

[MS-WORDSWCFQ]:

Word File Conversion Service Queue WCF Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@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.

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 is preliminary documentation for this technology. Since the documentation may change between this preliminary version and the final version, 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 |
|------------|------------------|----------------|--|
| 07/13/2009 | 0.1 | Major | Initial Availability |
| 08/28/2009 | 0.2 | Editorial | Revised and edited the technical content |

Table of Contents

| | |
|---|-----------|
| 1 Introduction | 8 |
| 1.1 Glossary | 8 |
| 1.2 References | 8 |
| 1.2.1 Normative References | 8 |
| 1.2.2 Informative References | 9 |
| 1.3 Protocol Overview (Synopsis) | 9 |
| 1.4 Relationship to Other Protocols | 10 |
| 1.5 Prerequisites/Preconditions | 11 |
| 1.6 Applicability Statement | 11 |
| 1.7 Versioning and Capability Negotiation | 11 |
| 1.8 Vendor-Extensible Fields | 11 |
| 1.9 Standards Assignments | 12 |
| 2 Messages | 13 |
| 2.1 Transport | 13 |
| 2.2 Common Message Syntax | 13 |
| 2.2.1 Namespaces | 13 |
| 2.2.2 Messages | 14 |
| 2.2.3 Elements | 14 |
| 2.2.4 Complex Types | 14 |
| 2.2.4.1 SPUserToken (from namespace http://schemas.datacontract.org/2004/07/Microsoft.SharePoint) | 14 |
| 2.2.4.2 ArrayOfguid (from namespace http://schemas.microsoft.com/2003/10/Serialization/Arrays) | 14 |
| 2.2.5 Simple Types | 15 |
| 2.2.5.1 guid (from namespace http://schemas.microsoft.com/2003/10/Serialization/) .. | 15 |
| 2.2.6 Attributes | 15 |
| 2.2.7 Groups | 15 |
| 2.2.8 Attribute Groups | 15 |
| 2.3 XML Structures | 15 |
| 2.3.1 Namespaces | 15 |
| 2.3.2 Elements | 16 |
| 2.3.2.1 jobSettings (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs) | 16 |
| 2.3.3 Complex Types | 17 |
| 2.3.3.1 CT_BalloonState (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs) | 18 |
| 2.3.3.2 CT_Bookmarks (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs) | 18 |
| 2.3.3.3 CT_Boolean (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs) | 18 |
| 2.3.3.4 CT_CompatibilityMode (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs) | 18 |

| | | |
|----------|--|----|
| 2.3.3.5 | CT_DefaultLanguage (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 19 |
| 2.3.3.6 | CT_FixedFormatSettings (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 19 |
| 2.3.3.7 | CT_OutputQuality (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 20 |
| 2.3.3.8 | CT_RevisionState (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 20 |
| 2.3.3.9 | CT_SaveBehavior (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 20 |
| 2.3.3.10 | CT_SaveFormat (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 21 |
| 2.3.4 | Simple Types | 21 |
| 2.3.4.1 | ST_BalloonState (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 21 |
| 2.3.4.2 | ST_Bookmarks (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 22 |
| 2.3.4.3 | ST_Boolean (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 22 |
| 2.3.4.4 | ST_CompatibilityMode (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 23 |
| 2.3.4.5 | ST_DefaultLanguage (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 24 |
| 2.3.4.6 | ST_Empty (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 24 |
| 2.3.4.7 | ST_OutputQuality (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 24 |
| 2.3.4.8 | ST_RevisionState (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 25 |
| 2.3.4.9 | ST_SaveBehavior (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 26 |
| 2.3.4.10 | ST_SaveFormat (from namespace http://schemas.microsoft.com/office/server/word/2007/12/conversionSettin gs)..... | 26 |
| 2.3.5 | Attributes | 27 |
| 2.3.6 | Groups | 27 |
| 2.3.7 | Attribute Groups | 27 |

| | |
|--|-----------|
| 3 Protocol Details | 28 |
| 3.1 Server Details | 28 |
| 3.1.1 Abstract Data Model | 28 |
| 3.1.2 Timers | 29 |
| 3.1.3 Initialization | 29 |
| 3.1.4 Message Processing Events and Sequencing Rules | 29 |
| 3.1.4.1 CancelJob | 30 |
| 3.1.4.1.1 Messages | 30 |
| 3.1.4.1.1.1 IQueueApp_CancelJob_InputMessage | 30 |
| 3.1.4.1.1.2 IQueueApp_CancelJob_OutputMessage | 30 |
| 3.1.4.1.2 Elements | 31 |
| 3.1.4.1.2.1 CancelJob | 31 |
| 3.1.4.1.2.2 CancelJobResponse | 31 |
| 3.1.4.2 GetJobStatus | 31 |
| 3.1.4.2.1 Messages | 32 |
| 3.1.4.2.1.1 IQueueApp_GetJobStatus_InputMessage | 32 |
| 3.1.4.2.1.2 IQueueApp_GetJobStatus_OutputMessage | 32 |
| 3.1.4.2.2 Elements | 32 |
| 3.1.4.2.2.1 GetJobStatus | 32 |
| 3.1.4.2.2.2 GetJobStatusResponse | 33 |
| 3.1.4.2.3 Complex Types | 33 |
| 3.1.4.2.3.1 ArrayOfConversionItemInternal (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Service) | 33 |
| 3.1.4.2.3.2 ConversionItemInternal (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Service) | 33 |
| 3.1.4.2.4 Simple Types | 34 |
| 3.1.4.2.4.1 ConversionItemState (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver) | 34 |
| 3.1.4.3 GetAllJobs | 35 |
| 3.1.4.3.1 Messages | 35 |
| 3.1.4.3.1.1 IQueueApp_GetAllJobs_InputMessage | 35 |
| 3.1.4.3.1.2 IQueueApp_GetAllJobs_OutputMessage | 35 |
| 3.1.4.3.2 Elements | 36 |
| 3.1.4.3.2.1 GetAllJobs | 36 |
| 3.1.4.3.2.2 GetAllJobsResponse | 36 |
| 3.1.4.4 CreateNewJob | 36 |
| 3.1.4.4.1 Messages | 38 |
| 3.1.4.4.1.1 IQueueApp_CreateNewJob_InputMessage | 38 |
| 3.1.4.4.1.2 IQueueApp_CreateNewJob_OutputMessage | 38 |
| 3.1.4.4.2 Elements | 38 |
| 3.1.4.4.2.1 CreateNewJob | 38 |
| 3.1.4.4.2.2 CreateNewJobResponse | 38 |
| 3.1.4.4.3 Complex Types | 39 |
| 3.1.4.4.3.1 ArrayOfstring (from namespace http://schemas.microsoft.com/2003/10/Serialization/Arrays) | 39 |
| 3.1.4.4.3.2 ConversionJobData (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Service) | 39 |

| | | |
|-------------|--|-----------|
| 3.1.4.4.3.3 | ConversionJobSettings (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 39 |
| 3.1.4.4.3.4 | FixedFormatSettings (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 40 |
| 3.1.4.4.4 | Simple Types | 41 |
| 3.1.4.4.4.1 | CompatibilityMode (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 41 |
| 3.1.4.4.4.2 | BalloonState (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 42 |
| 3.1.4.4.4.3 | FixedFormatBookmark (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 42 |
| 3.1.4.4.4.4 | FixedFormatQuality (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 43 |
| 3.1.4.4.4.5 | SaveFormat (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 43 |
| 3.1.4.4.4.6 | SaveBehavior (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 44 |
| 3.1.4.4.4.7 | RevisionState (from namespace http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Ser ver.Conversions) | 45 |
| 3.1.4.5 | Ping | 45 |
| 3.1.4.5.1 | Messages | 46 |
| 3.1.4.5.1.1 | IQueueApp_Ping_InputMessage | 46 |
| 3.1.4.5.1.2 | IQueueApp_Ping_OutputMessage | 46 |
| 3.1.4.5.2 | Elements | 46 |
| 3.1.4.5.2.1 | Ping | 46 |
| 3.1.4.5.2.2 | PingResponse | 46 |
| 3.1.4.5.3 | Complex Types | 47 |
| 3.1.4.5.3.1 | DateTimeOffset (from namespace http://schemas.datacontract.org/2004/07/System) | 47 |
| 3.1.4.6 | GetAllActiveJobs | 47 |
| 3.1.4.6.1 | Messages | 48 |
| 3.1.4.6.1.1 | IQueueApp_GetAllActiveJobs_InputMessage | 48 |
| 3.1.4.6.1.2 | IQueueApp_GetAllActiveJobs_OutputMessage | 48 |
| 3.1.4.6.2 | Elements | 48 |
| 3.1.4.6.2.1 | GetAllActiveJobs | 48 |
| 3.1.4.6.2.2 | GetAllActiveJobsResponse | 48 |
| 3.1.5 | Timer Events | 49 |
| 3.1.6 | Other Local Events | 49 |
| 4 | Protocol Examples | 50 |
| 4.1 | Creating a Conversion Job | 50 |
| 4.1.1 | Request | 50 |
| 4.1.2 | Stored Procedure Call | 51 |
| 4.1.3 | Response | 52 |

| | | |
|-----------|--|-----------|
| 4.2 | Getting the Status of a Conversion Job | 53 |
| 4.2.1 | Request | 53 |
| 4.2.2 | Stored Procedure Call | 53 |
| 4.2.3 | Stored Procedure Result Set | 53 |
| 4.2.4 | Response | 54 |
| 5 | Security | 56 |
| 5.1 | Security Considerations for Implementers | 56 |
| 5.2 | Index of Security Parameters | 56 |
| 6 | Appendix A: Full WSDL | 57 |
| 7 | Appendix B: Full Conversion Job Settings XML Schema | 66 |
| 8 | Appendix C: Product Behavior | 69 |
| 9 | Change Tracking..... | 70 |
| 10 | Index | 71 |

1 Introduction

This document specifies the Word File Conversion Service Queue WCF Service Protocol that is used to create, retrieve, and update a queue of documents that are intended to be converted between file formats.

1.1 Glossary

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

Coordinated Universal Time (UTC)
HTTP (Hypertext Transfer Protocol)
HTTPS (Hypertext Transfer Protocol over Secure Sockets Layer)
Transmission Control Protocol (TCP)
UTC (Coordinated Universal Time)
XML (Extensible Markup Language)

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

absolute URL
back-end database server
Boolean
GUID
partition
result set
SOAP (Simple Object Access Protocol)
SOAP action
SOAP body
SOAP fault
stored procedure
URL (Uniform Resource Locator)
WSDL (Web Services Description Language)
WSDL message
WSDL operation
XML namespace
XML namespace prefix
XML Schema

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MC-NMF] Microsoft Corporation, "[.NET Message Framing Protocol Specification](#)", October 2007.

[MS-WORDSSP] Microsoft Corporation, "[Word Services Stored Procedures Protocol Specification](#)", July 2009.

[MS-WORDSWCFW] Microsoft Corporation, "[Word File Conversion Service Worker WCF Service Protocol Specification](#)", July 2009.

[MS-WSSFO] Microsoft Corporation, "[Windows SharePoint Services \(WSS\): File Operations Database Communications Protocol Specification](#)", April 2008.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatté, S., and Winer, D., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

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

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

[WSA1.0 Core] Gudgin, M., Ed., Hadley, M., Ed., and Rogers, Tony, Ed., "Web Services Addressing 1.0 - Core", W3C Recommendation 9 May 2006, <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/ws-addr-core.pdf>.

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

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/REC-xml-names/>

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

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

1.2.2 Informative References

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

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

1.3 Protocol Overview (Synopsis)

This protocol enables a queue client to manage a queue of conversion jobs maintained by a queue server in a database. The protocol enables the queue client to add new conversion jobs, cancel existing conversion jobs, and retrieve a list of conversion jobs and the status of conversion items

within the conversion job. The communication is always initiated by the queue client using different operations.

A typical scenario for using this protocol is an application that performs bulk conversion jobs on an existing set of files. Such an application could use this protocol to store a list of documents that need to be converted and also provide the status of those conversion items. The application would use an implementation of the Worker WCF Service Protocol, as specified in [\[MS-WORDSWCFW\]](#) to initiate and perform the actual conversion of one file to another.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** messaging protocol for formatting requests and responses either as specified in [\[SOAP1.1\]](#) or as specified in [\[SOAP1.2/1\]](#) and [\[SOAP1.2/2\]](#). It transmits these messages using the **HTTP** protocol as specified in [\[RFC2616\]](#) or the **HTTPS** protocol as specified in [\[RFC2818\]](#) or the **TCP** protocol as specified in [\[MC-NMF\]](#).

The following diagram shows the underlying messaging and transport stack that the protocol uses:

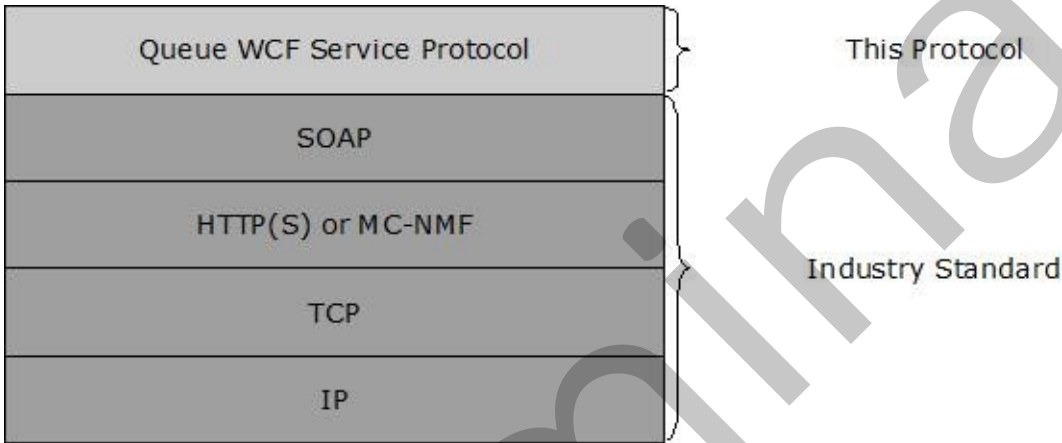


Figure 1: This protocol in relation to underlying messaging and transport stack protocols

While the queue server is used to manage a list of conversion jobs, the actual conversions are handled by the worker service protocol, as specified in [\[MS-WORDSWCFW\]](#). The worker client polls the database for conversion items to be converted and then sends a request to the worker server to perform the conversion of each conversion item. The worker server uses the settings and URLs previously specified by a queue client to convert an input file to an output file. After the conversion item is finished, the worker server updates the status of the conversion item in the database and stores any output. The queue server, worker client, and worker server read from and write to the same queue database using the database protocol specified in [\[MS-WORDSSP\]](#).

The following diagram shows the relationship between this protocol (the queue service), the worker service, and the database protocol. Dotted lines indicate interactions that are implementation-specific and not significant for interoperability in relation to these protocols.

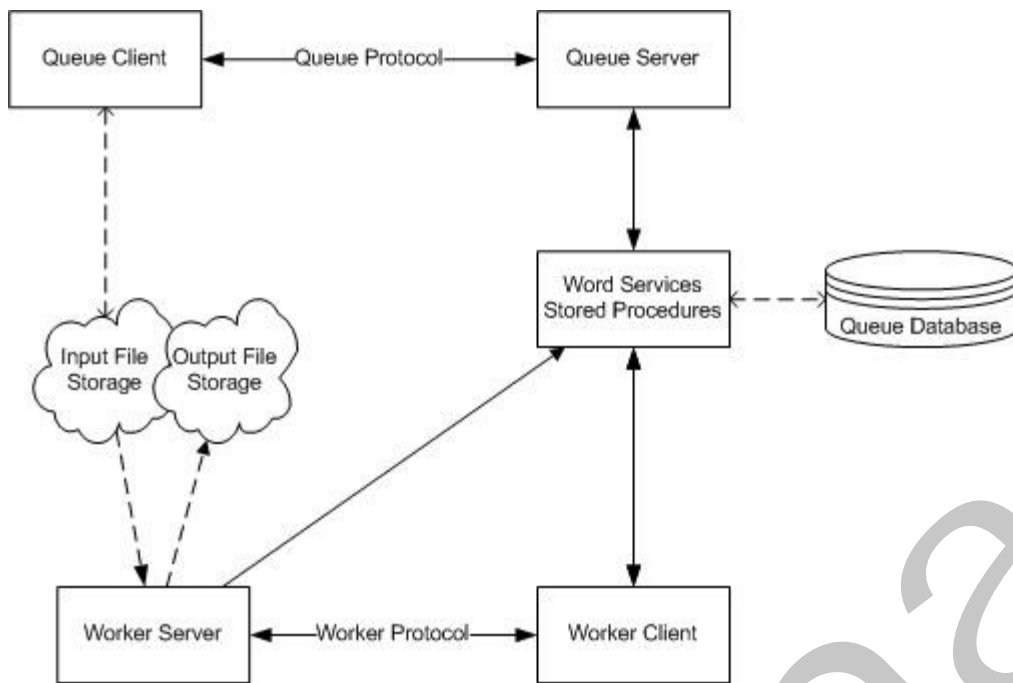


Figure 2: This protocol in relation to worker service and database protocol

1.5 Prerequisites/Preconditions

The queue server is required to use the same database protocol as the worker client and worker server.

The queue client is required to provide the input and output locations of a conversion item as **URLs** that are in a format that is understood by the worker server.

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

1.6 Applicability Statement

This protocol enables communication between a protocol server which manages a queue of conversion jobs, and a protocol client which:

- Creates conversion jobs.
- Cancels conversion jobs.
- Provides access to the status of conversion jobs.

1.7 Versioning and Capability Negotiation

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

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over either HTTP, HTTPS, or TCP.

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

All protocol messages are Internet messages, and MUST be transported using either HTTP or TCP bindings at the transport level.

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a 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 |
|--------|---|--|
| sys | http://schemas.datacontract.org/2004/07/System | |
| ser | http://schemas.microsoft.com/2003/10/Serialization/ | |
| tns | http://tempuri.org/ | |
| xs | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1] [XMLSCHEMA2] |
| soap | http://schemas.xmlsoap.org/wsdl/soap/ | [SOAP1.1] |
| soap12 | http://schemas.xmlsoap.org/wsdl/soap12/ | [SOAP1.2/1] [SOAP1.2/2] |
| wsaw | http://www.w3.org/2006/05/addressing/wsdl | [WSA1.0 Core] |
| wsdl | http://schemas.xmlsoap.org/wsdl/ | [WSDL] |
| sera | http://schemas.microsoft.com/2003/10/Serialization/Arrays | |
| mowss | http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service | |
| mows | http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server | |
| mowsc | http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversi | |

| Prefix | Namespace URI | Reference |
|--------|--|-----------|
| | ons | |
| sp | http://schemas.datacontract.org/2004/07/Microsoft.SharePoint | |

2.2.2 Messages

None.

2.2.3 Elements

This specification does not define any common XML Schema element definitions.

2.2.4 Complex Types

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 |
|--------------|---|
| ArrayOfguid | The ArrayOfguid type is an array of GUID elements. |
| SPUserToken | The SPUserToken type provides user credentials for accessing the input and output files. |

2.2.4.1 SPUserToken (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.SharePoint>)

The **SPUserToken** type provides user credentials for accessing the input and output files.

```
<xs:complexType name="SPUserToken">
  <xs:sequence>
    <xs:element name="m_token" type="xs:base64Binary"/>
  </xs:sequence>
</xs:complexType>
```

m_token: A variable-length structure associated with a user. The format is specified in [\[MS-WSSFO\]](#) section 2.2.4.9.

2.2.4.2 ArrayOfguid (from namespace <http://schemas.microsoft.com/2003/10/Serialization/Arrays>)

The **ArrayOfguid** type is an array of GUID elements.

```
<xs:complexType name="ArrayOfguid">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="guid" type="ser:guid"/>
  </xs:sequence>
</xs:complexType>
```

guid: A GUID value.

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 |
|-------------|--|
| guid | A string representing a GUID or unique identifier. |

2.2.5.1 guid (from namespace <http://schemas.microsoft.com/2003/10/Serialization/>)

A string representing a GUID or unique identifier.

```
<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>
```

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..

2.3 XML Structures

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

This section contains namespaces, simple types, complex types, elements, and attributes used by this protocol when encoding the [ConversionJobSettings](#) as **XML** in the [CreateNewJob](#) **WSDL operation**.

2.3.1 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 |
|--------|--|--|
| xs | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1] [XMLSCHEMA2] |
| cs | http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings | |

2.3.2 Elements

The following table summarizes the set of common XML Schema element definitions defined by this section.

| Element | Description |
|--------------------|--|
| jobSettings | The jobSettings element represents a collection of preferred settings for all conversion items within a single conversion job |

2.3.2.1 jobSettings (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **jobSettings** element represents a collection of preferred settings for all conversion items within a single conversion job. This element is functionally equivalent to and can be directly derived from the [ConversionJobSettings](#) type in the [CreateNewJob](#) WSDL operation.

The worker server MAY [ignore](#) some or all of these preferences.

```
<xs:element name="jobSettings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="revisionState" type="cs:CT_RevisionState"/>
      <xs:element name="updateFields" type="cs:CT_Boolean"/>
      <xs:element name="defaultLanguage" type="cs:CT_DefaultLanguage"/>
      <xs:element name="embedFonts" type="cs:CT_Boolean"/>
      <xs:element name="subsetEmbeddedFonts" type="cs:CT_Boolean"/>
      <xs:element name="doNotIncludeSystemFonts" type="cs:CT_Boolean"/>
      <xs:element name="compatibilityMode" type="cs:CT_CompatibilityMode"/>
      <xs:element name="addThumbNail" type="cs:CT_Boolean"/>
      <xs:element name="saveFormat" type="cs:CT_SaveFormat"/>
      <xs:element name="saveBehavior" type="cs:CT_SaveBehavior"/>
      <xs:element name="fixedFormatSettings" type="cs:CT_FixedFormatSettings"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

revisionState: Specifies a preference for the visibility of revision marking and comments in the output document. This element is functionally equivalent to and can be directly derived from the RevisionState child element of the ConversionJobSettings type.

updateFields: Specifies a preference for whether or not fields are automatically updated. This element is functionally equivalent to and can be directly derived from the UpdateFields child element of the ConversionJobSettings type.

defaultLanguage: Specifies a preference for the language used when resolving language dependent ambiguities. This element is functionally equivalent to and can be directly derived from the DefaultLanguage child element of the ConversionJobSettings type.

embedFonts: Specifies a preference for whether fonts used within the document are obfuscated and saved within the output file. This element is functionally equivalent to and can be directly derived from the EmbedFonts child element of the ConversionJobSettings type.

subsetEmbeddedFonts: Specifies a preference that only characters required for a document are included in the embedded font. This element is functionally equivalent to and can be directly derived from the SubsetEmbeddedFonts child element of the ConversionJobSettings type.

doNotIncludeSystemFonts: Specifies a preference for whether common fonts are not included in the output file. This element is functionally equivalent to and can be directly derived from the DoNotEmbedSystemFonts child element of the ConversionJobSettings type.

compatibilityMode: Specifies a preference for the compatibility of the output document. This element is functionally equivalent to and can be directly derived from the CompatibilityMode child element of the ConversionJobSettings type.

addThumbNail: Specifies a preference for whether the output file is saved with an added thumbnail. This element is functionally equivalent to and can be directly derived from the AddThumbnail child element of the ConversionJobSettings type.

saveFormat: Specifies a preference for the file format of the document after it has been converted. This element is functionally equivalent to and can be directly derived from the OutputFormat child element of the ConversionJobSettings type.

saveBehavior: Specifies a preference for what overwrite behavior occurs when the output file already exists. This element is functionally equivalent to and can be directly derived from the OutputSaveBehavior child element of the ConversionJobSettings type.

fixedFormatSettings: Specifies a collection of additional preferences used for fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the FixedFormatSettings child element of the ConversionJobSettings type.

2.3.3 Complex Types

The following table summarizes the set of common XML Schema complex type definitions defined by this section.

| Complex Type | Description |
|------------------------|--|
| CT_BalloonState | The CT_BalloonState type represents a single ST_BalloonState value. |
| CT_Bookmarks | The CT_Bookmarks type represents a single ST_Bookmarks value. |
| CT_Boolean | The CT_Boolean type represents a single ST_Boolean value. |
| CT_CompatibilityMode | The CT_CompatibilityMode type represents a single ST_CompatibilityMode value. |
| CT_DefaultLanguage | The CT_DefaultLanguage type represents a single ST_DefaultLanguage value. |
| CT_FixedFormatSettings | The CT_FixedFormatSettings type represents a collection of additional preferences used for fixed formats such as PDF and XPS. |

| Complex Type | Description |
|------------------|--|
| CT_OutputQuality | The CT_OutputQuality type represents a single ST_OutputQuality value. |
| CT_RevisionState | The CT_RevisionState type represents a single ST_RevisionState value. |
| CT_SaveBehavior | The CT_SaveBehavior type represents a single ST_SaveBehavior value. |
| CT_SaveFormat | The CT_SaveFormat type represents a single ST_SaveFormat value. |

2.3.3.1 CT_BalloonState (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_BalloonState** type represents a single [ST_BalloonState](#) value.

```
<xs:complexType name="CT_BalloonState">
  <xs:attribute name="val" use="required" type="cs:ST_BalloonState"/>
</xs:complexType>
```

val: An ST_BalloonState value.

2.3.3.2 CT_Bookmarks (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_Bookmarks** type represents a single [ST_Bookmarks](#) value.

```
<xs:complexType name="CT_Bookmarks">
  <xs:attribute name="val" use="required" type="cs:ST_Bookmarks"/>
</xs:complexType>
```

val: An ST_Bookmarks value.

2.3.3.3 CT_Boolean (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_Boolean** type represents a single [ST_Boolean](#) value.

```
<xs:complexType name="CT_Boolean">
  <xs:attribute name="val" use="required" type="cs:ST_Boolean"/>
</xs:complexType>
```

val: An ST_Boolean value.

2.3.3.4 CT_CompatibilityMode (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_CompatibilityMode** type represents a single [ST_CompatibilityMode](#) value.

```
<xs:complexType name="CT_CompatibilityMode">
  <xs:attribute name="val" use="required" type="cs:ST_CompatibilityMode"/>
</xs:complexType>
```

val: An ST_CompatibilityMode value.

2.3.3.5 CT_DefaultLanguage (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_DefaultLanguage** type represents a single [ST_DefaultLanguage](#) value.

```
<xs:complexType name="CT_DefaultLanguage">
  <xs:attribute name="val" use="required" type="cs:ST_DefaultLanguage"/>
</xs:complexType>
```

val: An ST_DefaultLanguage value.

2.3.3.6 CT_FixedFormatSettings (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_FixedFormatSettings** type represents a collection of additional preferences used for fixed formats such as PDF and XPS.

This type is functionally equivalent to and can be directly derived from the [FixedFormatSettings](#) type in the [CreateNewJob](#) WSDL operation.

```
<xs:complexType name="CT_FixedFormatSettings">
  <xs:sequence>
    <xs:element name="outputQuality" type="cs:CT_OutputQuality"/>
    <xs:element name="bookmarks" type="cs:CT_Bookmarks"/>
    <xs:element name="includeDocumentProperties" type="cs:CT_Boolean"/>
    <xs:element name="includeDocumentStructure" type="cs:CT_Boolean"/>
    <xs:element name="usePDFA" type="cs:CT_Boolean"/>
    <xs:element name="bitmapEmbeddedFonts" type="cs:CT_Boolean"/>
    <xs:element name="balloonState" type="cs:CT_BalloonState"/>
  </xs:sequence>
</xs:complexType>
```

outputQuality: Specifies a preference for the output quality of fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the **OutputQuality** child element of the FixedFormatSettings type.

bookmarks: Specifies a preference for the creation of bookmarks in fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the **Bookmarks** child element of the FixedFormatSettings type.

includeDocumentProperties: Specifies a preference to include document properties in fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the **IncludeDocumentProperties** child element of the FixedFormatSettings type.

includeDocumentStructure: Specifies a preference to include document structure tags in fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the **IncludeDocumentStructure** child element of the FixedFormatSettings type.

usePDFA: Specifies a preference to use the PDF/A format for the fixed format PDF. This element is functionally equivalent to and can be directly derived from the **UsePDFA** child element of the FixedFormatSettings type.

bitmapEmbeddedFonts: Specifies a preference to rasterize fonts when they cannot be embedded in fixed formats such as PDF or XPS. This element is functionally equivalent to and can be directly derived from the **BitmapEmbeddedFonts** child element of the FixedFormatSettings type.

balloonState: Specifies a preference for the visibility of markup balloons in fixed formats such as PDF and XPS. This element is functionally equivalent to and can be directly derived from the **BalloonState** child element of the FixedFormatSettings type.

2.3.3.7 CT_OutputQuality (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_OutputQuality** type represents a single [ST_OutputQuality](#) value.

```
<xs:complexType name="CT_OutputQuality">
  <xs:attribute name="val" use="required" type="cs:ST_OutputQuality"/>
</xs:complexType>
```

val: An ST_OutputQuality value.

2.3.3.8 CT_RevisionState (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_RevisionState** type represents a single [ST_RevisionState](#) value.

```
<xs:complexType name="CT_RevisionState">
  <xs:attribute name="val" use="required" type="cs:ST_RevisionState"/>
</xs:complexType>
```

val: An ST_RevisionState value.

2.3.3.9 CT_SaveBehavior (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_SaveBehavior** type represents a single [ST_SaveBehavior](#) value.

```
<xs:complexType name="CT_SaveBehavior">
  <xs:attribute name="val" use="required" type="cs:ST_SaveBehavior"/>
</xs:complexType>
```

val: An ST_SaveBehavior value.

2.3.3.10 CT_SaveFormat (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **CT_SaveFormat** type represents a single [ST_SaveFormat](#) value.

```
<xs:complexType name="CT_SaveFormat">
  <xs:attribute name="val" use="required" type="cs:ST_SaveFormat"/>
</xs:complexType>
```

val: An [ST_SaveFormat](#) value.

2.3.4 Simple Types

The following table summarizes the set of common XML Schema simple type definitions defined by this section.

| Simple Type | Description |
|----------------------|---|
| ST_BalloonState | The ST_BalloonState type specifies a preference for the visibility of markup balloons in the document. |
| ST_Bookmarks | The ST_Bookmarks type specifies a preference for the creation of bookmarks in fixed formats such as PDF and XPS. |
| ST_Boolean | The ST_Boolean type specifies a preference to enable or disable a Boolean setting. |
| ST_CompatibilityMode | The ST_CompatibilityMode type specifies a preference for the compatibility of the output document. |
| ST_DefaultLanguage | The ST_DefaultLanguage type specifies a preference for the language used when resolving language dependent ambiguities. |
| ST_Empty | The ST_Empty type specifies a value to have an empty string. |
| ST_OutputQuality | The ST_OutputQuality type specifies a preference for the output quality of fixed formats such as PDF and XPS. |
| ST_RevisionState | The ST_RevisionState type specifies a preference for the visibility of revision marking and comments in the output document. |
| ST_SaveBehavior | The ST_SaveBehavior type specifies a preference for what overwrite behavior occurs when the output file already exists. |
| ST_SaveFormat | The ST_SaveFormat type specifies a preference for the file format of the output document. |

2.3.4.1 ST_BalloonState (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_BalloonState** type specifies a preference for the visibility of markup balloons in the document.

This type is functionally equivalent to and can be directly derived from the [BalloonState](#) type in the [CreateNewJob](#) WSDL operation.

```

<xs:simpleType name="ST_BalloonState">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for ST_BalloonState:

| Value | mowsc:BalloonState Value | Meaning |
|-------|---------------------------|--|
| 0 | AlwaysUse | Prefer that balloons always display when present. |
| 1 | Inline | Prefer that balloons appear only when inline. |
| 2 | OnlyCommentsAndFormatting | Prefer that balloons appear only in comments and formatting. |

2.3.4.2 ST_Bookmarks (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_Bookmarks** type specifies a preference for the creation of bookmarks in fixed formats such as PDF and XPS.

This type is functionally equivalent to and can be directly derived from the [FixedFormatBookmark](#) type in the [CreateNewJob](#) WSDL operation.

```

<xs:simpleType name="ST_Bookmarks">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="2"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for ST_Bookmarks:

| Value | mowsc:FixedFormatBookmark Value | Meaning |
|-------|---------------------------------|---|
| 0 | None | Prefer not to convert source document's bookmarks or headings into bookmarks. |
| 1 | Headings | Prefer to convert source document's headings into bookmarks. |
| 2 | Bookmarks | Prefer to convert source document's bookmarks into bookmarks. |

2.3.4.3 ST_Boolean (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_Boolean** type specifies a preference to enable or disable a Boolean setting.

This type is functionally equivalent to and can be directly derived from the **xs:boolean** type specified in [\[XMLSCHEMA2\]](#).

```
<xs:simpleType name="ST_Boolean">
  <xs:restriction base="xs:string">
    <xs:enumeration value="True"/>
    <xs:enumeration value="False"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ST_Boolean:

| Value | xs:boolean Value | Meaning |
|-------|------------------|--|
| True | true | Specifies a preference to enable the setting. |
| False | false | Specifies a preference to disable the setting. |

2.3.4.4 ST_CompatibilityMode (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_CompatibilityMode** type specifies a preference for the compatibility of the output document.

This type is functionally equivalent to and can be directly derived from the [CompatibilityMode](#) type in the [CreateNewJob](#) WSDL operation.

```
<xs:simpleType name="ST_CompatibilityMode">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="-1"/>
    <xs:enumeration value="0"/>
    <xs:enumeration value="11"/>
    <xs:enumeration value="12"/>
    <xs:enumeration value="14"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ST_CompatibilityMode:

| Value | mowsc:CompatibilityMode Value | Meaning |
|-------|-------------------------------|---|
| -1 | AlwaysUpgrade | Prefer document is upgraded in compatibility to the latest Word format available. |
| 0 | MaintainCurrentSetting | Prefer document maintains the compatibility mode specified in the file. |
| 11 | Word97To2003 | Prefer document is compatible with Microsoft® Office Word 2003. |
| 12 | Word2007 | Prefer document is compatible with Microsoft® Office Word 2007. |

| Value | mowsc:CompatibilityMode Value | Meaning |
|-------|-------------------------------|--|
| 14 | Word2009 | Prefer document is compatible with Microsoft® Word 2010 Technical Preview. |

2.3.4.5 ST_DefaultLanguage (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_DefaultLanguage** type specifies a preference for the language used when resolving language dependent ambiguities.

The value SHOULD be an empty string as specified by the [ST_Empty](#) type.

The value MAY [<2>](#) be an **xs:integer** as specified in [\[XMLSCHEMA2\]](#).

```
<xs:simpleType name="ST_DefaultLanguage">
  <xs:union memberTypes="xs:integer cs:ST_Empty"/>
</xs:simpleType>
```

2.3.4.6 ST_Empty (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_Empty** type specifies a value to have an empty string.

This type is used by [ST_DefaultLanguage](#) to specify a preference to not explicitly specify the language used when resolving language-dependent ambiguities.

```
<xs:simpleType name="ST_Empty">
  <xs:restriction base="xs:string">
    <xs:enumeration value=""/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ST_Empty:

| Value | Meaning |
|-------|--|
| | Specifies that the protocol client does not have a preferred language for use when resolving language-dependent ambiguities. |

2.3.4.7 ST_OutputQuality (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_OutputQuality** type specifies a preference for the output quality of fixed formats such as PDF and XPS.

This type is functionally equivalent to and can be directly derived from the [FixedFormatQuality](#) type in the [CreateNewJob](#) WSDL operation.


```

<xs:simpleType name="ST_OutputQuality">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for ST_OutputQuality:

| Value | mowsc:FixedFormatQuality Value | Meaning |
|-------|--------------------------------|--|
| 0 | Standard | Prefer to optimize for printing. |
| 1 | Minimum | Prefer to optimize for online reading. |

2.3.4.8 ST_RevisionState (from namespace <http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_RevisionState** type specifies a preference for the visibility of revision marking and comments in the output document.

This type is functionally equivalent to and can be directly derived from the [RevisionState](#) type in the [CreateNewJob](#) WSDL operation.

```

<xs:simpleType name="ST_RevisionState">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for ST_RevisionState:

| Value | mowsc:RevisionState Value | Meaning |
|-------|---------------------------|---|
| 0 | Final | Prefer to show the final document without revision marking and comments. |
| 1 | Original | Prefer to show the original document without revision marking and comments. |
| 2 | FinalShowingMarkup | Prefer to show the final document with revision marking and comments. |
| 3 | OriginalShowingMarkup | Prefer to show the original document with revision marking and comments. |

2.3.4.9 ST_SaveBehavior (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_SaveBehavior** type specifies a preference for what overwrite behavior occurs when the output file already exists.

This type is functionally equivalent to and can be directly derived from the [SaveBehavior](#) type in the [CreateNewJob](#) WSDL operation.

```
<xs:simpleType name="ST_SaveBehavior">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="3"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ST_SaveBehavior:

| Value | mowsc:SaveBehavior Value | Meaning |
|-------|--------------------------|---|
| 0 | AppendIfPossible | Prefer to append to the file's version history if the output file already exists. If the location of the output file does not support version history, prefer to overwrite the existing file. |
| 1 | AlwaysOverwrite | Prefer to always overwrite the output file if it already exists. |
| 2 | AppendOnly | Prefer to append to the file's version history if the output file already exists. If the location of the output file does not support version history, prefer to do nothing. |
| 3 | NeverOverwrite | Prefer to do nothing if there is already a file at the output file location. |

2.3.4.10 ST_SaveFormat (from namespace

<http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings>)

The **ST_SaveFormat** type specifies a preference for the file format of the output document.

This type is functionally equivalent to and can be directly derived from the [SaveFormat](#) type in the [CreateNewJob](#) WSDL operation.

```
<xs:simpleType name="ST_SaveFormat">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="11"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ST_SaveFormat:

| Value | mowsc:SaveFormat Value | Meaning |
|-------|------------------------|---|
| 0 | Automatic | The format of the output file is preferred to be determined by the file extension. |
| 1 | Document | The output file is preferred to be an Office Open XML document. |
| 2 | DocumentMacroEnabled | The output file is preferred to be an Office Open XML document with macros enabled. |
| 3 | Document97 | The output file is preferred to be an Microsoft® Office Word 2003 compatible binary document. |
| 4 | Template | The output file is preferred to be an Office Open XML document template. |
| 5 | TemplateMacroEnabled | The output file is preferred to be an Office Open XML document template with macros enabled. |
| 6 | Template97 | The output file is preferred to be an Office Word 2003 compatible binary document template. |
| 7 | MHTML | The output file is preferred to be a single file Web page. |
| 8 | PDF | The output file is preferred to be a PDF file. |
| 9 | RTF | The output file is preferred to be a Rich Text Format (RTF) document. |
| 10 | XML | The output file is preferred to be a Word XML document. |
| 11 | XPS | The output file is preferred to be an XPS file. |

2.3.5 Attributes

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

2.3.6 Groups

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

2.3.7 Attribute Groups

This section does not define any common XML Schema attribute group definitions.

3 Protocol Details

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

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

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

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

3.1 Server Details

3.1.1 Abstract Data Model

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

The queue server reads and writes data to a queue database which resides on a **back-end database server**. The queue database is also used by the worker client (as specified in [\[MS-WORDSWCFW\]](#) section 3.2) and the worker server (as specified in [\[MS-WORDSWCFW\]](#) section 3.1). Both services read from and write to the queue database using the database protocol as specified in [\[MS-WORDSSP\]](#).

Conversion Job: A logical group of conversion items that share the same conversion settings. Conversion jobs are constructed by the queue client and maintained in the database server by the queue server. A conversion job contains one or more conversion item(s).

Conversion Item: A single conversion from one input file to one output file. A conversion item is maintained in the database server by the queue server, the worker client, and the worker server.

Conversion Item State: The status of a conversion item. The conversion item state is specified by the [ConversionItemState](#) type and specified by the corresponding enumeration values in [\[MS-WORDSSP\]](#) section 2.2.1.1. A conversion item is considered not finished if its state has a value of **NotStarted** or **InProgress**.

Input File: The URL of the source file for a conversion item. The URL MUST be provided by the queue client and MUST be in a format that the worker server understands and can use to retrieve the input file.

Output File: The URL of the output file from a conversion item. The URL MUST be provided by the queue client and MUST be in a format that the worker server understands and can use to store the output file.

The queue server operates on a table of conversion jobs and another table of conversion items in those conversion jobs in conjunction with the worker client and worker server. The database tables are used in the following ways:

- The queue client initiates a request **WSDL message** to the queue server for an operation that adds conversion jobs or cancels existing conversion items within a conversion job. As part of that operation, the queue server synchronously makes the requested change or changes to the database and returns the appropriate response WSDL message.
- The queue client initiates a request WSDL message to the queue server for an operation requesting a list of conversion jobs or the status of conversion items in a conversion job. The queue server synchronously retrieves the requested data from the database and returns the appropriate response WSDL message with the requested data.
- The worker client polls the database for a batch of conversion items that have not finished. This includes both conversion items that have not been started and conversion items that were started by a previous batch and are already in progress. Based directly on conversion items in the database and their status, the worker client synchronously updates the status of those conversion items in the database. This update includes changing the state of a conversion item to in progress or failed. For each conversion item that is changed to in progress, the worker client sends a request WSDL message to the worker server to asynchronously convert that conversion item.
- The worker server receives a request from a worker client to convert a conversion item, and will initiate an internal operation to perform an asynchronous conversion of the conversion item and then return the appropriate response WSDL message. The worker server synchronously updates the status of the conversion item in the database when the internal operation has finished.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

This specification includes the following WSDL operations:

| WSDL Operation | Description |
|------------------|--|
| CancelJob | This operation is used to cancel conversion items in a conversion job that have not finished. |
| GetJobStatus | This operation is used to get the status of all conversion items in a conversion job. |
| GetAllJobs | This operation is used to enumerate all conversion jobs. |
| CreateNewJob | This operation is used to add a new conversion job to the queue. |
| Ping | This operation is used to determine the responsiveness of the service. |
| GetAllActiveJobs | This operation is used to enumerate all conversion jobs which have one or more conversion items that are not started or in progress. |

3.1.4.1 CancelJob

This operation is used to cancel conversion items in a conversion job that have not finished.

```
<wsdl:operation name="CancelJob">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/CancelJob"
    message="tns:IQueueApp_CancelJob_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/CancelJobResponse"
    message="tns:IQueueApp_CancelJob_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **CancelJob** request WSDL message, as follows:

- The **jobId** MUST NOT be null.
- The **rawPartitionId** MUST NOT be null.

The protocol server MUST call the **proc_ConversionJobCancel** **stored procedure**, as specified in [\[MS-WORDSSP\]](#) section 3.1.5.5. The following table specifies the parameter values for the stored procedure.

| Parameter | Value |
|--------------|--|
| @JobId | This MUST be the conversion job id specified by the jobId element. |
| @PartitionId | This MUST be the partition id specified by the rawPartitionId element. |

The protocol server MUST respond with a **CancelJob** response WSDL message.

3.1.4.1.1 Messages

3.1.4.1.1.1 IQueueApp_CancelJob_InputMessage

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

The **SOAP action** value is:

```
http://tempuri.org/IQueueApp/CancelJob
```

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

3.1.4.1.1.2 IQueueApp_CancelJob_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/CancelJob
```

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

3.1.4.1.2 Elements

3.1.4.1.2.1 CancelJob

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

```
<xs:element name="CancelJob">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="jobId" type="ser:guid"/>
      <xs:element minOccurs="1" name="rawPartitionId" type="ser:guid"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

jobId: A GUID value identifying the conversion job.

rawPartitionId: A GUID value identifying the partition of the conversion job.

3.1.4.1.2.2 CancelJobResponse

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

```
<xs:element name="CancelJobResponse"/>
```

3.1.4.2 GetJobStatus

This operation is used to get the status of all conversion items in a conversion job.

```
<wsdl:operation name="GetJobStatus">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetJobStatus"
    message="tns:IQueueApp_GetJobStatus_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetJobStatusResponse"
    message="tns:IQueueApp_GetJobStatus_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **GetJobStatus** request WSDL message, as follows:

- The **jobId** MUST NOT be null.
- The **rawPartitionId** MUST NOT be null.

The protocol server MUST call the **proc_ConversionJobStatusGet** stored procedure, as specified in [\[MS-WORDSSP\]](#) section 3.1.5.8. The following table specifies the parameter values for the stored procedure.

| Parameter | Value |
|------------|--|
| @JobId | This MUST be the conversion job id specified by the jobId element. |
| @UserToken | This MUST be user token specified by the userToken element, as follows: <ul style="list-style-type: none">▪ If userToken is null, this value MUST be null. |

| Parameter | Value |
|--------------|---|
| | <ul style="list-style-type: none"> If userToken is not null, this value MUST be the value of the m_token child element. |
| @PartitionId | This MUST be the partition id specified by the m_partitionId element. |

The protocol server MUST respond with a **GetJobStatus** response WSDL message, as follows:

- If the stored procedure returns a valid **result set**, the protocol server MUST return an array containing a [ConversionItemInternal](#) element for each conversion item in the result set.
- If the stored procedure returns an empty result set, the protocol server MUST return a SOAP fault.

3.1.4.2.1 Messages

3.1.4.2.1.1 IQueueApp_GetJobStatus_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetJobStatus
```

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

3.1.4.2.1.2 IQueueApp_GetJobStatus_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetJobStatus
```

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

3.1.4.2.2 Elements

3.1.4.2.2.1 GetJobStatus

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

```
<xs:element name="GetJobStatus">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="jobId" type="ser:guid"/>
      <xs:element minOccurs="1" name="rawPartitionId" type="ser:guid"/>
      <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```


jobId: A GUID value identifying the conversion job.

rawPartitionId: A GUID value identifying the partition of the conversion job.

userToken: An [SPUserToken](#) value that represents the user credentials used to filter the list of conversion jobs.

3.1.4.2.2 GetJobStatusResponse

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

```
<xs:element name="GetJobStatusResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetJobStatusResult" nillable="true"
type="mowss:ArrayOfConversionItemInternal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetJobStatusResult: An [ArrayOfConversionItemInternal](#) value containing a [ConversionItemInternal](#) element for each conversion item in the conversion job.

3.1.4.2.3 Complex Types

3.1.4.2.3.1 ArrayOfConversionItemInternal (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service>)

The **ArrayOfConversionItemInternal** type represents an array of [ConversionItemInternal](#) elements.

```
<xs:complexType name="ArrayOfConversionItemInternal">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="ConversionItemInternal"
type="mowss:ConversionItemInternal"/>
  </xs:sequence>
</xs:complexType>
```

ConversionItemInternal: A [ConversionItemInternal](#) value.

3.1.4.2.3.2 ConversionItemInternal (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service>)

The **ConversionItemInternal** type represents a conversion item.

```
<xs:complexType name="ConversionItemInternal">
  <xs:sequence>
    <xs:element minOccurs="0" name="AppManId" type="ser:guid"/>
    <xs:element minOccurs="0" name="CompleteTime" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="ConversionId" type="xs:unsignedInt"/>
    <xs:element minOccurs="1" name="InputFile" type="xs:string"/>
    <xs:element minOccurs="0" name="JobId" type="ser:guid"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element minOccurs="0" name="MessageDetails" type="xs:string"/>
<xs:element minOccurs="1" name="MessageId" type="xs:int"/>
<xs:element minOccurs="1" name="OutputFile" type="xs:string"/>
<xs:element minOccurs="0" name="RestartCount" type="xs:int"/>
<xs:element minOccurs="0" name="StartTime" type="xs:dateTime"/>
<xs:element minOccurs="1" name="State" type="mows:ConversionItemState"/>
</xs:sequence>
</xs:complexType>

```

AppManId: This element is not used and MUST be ignored.

CompleteTime: This element is not used and MUST be ignored.

ConversionId: This element is not used and MUST be ignored.

InputFile: A string value that represents the **absolute URL** of the input file.

JobId: This element is not used and MUST be ignored.

MessageDetails: This element is not used and MUST be ignored.

MessageId: An integer value that represents the error condition that caused this conversion item to fail. If the **State** element does not have a value of **Failed**, this value MUST be ignored.

OutputFile: A string value that represents the absolute URL of the output file.

RestartCount: This element is not used and MUST be ignored.

State: A [ConversionItemState](#) value that represents the status of this conversion item.

3.1.4.2.4 Simple Types

3.1.4.2.4.1 ConversionItemState (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server>)

The **ConversionItemState** type is an enumeration that specifies the current status of a conversion item.

```

<xs:simpleType name="ConversionItemState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotStarted"/>
    <xs:enumeration value="InProgress"/>
    <xs:enumeration value="Canceled"/>
    <xs:enumeration value="Succeeded"/>
    <xs:enumeration value="Failed"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for ConversionItemState:

| Value | Meaning |
|------------|--|
| NotStarted | Processing of the conversion item has not started. |
| InProgress | Processing of the conversion item has started, but not finished. |

| Value | Meaning |
|-----------|--|
| Canceled | Processing of the conversion item was canceled. |
| Succeeded | Processing of the conversion item has successfully finished. |
| Failed | Processing of the conversion item failed. |

3.1.4.3 GetAllJobs

This operation is used to enumerate all conversion jobs.

```
<wsdl:operation name="GetAllJobs">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetAllJobs"
    message="tns:IQueueApp_GetAllJobs_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetAllJobsResponse"
    message="tns:IQueueApp_GetAllJobs_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **GetAllJobs** request WSDL message and the **partitionId** MUST NOT be null.

The protocol server MUST call the **proc_ConversionJobGetAll** stored procedure, as specified in [\[MS-WORDSSP\]](#) section 3.1.5.6. The following table specifies the parameter values for the stored procedure.

| Parameter | Value |
|--------------|---|
| @PartitionId | This MUST be the partition id specified by the partitionId element. |
| @UserToken | This MUST be user token specified by the userToken element, as follows: <ul style="list-style-type: none"> If userToken is null, this value MUST be null. If userToken is not null, this value MUST be the value of the m_token child element. |

The protocol server MUST respond with a **GetAllJobs** response WSDL message, containing all conversion job ids in the result set.

3.1.4.3.1 Messages

3.1.4.3.1.1 IQueueApp_GetAllJobs_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetAllJobs
```

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

3.1.4.3.1.2 IQueueApp_GetAllJobs_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetAllJobs
```

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

3.1.4.3.2 Elements

3.1.4.3.2.1 GetAllJobs

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

```
<xs:element name="GetAllJobs">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="partitionId" type="ser:guid"/>
      <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

partitionId: A GUID value identifying the partition of the conversion job.

userToken: An [SPUserToken](#) value that represents the user credentials used to filter the list of conversion jobs.

3.1.4.3.2.2 GetAllJobsResponse

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

```
<xs:element name="GetAllJobsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetAllJobsResult" nillable="true"
type="ser:ArrayOfguid"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetAllJobsResult: An [ArrayOfguid](#) value containing the conversion job identifiers for all conversion jobs returned.

3.1.4.4 CreateNewJob

This operation is used to add a new conversion job to the queue.

```
<wsdl:operation name="CreateNewJob">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/CreateNewJob"
message="tns:IQueueApp_CreateNewJob_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/CreateNewJobResponse"
message="tns:IQueueApp_CreateNewJob_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **CreateNewJob** request WSDL message as follows:

- The **jobData** element MUST NOT be null.
- The **inputFiles** element MUST NOT be null.
- The **outputFiles** element MUST NOT be null.
- The **inputFiles** element MUST have the same number of children elements as the **outputFiles** element.
- The children elements in the **inputFiles** and **outputFiles** elements MUST NOT be empty.

The protocol server MUST create a new GUID value to be used as the conversion job identifier.

The protocol server MUST call the **proc_ConversionJobAdd** stored procedure, as specified in [\[MS-WORDSSP\]](#) section 3.1.5.4. The following table specifies the parameter values for the stored procedure.

| Parameter | Value |
|--------------|---|
| @JobId | This MUST be the conversion job identifier created by the protocol server. |
| @JobXml | <p>This MUST be the list of conversion items encoded as XML according to the XML Schema specified in [MS-WORDSSP] section 2.2.6.4.2.</p> <p>The following pseudocode specifies how to create the JobXml from the inputFiles and outputFiles elements:</p> <pre>WRITE START ELEMENT job FOR ConversionItemId = 0 TO ConversionItemCount WRITE START ELEMENT ci WRITE ELEMENT id WITH ConversionItemId WRITE ELEMENT inputFile WITH inputFiles[ConversionItemId] WRITE ELEMENT outputFile WITH outputFiles[ConversionItemId] WRITE END ELEMENT ci ENDFOR WRITE END ELEMENT job</pre> |
| @UserToken | <p>This MUST be the user token specified by the m_userToken child element of jobData, as follows:</p> <ul style="list-style-type: none">▪ If m_userToken is null, this value MUST be null.▪ If m_userToken is not null, this value MUST be the value of the m_token child element. |
| @PartitionId | This MUST be the partition identifier specified by m_rawPartitionId child element of jobData . |
| @SettingsXml | This MUST be the m_settings child element of jobData encoded as XML according to the XML Schema specified in section 2.3 , with a root element of cs:jobSettings . |

The protocol server MUST respond with a **CreateNewJob** response WSDL message with a **CreateNewJobResult** element whose value is the conversion job identifier created by the protocol server.

3.1.4.4.1 Messages

3.1.4.4.1.1 IQueueApp_CreateNewJob_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/CreateNewJob
```

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

3.1.4.4.1.2 IQueueApp_CreateNewJob_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/CreateNewJob
```

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

3.1.4.4.2 Elements

3.1.4.4.2.1 CreateNewJob

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

```
<xs:element name="CreateNewJob">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="inputFiles" type="sera:ArrayOfstring"/>
      <xs:element minOccurs="1" name="outputFiles" type="sera:ArrayOfstring"/>
      <xs:element minOccurs="1" name="jobData" type="mowss:ConversionJobData"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

inputFiles: An [ArrayOfstring](#) value that specifies the input file URL of each conversion item in the conversion job.

outputFiles: An [ArrayOfstring](#) value that specifies the output file URL of each conversion item in the conversion job.

jobData: A [ConversionJobData](#) value that specifies properties of the conversion job.

3.1.4.4.2.2 CreateNewJobResponse

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

```
<xs:element name="CreateNewJobResponse">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element name="CreateNewJobResult" type="ser:guid"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

```

CreateNewJobResult: A GUID value that represents the conversion job identifier.

3.1.4.4.3 Complex Types

3.1.4.4.3.1 ArrayOfstring (from namespace <http://schemas.microsoft.com/2003/10/Serialization/Arrays>)

The **ArrayOfstring** type represents an array of string elements.

```

<xs:complexType name="ArrayOfstring">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" name="string" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

```

string: A string value.

3.1.4.4.3.2 ConversionJobData (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service>)

The **ConversionJobData** type specifies properties of the conversion job.

```

<xs:complexType name="ConversionJobData">
    <xs:sequence>
        <xs:element minOccurs="0" name="m_partitionId" type="ser:guid"/>
        <xs:element minOccurs="1" name="m_rawPartitionId" type="ser:guid"/>
        <xs:element minOccurs="1" name="m_settings" type="mowsc:ConversionJobSettings"/>
        <xs:element minOccurs="0" name="m_userToken" type="sp:SPUserToken"/>
    </xs:sequence>
</xs:complexType>

```

m_partitionId: This element is not used and MUST be ignored.

m_rawPartitionId: A GUID value identifying the partition of the conversion job.

m_settings: A [ConversionJobsettings](#) value that provides a collection of preferred settings for all conversion items within a single conversion job.

m_userToken: An [SPUserToken](#) value that provides user credentials for accessing the input and output files in the conversion job.

3.1.4.4.3.3 ConversionJobSettings (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This complex type specifies a preference for a collection of settings for all conversion items within a single conversion job.

The worker server MAY [<3>](#) ignore some or all of these preferences.

```
<xs:complexType name="ConversionJobSettings">
  <xs:sequence>
    <xs:element minOccurs="0" name="AddThumbnail" type="xs:boolean"/>
    <xs:element minOccurs="0" name="CompatibilityMode" type="mowsc:CompatibilityMode"/>
    <xs:element minOccurs="0" name="DefaultLanguage"/>
    <xs:element minOccurs="0" name="DoNotEmbedSystemFonts" type="xs:boolean"/>
    <xs:element minOccurs="0" name="EmbedFonts" type="xs:boolean"/>
    <xs:element minOccurs="1" name="FixedFormatSettings" type="mowsc:FixedFormatSettings"/>
    <xs:element minOccurs="0" name="OutputFormat" type="mowsc:SaveFormat"/>
    <xs:element minOccurs="0" name="OutputSaveBehavior" type="mowsc:SaveBehavior"/>
    <xs:element minOccurs="0" name="RevisionState" type="mowsc:RevisionState"/>
    <xs:element minOccurs="0" name="SubsetEmbeddedFonts" type="xs:boolean"/>
    <xs:element minOccurs="0" name="UpdateFields" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

AddThumbnail: Specifies a preference for whether the output file is saved with an added thumbnail.

CompatibilityMode: Specifies a preference for the compatibility of the output document.

DefaultLanguage: Specifies a preference for the language used when resolving language-dependent ambiguities.

This value SHOULD [<4>](#) be null.

DoNotEmbedSystemFonts: Specifies a preference for whether common fonts are not included in the output file.

EmbedFonts: Specifies a preference for whether fonts used within the document are obfuscated and saved within the output file.

FixedFormatSettings: Specifies a collection of additional preferences used for fixed formats such as PDF and XPS.

OutputFormat: Specifies a preference for the file format of the document after it has been converted.

OutputSaveBehavior: Specifies a preference for what overwrite behavior occurs when the output file already exists.

RevisionState: Specifies a preference for the visibility of revision marking and comments in the output document.

SubsetEmbeddedFonts: Specifies a preference that only characters required for a document are included in the embedded font.

UpdateFields: Specifies a preference for whether or not fields are automatically updated.

3.1.4.4.3.4 FixedFormatSettings (from namespace

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

The **FixedFormatSettings** type represents a collection of additional preferences used for fixed formats such as PDF and XPS.

The worker server MAY [ignore](#) some or all of these preferences.

```
<xs:complexType name="FixedFormatSettings">
  <xs:sequence>
    <xs:element minOccurs="0" name="BalloonState" type="mowsc:BalloonState"/>
    <xs:element minOccurs="0" name="BitmapEmbeddedFonts" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Bookmarks" type="mowsc:FixedFormatBookmark"/>
    <xs:element minOccurs="0" name="IncludeDocumentProperties" type="xs:boolean"/>
    <xs:element minOccurs="0" name="IncludeDocumentStructure" type="xs:boolean"/>
    <xs:element minOccurs="0" name="OutputQuality" type="mowsc:FixedFormatQuality"/>
    <xs:element minOccurs="0" name="UsePDFA" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

BalloonState: Specifies a preference for the visibility of markup balloons in fixed formats such as PDF and XPS.

BitmapEmbeddedFonts: Specifies a preference to rasterize fonts when they cannot be embedded in fixed formats such as PDF or XPS.

Bookmarks: Specifies a preference for the creation of bookmarks in fixed formats such as PDF and XPS.

IncludeDocumentProperties: Specifies a preference to include document properties in fixed formats such as PDF or XPS.

IncludeDocumentStructure: Specifies a preference to include document structure tags in fixed formats such as PDF or XPS.

OutputQuality: Specifies a preference for the output quality of fixed formats such as PDF and XPS.

UsePDFA: Specifies a preference to use the PDF/A format for the fixed format PDF.

3.1.4.4.4 Simple Types

3.1.4.4.4.1 CompatibilityMode (from namespace

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the compatibility of the output document.

```
<xs:simpleType name="CompatibilityMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Word97To2003"/>
    <xs:enumeration value="Word2007"/>
    <xs:enumeration value="Word2009"/>
    <xs:enumeration value="MaintainCurrentSetting"/>
    <xs:enumeration value="AlwaysUpgrade"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for CompatibilityMode:

| Value | Meaning |
|------------------------|---|
| Word97To2003 | Prefer document is compatible with Microsoft® Office Word 2003. |
| Word2007 | Prefer document is compatible with Microsoft® Office Word 2007. |
| Word2009 | Prefer document is compatible with Microsoft® Word 2010 Technical Preview. |
| MaintainCurrentSetting | Prefer document maintains the compatibility mode specified in the file. |
| AlwaysUpgrade | Prefer document is upgraded in compatibility to the latest Word format available. |

3.1.4.4.4.2 BalloonState (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the visibility of markup balloons in the document.

```
<xs:simpleType name="BalloonState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AlwaysUse"/>
    <xs:enumeration value="Inline"/>
    <xs:enumeration value="OnlyCommentsAndFormatting"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for BalloonState:

| Value | Meaning |
|---------------------------|--|
| AlwaysUse | Prefer that balloons always display when present. |
| Inline | Prefer that balloons appear only when inline. |
| OnlyCommentsAndFormatting | Prefer that balloons appear only in comments and formatting. |

3.1.4.4.4.3 FixedFormatBookmark (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the creation of bookmarks in fixed formats such as PDF and XPS.

```
<xs:simpleType name="FixedFormatBookmark">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Headings"/>
    <xs:enumeration value="Bookmarks"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for FixedFormatBookmark:

| Value | Meaning |
|-----------|---|
| None | Prefer not to convert source document's bookmarks or headings into bookmarks. |
| Headings | Prefer to convert source document's headings into bookmarks. |
| Bookmarks | Prefer to convert source document's bookmarks into bookmarks. |

3.1.4.4.4 FixedFormatQuality (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the output quality of fixed formats such as PDF and XPS.

```
<xs:simpleType name="FixedFormatQuality">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Standard"/>
    <xs:enumeration value="Minimum"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for FixedFormatQuality:

| Value | Meaning |
|----------|--|
| Standard | Prefer to optimize for printing. |
| Minimum | Prefer to optimize for online reading. |

3.1.4.4.5 SaveFormat (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the file format of the document after it has been converted.

```
<xs:simpleType name="SaveFormat">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Automatic"/>
    <xs:enumeration value="Document"/>
    <xs:enumeration value="DocumentMacroEnabled"/>
    <xs:enumeration value="Document97"/>
    <xs:enumeration value="Template"/>
    <xs:enumeration value="TemplateMacroEnabled"/>
    <xs:enumeration value="Template97"/>
    <xs:enumeration value="HTML"/>
    <xs:enumeration value="MHTML"/>
    <xs:enumeration value="PDF"/>
    <xs:enumeration value="RTF"/>
    <xs:enumeration value="XML"/>
    <xs:enumeration value="XPS"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for SaveFormat:

| Value | Meaning |
|----------------------|---|
| Automatic | The format of the output file is preferred to be determined by the file extension. |
| Document | The output file is preferred to be an Office Open XML document. |
| DocumentMacroEnabled | The output file is preferred to be an Office Open XML document with macros enabled. |
| Document97 | The output file is preferred to be an Microsoft® Office Word 2003 compatible binary document. |
| Template | The output file is preferred to be an Office Open XML document template. |
| TemplateMacroEnabled | The output file is preferred to be an Office Open XML document template with macros enabled. |
| Template97 | The output file is preferred to be an Office Word 2003 compatible binary document template. |
| HTML | The output file is preferred to be an HTML document. |
| MHTML | The output file is preferred to be a single file Web page. |
| PDF | The output file is preferred to be a PDF file. |
| RTF | The output file is preferred to be a Rich Text Format (RTF) document. |
| XML | The output file is preferred to be a Word XML document. |
| XPS | The output file is preferred to be an XPS file. |

3.1.4.4.4.6 SaveBehavior (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for what overwrite behavior occurs when the output file already exists.

```
<xs:simpleType name="SaveBehavior">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AppendIfPossible"/>
    <xs:enumeration value="AlwaysOverwrite"/>
    <xs:enumeration value="AppendOnly"/>
    <xs:enumeration value="NeverOverwrite"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for SaveBehavior:

| Value | Meaning |
|------------------|---|
| AppendIfPossible | Prefer to append to the file's version history if the output file already exists. If the location of the output file does not support version history, prefer to overwrite the existing file. |

| Value | Meaning |
|-----------------|--|
| AlwaysOverwrite | Prefer to always overwrite the output file if it already exists. |
| AppendOnly | Prefer to append to the file's version history if the output file already exists. If the location of the output file does not support version history, prefer to do nothing. |
| NeverOverwrite | Prefer to do nothing if there is already a file at the output file location. |

3.1.4.4.7 RevisionState (from namespace <http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions>)

This simple type specifies a preference for the visibility of revision marking and comments in the output document.

```
<xs:simpleType name="RevisionState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Final"/>
    <xs:enumeration value="FinalShowingMarkup"/>
    <xs:enumeration value="Original"/>
    <xs:enumeration value="OriginalShowingMarkup"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for RevisionState:

| Value | Meaning |
|-----------------------|---|
| Final | Prefer to show the final document without revision marking and comments. |
| FinalShowingMarkup | Prefer to show the final document with revision marking and comments. |
| Original | Prefer to show the original document without revision marking and comments. |
| OriginalShowingMarkup | Prefer to show the original document with revision marking and comments. |

3.1.4.5 Ping

This operation is used to determine the responsiveness of the service.

```
<wsdl:operation name="Ping">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/Ping"
    message="tns:IQueueApp_Ping_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/PingResponse"
    message="tns:IQueueApp_Ping_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **Ping** request WSDL message and the protocol server MUST increment a counter and MUST return a **Ping** response WSDL message, as follows:

- The **receiveTime** MUST contain the time the protocol server began the **Ping** operation after receiving the **WSDL** request message.
- The **PingResult** MUST contain the count of **Ping** operations the protocol server has performed.

- The **PingResult** MUST include the current **Ping** operation.
- The **responseTime** MUST contain the time the protocol ended the **Ping** operation before returning the WSDL response message.

3.1.4.5.1 Messages

3.1.4.5.1.1 IQueueApp_Ping_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/Ping
```

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

3.1.4.5.1.2 IQueueApp_Ping_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/Ping
```

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

3.1.4.5.2 Elements

3.1.4.5.2.1 Ping

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

```
<xs:element name="Ping"/>
```

3.1.4.5.2.2 PingResponse

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

```
<xs:element name="PingResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" name="PingResult" type="xs:int"/>
      <xs:element minOccurs="1" name="receiveTime" type="sys:DateTimeOffset"/>
      <xs:element minOccurs="1" name="responseTime" type="sys:DateTimeOffset"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

PingResult: An integer value that represents the count of **Ping** operations the protocol server has performed.

receiveTime: A [DateTimeOffset](#) value that represents the time the protocol server began the **Ping** operation.

responseTime: A [DateTimeOffset](#) value that represents the time the protocol server ended the **Ping** operation.

3.1.4.5.3 Complex Types

3.1.4.5.3.1 DateTimeOffset (from namespace <http://schemas.datacontract.org/2004/07/System>)

The **DateTimeOffset** type represents a point in time, typically expressed as a date and time of day, relative to **Coordinated Universal Time (UTC)**.

```
<xs:complexType name="DateTimeOffset">
  <xs:sequence>
    <xs:element name="DateTime" type="xs:dateTime"/>
    <xs:element name="OffsetMinutes" type="xs:short"/>
  </xs:sequence>
</xs:complexType>
```

DateTime: A **dateTime** value that represents the date and time component of the **DateTimeOffset**.

OffsetMinutes: A **short** value that represents the time's offset, in minutes, from **UTC**.

3.1.4.6 GetAllActiveJobs

This operation is used to enumerate all conversion jobs which have one or more conversion items that are not started or in progress.

```
<wsdl:operation name="GetAllActiveJobs">
  <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetAllActiveJobs"
    message="tns:IQueueApp_GetAllActiveJobs_InputMessage"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetAllActiveJobsResponse"
    message="tns:IQueueApp_GetAllActiveJobs_OutputMessage"/>
</wsdl:operation>
```

The protocol client sends a **GetAllActiveJobs** request WSDL message and the **partitionId** MUST NOT be null.

The protocol server MUST call the **proc_ConversionJobGetAllActive** stored procedure, as specified in [\[MS-WORDSSP\]](#) section 3.1.5.7. The following table specifies the parameter values for the stored procedure.

| Parameter | Value |
|--------------|--|
| @PartitionId | This MUST be the partition identifier specified by the partitionId element. |
| @UserToken | This MUST be user token specified by the userToken element, as follows: <ul style="list-style-type: none">If userToken is null, this value MUST be null.If userToken is not null, this value MUST be the value of the m_token child element. |

The protocol server MUST respond with a **GetAllActiveJobs** response WSDL message, containing all conversion job identifiers in the result set.

3.1.4.6.1 Messages

3.1.4.6.1.1 IQueueApp_GetAllActiveJobs_InputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetAllActiveJobs
```

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

3.1.4.6.1.2 IQueueApp_GetAllActiveJobs_OutputMessage

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

The SOAP action value is:

```
http://tempuri.org/IQueueApp/GetAllActiveJobs
```

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

3.1.4.6.2 Elements

3.1.4.6.2.1 GetAllActiveJobs

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

```
<xs:element name="GetAllActiveJobs">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="partitionId" type="ser:guid"/>
      <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

partitionId: A GUID value identifying the partition of the conversion job.

userToken: An [SPUserToken](#) value that represents the user credentials used to filter the list of conversion jobs.

3.1.4.6.2.2 GetAllActiveJobsResponse

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

```
<xs:element name="GetAllActiveJobsResponse">
  <xs:complexType>
    <xs:sequence>
```



```
<xs:element minOccurs="0" name="GetAllActiveJobsResult" nillable="true"
type="sera:ArrayOfguid"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

GetAllActiveJobsResult: An [ArrayOfguid](#) value containing the conversion job identifiers for all conversion jobs returned.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

4.1 Creating a Conversion Job

This example demonstrates how a protocol client adds a new conversion job to the queue.

The protocol client has five documents to convert to PDF. Each document is a single conversion item having both an input file and an output file.

The input file URLs for the five conversion items are as follows:

1. `http://server/loremipsum/Aenean nec lorem.docx`
2. `http://server/loremipsum/Fusce aliquet pede non pede.docx`
3. `http://server/loremipsum/Lorem ipsum dolor sit amet.docx`
4. `http://server/loremipsum/Nunc viverra imperdiet enim.docx`
5. `http://server/loremipsum/Pellentesque habitant morbi.docx`

The corresponding output file URLs for the five conversion items are as follows:

1. `http://server/loremipsum/Aenean nec lorem.pdf`
2. `http://server/loremipsum/Fusce aliquet pede non pede.pdf`
3. `http://server/loremipsum/Lorem ipsum dolor sit amet.pdf`
4. `http://server/loremipsum/Nunc viverra imperdiet enim.pdf`
5. `http://server/loremipsum/Pellentesque habitant morbi.pdf`

4.1.1 Request

The protocol client then sends the following request WSDL message:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:sera="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  xmlns:mowss="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service"
  xmlns:mowsc="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Convertio
ns"
>
  <s:Body>
    <CreateNewJob xmlns="http://tempuri.org/">
      <inputFiles>
        <sera:string>http://server/loremipsum/Aenean nec lorem.docx</sera:string>
        <sera:string>http://server/loremipsum/Fusce aliquet pede non pede.docx</sera:string>
        <sera:string>http://server/loremipsum/Lorem ipsum dolor sit amet.docx</sera:string>
        <sera:string>http://server/loremipsum/Nunc viverra imperdiet enim.docx</sera:string>
        <sera:string>http://server/loremipsum/Pellentesque habitant morbi.docx</sera:string>
      </inputFiles>
      <outputFiles>
        <sera:string>http://server/loremipsum/Aenean nec lorem.pdf</sera:string>
        <sera:string>http://server/loremipsum/Fusce aliquet pede non pede.pdf</sera:string>
        <sera:string>http://server/loremipsum/Lorem ipsum dolor sit amet.pdf</sera:string>
      </outputFiles>
    </CreateNewJob>
  </s:Body>
</s:Envelope>
```

```

    <sera:string>http://server/loremipsum/Nunc viverra imperdiet enim.pdf</sera:string>
    <sera:string>http://server/loremipsum/Pellentesque habitant morbi.pdf</sera:string>
  </outputFiles>
  <jobData>
    <mowss:m_rawPartitionId>0C37852B-34D0-418E-91C6-2AC25AF4BE5B</mowss:m_rawPartitionId>
    <mowss:m_settings>
      <mowsc:AddThumbnail>false</mowsc:AddThumbnail>
      <mowsc:CompatibilityMode>MaintainCurrentSetting</mowsc:CompatibilityMode>
      <mowsc:DoNotEmbedSystemFonts>true</mowsc:DoNotEmbedSystemFonts>
      <mowsc:EmbedFonts>false</mowsc:EmbedFonts>
      <mowsc:FixedFormatSettings>
        <mowsc:BalloonState>OnlyCommentsAndFormatting</mowsc:BalloonState>
        <mowsc:BitmapEmbeddedFonts>true</mowsc:BitmapEmbeddedFonts>
        <mowsc:Bookmarks>None</mowsc:Bookmarks>
        <mowsc:IncludeDocumentProperties>true</mowsc:IncludeDocumentProperties>
        <mowsc:IncludeDocumentStructure>true</mowsc:IncludeDocumentStructure>
        <mowsc:OutputQuality>Standard</mowsc:OutputQuality>
        <mowsc:UsePDFA>false</mowsc:UsePDFA>
      </mowsc:FixedFormatSettings>
      <mowsc:OutputFormat>PDF</mowsc:OutputFormat>
      <mowsc:OutputSaveBehavior>AppendIfPossible</mowsc:OutputSaveBehavior>
      <mowsc:RevisionState>FinalShowingMarkup</mowsc:RevisionState>
      <mowsc:SubsetEmbeddedFonts>false</mowsc:SubsetEmbeddedFonts>
      <mowsc:UpdateFields>false</mowsc:UpdateFields>
    </mowss:m_settings>
  </jobData>
</CreateNewJob>
</s:Body>
</s:Envelope>

```

4.1.2 Stored Procedure Call

The protocol server receives the request and assigns the conversion job an identifier of {E66580F8-1696-4B80-A571-DA8B8C59071D} and adds the conversion job and its conversion items to the database. It does this by calling the **proc_ConversionJobAdd** stored procedure using the following parameters:

| Parameter | Value |
|-----------|---|
| @JobId | E66580F8-1696-4B80-A571-DA8B8C59071D |
| @JobXml | <pre> <job xmlns="http://schemas.microsoft.com/office/server/word/2007/12/conversionJob"> <ci> <id val="1" /> <inputFile val="http://server/loremipsum/Aenean nec lorem.docx" /> <outputFile val="http://server/loremipsum/Aenean nec lorem.pdf" /> </ci> <ci> <id val="2" /> <inputFile val="http://server/loremipsum/Fusce aliquet pede non pede.docx" /> <outputFile val="http://server/loremipsum/Fusce aliquet pede non pede.pdf" /> </ci> </job> </pre> |

| Parameter | Value |
|--------------|--|
| | <pre> <id val="3" /> <inputFile val="http://server/loremipsum/Lorem ipsum dolor sit amet.docx" /> <outputFile val="http://server/loremipsum/Lorem ipsum dolor sit amet.pdf" /> </ci> <ci> <id val="4" /> <inputFile val="http://server/loremipsum/Nunc viverra imperdiet enim.docx" /> <outputFile val="http://server/loremipsum/Nunc viverra imperdiet enim.pdf" /> </ci> <ci> <id val="5" /> <inputFile val="http://server/loremipsum/Pellentesque habitant morbi.docx" /> <outputFile val="http://server/loremipsum/Pellentesque habitant morbi.pdf" /> </ci> </job> </pre> |
| @SettingsXml | <pre> <jobSettings xmlns="http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings"> <revisionState val="2" /> <updateFields val="False" /> <defaultLanguage val="" /> <embedFonts val="False" /> <subsetEmbeddedFonts val="False" /> <doNotIncludeSystemFonts val="True" /> <compatibilityMode val="0" /> <addThumbNail val="False"/> <saveFormat val="9" /> <saveBehavior val="0" /> <fixedFormatSettings> <outputQuality val="0" /> <bookmarks val="0" /> <includeDocumentProperties val="True" /> <includeDocumentStructure val="True" /> <usePDFA val="False" /> <bitmapEmbeddedFonts val="True" /> <balloonState val="2" /> </fixedFormatSettings> </jobSettings> </pre> |
| @PartitionId | 0C37852B-34D0-418E-91C6-2AC25AF4BE5B |
| @UserToken | NULL |

4.1.3 Response

The protocol server then sends the following response WSDL message:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>

```

```

    <CreateNewJobResponse xmlns="http://tempuri.org/">
      <CreateNewJobResult>E66580F8-1696-4B80-A571-DA8B8C59071D</CreateNewJobResult>
    </CreateNewJobResponse>
  </s:Body>
</s:Envelope>

```

4.2 Getting the Status of a Conversion Job

This example demonstrates how a protocol client would get the status of the conversion job added to the queue in section 4.1, which has five conversion items and a conversion job identifier of {E66580F8-1696-4B80-A571-DA8B8C59071D}.

4.2.1 Request

The protocol client sends the following request WSDL message:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetJobStatus xmlns="http://tempuri.org/">
      <jobId>E66580F8-1696-4B80-A571-DA8B8C59071D</jobId>
      <rawPartitionId>0C37852B-34D0-418E-91C6-2AC25AF4BE5B</rawPartitionId>
    </GetJobStatus>
  </s:Body>
</s:Envelope>

```

4.2.2 Stored Procedure Call

The protocol server receives the request and gets the status of the conversion items in the conversion job from the database. It does this by calling the **proc_ConversionJobStatusGet** stored procedure using the following parameters:

| Parameter | Value |
|--------------|--------------------------------------|
| @JobId | E66580F8-1696-4B80-A571-DA8B8C59071D |
| @UserToken | NULL |
| @PartitionId | 0C37852B-34D0-418E-91C6-2AC25AF4BE5B |

4.2.3 Stored Procedure Result Set

The stored procedure sets the out parameter **@Error** to a value of 0, indicating success, and returns the following result set:

| Conversion Id | InputFile | OutputFile | Message Id | State |
|---------------|---|--|------------|-------|
| 1 | http://server/loremipsum/Aenean nec lorem.docx | http://server/loremipsum/Aenean nec lorem.pdf | 0 | 3 |
| 2 | http://server/loremipsum/Fusce aliquet pede non pede.docx | http://server/loremipsum/Fusce aliquet pede non pede.pdf | 0 | 3 |
| 3 | http://server/loremipsum/Lorem | http://server/loremipsum/Lorem | 99 | 4 |

| Conversion Id | InputFile | OutputFile | Message Id | State |
|---------------|---|--|------------|-------|
| | ipsum dolor sit amet.docx | ipsum dolor sit amet.pdf | | |
| 4 | http://server/loremipsum/Nunc viverra imperdiet enim.docx | http://server/loremipsum/Nunc viverra imperdiet enim.pdf | NULL | 1 |
| 5 | http://server/loremipsum/Pellentesque habitant morbi.docx | http://server/loremipsum/Pellentesque habitant morbi.pdf | NULL | 0 |

4.2.4 Response

The protocol server then sends the following response WSDL message:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mowss="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service"
>
  <s:Body>
    <GetJobStatusResponse xmlns="http://tempuri.org/">
      <GetJobStatusResult>
        <mowss:ConversionItemInternal>
          <mowss:InputFile>http://server/loremipsum/Aenean nec lorem.docx</mowss:InputFile>
          <mowss:MessageId>0</mowss:MessageId>
          <mowss:OutputFile>http://server/loremipsum/Aenean nec lorem.pdf</mowss:OutputFile>
          <mowss:State>Succeeded</mowss:State>
        </mowss:ConversionItemInternal>
        <mowss:ConversionItemInternal>
          <mowss:InputFile>http://server/loremipsum/Fusce aliquet pede non
pede.docx</mowss:InputFile>
          <mowss:MessageId>0</mowss:MessageId>
          <mowss:OutputFile>http://server/loremipsum/Fusce aliquet pede non
pede.pdf</mowss:OutputFile>
          <mowss:State>Succeeded</mowss:State>
        </mowss:ConversionItemInternal>
        <mowss:ConversionItemInternal>
          <mowss:InputFile>http://server/loremipsum/Lorem ipsum dolor sit
amet.docx</mowss:InputFile>
          <mowss:MessageId>99</mowss:MessageId>
          <mowss:OutputFile>http://server/loremipsum/Lorem ipsum dolor sit
amet.pdf</mowss:OutputFile>
          <mowss:State>Failed</mowss:State>
        </mowss:ConversionItemInternal>
        <mowss:ConversionItemInternal>
          <mowss:InputFile>http://server/loremipsum/Nunc viverra imperdiet
enim.docx</mowss:InputFile>
          <mowss:MessageId>-1</mowss:MessageId>
          <mowss:OutputFile>http://server/loremipsum/Nunc viverra imperdiet
enim.pdf</mowss:OutputFile>
          <mowss:State>InProgress</mowss:State>
        </mowss:ConversionItemInternal>
        <mowss:ConversionItemInternal>
          <mowss:InputFile>http://server/loremipsum/Pellentesque habitant
morbi.docx</mowss:InputFile>
          <mowss:MessageId>-1</mowss:MessageId>
          <mowss:OutputFile>http://server/loremipsum/Pellentesque habitant
morbi.pdf</mowss:OutputFile>
          <mowss:State>NotStarted</mowss:State>
        </mowss:ConversionItemInternal>
      </GetJobStatusResult>
    </GetJobStatusResponse>
  </s:Body>
</s:Envelope>
```

```

        </mowss:ConversionItemInternal>
    </GetJobStatusResult>
</GetJobStatusResponse>
</s:Body>
</s:Envelope>

```

The following table is an example of how the protocol client could interpret the status of each item.

| Item | Status |
|------|---|
| 1 | <p>The conversion of input file</p> <p><code>http://server/loremipsum/Aenean nec lorem.docx</code></p> <p>to output file</p> <p><code>http://server/loremipsum/Aenean nec lorem.pdf</code></p> <p>successfully finished.</p> |
| 2 | <p>The conversion of input file</p> <p><code>http://server/loremipsum/Fusce aliquet pede non pede.docx</code></p> <p>to output file</p> <p><code>http://server/loremipsum/Fusce aliquet pede non pede.pdf</code></p> <p>successfully finished.</p> |
| 3 | <p>The conversion of input file</p> <p><code>http://server/loremipsum/Lorem ipsum dolor sit amet.docx</code></p> <p>to output file</p> <p><code>http://server/loremipsum/Lorem ipsum dolor sit amet.pdf</code></p> <p>failed with error code of 99.</p> |
| 4 | <p>The conversion of input file</p> <p><code>http://server/loremipsum/Nunc viverra imperdiet enim.docx</code></p> <p>to output file</p> <p><code>http://server/loremipsum/Nunc viverra imperdiet enim.pdf</code></p> <p>is in progress.</p> |
| 5 | <p>The conversion of input file</p> <p><code>http://server/loremipsum/Pellentesque habitant morbi.docx</code></p> <p>to output file</p> <p><code>http://server/loremipsum/Pellentesque habitant morbi.pdf</code></p> <p>has not started.</p> |

5 Security

5.1 Security Considerations for Implementers

This protocol does not introduce any additional security considerations beyond those that apply to its underlying protocols.

5.2 Index of Security Parameters

| Security Parameter | Section |
|--------------------|-------------------------|
| User Token | 2.2.4.1 |

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  name="WordServiceApplication"
  targetNamespace="http://tempuri.org/"
  xmlns:mows="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server"
  xmlns:mowsc="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions"
  xmlns:mowss="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service"
  xmlns:ser="http://schemas.microsoft.com/2003/10/Serialization/"
  xmlns:sera="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  xmlns:sp="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
  xmlns:sys="http://schemas.datacontract.org/2004/07/System"
  xmlns:tns="http://tempuri.org/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <wsdl:types>
    <xs:schema targetNamespace="http://tempuri.org/Imports">
      <xs:import namespace="http://tempuri.org/" />
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System" />
      <xs:import namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint" />
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service" />
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server" />
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions"
      />
    </xs:schema>
    <xs:schema targetNamespace="http://tempuri.org/" elementFormDefault="qualified">
      <xs:import namespace="http://schemas.datacontract.org/2004/07/System" />
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:import namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint" />
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service" />
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
      <xs:element name="Ping" />
      <xs:element name="PingResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" name="PingResult" type="xs:int" />
            <xs:element minOccurs="1" name="receiveTime" type="sys:DateTimeOffset" />
            <xs:element minOccurs="1" name="responseTime" type="sys:DateTimeOffset" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CancelJob">
        <xs:complexType>
          <xs:sequence>
```

```

        <xs:element minOccurs="1" name="jobId" type="ser:guid" />
        <xs:element minOccurs="1" name="rawPartitionId" type="ser:guid" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="CancelJobResponse" />
<xs:element name="GetJobStatus">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" name="jobId" type="ser:guid" />
            <xs:element minOccurs="1" name="rawPartitionId" type="ser:guid" />
            <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetJobStatusResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="GetJobStatusResult" nillable="true"
type="mowss:ArrayOfConversionItemInternal" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateNewJob">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" name="inputFiles" type="sera:ArrayOfstring" />
            <xs:element minOccurs="1" name="outputFiles" type="sera:ArrayOfstring" />
            <xs:element minOccurs="1" name="jobData" type="mowss:ConversionJobData" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CreateNewJobResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CreateNewJobResult" type="ser:guid" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAllJobs">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" name="partitionId" type="ser:guid" />
            <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAllJobsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="GetAllJobsResult" nillable="true"
type="sera:ArrayOfguid" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetAllActiveJobs">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" name="partitionId" type="ser:guid" />

```

```

        <xs:element minOccurs="0" name="userToken" type="sp:SPUserToken" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetAllActiveJobsResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="GetAllActiveJobsResult" nillable="true"
type="sera:ArrayOfguid" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
<xs:schema targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified">
  <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:schema>
<xs:schema targetNamespace="http://schemas.datacontract.org/2004/07/System"
elementFormDefault="qualified">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
  <xs:complexType name="DateTimeOffset">
    <xs:sequence>
      <xs:element name="DateTime" type="xs:dateTime" />
      <xs:element name="OffsetMinutes" type="xs:short" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint"
elementFormDefault="qualified">
  <xs:complexType name="SPUserToken">
    <xs:sequence>
      <xs:element name="m_token" type="xs:base64Binary" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
<xs:schema
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Service
" elementFormDefault="qualified">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server" />
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions"
/>
  <xs:import namespace="http://schemas.datacontract.org/2004/07/Microsoft.SharePoint" />
  <xs:complexType name="ArrayOfConversionItemInternal">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" name="ConversionItemInternal"
type="mowss:ConversionItemInternal" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ConversionItemInternal">
    <xs:sequence>
      <xs:element minOccurs="0" name="AppManId" type="ser:guid" />
      <xs:element minOccurs="0" name="CompleteTime" type="xs:dateTime" />

```

```

        <xs:element minOccurs="0" name="ConversionId" type="xs:unsignedInt" />
        <xs:element minOccurs="1" name="InputFile" type="xs:string" />
        <xs:element minOccurs="0" name="JobId" type="ser:guid" />
        <xs:element minOccurs="0" name="MessageDetails" type="xs:string" />
        <xs:element minOccurs="1" name="MessageId" type="xs:int" />
        <xs:element minOccurs="1" name="OutputFile" type="xs:string" />
        <xs:element minOccurs="0" name="RestartCount" type="xs:int" />
        <xs:element minOccurs="0" name="StartTime" type="xs:dateTime" />
        <xs:element minOccurs="1" name="State" type="mows:ConversionItemState" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ConversionJobData">
    <xs:sequence>
        <xs:element minOccurs="0" name="m_partitionId" type="ser:guid" />
        <xs:element minOccurs="1" name="m_rawPartitionId" type="ser:guid" />
        <xs:element minOccurs="1" name="m_settings" type="mowsc:ConversionJobSettings" />
        <xs:element minOccurs="0" name="m_userToken" type="sp:SPUserToken" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
<xs:schema
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server"
elementFormDefault="qualified">
    <xs:simpleType name="ConversionItemState">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NotStarted" />
            <xs:enumeration value="InProgress" />
            <xs:enumeration value="Canceled" />
            <xs:enumeration value="Succeeded" />
            <xs:enumeration value="Failed" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
<xs:schema targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified">
    <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    <xs:complexType name="ArrayOfstring">
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="unbounded" name="string" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ArrayOfguid">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="guid" type="ser:guid" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
<xs:schema
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Word.Server.Conversions"
elementFormDefault="qualified">
    <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    <xs:complexType name="ConversionJobSettings">
        <xs:sequence>
            <xs:element minOccurs="0" name="AddThumbnail" type="xs:boolean" />
            <xs:element minOccurs="0" name="CompatibilityMode" type="mowsc:CompatibilityMode" />
            <xs:element minOccurs="0" name="DefaultLanguage" />
            <xs:element minOccurs="0" name="DoNotEmbedSystemFonts" type="xs:boolean" />
            <xs:element minOccurs="0" name="EmbedFonts" type="xs:boolean" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

```

        <xs:element minOccurs="1" name="FixedFormatSettings"
type="mowsc:FixedFormatSettings" />
        <xs:element minOccurs="0" name="OutputFormat" type="mowsc:SaveFormat" />
        <xs:element minOccurs="0" name="OutputSaveBehavior" type="mowsc:SaveBehavior" />
        <xs:element minOccurs="0" name="RevisionState" type="mowsc:RevisionState" />
        <xs:element minOccurs="0" name="SubsetEmbeddedFonts" type="xs:boolean" />
        <xs:element minOccurs="0" name="UpdateFields" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="CompatibilityMode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Word97To2003" />
        <xs:enumeration value="Word2007" />
        <xs:enumeration value="Word2009" />
        <xs:enumeration value="MaintainCurrentSetting" />
        <xs:enumeration value="AlwaysUpgrade" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="FixedFormatSettings">
    <xs:sequence>
        <xs:element minOccurs="0" name="BalloonState" type="mowsc:BalloonState" />
        <xs:element minOccurs="0" name="BitmapEmbeddedFonts" type="xs:boolean" />
        <xs:element minOccurs="0" name="Bookmarks" type="mowsc:FixedFormatBookmark" />
        <xs:element minOccurs="0" name="IncludeDocumentProperties" type="xs:boolean" />
        <xs:element minOccurs="0" name="IncludeDocumentStructure" type="xs:boolean" />
        <xs:element minOccurs="0" name="OutputQuality" type="mowsc:FixedFormatQuality" />
        <xs:element minOccurs="0" name="UsePDFa" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="BalloonState">
    <xs:restriction base="xs:string">
        <xs:enumeration value="AlwaysUse" />
        <xs:enumeration value="Inline" />
        <xs:enumeration value="OnlyCommentsAndFormatting" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FixedFormatBookmark">
    <xs:restriction base="xs:string">
        <xs:enumeration value="None" />
        <xs:enumeration value="Headings" />
        <xs:enumeration value="Bookmarks" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FixedFormatQuality">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Standard" />
        <xs:enumeration value="Minimum" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SaveFormat">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Automatic" />
        <xs:enumeration value="Document" />
        <xs:enumeration value="DocumentMacroEnabled" />
        <xs:enumeration value="Document97" />
        <xs:enumeration value="Template" />
        <xs:enumeration value="TemplateMacroEnabled" />
        <xs:enumeration value="Template97" />
        <xs:enumeration value="HTML" />
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="MHTML" />
        <xs:enumeration value="PDF" />
        <xs:enumeration value="RTF" />
        <xs:enumeration value="XML" />
        <xs:enumeration value="XPS" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SaveBehavior">
    <xs:restriction base="xs:string">
        <xs:enumeration value="AppendIfPossible" />
        <xs:enumeration value="AlwaysOverwrite" />
        <xs:enumeration value="AppendOnly" />
        <xs:enumeration value="NeverOverwrite" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RevisionState">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Final" />
        <xs:enumeration value="FinalShowingMarkup" />
        <xs:enumeration value="Original" />
        <xs:enumeration value="OriginalShowingMarkup" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
</wsdl:types>
<wsdl:message name="IQueueApp_Ping_InputMessage">
    <wsdl:part name="parameters" element="tns:Ping" />
</wsdl:message>
<wsdl:message name="IQueueApp_Ping_OutputMessage">
    <wsdl:part name="parameters" element="tns:PingResponse" />
</wsdl:message>
<wsdl:message name="IQueueApp_CancelJob_InputMessage">
    <wsdl:part name="parameters" element="tns:CancelJob" />
</wsdl:message>
<wsdl:message name="IQueueApp_CancelJob_OutputMessage">
    <wsdl:part name="parameters" element="tns:CancelJobResponse" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetJobStatus_InputMessage">
    <wsdl:part name="parameters" element="tns:GetJobStatus" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetJobStatus_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetJobStatusResponse" />
</wsdl:message>
<wsdl:message name="IQueueApp_CreateNewJob_InputMessage">
    <wsdl:part name="parameters" element="tns:CreateNewJob" />
</wsdl:message>
<wsdl:message name="IQueueApp_CreateNewJob_OutputMessage">
    <wsdl:part name="parameters" element="tns:CreateNewJobResponse" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetAllJobs_InputMessage">
    <wsdl:part name="parameters" element="tns:GetAllJobs" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetAllJobs_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetAllJobsResponse" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetAllActiveJobs_InputMessage">
    <wsdl:part name="parameters" element="tns:GetAllActiveJobs" />
</wsdl:message>
<wsdl:message name="IQueueApp_GetAllActiveJobs_OutputMessage">

```

```

        <wsdl:part name="parameters" element="tns:GetAllActiveJobsResponse" />
    </wsdl:message>
    <wsdl:portType name="IQueueApp">
        <wsdl:operation name="Ping">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/Ping"
message="tns:IQueueApp_Ping_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/PingResponse"
message="tns:IQueueApp_Ping_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="CancelJob">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/CancelJob"
message="tns:IQueueApp_CancelJob_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/CancelJobResponse"
message="tns:IQueueApp_CancelJob_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="GetJobStatus">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetJobStatus"
message="tns:IQueueApp_GetJobStatus_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetJobStatusResponse"
message="tns:IQueueApp_GetJobStatus_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="CreateNewJob">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/CreateNewJob"
message="tns:IQueueApp_CreateNewJob_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/CreateNewJobResponse"
message="tns:IQueueApp_CreateNewJob_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="GetAllJobs">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetAllJobs"
message="tns:IQueueApp_GetAllJobs_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetAllJobsResponse"
message="tns:IQueueApp_GetAllJobs_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="GetAllActiveJobs">
            <wsdl:input wsaw:Action="http://tempuri.org/IQueueApp/GetAllActiveJobs"
message="tns:IQueueApp_GetAllActiveJobs_InputMessage" />
            <wsdl:output wsaw:Action="http://tempuri.org/IQueueApp/GetAllActiveJobsResponse"
message="tns:IQueueApp_GetAllActiveJobs_OutputMessage" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="WordServer-Endpoint-Queue-http" type="tns:IQueueApp">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="Ping">
            <soap:operation soapAction="http://tempuri.org/IQueueApp/Ping" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="CancelJob">
            <soap:operation soapAction="http://tempuri.org/IQueueApp/CancelJob" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

```

```

<wsdl:operation name="GetJobStatus">
  <soap:operation soapAction="http://tempuri.org/IQueueApp/GetJobStatus" style="document"
/>
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateNewJob">
  <soap:operation soapAction="http://tempuri.org/IQueueApp/CreateNewJob" style="document"
/>
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllJobs">
  <soap:operation soapAction="http://tempuri.org/IQueueApp/GetAllJobs" style="document"
/>
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllActiveJobs">
  <soap:operation soapAction="http://tempuri.org/IQueueApp/GetAllActiveJobs"
style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="WordServer-Endpoint-Queue-net.tcp" type="tns:IQueueApp">
  <soap12:binding transport="http://schemas.microsoft.com/soap/tcp" />
  <wsdl:operation name="Ping">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/Ping" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CancelJob">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/CancelJob" style="document"
/>
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>

```



```

        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetJobStatus">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/GetJobStatus"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateNewJob">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/CreateNewJob"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllJobs">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/GetAllJobs" style="document"
/>
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAllActiveJobs">
    <soap12:operation soapAction="http://tempuri.org/IQueueApp/GetAllActiveJobs"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>

```

7 Appendix B: Full Conversion Job Settings XML Schema

For ease of implementation, the full XML Schema specified in section 2.3 is provided as follows.

```
<?xml version="1.0"?>
<xs:schema
  targetNamespace="http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings"
  xmlns:cs="http://schemas.microsoft.com/office/server/word/2007/12/conversionSettings"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
>
  <xs:simpleType name="ST_Boolean">
    <xs:restriction base="xs:string">
      <xs:enumeration value="True" />
      <xs:enumeration value="False" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CT_Boolean">
    <xs:attribute name="val" use="required" type="cs:ST_Boolean" />
  </xs:complexType>
  <xs:simpleType name="ST_RevisionState">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="3" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CT_RevisionState">
    <xs:attribute name="val" use="required" type="cs:ST_RevisionState" />
  </xs:complexType>
  <xs:simpleType name="ST_Empty">
    <xs:restriction base="xs:string">
      <xs:enumeration value="" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_DefaultLanguage">
    <xs:union memberTypes="xs:integer cs:ST_Empty" />
  </xs:simpleType>
  <xs:complexType name="CT_DefaultLanguage">
    <xs:attribute name="val" use="required" type="cs:ST_DefaultLanguage" />
  </xs:complexType>
  <xs:simpleType name="ST_CompatibilityMode">
    <xs:restriction base="xs:integer">
      <xs:enumeration value="-1" />
      <xs:enumeration value="0" />
      <xs:enumeration value="11" />
      <xs:enumeration value="12" />
      <xs:enumeration value="14" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CT_CompatibilityMode">
    <xs:attribute name="val" use="required" type="cs:ST_CompatibilityMode" />
  </xs:complexType>
  <xs:simpleType name="ST_SaveFormat">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="11" />
    </xs:restriction>
  </xs:simpleType>
```

```

    </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_SaveFormat">
  <xs:attribute name="val" use="required" type="cs:ST_SaveFormat" />
</xs:complexType>
<xs:simpleType name="ST_SaveBehavior">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="3" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_SaveBehavior">
  <xs:attribute name="val" use="required" type="cs:ST_SaveBehavior" />
</xs:complexType>
<xs:simpleType name="ST_OutputQuality">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_OutputQuality">
  <xs:attribute name="val" use="required" type="cs:ST_OutputQuality" />
</xs:complexType>
<xs:simpleType name="ST_Bookmarks">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="2" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_Bookmarks">
  <xs:attribute name="val" use="required" type="cs:ST_Bookmarks" />
</xs:complexType>
<xs:simpleType name="ST_BalloonState">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="2" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_BalloonState">
  <xs:attribute name="val" use="required" type="cs:ST_BalloonState" />
</xs:complexType>
<xs:complexType name="CT_FixedFormatSettings">
  <xs:sequence>
    <xs:element name="outputQuality" type="cs:CT_OutputQuality" />
    <xs:element name="bookmarks" type="cs:CT_Bookmarks" />
    <xs:element name="includeDocumentProperties" type="cs:CT_Boolean" />
    <xs:element name="includeDocumentStructure" type="cs:CT_Boolean" />
    <xs:element name="usePDFa" type="cs:CT_Boolean" />
    <xs:element name="bitmapEmbeddedFonts" type="cs:CT_Boolean" />
    <xs:element name="balloonState" type="cs:CT_BalloonState" />
  </xs:sequence>
</xs:complexType>
<xs:element name="jobSettings">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="revisionState" type="cs:CT_RevisionState" />
      <xs:element name="updateFields" type="cs:CT_Boolean" />
      <xs:element name="defaultLanguage" type="cs:CT_DefaultLanguage" />
      <xs:element name="embedFonts" type="cs:CT_Boolean" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
<xs:element name="subsetEmbeddedFonts" type="cs:CT_Boolean" />
<xs:element name="doNotIncludeSystemFonts" type="cs:CT_Boolean" />
<xs:element name="compatibilityMode" type="cs:CT_CompatibilityMode" />
<xs:element name="addThumbNail" type="cs:CT_Boolean" />
<xs:element name="saveFormat" type="cs:CT_SaveFormat" />
<xs:element name="saveBehavior" type="cs:CT_SaveBehavior" />
<xs:element name="fixedFormatSettings" type="cs:CT_FixedFormatSettings" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

8 Appendix C: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft® Office 2010 Technical Preview

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

[<1> Section 2.3.2.1:](#) Office 2010 Technical Preview ignores preferences that cannot be used in combination with each other. For example, if the **embedFonts** setting is false, the **doNotIncludeSystemFonts** setting is ignored.

[<2> Section 2.3.4.5:](#) Office 2010 Technical Preview uses an integer value instead of an empty string. Implementers should ignore this value in Office 2010 Technical Preview as Office 2010 will represent this information differently.

[<3> Section 3.1.4.4.3.3:](#) Office 2010 Technical Preview ignores preferences that cannot be used in combination with each other. For example, if the **EmbedFonts** setting is false, the **DoNotEmbedSystemFonts** setting is ignored.

[<4> Section 3.1.4.4.3.3:](#) Office 2010 Technical Preview uses values other than null. Implementers should ignore this element as Office 2010 will implement this feature differently than Office 2010 Technical Preview.

[<5> Section 3.1.4.4.3.4:](#) Office 2010 Technical Preview ignores preferences that cannot be used in combination with each other. For example, if the **UsePDFA** setting is false, the **BitmapEmbeddedFonts** setting is ignored and fonts that cannot be embedded are always bitmapped.

9 Change Tracking

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

10 Index

A

[Applicability](#) 11
Attribute Groups ([section 2.2.8](#) 15, [section 2.3.7](#) 27)
Attributes ([section 2.2.6](#) 15, [section 2.3.5](#) 27)

C

[Capability Negotiation](#) 11
[Change tracking](#) 70
Complex types ([section 2.2.4](#) 14, [section 2.3.3](#) 17)

D

[Details](#) 28

E

Examples
[overview](#) 50

F

Fields
[vendor-extensible](#) 11

G

[Glossary](#) 8
Groups ([section 2.2.7](#) 15, [section 2.3.6](#) 27)

I

Implementer
[security considerations](#) 56
[Index of security parameters](#) 56
[Informative references](#) 9
[Introduction](#) 8

M

Messages
[attribute groups](#) 15
[attributes](#) 15
[complex types](#) 14
[elements](#) 14
[groups](#) 15
[namespaces](#) 13
[simple types](#) 15
[syntax](#) 13
[transport](#) 13
[XML structures](#) 15
Messages overview ([section 2](#) 13, [section 2.2.2](#) 14)

N

Namespaces ([section 2.2.1](#) 13, [section 2.3.1](#) 15)
[Normative references](#) 8

O

[Overview](#) 9

P

Parameters
[security index](#) 56
[Preconditions](#) 11
[Prerequisites](#) 11
[Product behavior](#) 69

R

References
[informative](#) 9
[normative](#) 8
[overview](#) 8
[Relationship to other protocols](#) 10

S

Security
[implementer considerations](#) 56
[overview](#) 56
[parameter index](#) 56
Simple types ([section 2.2.5](#) 15, [section 2.3.4](#) 21)
[Standards assignments](#) 12
Structures
[XML](#) 15
[Syntax](#) 13

T

[Tracking changes](#) 70
[Transport](#) 13
Types
complex ([section 2.2.4](#) 14, [section 2.3.3](#) 17)
simple ([section 2.2.5](#) 15, [section 2.3.4](#) 21)

V

[Vendor-extensible fields](#) 11
[Versioning](#) 11

W

[WSDL](#) 57
[full](#) 57

X

[XML structures](#) 15
[attribute groups](#) 27
[attributes](#) 27
[complex types](#) 17
[elements](#) 16
[groups](#) 27

[namespaces](#) 15
[simple types](#) 21

Preliminary