs:enumeration value="WssViewAccessDenied"/>
        <xs:enumeration value="WssListAccessDenied"/>
        <xs:enumeration value="InvalidPageNumber"/>
        <xs:enumeration value="InvalidResolution"/>
        <xs:enumeration value="InvalidFileUrl"/>
        <xs:enumeration value="RasterizationError"/>
        <xs:enumeration value="InvalidFileExtension"/>
        <xs:enumeration value="EmptyServerData"/>
        <xs:enumeration value="OpenServerFileFailed"/>
        <xs:enumeration value="NoReadPermissionToFile"/>
        <xs:enumeration value="KeyNotFoundFromCollection"/>
        <xs:enumeration value="ExceededMaxDiagramSize"/>
        <xs:enumeration value="IncompatibleVersion"/>
        <xs:enumeration value="DataProviderNotTrusted"/>
        <xs:enumeration value="DataBindingConflict"/>
        <xs:enumeration value="UnableToGetExternalDataSourceCredentials"/>
        <xs:enumeration value="UnattendedAccountTypeMismatch"/>
        <xs:enumeration value="RecalcTimedOut"/>
        <xs:enumeration value="LicenseExpired"/>
        <xs:enumeration value="InvalidZoomLevel"/>
        <xs:enumeration value="VSDXFileLoadGenericError"/>
        <xs:enumeration value="VSDXFileLoadPermissionError"/>
```

```xml
        <xs:enumeration value="VSDXFileSizeError"/>
        <xs:enumeration value="WebPartConfigError"/>
        <xs:enumeration value="EmptyRasterPageRequest"/>
        <xs:enumeration value="EmptyRasterPageItemRequest"/>
        <xs:enumeration value="VDWNotSupportExternalList"/>
        <xs:enumeration value="ExternalListConfigurationNotSupported"/>
        <xs:enumeration value="IncompatibleDomain_15"/>
        <xs:enumeration value="FileNotLockedOnServer"/>
        <xs:enumeration value="FileAlreadyCheckedOutOnServer"/>
        <xs:enumeration value="FileAlreadyLockedOnServer"/>
        <xs:enumeration value="DocumentCheckoutRequired"/>
        <xs:enumeration value="ExcelFileNotInSharePoint"/>
        <xs:enumeration value="PageDrawingAreaSizeUnsupported"/>
        <xs:enumeration value="PageImageSizeUnsupported"/>
        <xs:enumeration value="PageFailToRaster"/>
        <xs:enumeration value="CannotRetrieveExternalListData"/>
      </xs:restriction>
  </xs:simpleType>
  <xs:element name="FaultCode" nillable="true" type="tns:FaultCode"/>
  <xs:complexType name="VectorDiagramRequestContract">
    <xs:complexContent mixed="false">
      <xs:extension base="tns:BaseRequestContract">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="VectorDiagramRequestContract" nillable="true"
type="tns:VectorDiagramRequestContract"/>
  <xs:complexType name="VectorDiagram">
    <xs:complexContent mixed="false">
      <xs:extension base="tns:DiagramBase">
        <xs:sequence>
          <xs:element minOccurs="0" name="OutputPage" nillable="true"
type="tns:RenderedPage"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="VectorDiagram" nillable="true" type="tns:VectorDiagram"/>
  <xs:complexType name="RenderedPage">
    <xs:sequence>
      <xs:element minOccurs="0" name="Resources" nillable="true"
type="tns:ArrayOfDiagramResource"/>
      <xs:element minOccurs="0" name="ShapeInfo" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Xaml" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="RenderedPage" nillable="true" type="tns:RenderedPage"/>
  <xs:complexType name="ArrayOfDiagramResource">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="DiagramResource" nillable="true"
type="tns:DiagramResource"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfDiagramResource" nillable="true"
type="tns:ArrayOfDiagramResource"/>
  <xs:complexType name="DiagramResource">
    <xs:sequence>
      <xs:element minOccurs="0" name="ByteData" nillable="true" type="xs:base64Binary"/>
      <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="DiagramResource" nillable="true" type="tns:DiagramResource"/>
  <xs:complexType name="RasterPageRequest">
    <xs:sequence>
      <xs:element minOccurs="0" name="DataModeName" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
      <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageTag"/>
```

```xml
        <xs:element minOccurs="0" name="FileUrl" nillable="true" type="xs:anyURI"/>
        <xs:element minOccurs="0" name="PageId" type="xs:long"/>
        <xs:element minOccurs="0" name="RefreshCookie" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageRequest" nillable="true" type="tns:RasterPageRequest"/>
    <xs:complexType name="RasterPageTag">
      <xs:sequence>
        <xs:element minOccurs="0" name="DataModeName" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="FileETag" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="OutputETag" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="SequenceId" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageTag" nillable="true" type="tns:RasterPageTag"/>
    <xs:complexType name="RasterPageResponse">
      <xs:sequence>
        <xs:element minOccurs="0" name="DataSources" nillable="true"
type="tns:ArrayOfAddonDataSource"/>
        <xs:element minOccurs="0" name="EntityTag" nillable="true" type="tns:RasterPageTag"/>
        <xs:element minOccurs="0" name="PageInfo" nillable="true" type="tns:RasterPageInfo"/>
        <xs:element minOccurs="0" name="ResponseCode" type="tns:RasterPageResponseCode"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageResponse" nillable="true" type="tns:RasterPageResponse"/>
    <xs:complexType name="RasterPageInfo">
      <xs:sequence>
        <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="ZoomValues" nillable="true" type="q1:ArrayOfint"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageInfo" nillable="true" type="tns:RasterPageInfo"/>
    <xs:simpleType name="RasterPageResponseCode">
      <xs:restriction base="xs:string">
        <xs:enumeration value="NotModified"/>
        <xs:enumeration value="OK"/>
        <xs:enumeration value="DataSources"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:element name="RasterPageResponseCode" nillable="true"
type="tns:RasterPageResponseCode"/>
    <xs:complexType name="RasterPageItemRequest">
      <xs:sequence>
        <xs:element minOccurs="0" name="EntityTag" nillable="true"
type="tns:RasterPageItemTag"/>
        <xs:element minOccurs="0" name="FileUrl" nillable="true" type="xs:anyURI"/>
        <xs:element minOccurs="0" name="ItemName" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="SequenceId" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageItemRequest" nillable="true" type="tns:RasterPageItemRequest"/>
    <xs:complexType name="RasterPageItemTag">
      <xs:sequence>
        <xs:element minOccurs="0" name="DataModeName" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="FileETag" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageItemTag" nillable="true" type="tns:RasterPageItemTag"/>
    <xs:complexType name="RasterPageItemResponse">
      <xs:sequence>
        <xs:element minOccurs="0" name="EntityTag" nillable="true"
type="tns:RasterPageItemTag"/>
        <xs:element minOccurs="0" name="ItemData" nillable="true" type="xs:base64Binary"/>
        <xs:element minOccurs="0" name="ResponseCode" type="tns:RasterPageItemResponseCode"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="RasterPageItemResponse" nillable="true"
type="tns:RasterPageItemResponse"/>
    <xs:simpleType name="RasterPageItemResponseCode">
```

```
    <xs:restriction base="xs:string">
      <xs:enumeration value="NotModified"/>
      <xs:enumeration value="OK"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="RasterPageItemResponseCode" nillable="true"
type="tns:RasterPageItemResponseCode"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"/>
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
</xs:schema>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfint">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="int" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfint" nillable="true" type="q1:ArrayOfint"/>
</xs:schema>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns1="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>
  <xs:element name="QName" nillable="true" type="xs:QName"/>
  <xs:element name="short" nillable="true" type="xs:short"/>
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
  <xs:element name="char" nillable="true" type="tns1:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns1:duration"/>
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns1:guid"/>
  <xs:simpleType name="guid">
```

```
      <xs:restriction base="xs:string">
        <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:attribute name="FactoryType" type="xs:QName"/>
    <xs:attribute name="Id" type="xs:ID"/>
    <xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
```

7.4 http://tempuri.org/ Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er"/>
  <xs:element name="GetRasterDiagram">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="rasterDiagramRequestContract" nillable="true"
type="tns:RasterDiagramRequestContract"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRasterDiagramResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterDiagramResult" nillable="true" type="tns:RasterDiagram"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetVectorDiagram">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="vectorDiagramRequest" nillable="true"
type="tns:VectorDiagramRequestContract"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetVectorDiagramResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetVectorDiagramResult" nillable="true" type="tns:VectorDiagram"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRasterPage">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="rasterPageRequestContract" nillable="true"
type="tns:RasterPageRequest"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRasterPageResponse">
```

```
      <xs:complexType>
        <xs:sequence>
          <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterPageResult" nillable="true" type="tns:RasterPageResponse"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetRasterPageItem">
      <xs:complexType>
        <xs:sequence>
          <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="rasterPageItemRequestContract" nillable="true"
type="tns:RasterPageItemRequest"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetRasterPageItemResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Visio.Server.GraphicsServ
er" minOccurs="0" name="GetRasterPageItemResult" nillable="true"
type="tns:RasterPageItemResponse"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SaveComments">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="fileUrl" nillable="true" type="xs:anyURI"/>
          <xs:element minOccurs="0" name="commentData" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SaveCommentsResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="SaveCommentsResult" nillable="true"
type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
</xs:schema>
```

8   Appendix C: Product Behavior

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

- Microsoft SharePoint Server 2010

- Microsoft SharePoint Server 2013

- Microsoft SharePoint Server 2016 Preview

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

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

<1> Section 2.2.9.1: This element and its complex types are available only in SharePoint Server 2013.

<2> Section 2.2.9.2: This element and its complex types are available only in SharePoint Server 2013.

<3> Section 2.2.9.3: This element and its complex types are available only in SharePoint Server 2013.

<4> Section 2.2.9.4: This element and its complex types are available only in SharePoint Server 2013.

<5> Section 3.1.1: This model is available only in SharePoint Server 2013.

<6> Section 3.1.2: This timer is available only in SharePoint Server 2013.

<7> Section 3.1.4: The **GetRasterPage** operation is available only in SharePoint Server 2013.

<8> Section 3.1.4: The **GetRasterPageItem** operation is available only in SharePoint Server 2013.

<9> Section 3.1.4: This operation is available only in SharePoint Server 2013.

<10> Section 3.1.4.3: The **GetRasterPage** operation is available only in SharePoint Server 2013.

<11> Section 3.1.4.4: The **GetRasterPageItem** operation is available only in SharePoint Server 2013.

<12> Section 3.1.4.4: In SharePoint Server 2010 and SharePoint Server 2013, the server responds with a **VisioGraphicsServiceFault** complex type SOAP fault only when the request message is null.

<13> Section 3.1.4.5: This operation is available only in SharePoint Server 2013.

<14> Section 3.1.5: This timer event is available only in SharePoint Server 2013.

9   Change Tracking

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

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.

- The removal of a document from the documentation set.

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

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

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

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

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

10 Index