

# [MS-BDSRR]: Business Document Scanning: Scan Repository Capabilities and Status Retrieval 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 [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard

specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
12/05/2008	0.1	Major	Initial Availability
01/16/2009	0.1.1	Editorial	Revised and edited the technical content.
02/27/2009	0.2	Minor	Updated the technical content.
04/10/2009	1.0	Major	Updated and revised the technical content.
05/22/2009	2.0	Major	Updated and revised the technical content.
07/02/2009	3.0	Major	Updated and revised the technical content.
08/14/2009	3.1	Minor	Updated the technical content.
09/25/2009	4.0	Major	Updated and revised the technical content.
11/06/2009	5.0	Major	Updated and revised the technical content.
12/18/2009	5.0.1	Editorial	Revised and edited the technical content.
01/29/2010	5.1	Minor	Updated the technical content.
03/12/2010	6.0	Major	Updated and revised the technical content.
04/23/2010	6.0.1	Editorial	Revised and edited the technical content.
06/04/2010	6.0.2	Editorial	Revised and edited the technical content.
07/16/2010	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	6.0.2	No change	No changes to the meaning, language, or formatting of the technical content.

# Contents

<b>1 Introduction</b>	<b>8</b>
1.1 Glossary	8
1.2 References	9
1.2.1 Normative References	9
1.2.2 Informative References	10
1.3 Overview	10
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
<b>2 Messages</b>	<b>13</b>
2.1 Transport	13
2.2 Common Message Syntax	13
2.2.1 Namespaces	13
2.2.2 Messages	14
2.2.2.1 Error Codes	14
2.2.3 Elements	15
2.2.3.1 dsc:Dialect	17
2.2.3.1.1 dsc:Dialect, Child of dsc:FilterStatusType	17
2.2.3.1.2 dsc:Dialect, Child of dsc:FilterType	17
2.2.3.2 dsc:Document	17
2.2.3.3 dsc:Documents	17
2.2.3.4 dsc:DocumentDescription	17
2.2.3.5 dsc:DocumentId	18
2.2.3.6 dsc:Filter	18
2.2.3.7 dsc:FilterCompletedStatuses	18
2.2.3.8 dsc:FilterConfig	18
2.2.3.9 dsc:Filters	18
2.2.3.10 dsc:FilterState	18
2.2.3.11 dsc:FilterStateReason	18
2.2.3.12 dsc:FilterStateReasons	19
2.2.3.13 dsc:FilterStatus	19
2.2.3.14 dsc:FilterStatuses	19
2.2.3.15 dsc:Format	19
2.2.3.16 dsc:ImagesReceived	19
2.2.3.17 dsc:JobCompletedState	19
2.2.3.18 dsc:JobCompletedStateReasons	20
2.2.3.19 dsc:JobCompletedTime	20
2.2.3.20 dsc:JobCreatedTime	20
2.2.3.21 dsc:JobDescription	20
2.2.3.22 dsc:JobOriginatingUserName	20
2.2.3.23 dsc:JobState	20
2.2.3.24 dsc:JobStateReason	21
2.2.3.25 dsc:JobStateReasons	21
2.2.3.26 dsc:JobStatus	21
2.2.3.27 dsc:JobSummary	21
2.2.3.28 dsc:JobToken	21

2.2.3.29	dsc:PSP_DisplayName .....	21
2.2.3.30	dsc:PSP_Identifier.....	22
2.2.3.31	dsc:RepositoryConfiguration.....	22
2.2.3.32	dsc:RepositoryState .....	22
2.2.3.33	dsc:RepositoryStateReason .....	22
2.2.3.34	dsc:RepositoryStateReasons .....	22
2.2.3.35	dsc:RepositoryStatus .....	22
2.2.4	Complex Types .....	23
2.2.4.1	psp:anyURIExtType .....	24
2.2.4.2	dsc:DateTimeExtType .....	24
2.2.4.3	dsc:DocumentDescriptionType .....	25
2.2.4.4	dsc:DocumentFormatExtType.....	25
2.2.4.5	dsc:DocumentType .....	25
2.2.4.6	dsc:DocumentsType.....	25
2.2.4.7	dsc:FilterConfigType .....	26
2.2.4.8	dsc:FilterStateReasonsType .....	26
2.2.4.9	dsc:FilterStateReasonType.....	26
2.2.4.10	dsc:FilterStateType .....	26
2.2.4.11	dsc:FilterStatusesType .....	27
2.2.4.12	dsc:FilterStatusType.....	27
2.2.4.13	dsc:FiltersType .....	27
2.2.4.14	dsc:FilterType .....	27
2.2.4.15	dsc:IntOneExtType .....	28
2.2.4.16	dsc:IntZeroExtType.....	28
2.2.4.17	dsc:ListOfSummarysType.....	28
2.2.4.18	dsc:JobDescriptionType .....	28
2.2.4.19	dsc:JobStateReasonsType .....	29
2.2.4.20	dsc:JobStateReasonType .....	29
2.2.4.21	dsc:JobStateType .....	29
2.2.4.22	dsc:JobStatusType .....	29
2.2.4.23	dsc:JobSummaryType .....	30
2.2.4.24	dsc:RepositoryConfigurationType .....	30
2.2.4.25	dsc:RepositoryStateType .....	30
2.2.4.26	dsc:RepositoryStateReasonsType.....	31
2.2.4.27	dsc:RepositoryStateReasonType .....	31
2.2.4.28	dsc:RepositoryStatusType .....	31
2.2.4.29	dsc:String255ExtType .....	31
2.2.5	Simple Types .....	32
2.2.5.1	dsc:DocumentFormatBaseType .....	33
2.2.5.2	dsc:DocumentFormatWKVType.....	33
2.2.5.3	dsc:FilterStateBaseType .....	34
2.2.5.4	dsc:FilterStateReasonBaseType .....	35
2.2.5.5	dsc:FilterStateReasonWKVType .....	35
2.2.5.6	dsc:FilterStateWKVType .....	36
2.2.5.7	dsc:IntOneBaseType .....	36
2.2.5.8	dsc:IntZeroBaseType .....	37
2.2.5.9	dsc:JobStateBaseType.....	37
2.2.5.10	dsc:JobStateReasonBaseType.....	37
2.2.5.11	dsc:JobStateReasonWKVType.....	37
2.2.5.12	dsc:JobStateWKVType .....	39
2.2.5.13	dsc:KeywordNsExtensionPatternType .....	39
2.2.5.14	dsc:MimeExtensionPatternType .....	40
2.2.5.15	dsc:RepositoryStateBaseType.....	40

2.2.5.16	dsc:RepositoryStateReasonBaseType .....	40
2.2.5.17	dsc:RepositoryStateReasonWKVType.....	40
2.2.5.18	dsc:RepositoryStateWKVType.....	41
2.2.5.19	dsc:String255BaseType .....	41
2.2.6	Attributes.....	41
2.2.6.1	Name.....	42
2.2.6.2	Valid.....	42
2.2.7	Groups.....	42
2.2.8	Attribute Groups .....	42
<b>3</b>	<b>Protocol Details.....</b>	<b>43</b>
3.1	Common Details .....	43
3.1.1	Abstract Data Model .....	43
3.1.2	Timers .....	43
3.1.3	Initialization .....	43
3.1.4	Message Processing Events and Sequencing Rules.....	43
3.1.4.1	Common SOAP Faults.....	43
3.1.4.1.1	wsa:ActionNotSupported .....	43
3.1.4.1.2	dsc:InvalidArgs.....	44
3.1.4.1.3	dsc:OperationFailed.....	44
3.1.4.1.4	dsc:ServerErrorTemporaryError .....	44
3.1.4.1.5	dsc:ServerErrorInternalError .....	44
3.1.4.1.6	dsc:ClientErrorJobTokenNotFound .....	45
3.2	RepositoryConfigurationServiceType Server Details.....	45
3.2.1	Abstract Data Model .....	45
3.2.2	Timers .....	47
3.2.3	Initialization .....	47
3.2.4	Message Processing Events and Sequencing Rules.....	47
3.2.4.1	GetRepositoryElements .....	47
3.2.4.1.1	Messages.....	48
3.2.4.1.1.1	GetRepositoryElementsRequestMsg .....	49
3.2.4.1.1.2	GetRepositoryElementsResponseMsg .....	49
3.2.4.1.2	Elements.....	49
3.2.4.1.2.1	dsc:ElementData .....	50
3.2.4.1.2.2	dsc:GetRepositoryElementsRequest .....	50
3.2.4.1.2.3	dsc:GetRepositoryElementsResponse .....	50
3.2.4.1.2.4	dsc:Name.....	50
3.2.4.1.2.5	dsc:RepositoryElements .....	50
3.2.4.1.2.6	dsc:RequestedElements .....	51
3.2.4.1.3	Complex Types .....	51
3.2.4.1.3.1	dsc:GetRepositoryElementsRequestType .....	51
3.2.4.1.3.2	dsc:GetRepositoryElementsResponseType .....	52
3.2.4.1.3.3	dsc:RepositoryElementDataType .....	52
3.2.4.1.3.4	dsc:RepositoryElementsType.....	52
3.2.4.1.3.5	dsc:RequestedRepositoryElementsType.....	53
3.2.4.2	CancelPostScanJob .....	53
3.2.4.2.1	Messages.....	54
3.2.4.2.1.1	CancelPostScanJobRequestMsg .....	54
3.2.4.2.1.2	CancelPostScanJobResponseMsg .....	54
3.2.4.2.2	Elements.....	55
3.2.4.2.2.1	CancelPostScanJobRequest .....	55
3.2.4.2.2.2	CancelPostScanJobResponse .....	55
3.2.4.2.3	Complex Types .....	55

3.2.4.2.3.1	CancelPostScanJobRequestType .....	55
3.2.4.2.3.2	CancelPostScanJobResponseType .....	56
3.2.4.3	GetPostScanJobElements .....	56
3.2.4.3.1	Messages .....	57
3.2.4.3.1.1	GetPostScanJobElementsRequestMsg .....	57
3.2.4.3.1.2	GetPostScanJobElementsResponseMsg .....	57
3.2.4.3.2	Elements.....	58
3.2.4.3.2.1	dsc:ElementData .....	58
3.2.4.3.2.2	dsc:GetPostScanJobElementsRequest.....	58
3.2.4.3.2.3	dsc:GetPostScanJobElementsResponse.....	59
3.2.4.3.2.4	dsc:Name .....	59
3.2.4.3.2.5	dsc:JobElements .....	59
3.2.4.3.2.6	dsc:RequestedElements .....	59
3.2.4.3.3	Complex Types .....	59
3.2.4.3.3.1	dsc:GetPostScanJobElementsRequestType .....	60
3.2.4.3.3.2	dsc:GetPostScanJobElementsResponseType .....	60
3.2.4.3.3.3	dsc:JobElementDataType.....	61
3.2.4.3.3.4	dsc:JobElementsType.....	61
3.2.4.3.3.5	dsc:RequestedJobElementsType .....	61
3.2.4.4	GetActiveJobs .....	62
3.2.4.4.1	Messages .....	62
3.2.4.4.1.1	GetActiveJobsRequestMsg.....	63
3.2.4.4.1.2	GetActiveJobsResponseMsg .....	63
3.2.4.4.2	Elements.....	63
3.2.4.4.2.1	dsc:ActiveJobs .....	63
3.2.4.4.2.2	dsc:GetActiveJobsRequest .....	64
3.2.4.4.2.3	dsc:GetActiveJobsResponse .....	64
3.2.4.4.3	Complex Types .....	64
3.2.4.4.3.1	dsc:GetActiveJobsRequestType .....	64
3.2.4.4.3.2	dsc:GetActiveJobsResponseType .....	64
3.2.4.5	GetJobHistory .....	65
3.2.4.5.1	Messages .....	65
3.2.4.5.1.1	GetJobHistoryRequestMsg.....	65
3.2.4.5.1.2	GetJobHistoryResponseMsg.....	66
3.2.4.5.2	Elements.....	66
3.2.4.5.2.1	dsc:JobHistory .....	66
3.2.4.5.2.2	dsc:GetJobHistoryRequest .....	67
3.2.4.5.2.3	dsc:GetJobHistoryResponse .....	67
3.2.4.5.3	Complex Types .....	67
3.2.4.5.3.1	dsc:GetJobHistoryRequestType .....	67
3.2.4.5.3.2	dsc:GetJobHistoryResponseType .....	67
3.2.4.6	RepositoryElementsChangeEvent .....	68
3.2.4.6.1	Messages .....	68
3.2.4.6.1.1	RepositoryElementsChangeEventMsg .....	68
3.2.4.6.2	Elements.....	69
3.2.4.6.2.1	dsc:ElementChanges.....	69
3.2.4.6.2.2	dsc:RepositoryElementsChangeEvent .....	69
3.2.4.6.3	Complex Types .....	69
3.2.4.6.3.1	dsc:ElementChangesType .....	69
3.2.4.6.3.2	dsc:RepositoryElementsChangeEventType .....	70
3.2.4.7	RepositoryStatusSummaryEvent.....	70
3.2.4.7.1	Messages .....	71
3.2.4.7.1.1	dsc:RepositoryStatusSummaryEventMsg .....	71

3.2.4.7.2	Elements.....	71
3.2.4.7.2.1	dsc:RepositoryStatusSummaryEvent.....	71
3.2.4.7.2.2	dsc:StatusSummary .....	71
3.2.4.7.3	Complex Types .....	72
3.2.4.7.3.1	dsc:RepositoryStatusSummaryEventType .....	72
3.2.4.7.3.2	dsc:StatusSummaryType.....	72
3.2.4.8	PostScanJobStatusEvent.....	72
3.2.4.8.1	Messages .....	73
3.2.4.8.1.1	dsc:PostScanJobStatusEventMsg .....	73
3.2.4.8.2	Elements.....	73
3.2.4.8.2.1	dsc:PostScanJobStatusEvent.....	73
3.2.4.8.3	Complex Types .....	74
3.2.4.8.3.1	dsc:PostScanJobStatusEventType .....	74
3.2.4.9	PostScanJobEndStateEvent .....	74
3.2.4.9.1	Messages .....	75
3.2.4.9.1.1	dsc:PostScanJobEndStateEventMsg .....	75
3.2.4.9.2	Elements.....	75
3.2.4.9.2.1	dsc:PostScanJobEndState .....	75
3.2.4.9.2.2	dsc:PostScanJobEndStateEvent .....	75
3.2.4.9.3	Complex Types .....	76
3.2.4.9.3.1	dsc:PostScanJobEndStateType .....	76
3.2.4.9.3.2	dsc:PostScanJobEndStateEventType .....	76
3.2.5	Timer Events .....	77
3.2.6	Other Local Events .....	77
3.3	RepositoryConfigurationServiceType Client Details .....	77
<b>4</b>	<b>Protocol Examples.....</b>	<b>78</b>
4.1	WSDL Message Examples .....	78
4.2	SOAP Fault Example.....	83
<b>5</b>	<b>Security.....</b>	<b>84</b>
5.1	Security Considerations for Implementers.....	84
5.2	Index of Security Parameters .....	84
<b>6</b>	<b>Appendix A: Full WSDL.....</b>	<b>85</b>
6.1	RepositoryConfigurationServiceType WSDL .....	85
6.2	Full XSD .....	89
6.2.1	dsc.....	89
6.2.2	psp.....	105
6.2.3	wscn.....	107
<b>7</b>	<b>Appendix B: Product Behavior.....</b>	<b>116</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>118</b>
<b>9</b>	<b>Index .....</b>	<b>119</b>

# 1 Introduction

This is a specification of the Business Document Scanning: Scan Repository Capabilities and Status Retrieval (BDSRR) Protocol. This protocol is used to query a **server** for the capabilities and status of the **scan repository**.

## 1.1 Glossary

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

**certificate**  
**client**  
**globally unique identifier (GUID)**  
**server**  
**SOAP action**  
**SOAP body**  
**SOAP fault**  
**SOAP message**  
**Uniform Resource Locator (URL)**  
**URI**  
**Web Services Description Language (WSDL)**  
**WSDL message**  
**WSDL operation**  
**XML**  
**XML namespace**  
**XML schema (XSD)**

The following terms are specific to this document:

**filter:** A processing object in the **scan repository** identified by **scan documents** that is a unique **URI**. This object will perform a specific operation in a **PostScan job**. The operations performed by each **filter** are governed by the **filter** dialect **URI** and **filter-specific instructions**.

**filter-specific instructions:** A set of data elements that tell a specific **filter** how to process the **scan document(s)** that are part of a **PostScan job**. The definition of these data elements is specific to each **filter**.

**PostScan job:** A unit of work within the **scan repository**.

**PostScan process:** The entity that contains a **scan ticket** used by a **scan device** and the PostScan instructions used by a **scan repository server** when processing image data sent by a **scan device**.

**scan device:** A scanner, copier, or multifunction peripheral that supports the Devices Profile for Web Services [\[DPWS\]](#).

**scan document:** A single image file created by a **scan device** and transferred to the **scan repository server** during the processing of a **PostScan job**.

**scan repository:** A service that supports processing **PostScan jobs** based on data and instructions in a **PostScan process**.



**scan ticket:** An element that communicates the appropriate settings that should be used by a **scan device** when creating a **scan document**.

**WSDL Event:** A one-way **WSDL operation** from the **server** to the **client**.

**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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[DPWS] Chans, S., Conti, D., Schlimmer, J., et al., "Devices Profile for Web Services", February 2006, <http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf>

[IANAPORT] Internet Assigned Numbers Authority, "Port Numbers", November 2006, <http://www.iana.org/assignments/port-numbers>

[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>

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, <http://www.ietf.org/rfc/rfc2396.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>

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

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

[WSAddressing] Box, D., Christensen, E., Ferguson, D., et al., "Web Services Addressing (WS-Addressing)", August 2004, <http://www.w3.org/Submission/ws-addressing/>

If you have any trouble finding [WSAddressing], please check [here](#).

[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>

[WSEVENTING] Box, D., Cabrera, L.F., Critchley, C., et al., "Web Services Eventing (WS-Eventing)", March 2006, <http://www.w3.org/Submission/WS-Eventing>

[WSPOLICY] Bajaj, S., Box, D., Chappell, D., et al., "Web Services Policy Framework (WS-POLICY) and Web Services Policy Attachment (WS-PolicyAttachment)", March 2006, <http://schemas.xmlsoap.org/ws/2004/09/policy>

[WSS1] Nadalin, A., Kaler, C., Hallam-Baker, P., et al., "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", March 2004, <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

[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-ADLS] Microsoft Corporation, "[Active Directory Lightweight Directory Services Schema](#)", June 2007.

[MS-ADSC] Microsoft Corporation, "[Active Directory Schema Classes](#)", June 2007.

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

[MSDN-SPS] Microsoft Corporation, "Scan Process Schema", <http://msdn.microsoft.com/en-us/library/ff547951.aspx>

[MSDN-WSDSP] Microsoft Corporation, "Distributed Scan Processing Web Service Protocol Summary", <http://msdn.microsoft.com/en-us/library/ff540624.aspx>

[MSDN-WSSCAN] Microsoft Corporation, "Scan Service (WS-Scan) Schema", <http://msdn.microsoft.com/en-us/library/ff547963.aspx>

[WS-Discovery] Beatty, J., Kakivaya, G., Kemp D., et al., "Web Services Dynamic Discovery (WS-Discovery)", April 2005, <http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>

If you have any trouble finding [WS-Discovery], please check [here](#).

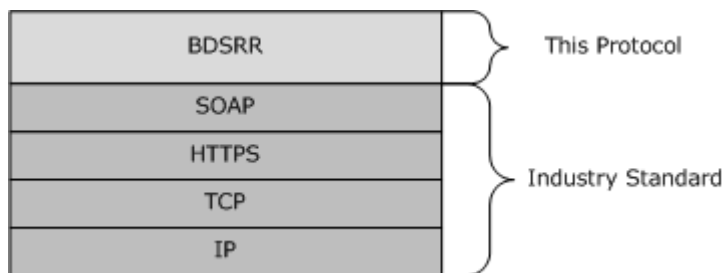
## 1.3 Overview

The Business Document Scanning: Scan Repository Capabilities and Status Retrieval (BDSRR) Protocol is modeled as a **client**, a scan repository server, and the conversations between those elements that support document scanning, processing, and storage in a business or enterprise setting. The BDSRR protocol is a means by which a client application can determine the current status and configuration of a scan repository server. The protocol contains operations and events that describe which **filters** are supported by a scan repository and the state of the repository, in addition to information about active and completed **PostScan jobs**.

## 1.4 Relationship to Other Protocols

The BDSRR protocol uses the **SOAP messages** protocol for formatting requests and responses as specified in [\[SOAP1.2-1/2003\]](#) and [\[SOAP1.2-2/2003\]](#). The BDSRR protocol transmits these messages using the HTTPS protocol as specified in [\[RFC2818\]](#).

The following diagram shows the underlying messaging and [transport](#) stack that the protocol uses:



**Figure 1: The BDSRR messaging and transport stack**

[\[RFC2396\]](#) is used to define a **Uniform Resource Identifier (URI)**, which is used as the basis for the [dsc:Dialect](#) elements.

## 1.5 Prerequisites/Preconditions

It is assumed that a client of this protocol has obtained the name of a scan repository that supports this protocol before it is invoked. The **PostScan process** is typically defined by a user or administration from a client and is stored in a directory service. There are various ways a client can accomplish this. For information, see [\[MS-ADLS\]](#), [\[MS-ADSC\]](#), and [\[WS-Discovery\]](#).

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

## 1.6 Applicability Statement

The Business Document Scanning: Scan Repository Capabilities and Status Retrieval (BDSRR) Protocol is applicable only for operations between a system functioning as a client and a system functioning as a scan repository server. This protocol is intended to scale from office use, in which a scan repository is shared between a small group of users, to enterprise use, in which multiple scan repositories are employed in a distributed environment.

## 1.7 Versioning and Capability Negotiation

The BDSRR protocol has no versioning mechanism or means of negotiating capabilities.

## 1.8 Vendor-Extensible Fields

The BDSRR protocol contains a number of vendor-extensible fields. These fields use an extension mechanism based on **XML** type `xs:NMTOKEN`. Each of these elements can be extended by using either of a pair of XML pattern definitions. The definitions, which allow keywords from additional namespaces, are defined as follows.

```
<xs:simpleType name="KeywordNsExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+:[\w_\-\.]+"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="MimeExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+[\w;/;_\-\|\\\+\.]*"/>
  </xs:restriction>
</xs:simpleType>
```

The following table lists all the elements that support these extension patterns.<1>

Simple Type	Reference
dsc:DocumentFormatBaseType	Section <a href="#">2.2.5.1</a>
dsc:FilterStateBaseType	Section <a href="#">2.2.5.3</a>
dsc:FilterStateReasonBaseType	Section <a href="#">2.2.5.4</a>
dsc:JobStateBaseType	Section <a href="#">2.2.5.9</a>
dsc:JobStateReasonBaseType	Section <a href="#">2.2.5.10</a>
dsc:RepositoryStateBaseType	Section <a href="#">2.2.5.15</a>
dsc:RepositoryStateReasonBaseType	Section <a href="#">2.2.5.16</a>

## 1.9 Standards Assignments

Parameter	Value	Reference
UDP Port	5362	<a href="#">[IANAPORT]</a>
TCP Port	5362	<a href="#">[IANAPORT]</a>

## 2 Messages

### 2.1 Transport

The BDSRR protocol can be used over any transport protocol that supports transmitting messages specified in [\[SOAP1.2-1/2003\]](#) and [\[SOAP1.2-2/2003\]](#).

### 2.2 Common Message Syntax

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

#### 2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although the BDSRR protocol 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
wSDL	<a href="http://schemas.xmlsoap.org/wSDL/">http://schemas.xmlsoap.org/wSDL/</a>	<a href="#">[WSDL]</a>
wSOAP1.2	<a href="http://schemas.xmlsoap.org/wSDL/soap12/">http://schemas.xmlsoap.org/wSDL/soap12/</a>	<a href="#">[SOAP1.2-1/2003]</a>
http	<a href="http://schemas.xmlsoap.org/wSDL/http/">http://schemas.xmlsoap.org/wSDL/http/</a>	<a href="#">[RFC2616]</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a> , <a href="#">[XMLSCHEMA2]</a>
wSDP	<a href="http://schemas.xmlsoap.org/ws/2006/02/devprof">http://schemas.xmlsoap.org/ws/2006/02/devprof</a>	<a href="#">[DPWS]</a>
wSP	<a href="http://schemas.xmlsoap.org/ws/2004/09/policy">http://schemas.xmlsoap.org/ws/2004/09/policy</a>	<a href="#">[WSPOLICY]</a>
wSD	<a href="http://schemas.xmlsoap.org/ws/2005/04/discovery">http://schemas.xmlsoap.org/ws/2005/04/discovery</a>	<a href="#">[WS-Discovery]</a>
wSE	<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing">http://schemas.xmlsoap.org/ws/2004/08/eventing</a>	<a href="#">[WSEVENTING]</a>
wSA	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	<a href="#">[WSAddressing1]</a>
wSU	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>	<a href="#">[WSS1]</a>
psp	<a href="http://schemas.microsoft.com/windows/2008/02/imaging/postscan/psp">http://schemas.microsoft.com/windows/2008/02/imaging/postscan/psp</a>	<a href="#">[MSDN-SPS]</a>
wscn	<a href="http://schemas.microsoft.com/windows/2006/08/wdp/scan">http://schemas.microsoft.com/windows/2006/08/wdp/scan</a>	<a href="#">[MSDN-WSSCAN]</a>
dsc	<a href="http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/configuration">http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/configuration</a>	Section <a href="#">6.2.1</a>

## 2.2.2 Messages

### 2.2.2.1 Error Codes

Error codes are returned in the soap:Fault element. A vendor MAY [<2>](#) create a subset of or extend these error codes by supporting private error subcodes. All fault messages defined in this specification MUST be sent according to the rules specified in [\[WSAddressing\]](#) section 4. The messages are sent to the [fault endpoint], if present and valid. Otherwise, they are sent to the [reply endpoint], if present. If neither are present, faults may be sent to the [source endpoint].

Endpoints compliant with this specification MUST include required message information headers on all fault messages. Fault messages are correlated as replies using the [relationship] property as defined in WS-Addressing. The following [action] property designates fault messages.

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
```

Properties are bound to a [\[SOAP1.2-1/2003\]](#) fault as follows.

```
<S:Envelope>
  <S:Header>
    <wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/addressing/fault</wsa:Action>
    <!-- Headers excluded for clarity -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value>
          [Code]
        </S:Value>
        <S:Subcode>
          <S:Value>
            [Subcode]
          </S:Value>
        </S:Subcode>
      </S:Code>
      <S:Reason>
        <S:Text xml:lang="en">
          [Reason]
        </S:Text>
      </S:Reason>
      <S:Detail>
        [Detail]
      </S:Detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

The definitions of faults use the following properties.

[Code]	Fault code
[Subcode]	The fault subcode.

[Code]	Fault code
[Reason]	The English language reason element.
[Detail]	The detail element. If absent, no detail element is defined for the fault.

### 2.2.3 Elements

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

Element	Description
<a href="#">dsc:Dialect</a>	A uniform resource identifier (URI) that represents the current filter.
<a href="#">dsc:Document</a>	An element that represents an individual <b>scan document</b> with a PostScan job.
<a href="#">dsc:Documents</a>	An element that contains zero or more <a href="#">dsc:Document</a> elements which represent all of the documents associated with a PostScan job.
<a href="#">dsc:DocumentDescription</a>	An element that describes the description information of a particular scan document.
<a href="#">dsc:DocumentId</a>	The Id of a scan document within a collection of scan documents that are part of a PostScan job.
<a href="#">dsc:Filter</a>	An element that represents a single filter supported by the scan repository.
<a href="#">dsc:FilterCompletedStatuses</a>	An element that contains the final processing status of all filters in the completed PostScan job.
<a href="#">dsc:FilterConfig</a>	The unique configuration XML for the filter dialect.
<a href="#">dsc:Filters</a>	An element that contains a list of the supported filters for this scan repository.
<a href="#">dsc:FilterState</a>	An element that represents the current processing state of a filter in the current PostScan job.
<a href="#">dsc:FilterStateReason</a>	An element that represents additional detailed information regarding the current processing state of a filter in the current PostScan job.
<a href="#">dsc:FilterStateReasons</a>	A collection of all of the additional detailed information regarding the current processing state of a filter in the current PostScan job.
<a href="#">dsc:FilterStatus</a>	An element that contains the current filter dialect and processing state.
<a href="#">dsc:FilterStatuses</a>	A collection of all of the statuses for all filters in the current PostScan job.
<a href="#">dsc:Format</a>	An element that contains the image file format of the scan document.
<a href="#">dsc:ImagesReceived</a>	An element that contains the total number of scan documents in the current PostScan job.

<b>Element</b>	<b>Description</b>
<a href="#">dsc:JobCompletedState</a>	An element that contains the final state of the PostScan job.
<a href="#">dsc:JobCompletedStateReasons</a>	A collection of all of the additional detailed information regarding the final processing state of a PostScan job.
<a href="#">dsc:JobCompletedTime</a>	An element that contains the amount of time that it took the PostScan job to complete processing.
<a href="#">dsc:JobCreatedTime</a>	An element that contains the amount of time that it took to create the PostScan job.
<a href="#">dsc:JobDescription</a>	An element that contains descriptive information about the PostScan job.
<a href="#">dsc:JobOriginatingUserName</a>	An element that contains the username that initiated the PostScan job.
<a href="#">dsc:JobState</a>	An element that contains the current processing state of a PostScan job.
<a href="#">dsc:JobStateReason</a>	An element that represents additional detailed information regarding the current processing state of the current PostScan job.
<a href="#">dsc:JobStateReasons</a>	A collection of all of the additional detailed information regarding the current processing state of the current PostScan job.
<a href="#">dsc:JobStatus</a>	An element that contains all of the information about the status of a PostScan job.
<a href="#">dsc:JobSummary</a>	An element that contains a subset of status and descriptive information about a PostScan job.
<a href="#">dsc:JobToken</a>	An element that contains a unique identifier for a PostScan job. The identifier is only guaranteed to be unique within the current scan repository.
<a href="#">dsc:PSP_DisplayName</a>	An element that contains the description of the PostScan process that provided the image acquisition settings and instructions of this PostScan job.
<a href="#">dsc:PSP_Identifier</a>	An element that contains the unique identifier of the PostScan process that provided the image acquisition settings and instructions of this PostScan job.
<a href="#">dsc:RepositoryConfiguration</a>	An element that contains the current configuration of the scan repository. This includes a collection of the supported filters.
<a href="#">dsc:RepositoryState</a>	An element that contains the current state of the scan repository.
<a href="#">dsc:RepositoryStateReason</a>	An element that represents additional detailed information regarding the current state of the scan repository.
<a href="#">dsc:RepositoryStateReasons</a>	A collection of all of the additional detailed information regarding the current state of the scan repository.
<a href="#">dsc:RepositoryStatus</a>	An element that contains all of the information about the status of the scan repository.



### 2.2.3.1 dsc:Dialect

The dsc:Dialect element is used to specify the name of a filter to be processed in a PostScan job or the name of a filter supported by the scan repository.

```
<xs:element name="Dialect" type="xs:QName" />
```

#### 2.2.3.1.1 dsc:Dialect, Child of dsc:FilterStatusType

The dsc:Dialect element, as defined in [dsc:FilterStatusType](#), specifies the name of a filter that will be used while processing the PostScan job.

```
<xs:element name="Dialect" type="psp:anyURIExtType"/>
```

#### 2.2.3.1.2 dsc:Dialect, Child of dsc:FilterType

The dsc:Dialect element, as defined in [dsc:FilterType](#), specifies the name of a specific filter supported by the scan repository.

```
<xs:element name="Dialect" type="xs:QName" />
```

### 2.2.3.2 dsc:Document

The dsc:Document element is used to represent an individual document in a PostScan job. The dsc:Document element contains descriptive information about the scan document.

```
<xs:element name="Document" type="dsc:DocumentType"
minOccurs="0" maxOccurs="unbounded"/>
```

### 2.2.3.3 dsc:Documents

The dsc:Documents element contains all of the current [dsc:Document](#) elements in a PostScan job.

```
<xs:element name="Documents" type="dsc:DocumentsType"/>
```

### 2.2.3.4 dsc:DocumentDescription

The dsc:DocumentDescription element contains the [dsc:DocumentId](#) and [dsc:Format](#) of a PostScan job.

```
<xs:element name="DocumentDescription"
type="dsc:DocumentDescriptionType"/>
```

### 2.2.3.5 dsc:DocumentId

The dsc:DocumentId element is used to store the numerical identifier of a scan document in a PostScan job. This Id will be unique within the confines of each PostScan job.

```
<xs:element name="DocumentId" type="dsc:IntOneExtType"/>
```

### 2.2.3.6 dsc:Filter

The dsc:Filter element is used to describe a single filter supported by the scan repository.

```
<xs:element name="Filter" type="dsc:FilterType" maxOccurs="unbounded"/>
```

### 2.2.3.7 dsc:FilterCompletedStatuses

The dsc:FilterCompletedStatuses element contains a collection of the final processing status of all filters in the completed PostScan job.

```
<xs:element name="FilterCompletedStatuses" type="dsc:FilterStatusesType"/>
```

### 2.2.3.8 dsc:FilterConfig

The dsc:FilterConfig element contains the unique configuration XML for the current filter dialect.

```
<xs:element name="FilterConfig" type="dsc:FilterConfigType"/>
```

### 2.2.3.9 dsc:Filters

The dsc:Filters element is used to return the list of filters supported by the scan repository.

```
<xs:element name="Filters" type="dsc:FiltersType" minOccurs="0"/>
```

### 2.2.3.10 dsc:FilterState

The dsc:FilterState element is used to indicate the current processing state of a filter in a PostScan job.

```
<xs:element name="FilterState" type="dsc:FilterStateType"/>
```

### 2.2.3.11 dsc:FilterStateReason

The dsc:FilterStateReason element indicates a single piece of detailed information regarding the current processing state of a filter in a PostScan job.

```
<xs:element name="FilterStateReason" type="dsc:FilterStateReasonType"
maxOccurs="unbounded"/>
```

### 2.2.3.12 dsc:FilterStateReasons

The dsc:FilterStateReasons element contains a collection of [dsc:FilterStateReason](#) elements.

```
<xs:element name="FilterStateReasons" type="dsc:FilterStateReasonsType"
minOccurs="0"/>
```

### 2.2.3.13 dsc:FilterStatus

The dsc:FilterStatus element contains the dialect and processing status of a filter.

```
<xs:element name="FilterStatus" type="dsc:FilterStatusType"
maxOccurs="unbounded"/>
```

### 2.2.3.14 dsc:FilterStatuses

The dsc:FilterStatuses element contains a collection of [dsc:FilterStatus](#) elements.

```
<xs:element name="FilterStatuses" type="dsc:FilterStatusesType"/>
```

### 2.2.3.15 dsc:Format

The dsc:Format element contains the image file format of the current scan document.

```
<xs:element name="Format" type="dsc:DocumentFormatExtType"/>
```

### 2.2.3.16 dsc:ImagesReceived

The dsc:ImagesReceived element contains the number of scan documents received for a PostScan job.

```
<xs:element name="ImagesReceived" type="dsc:IntZeroExtType"/>
```

### 2.2.3.17 dsc:JobCompletedState

The dsc:JobCompletedState element contains the final state of the PostScan job.

```
<xs:element name="JobCompletedState" type="dsc:JobStateType"/>
```

### 2.2.3.18 dsc:JobCompletedStateReasons

The dsc:JobCompletedStateReasons element contains a collection of [JobStateReason](#) elements that represent the final detailed state information for the PostScan job.

```
<xs:element name="JobCompletedStateReasons" type="dsc:JobStateReasonsType"/>
```

### 2.2.3.19 dsc:JobCompletedTime

The dsc:JobCompletedTime element contains the amount of time for the PostScan job to complete processing.

```
<xs:element name="JobCompletedTime" type="dsc:DateTimeExtType" minOccurs="0"/>
```

### 2.2.3.20 dsc:JobCreatedTime

The dsc:JobCreatedTime element contains the amount of time to create the PostScan job.

```
<xs:element name="JobCreatedTime" type="dsc:DateTimeExtType" minOccurs="0"/>
```

### 2.2.3.21 dsc:JobDescription

The dsc:JobDescription element contains information about the user and PostScan process that is used to submit and create the PostScan job.

```
<xs:element name="JobDescription" type="dsc:JobDescriptionType"/>
```

### 2.2.3.22 dsc:JobOriginatingUserName

The dsc:JobOriginatingUserName element contains the username that initiated the PostScan job.

```
<xs:element name="JobOriginatingUserName" type="dsc:String255ExtType"/>
```

### 2.2.3.23 dsc:JobState

The dsc:JobState element contains the current processing state of a PostScan job.

```
<xs:element name="JobState" type="dsc:JobStateType"/>
```

### 2.2.3.24 dsc:JobStateReason

The dsc:JobStateReason element contains a single piece of additional detailed information regarding the current processing state of the current PostScan job.

```
<xs:element name="JobStateReason" type="dsc:JobStateReasonType"
maxOccurs="unbounded"/>
```

### 2.2.3.25 dsc:JobStateReasons

The dsc:JobStateReasons element contains a collection of [dsc:JobStateReason](#) elements.

```
<xs:element name="JobStateReasons" type="dsc:JobStateReasonsType"
minOccurs="0"/>
```

### 2.2.3.26 dsc:JobStatus

The dsc:JobStatus element contains all of the information about the status of a PostScan job.

```
<xs:element name="JobStatus" type="dsc:JobStatusType"/>
```

### 2.2.3.27 dsc:JobSummary

The dsc:JobSummary element contains a subset of status and descriptive information about a PostScan job.

```
<xs:element name="JobSummary" type="dsc:JobSummaryType" minOccurs="0"
maxOccurs="unbounded"/>
```

### 2.2.3.28 dsc:JobToken

The dsc:JobToken element contains a unique identifier for a PostScan job. The identifier is only guaranteed to be unique within the current scan repository. Clients should view this element as opaque and not modify it in any way. [<3>](#)

```
<xs:element name="JobToken" type="dsc:String255ExtType"/>
```

### 2.2.3.29 dsc:PSP\_DisplayName

The dsc:PSP\_DisplayName element contains the description of the PostScan process that provided the image acquisition settings and instructions of this PostScan job.

```
<xs:element name="PSP_DisplayName" type="dsc:String255ExtType"/>
```

### 2.2.3.30 dsc:PSP\_Identifier

The dsc:PSP\_Identifier element contains the unique identifier of the PostScan process that provided the image acquisition settings and instructions of this PostScan job.

```
<xs:element name="PSP_Identifier" type="dsc:String255ExtType"/>
```

### 2.2.3.31 dsc:RepositoryConfiguration

The dsc:RepositoryConfiguration element contains the current configuration of the scan repository. This includes a collection of the supported filters.

```
<xs:element name="RepositoryConfiguration" type="dsc:RepositoryConfigurationType"/>
```

### 2.2.3.32 dsc:RepositoryState

The dsc:RepositoryState element contains the current state of the scan repository.

```
<xs:element name="RepositoryState" type="dsc:RepositoryStateType"/>
```

### 2.2.3.33 dsc:RepositoryStateReason

The dsc:RepositoryStateReason element contains a single piece of additional detailed information regarding the current state of the scan repository.

```
<xs:element name="RepositoryStateReason" type="dsc:RepositoryStateReasonType" maxOccurs="unbounded"/>
```

### 2.2.3.34 dsc:RepositoryStateReasons

The dsc:RepositoryStateReasons element contains a collection of [dsc:RepositoryStateReason](#) elements.

```
<xs:element name="RepositoryStateReasons" type="dsc:RepositoryStateReasonsType" minOccurs="0"/>
```

### 2.2.3.35 dsc:RepositoryStatus

The dsc:RepositoryStatus element contains all of the information about the status of the scan repository.

```
<xs:element name="RepositoryStatus" type="dsc:RepositoryStatusType"/>
```

## 2.2.4 Complex Types

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

Complex Type	Description
<a href="#">anyURIExtType</a>	An extensible type that allows extended attributes on an xs:anyURI type (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ).
<a href="#">dsc:DateTimeExtType</a>	An extensible type that allows extended attributes on an xs:dateTime type (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ).
<a href="#">dsc:DocumentDescriptionType</a>	A type that defines the descriptive information for a scan document within a PostScan job.
<a href="#">dsc:DocumentFormatExtType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:DocumentFormatBaseType</a> type.
<a href="#">dsc:DocumentType</a>	A type that defines all the information for a scan document within a PostScan job.
<a href="#">dsc:DocumentsType</a>	A type that defines a collection of zero or more <a href="#">dsc:DocumentType</a> elements.
<a href="#">dsc:FilterConfigType</a>	A complex type that allows filters to specify any <a href="#">valid</a> XML as configuration information.
<a href="#">dsc:FilterStateReasonsType</a>	A type that defines a collection of one or more <a href="#">dsc:FilterStateReasonType</a> elements.
<a href="#">dsc:FilterStateReasonType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:FilterStateReasonBaseType</a> type.
<a href="#">dsc:FilterStateType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:FilterStateBaseType</a> type.
<a href="#">dsc:FilterStatusesType</a>	A type that defines a collection of one or more <a href="#">dsc:FilterStatusType</a> elements.
<a href="#">dsc:FilterStatusType</a>	A complex type that includes the dialect and processing state information for a single filter within a PostScan job.
<a href="#">dsc:FiltersType</a>	A type that defines a collection of one or more <a href="#">dsc:FilterType</a> elements.
<a href="#">dsc:FilterType</a>	A complex type that includes the dialect and configuration for a single filter within the scan repository.
<a href="#">dsc:IntOneExtType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:IntOneBaseType</a> type.
<a href="#">dsc:IntZeroExtType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:IntZeroBaseType</a> type.
<a href="#">dsc:ListOfSummariesType</a>	A type that defines a collection of zero or more <a href="#">dsc:JobSummaryType</a> elements.
<a href="#">dsc:JobDescriptionType</a>	A type that defines the descriptive information about a PostScan job.

Complex Type	Description
<a href="#">dsc:JobStateReasonsType</a>	A type that defines a collection of one or more <a href="#">dsc:JobStateReasonType</a> elements.
<a href="#">dsc:JobStateReasonType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:JobStateReasonBaseType</a> type.
<a href="#">dsc:JobStateType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:JobStateBaseType</a> type.
<a href="#">dsc:JobStatusType</a>	A complex type that contains all of the processing state information for a PostScan job.
<a href="#">dsc:JobSummaryType</a>	A complex type that defines a subset of status and descriptive information about a PostScan job.
<a href="#">dsc:RepositoryConfigurationType</a>	A type that defines the elements that make up the configuration of the scan repository.
<a href="#">dsc:RepositoryStateType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:RepositoryStateBaseType</a> type.
<a href="#">dsc:RepositoryStateReasonsType</a>	A type that defines a collection of one or more <a href="#">dsc:RepositoryStateReasonType</a> elements.
<a href="#">dsc:RepositoryStateReasonType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:RepositoryStateReasonBaseType</a> type.
<a href="#">dsc:RepositoryStatusType</a>	A complex type that contains all of the state information for the scan repository.
<a href="#">dsc:String255ExtType</a>	An extensible type that allows extended attributes on an <a href="#">dsc:String255BaseType</a> type.

#### 2.2.4.1 psp:anyURIExtType

The psp:anyURIExtType type allows extended attributes on an xs:anyURI type (as specified in [XMLSCHEMA1](#) and [XMLSCHEMA2](#)).

```
<xs:complexType name="anyURIExtType">
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.2 dsc:DateTimeExtType

The dsc:DateTimeExtType type allows extended attributes on an xs:dateTime type (as specified in [XMLSCHEMA1](#) and [XMLSCHEMA2](#)). The server SHOULD use the current local time when setting time values.

```
<xs:complexType name="DateTimeExtType">
  <xs:simpleContent>
```



```

    <xs:extension base="xs:dateTime">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

### 2.2.4.3 dsc:DocumentDescriptionType

The dsc:DocumentDescriptionType type defines the descriptive information for a scan document within a PostScan job.

```

<xs:complexType name="DocumentDescriptionType">
  <xs:sequence>
    <xs:element name="DocumentId" type="dsc:IntOneExtType"/>
    <xs:element name="Format" type="dsc:DocumentFormatExtType" />
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

### 2.2.4.4 dsc:DocumentFormatExtType

The dsc:DocumentFormatExtType type allows extended attributes on an [dsc:DocumentFormatBaseType](#) type.

```

<xs:complexType name="DocumentFormatExtType">
  <xs:simpleContent>
    <xs:extension base="dsc:DocumentFormatBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

### 2.2.4.5 dsc:DocumentType

The dsc:DocumentType type defines all of the information for a scan document within a PostScan job.

```

<xs:complexType name="DocumentType">
  <xs:sequence>
    <xs:element name="DocumentDescription" type="dsc:DocumentDescriptionType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

### 2.2.4.6 dsc:DocumentsType

The dsc:DocumentsType type defines a collection of zero or more [dsc:DocumentType](#) elements.

```

<xs:complexType name="DocumentsType">
  <xs:sequence>

```

```

    <xs:element name="Document" type="dsc:DocumentType" minOccurs="0"
      maxOccurs="unbounded" />
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

### 2.2.4.7 dsc:FilterConfigType

The dsc:FilterConfigType type allows filters to specify any [valid](#) XML as configuration information.

```

<xs:complexType name="FilterConfigType">
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

### 2.2.4.8 dsc:FilterStateReasonsType

The dsc:FilterStateReasonsType type defines a collection of one or more [dsc:FilterStateReasonType](#) elements.

```

<xs:complexType name="FilterStateReasonsType">
  <xs:sequence>
    <xs:element name="FilterStateReason" type="dsc:FilterStateReasonType"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

### 2.2.4.9 dsc:FilterStateReasonType

The dsc:FilterStateReasonType type allows extended attributes on an [dsc:FilterStateReasonBaseType](#) type.

```

<xs:complexType name="FilterStateReasonType">
  <xs:simpleContent>
    <xs:extension base="dsc:FilterStateReasonBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

### 2.2.4.10 dsc:FilterStateType

The dsc:FilterStateType type allows extended attributes on an [dsc:FilterStateBaseType](#) type.

```

<xs:complexType name="FilterStateType">
  <xs:simpleContent>
    <xs:extension base="dsc:FilterStateBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

#### 2.2.4.11 dsc:FilterStatusesType

The dsc:FilterStatusesType type defines a collection of one or more [dsc:FilterStatusType](#) elements.

```

<xs:complexType name="FilterStatusesType">
  <xs:sequence>
    <xs:element name="FilterStatus" type="dsc:FilterStatusType"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.12 dsc:FilterStatusType

The dsc:FilterStatusType type includes the dialect and processing state information for a single filter within a PostScan job.

```

<xs:complexType name="FilterStatusType">
  <xs:sequence>
    <xs:element name="Dialect" type="psp:anyURIExtType"/>
    <xs:element name="FilterState" type="dsc:FilterStateType"/>
    <xs:element name="FilterStateReasons" type="dsc:FilterStateReasonsType"
      minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.13 dsc:FiltersType

The dsc:FiltersType type defines a collection of one or more [dsc:FilterType](#) elements.

```

<xs:complexType name="FiltersType">
  <xs:sequence>
    <xs:element name="Filter" type="dsc:FilterType" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.14 dsc:FilterType

The dsc:FilterType includes the dialect and configuration for a single filter within the scan repository.

```

<xs:complexType name="FilterType">
  <xs:sequence>
    <xs:element name="Dialect" type="xs:QName"/>
    <xs:element name="FilterConfig" type="dsc:FilterConfigType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>

```

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

#### 2.2.4.15 dsc:IntOneExtType

The dsc:IntOneExtType type allows extended attributes on an [dsc:IntOneBaseType](#) type.

```
<xs:complexType name="IntOneExtType">
  <xs:simpleContent>
    <xs:extension base="dsc:IntOneBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.16 dsc:IntZeroExtType

The dsc:IntZeroExtType type allows extended attributes on an [dsc:IntZeroBaseType](#) type.

```
<xs:complexType name="IntZeroExtType">
  <xs:simpleContent>
    <xs:extension base="dsc:IntZeroBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.17 dsc:ListOfSummarysType

The dsc:ListOfSummarysType type defines a collection of zero or more [dsc:JobSummaryType](#) elements.

```
<xs:complexType name="ListOfSummarysType">
  <xs:sequence>
    <xs:element name="JobSummary" type="dsc:JobSummaryType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

#### 2.2.4.18 dsc:JobDescriptionType

The dsc:JobDescriptionType type defines the descriptive information about a PostScan job.

```
<xs:complexType name="JobDescriptionType">
  <xs:sequence>
    <xs:element name="PSP_Identifier" type="dsc:String255ExtType"/>
    <xs:element name="PSP_DisplayName" type="dsc:String255ExtType"/>
    <xs:element name="JobOriginatingUserName" type="dsc:String255ExtType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

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

#### 2.2.4.19 dsc:JobStateReasonsType

The dsc:JobStateReasonsType type defines a collection of one or more [dsc:JobStateReasonType](#) elements.

```
<xs:complexType name="JobStateReasonsType">
  <xs:sequence>
    <xs:element name="JobStateReason" type="dsc:JobStateReasonType"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

#### 2.2.4.20 dsc:JobStateReasonType

The dsc:JobStateReasonType type allows extended attributes on an [dsc:JobStateReasonBaseType](#) type.

```
<xs:complexType name="JobStateReasonType">
  <xs:simpleContent>
    <xs:extension base="dsc:JobStateReasonBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.21 dsc:JobStateType

The dsc:JobStateType type allows extended attributes on an [dsc:JobStateBaseType](#) type.

```
<xs:complexType name="JobStateType">
  <xs:simpleContent>
    <xs:extension base="dsc:JobStateBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.22 dsc:JobStatusType

The dsc:JobStatusType type contains all of the processing state information for a PostScan job.

```
<xs:complexType name="JobStatusType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
    <xs:element name="JobState" type="dsc:JobStateType"/>
    <xs:element name="JobStateReasons" type="dsc:JobStateReasonsType"
      minOccurs="0"/>
    <xs:element name="FilterStatuses" type="dsc:FilterStatusesType"/>
  </xs:sequence>
</xs:complexType>
```

```

    <xs:element name="ImagesReceived" type="dsc:IntZeroExtType"/>
    <xs:element name="JobCreatedTime" type="dsc:DateTimeExtType"
      minOccurs="0"/>
    <xs:element name="JobCompletedTime" type="dsc:DateTimeExtType"
      minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.23 dsc:JobSummaryType

The dsc:JobSummaryType type defines a subset of status and descriptive information about a PostScan job.

```

<xs:complexType name="JobSummaryType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
    <xs:element name="PSP_Identifier" type="dsc:String255ExtType"/>
    <xs:element name="PSP_DisplayName" type="dsc:String255ExtType"/>
    <xs:element name="JobOriginatingUserName" type="dsc:String255ExtType"/>
    <xs:element name="JobState" type="dsc:JobStateType"/>
    <xs:element name="JobStateReasons" type="dsc:JobStateReasonsType"
      minOccurs="0"/>
    <xs:element name="FilterStatuses" type="dsc:FilterStatusesType"/>
    <xs:element name="ImagesReceived" type="dsc:IntZeroExtType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.24 dsc:RepositoryConfigurationType

The dsc:RepositoryConfigurationType type defines the elements that make up the configuration of the scan repository.

```

<xs:complexType name="RepositoryConfigurationType">
  <xs:sequence>
    <xs:element name="Filters" type="dsc:FiltersType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### 2.2.4.25 dsc:RepositoryStateType

The dsc:RepositoryStateType type allows extended attributes on an [dsc:RepositoryStateBaseType](#) type.

```

<xs:complexType name="RepositoryStateType">
  <xs:simpleContent>
    <xs:extension base="dsc:RepositoryStateBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

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

#### 2.2.4.26 dsc:RepositoryStateReasonsType

The dsc:RepositoryStateReasonsType type defines a collection of one or more [dsc:RepositoryStateReasonType](#) elements.

```
<xs:complexType name="RepositoryStateReasonsType">
  <xs:sequence>
    <xs:element name="RepositoryStateReason"
      type="dsc:RepositoryStateReasonType"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

#### 2.2.4.27 dsc:RepositoryStateReasonType

The dsc:RepositoryStateReasonType type allows extended attributes on an [dsc:RepositoryStateReasonBaseType](#) type.

```
<xs:complexType name="RepositoryStateReasonType">
  <xs:simpleContent>
    <xs:extension base="dsc:RepositoryStateReasonBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

#### 2.2.4.28 dsc:RepositoryStatusType

The dsc:RepositoryStatusType type contains all of the state information for the scan repository.

```
<xs:complexType name="RepositoryStatusType">
  <xs:sequence>
    <xs:element name="RepositoryState" type="dsc:RepositoryStateType"/>
    <xs:element name="RepositoryStateReasons"
      type="dsc:RepositoryStateReasonsType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

#### 2.2.4.29 dsc:String255ExtType

The dsc:String255ExtType type allows extended attributes on an [dsc:String255BaseType](#) type and restricts the length to 255 characters.

```
<xs:complexType name="String255ExtType">
  <xs:simpleContent>
    <xs:extension base="dsc:String255BaseType">
```

```

        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>

```

## 2.2.5 Simple Types

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

Simple Type	Description
<a href="#">dsc:DocumentFormatBaseType</a>	A simple type that defines a union of types <a href="#">dsc:DocumentFormatWKVType</a> and <a href="#">dsc:MimeExtensionPatternType</a> .
<a href="#">dsc:DocumentFormatWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent the predefined (or well-known) values of a scan documents format.
<a href="#">dsc:FilterStateBaseType</a>	A simple type that defines a union of types <a href="#">dsc:FilterStateWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:FilterStateReasonBaseType</a>	A simple type that defines a union of types <a href="#">dsc:FilterStateReasonWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:FilterStateReasonWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent the predefined (or well-known) values of detailed information regarding the processing state of a filter.
<a href="#">dsc:FilterStateWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent the predefined (or well-known) values of the processing state of a filter.
<a href="#">dsc:IntOneBaseType</a>	A simple type that restricts the built-in type xs:int (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) to a minimum value of one.
<a href="#">dsc:IntZeroBaseType</a>	A simple type that restricts the built-in type xs:int (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) to a minimum value of zero.
<a href="#">dsc:JobStateBaseType</a>	A simple type that defines a union of types <a href="#">dsc:JobStateWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:JobStateReasonBaseType</a>	A simple type that defines a union of types <a href="#">dsc:JobStateReasonWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:JobStateReasonWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent the predefined (or well-known) values of detailed information regarding the processing state of a PostScan job.
<a href="#">dsc:JobStateWKVType</a>	A simple type that defines an enumerated list of built-in type



Simple Type	Description
	xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) that represent the predefined (or well-known) values of the processing state of a PostScan job.
<a href="#">dsc:KeywordNsExtensionPatternType</a>	A simple type that restricts built-in type xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) with a pattern based on a regular expression that allows extensible keywords.
<a href="#">dsc:MimeExtensionPatternType</a>	A simple type that restricts built-in type xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) with a pattern based on a regular expression that allows extensible mime application types.
<a href="#">dsc:RepositoryStateBaseType</a>	A simple type that defines a union of types <a href="#">dsc:RepositoryStateWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:RepositoryStateReasonBaseType</a>	A simple type that defines a union of types <a href="#">dsc:RepositoryStateReasonWKVType</a> and <a href="#">dsc:KeywordNsExtensionPatternType</a> .
<a href="#">dsc:RepositoryStateReasonWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) that represent the predefined (or well-known) values of detailed information regarding the state of the scan repository.
<a href="#">dsc:RepositoryStateWKVType</a>	A simple type that defines an enumerated list of built-in type xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) that represent the predefined (or well known) values of the scan repository.
<a href="#">dsc:String255BaseType</a>	A simple type that restricts the built-in type xs:string (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) to a maximum length of 255 characters and preserves white space.

### 2.2.5.1 dsc:DocumentFormatBaseType

The [dsc:DocumentFormatBaseType](#) type defines a union of types [dsc:DocumentFormatWKVType](#) and [dsc:MimeExtensionPatternType](#).

```
<xs:simpleType name="DocumentFormatBaseType">
  <xs:union memberTypes="dsc:DocumentFormatWKVType
    dsc:MimeExtensionPatternType"/>
</xs:simpleType>
```

### 2.2.5.2 dsc:DocumentFormatWKVType

The [dsc:DocumentFormatWKVType](#) type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [XMLSCHEMA1] and [XMLSCHEMA2]) that represent the predefined (or well-known) values of a scan documents format.

```
<xs:simpleType name="DocumentFormatWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="dib"/>
    <xs:enumeration value="exif"/>
  </xs:restriction>
</xs:simpleType>
```

```

<xs:enumeration value="jbig"/>
<xs:enumeration value="jfif"/>
<xs:enumeration value="jpeg2k"/>
<xs:enumeration value="pdf-a"/>
<xs:enumeration value="png"/>
<xs:enumeration value="tiff-single-uncompressed"/>
<xs:enumeration value="tiff-single-g4"/>
<xs:enumeration value="tiff-single-g3mh"/>
<xs:enumeration value="tiff-single-jpeg-tn2"/>
<xs:enumeration value="tiff-multi-uncompressed"/>
<xs:enumeration value="tiff-multi-g4"/>
<xs:enumeration value="tiff-multi-g3mh"/>
<xs:enumeration value="tiff-multi-jpeg-tn2"/>
<xs:enumeration value="xps"/>
</xs:restriction>
</xs:simpleType>

```

Value	Description
dib	Windows Device Independent Bitmap
exif	Exchangeable Image File Format Version 2.x
jbig	ISO/IEC 11544:1993 Standard - Coded representation of picture and audio information -- Progressive bi-level image compression
jpeg2k	JPEG 2000 standards based file format and compression
pdf-a	PDF/A format: Standard based on ISO/CD 19005-1
png	Portable Networks Graphics (PNG) format. This format only supports PNG compression type
tiff-single-uncompressed	Single-page TIFF file with no compression type
tiff-single-g4	Single-page TIFF file with g4 compression type
tiff-single-g3mh	Single-page TIFF file with g3mh compression type
tiff-multi-uncompressed	Multiple-page TIFF file with no compression type
tiff-multi-g4	Multiple-page TIFF file with g4 compression type
tiff-multi-g3mh	Multiple page TIFF file with g3mh compression type
xps	XML Paper Specification

### 2.2.5.3 dsc:FilterStateBaseType

The `dsc:FilterStateBaseType` type defines a union of types [dsc:FilterStateWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```

<xs:simpleType name="FilterStateBaseType">
  <xs:union memberTypes="dsc:FilterStateWKVType
    dsc:KeywordNsExtensionPatternType"/>

```

```
</xs:simpleType>
```

#### 2.2.5.4 dsc:FilterStateReasonBaseType

The `dsc:FilterStateReasonBaseType` type defines a union of types [dsc:FilterStateReasonWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```
<xs:simpleType name="FilterStateReasonBaseType">  
  <xs:union memberTypes="dsc:FilterStateReasonWKVType  
    dsc:KeywordNsExtensionPatternType"/>  
</xs:simpleType>
```

#### 2.2.5.5 dsc:FilterStateReasonWKVType

The `dsc:FilterStateReasonWKVType` type defines an enumerated list of built-in type `xs:NMTOKEN` (as specified in [XMLSHEMA1](#) and [XMLSHEMA2](#)) that represent the predefined (or well-known) values of detailed information regarding the processing state of a filter.

```
<xs:simpleType name="FilterStateReasonWKVType">  
  <xs:restriction base="xs:NMTOKEN">  
    <xs:maxLength value="255"/>  
    <xs:enumeration value="InvalidArgs"/>  
    <xs:enumeration value="OutOfDiskSpace"/>  
    <xs:enumeration value="ServerOutOfMemory"/>  
    <xs:enumeration value="UnknownError"/>  
    <xs:enumeration value="FileShareAccessDenied"/>  
    <xs:enumeration value="FileShareOutOfDiskSpace"/>  
    <xs:enumeration value="SharePointAccessDenied"/>  
    <xs:enumeration value="SharePointOutOfDiskSpace"/>  
    <xs:enumeration value="InvalidSMTPServer"/>  
    <xs:enumeration value="UnableToSendEmail"/>  
  </xs:restriction>  
</xs:simpleType>
```

Value	Description
InvalidArgs	The filter does not understand something in the <code>psp:Instructions</code> elements
OutOfDiskSpace	The filter encountered an "Out of Disk Space" error while processing the scan document(s)
ServerOutOfMemory	The filter ran out of memory while processing the scan document(s)
UnknownError	The filter encountered an unknown problem while processing the scan document(s)
FileShareAccessDenied	The FileShare filter encountered an "Access Denied" error while processing the scan document(s)
FileShareOutOfDiskSpace	The FileShare filter encountered an "Out of Disk Space" error while processing the scan document(s)
SharePointAccessDenied	The SharePoint filter encountered an "Access Denied" error while processing the scan document(s)

Value	Description
SharePointOutOfDiskSpace	The SharePoint filter encountered an "Out of Disk Space" error while processing the scan document(s)
InvalidSMTPServer	The e-mail filter could not connect to the specified Simple Mail Transfer Protocol (SMTP) server while processing the scan document(s)
UnableToSendEmail	The e-mail filter encountered an SMTP mail failure while processing the scan document(s)

### 2.2.5.6 dsc:FilterStateWKVType

The dsc:FilterStateWKVType type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent the predefined (or well-known) values of the processing state of a filter.

```
<xs:simpleType name="FilterStateWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="Pending"/>
    <xs:enumeration value="Processing"/>
    <xs:enumeration value="Canceled"/>
    <xs:enumeration value="CompletedSuccessfully"/>
    <xs:enumeration value="CompletedWithErrors"/>
    <xs:enumeration value="CompletedWithWarnings"/>
  </xs:restriction>
</xs:simpleType>
```

Value	Description
Pending	The PostScan job has been initialized and the filter is waiting to be processed
Processing	The filter is processing the scan documents
Canceled	The filter was canceled as part of the PostScan job, either by a client using the <a href="#">CancelPostScanJob</a> operation or by means outside the scope of this protocol
CompletedSuccessfully	The filter finished processing without any warnings or errors
CompletedWithErrors	The filter finished processing with at least one filter error
CompletedWithWarnings	The filter finished processing with at least one warning. The filter was expected to successfully process the scan documents.

### 2.2.5.7 dsc:IntOneBaseType

The dsc:IntOneBaseType type restricts the built-in type xs:int (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) to a minimum value of one.

```
<xs:simpleType name="IntOneBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.5.8 dsc:IntZeroBaseType

The dsc:IntZeroBaseType type restricts the built-in type xs:int (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) to a minimum value of zero.

```
<xs:simpleType name="IntZeroBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.5.9 dsc:JobStateBaseType

The dsc:JobStateBaseType type defines a union of types [dsc:JobStateWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```
<xs:simpleType name="JobStateBaseType">
  <xs:union memberTypes="dsc:JobStateWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
```

### 2.2.5.10 dsc:JobStateReasonBaseType

The dsc:JobStateReasonBaseType type defines a union of types [dsc:JobStateReasonWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```
<xs:simpleType name="JobStateReasonBaseType">
  <xs:union memberTypes="
    dsc:JobStateReasonWKVType dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
```

### 2.2.5.11 dsc:JobStateReasonWKVType

The dsc:JobStateReasonWKVType type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent the predefined (or well-known) values of detailed information regarding the processing state of a PostScan job. [<4>](#)

```
<xs:simpleType name="JobStateReasonWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="None"/>
    <xs:enumeration value="PostScanJobTimedOut"/>
    <xs:enumeration value="NoValidPostScanInstructionsFound"/>
    <xs:enumeration value="CreatePostScanJobFailed"/>
    <xs:enumeration value="SendImageFailed"/>
    <xs:enumeration value="PostScanJobCanceled"/>
    <xs:enumeration value="PostScanJobProcessingFailed"/>
    <xs:enumeration value="PostScanJobCompletedSuccessfully"/>
    <xs:enumeration value="PostScanJobCompletedWithErrors"/>
    <xs:enumeration value="PostScanJobCompletedWithWarnings"/>
  </xs:restriction>
</xs:simpleType>
```

Value	Description
None	No additional information is available concerning the current PostScan job state.
PostScanJobTimedOut	<p>The PostScan job completed after no <b>EndPostScanJob</b> operation followed the <b>CreatePostScanJob</b> or <b>SendImage</b> operations within the 10-minute time limit.</p> <p>The <b>CreatePostScanJob</b> operation is part of the Distributed Scan Processing Web Service. It is used by a <b>scan device</b> to start a PostScan job on a scan repository. Each <b>CreatePostScanJob</b> operation is composed of a <b>CreatePostScanJobRequest</b> element from the scan device to the scan repository and a <b>CreatePostScanJobResponse</b> element from the scan repository back to the scan device. See <a href="#">[MSDN-WSDSP]</a> for more details on these elements.</p> <p>The <b>EndPostScanJob</b> operation is part of the Distributed Scan Processing Web Service. It is used by a scan device to notify a scan repository that all the scan documents for a previously created PostScan job have been transferred. Each <b>EndPostScanJob</b> operation is composed of a <b>EndPostScanJobRequest</b> element from the scan device to the scan repository and a <b>EndPostScanJobResponse</b> element from the scan repository back to the scan device. See <a href="#">[MSDN-WSDSP]</a> for more details on these elements.</p>
NoValidPostScanInstructionsFound	The scan repository does not recognize any of the filter dialects in the PostScan job instructions.
CreatePostScanJobFailed	The PostScan job was aborted when the <b>CreatePostScanJob</b> operation failed.
SendImageFailed	<p>The PostScan job was aborted when a <b>SendImage</b> operation failed.</p> <p>The <b>SendImage</b> operation is part of the Distributed Scan Processing Web Service. It is used by a scan device to send one or more scan documents to a scan repository. Each <b>SendImage</b> operation is composed of a <b>SendImageRequest</b> element from the scan device to the scan repository and a <b>SendImageResponse</b> element from the scan repository back to the scan device. See <a href="#">[MSDN-WSDSP]</a> for more details on these elements.</p>
PostScanJobCanceled	The PostScan job was canceled.
PostScanJobProcessingFailed	The PostScan job failed processing with an unrecoverable error.
PostScanJobCompletedSuccessfully	The PostScan job finished processing without any warnings or errors.
PostScanJobCompletedWithErrors	The PostScan job finished processing with at least one filter error.
PostScanJobCompletedWithWarnings	The PostScan job finished processing with at least one warning. The PostScan job scan documents are expected to be successfully processed.

### 2.2.5.12 dsc:JobStateWKVType

The dsc:JobStateWKVType type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent the predefined (or well-known) values of the processing state of a PostScan job. [<5>](#)

```
<xs:simpleType name="JobStateWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="Aborted"/>
    <xs:enumeration value="Canceled"/>
    <xs:enumeration value="Completed"/>
    <xs:enumeration value="Creating"/>
    <xs:enumeration value="Pending"/>
    <xs:enumeration value="Pending-Held"/>
    <xs:enumeration value="Processing"/>
    <xs:enumeration value="Started"/>
    <xs:enumeration value="Terminating"/>
  </xs:restriction>
</xs:simpleType>
```

Value	Description
Aborted	The PostScan job was aborted by the scan repository
Canceled	The PostScan job was canceled either by a client using the <a href="#">CancelPostScanJob</a> operation or by means outside the scope of this protocol
Completed	The PostScan job is finished processing and all of the scan documents have been processed by the appropriate filters
Creating	The PostScan job is being initialized
Pending	The PostScan job has been initialized and is waiting to be processed
Pending-Held	The PostScan job is waiting to be processed but is unavailable for scheduling. This state can only be reached by methods outside the scope of this protocol
Processing	The PostScan job scan documents are being processed by the appropriate filter(s)
Started	The scan repository has begun processing the PostScan job. This is a transient state and will usually be seen only within a <a href="#">PostScanJobStatusEvent</a>
Terminating	The PostScan job was canceled either by a client using the CancelPostScanJob operation or aborted by means outside the scope of this protocol

### 2.2.5.13 dsc:KeywordNsExtensionPatternType

The dsc:KeywordNsExtensionPatternType type restricts built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) with a pattern based on a regular expression that allows extensible keywords.

```
<xs:simpleType name="KeywordNsExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+:[\w_\-\.\.]+"/>
  </xs:restriction>
</xs:simpleType>
```

#### 2.2.5.14 dsc:MimeExtensionPatternType

The dsc:MimeExtensionPatternType type restricts built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) with a pattern based on a regular expression that allows extensible keywords.

```
<xs:simpleType name="MimeExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+[\w:/;_-\|\\\+\.]*"/>
  </xs:restriction>
</xs:simpleType>
```

#### 2.2.5.15 dsc:RepositoryStateBaseType

The dsc:RepositoryStateBaseType type defines a union of types [dsc:RepositoryStateWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```
<xs:simpleType name="RepositoryStateBaseType">
  <xs:union memberTypes="dsc:RepositoryStateWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
```

#### 2.2.5.16 dsc:RepositoryStateReasonBaseType

The dsc:RepositoryStateReasonBaseType type defines a union of types [dsc:RepositoryStateReasonWKVType](#) and [dsc:KeywordNsExtensionPatternType](#).

```
<xs:simpleType name="RepositoryStateReasonBaseType">
  <xs:union memberTypes="dsc:RepositoryStateReasonWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
```

#### 2.2.5.17 dsc:RepositoryStateReasonWKVType

The dsc:RepositoryStateReasonWKVType type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent the predefined (or well-known) values of detailed information regarding the state of the scan repository.

```
<xs:simpleType name="RepositoryStateReasonWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="None"/>
  </xs:restriction>
</xs:simpleType>
```

Value	Description
None	No additional information is available concerning the current scan repository state



### 2.2.5.18 dsc:RepositoryStateWKVType

The dsc:RepositoryStateWKVType type defines an enumerated list of built-in type xs:NMTOKEN (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent the predefined (or well-known) values of state of the scan repository. [<6>](#)

```
<xs:simpleType name="RepositoryStateWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="Idle"/>
    <xs:enumeration value="Processing"/>
    <xs:enumeration value="Stopped"/>
  </xs:restriction>
</xs:simpleType>
```

Value	Description
Idle	The scan repository server is available and can start processing a new job
Processing	The scan repository server is currently processing jobs
Stopped	No PostScan jobs can be processed and intervention is needed

### 2.2.5.19 dsc:String255BaseType

The dsc:String255BaseType type restricts the built-in type xs:string (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) to a maximum length of 255 characters and preserves white space.

```
<xs:simpleType name="String255BaseType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="255"/>
    <xs:whiteSpace value="preserve"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.2.6 Attributes

An attribute is considered a common attribute if it is used (directly or indirectly) by more than one **WSDL message**.

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

Attribute	Description
<a href="#">Name</a>	An attribute that contains the qualified name for either a Repository element or a PostScan job element. This name is used in the <a href="#">GetRepositoryElements</a> response or the <a href="#">GetPostScanJobElements</a> response to identify the data element in the current <a href="#">dsc:ElementData</a> element, as specified in sections <a href="#">3.2.4.1.2.1</a> and <a href="#">3.2.4.3.2.1</a> .
<a href="#">Valid</a>	A <b>Boolean</b> attribute that indicates whether the <b>Name</b> attribute in the current <a href="#">dsc:ElementData</a> element, as specified in sections <a href="#">3.2.4.1.2.1</a> and <a href="#">3.2.4.3.2.1</a> , is a recognized qualified name for this scan repository.

### 2.2.6.1 Name

The **Name** attribute contains the qualified name for either a Repository element or a PostScan job element. This name is used in the [GetRepositoryElements](#) response or the [GetPostScanJobElements](#) response to identify the data element in the current [dsc:ElementData](#) element as specified in sections [3.2.4.1.2.1](#) and [3.2.4.3.2.1](#).

```
<xs:attribute name="Name" type="xs:QName"/>
```

### 2.2.6.2 Valid

The **Valid** attribute indicates if the **Name** attribute in the current [dsc:ElementData](#) element, as specified in sections [3.2.4.1.2.1](#) and [3.2.4.3.2.1](#), is a recognized qualified name for this scan repository.

```
<xs:attribute name="Valid" type="xs:boolean"/>
```

### 2.2.7 Groups

None.

### 2.2.8 Attribute Groups

None.

## 3 Protocol Details

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

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

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

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

### 3.1 Common Details

#### 3.1.1 Abstract Data Model

None.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing Rules

##### 3.1.4.1 Common SOAP Faults

The following sections list error codes common to one or more operations.

###### 3.1.4.1.1 wsa:ActionNotSupported

The wsa:ActionNotSupported fault is sent when a client requests an operation that is not supported by the current service.

[Code]	soap:Sender
[Subcode]	wsa:ActionNotSupported
[Reason]	The [action] cannot be processed at the receiver
[Detail]	The invalid operation name

### 3.1.4.1.2 dsc:InvalidArgs

The dsc:InvalidArgs fault is sent when a client sends an invalid argument as part of an operation. The invalid argument could be any of the following: not enough in args, too many in args, no arg by that name, one or more in args are of the wrong data type.

[Code]	soap:Sender
[Subcode]	dsc:InvalidArgs
[Reason]	At least one input argument is invalid
[Detail]	The invalid argument

### 3.1.4.1.3 dsc:OperationFailed

The dsc:OperationFailed fault may be returned if the current state of the scan repository service prevents invoking the operation. This could be due to a change in server configuration or the scan repository service being disabled.

[Code]	soap:Receiver
[Subcode]	dsc:OperationFailed
[Reason]	Scan repository service cannot perform the requested operation
[Detail]	None

### 3.1.4.1.4 dsc:ServerErrorTemporaryError

The dsc:ServerErrorTemporaryError fault is sent when the server experiences a temporary error that occurs while the scan repository processes the operation. The client MAY try the unmodified request again at some later point in time with an expectation that the temporary internal error condition may have been cleared. If there is a more specific error defined that applies to a temporary error, such as disk full, that code SHOULD [<7>](#) be used.

[Code]	soap:Receiver
[Subcode]	dsc:ServerErrorTemporaryError
[Reason]	The service had an unexpected error
[Detail]	None

### 3.1.4.1.5 dsc:ServerErrorInternalError

The dsc:ServerErrorInternalError fault is sent when the scan repository encounters an unexpected condition that prevented it from fulfilling the request. This error differs from [dsc:ServerErrorTemporaryError](#) in that it implies a more permanent type of internal error and resending the operation will return the same fault.

[Code]	soap:Receiver
[Subcode]	dsc:ServerErrorInternalError

<b>[Code]</b>	<b>soap:Receiver</b>
[Reason]	The service had an unexpected error
[Detail]	None

### 3.1.4.1.6 dsc:ClientErrorJobTokenNotFound

The dsc:ClientErrorJobTokenNotFound fault is sent when the scan repository cannot find a PostScan job matching the [dsc:JobToken](#) argument for either an [CancelPostScanJob](#) operation or an [dsc:GetPostScanJobElements](#) operation.

<b>[Code]</b>	<b>soap:Sender</b>
[Subcode]	dsc:ClientErrorJobTokenNotFound
[Reason]	A PostScan job identified by the specified dsc:JobToken argument could not be found.
[Detail]	JobToken: Specified in JobToken. <a href="#">&lt;8&gt;</a>

## 3.2 RepositoryConfigurationServiceType Server Details

### 3.2.1 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.

This protocol depends on an abstract data model that maintains information about the scan repository, PostScan jobs, and related objects. These objects represent processing filters and PostScan jobs within the scan repository, and they are used in this protocol to communicate the capabilities and status of the Repository and the processing status of any PostScan jobs.

A scan repository **MUST** behave as if it hosted the following objects in the hierarchy specified below:

**Repository Configuration:** The object that contains the **List of Filters** supported by the scan repository.

**List of Filters:** A filter is a processing object in the scan repository. The definition of a filter consists of a dialect and filter Configuration. Each filter can be accessed by a PostScan job to perform actions upon the scan documents of the job.

**Repository Status:** The object that contains the state of the scan repository and a list of state reasons.

**List of PostScan jobs:** A PostScan job is the basic unit of work within the scan repository. Each PostScan job will contain one or more scan documents and invoke filters to perform actions on these scan documents. Each PostScan job object **MUST** maintain the following data elements:

- A **Job Status** object
- A list of **Filter Status** objects
- A **Job Description**

- A list of scan documents

**Job Status:** The object that contains information about the status of a PostScan job. Each <Job Status> object MUST maintain the following data elements:

- A token that uniquely identifies the PostScan job.
- The state of the PostScan job.
- A list of state reasons that give detailed information about the state of the PostScan job.

**Filter Status:** The object that contains information about the status of a single filter instance within a PostScan job. Each <Filter Status> object MUST maintain the following data elements:

- The dialect of the filter. This dialect will match one of the filters contained in the **Repository Configuration**.
- The state of the filter.
- A list of state reasons that give detailed information about the state of the filter.

**Job Description:** The object that contains descriptive information about a PostScan job. Each **Job Description** object MUST maintain the following data elements:

- An **Identifier** that uniquely identifies the **PostScan Process** associated with the PostScan job.
- The name of the **PostScan Process** associated with the PostScan job.
- The name of the user that submitted the PostScan job.

**Scan Document:** The object that contains information about a single scanned image that is part of a PostScan job. The scan document object MUST maintain the following data elements:

- A Id that is unique within the context of a PostScan job.
- The description of the scan document.
- The image file format of the scan document.

**PostScan Process:** The object that contains all the configuration and processing instructions required to scan documents at a scan device and then process the scanned documents at the scan repository. The data that make up a **PostScan Process** include a unique identifier, image acquisition settings, the name of a scan repository, and **filter-specific instructions** that specify how to process the scanned documents at the scan repository. The user selects a **PostScan Process** at the scan device. The scan device uses the information from the selected **PostScan Process** to scan the documents, find the scan repository, and tell the scan repository how to process the scanned images transferred as part of the PostScan job.

**Filter Instructions:** A set of data elements that a filter uses to determine how to process the scan documents of the scan job. The data elements are unique to each filter definition. This filter is identified by a dialect which will match one of the filters in the **Repository Configuration**.

The distributed scan environment typically consists of a management client, a directory service, a scan device, and a scan repository service. The management client is used by the administrator to create **PostScan Process** objects and query the scan repository service for configuration and status

information. The directory service is where the **PostScan Process** objects are stored for accessing by the scan device and the scan repository service. The scan device will find **PostScan Process** objects for a user and based on one of these create a PostScan job in the scan repository. The data within a **PostScan Process** is used to determine how to scan the documents at the scan device, which scan repository service to work with, and then how to process the scanned documents once they are sent to the scan repository server.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

The server MUST listen on the well-known User Datagram Protocol (UDP) and TCP ports for this protocol as defined in section [1.9](#).

### 3.2.4 Message Processing Events and Sequencing Rules

This section contains 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 [\[WSDL\]](#).

Unless otherwise specified, operations of this protocol MUST return an [dsc:InvalidArgs](#) SOAP fault upon receiving incorrectly-formed arguments.

Operation	Description
<a href="#">CancelPostScanJob</a>	This operation deletes a currently processing (active) PostScan job from the scan repository.
<a href="#">GetActiveJobs</a>	This operation returns a list of all currently processing (active) PostScan jobs from the scan repository.
<a href="#">GetJobHistory</a>	This operation returns a list of all completed PostScan jobs from the scan repository.
<a href="#">GetPostScanJobElements</a>	This operation returns data elements of a PostScan job from the scan repository.
<a href="#">GetRepositoryElements</a>	This operation returns data elements from the scan repository.
<a href="#">PostScanJobEndStateEvent</a>	An event generated by the scan repository upon PostScan job completion.
<a href="#">PostScanJobStatusEvent</a>	An event generated by the scan repository upon change of status of a PostScan job.
<a href="#">RepositoryElementsChangeEvent</a>	An event generated by the scan repository upon change of one or more elements in the repository configuration.
<a href="#">RepositoryStatusSummaryEvent</a>	An event generated by the scan repository upon change of the repository status.

#### 3.2.4.1 GetRepositoryElements

The GetRepositoryElements operation is used to query the scan repository for its configuration and/or status.

```

<wsdl:operation name="GetRepositoryElements">
  <wsdl:input message="dsc:GetRepositoryElementsRequestMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
    configuration/GetRepositoryElements"/>
  <wsdl:output message="dsc:GetRepositoryElementsResponseMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
    configuration/GetRepositoryElementsResponse"/>
</wsdl:operation>

```

Upon receiving a [dsc:GetRepositoryElementsRequestMsg](#), the server MUST respond with an [dsc:GetRepositoryElementsResponseMsg](#) or a SOAP fault as follows:

- If the server cannot validate the arguments of the `dsc:GetRepositoryElementsRequestMsg`, the server MUST return a [dsc:InvalidArgs](#) SOAP fault.
- If the server encounters an unexpected error from which it can recover while processing the `dsc:GetRepositoryElementsRequestMsg`, the server MUST return a [dsc:ServerErrorTemporaryError](#) SOAP fault.
- If the server encounters an unexpected error from which it cannot recover while processing the `dsc:GetRepositoryElementsRequestMsg`, the server MUST return a [dsc:ServerErrorInternalError](#) SOAP fault depending on the severity of the unexpected error.
- If the server can successfully process the `dsc:GetRepositoryElementsRequestMsg`, the server MUST return a `dsc:GetRepositoryElementsResponseMsg` with the following requirements:
  - The `dsc:GetRepositoryElementsResponseMsg` MUST have one [dsc:RepositoryElements](#) child, which MUST contain as many [dsc:ElementData](#) children as there were [dsc:Name](#) children in the request message. The `dsc:ElementData` children MUST be in the same order as the `dsc:Name` children in the request message.
  - Each `dsc:ElementData` child corresponding to an `dsc:Name` child of the request's [dsc:RequestedElements](#) MUST have the following characteristics:
    - The attributes of the `dsc:ElementData` child MUST be set as follows:
      - The **Name** attribute MUST be set to the same value as the value of the corresponding `dsc:Name` element of the request.
      - If the requested element exists on the server, the **Valid** attribute MUST be set to "true", otherwise, if the requested element is not known by the server, the **Valid** attribute MUST be set to "false".<9>
      - Vendor-defined attributes MAY be added.
    - The contents of the element MUST be set as follows:
      - If the element is a configuration element, an [dsc:RepositoryConfiguration](#) child element MUST be added.
      - If the element is a status element, an [dsc:RepositoryStatus](#) element MUST be added.
      - Vendor-defined child elements MAY be added.

### 3.2.4.1.1 Messages

The following WSDL message definitions are specific to this operation.



### 3.2.4.1.1.1 GetRepositoryElementsRequestMsg

The GetRepositoryElementsRequestMsg message is a request WSDL message for a [GetRepositoryElements](#) WSDL operation.

```
<wsdl:message name="GetRepositoryElementsRequestMsg">
  <wsdl:part name="body"
    element="dsc:GetRepositoryElementsRequest"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetRepositoryElements
```

The SOAP body contains an [dsc:GetRepositoryElementsRequest](#) element.

### 3.2.4.1.1.2 GetRepositoryElementsResponseMsg

The GetRepositoryElementsResponseMsg message is a response WSDL message for a [GetRepositoryElements](#) WSDL operation.

```
<wsdl:message name="GetRepositoryElementsResponseMsg">
  <wsdl:part name="body" element="dsc:GetRepositoryElementsResponse"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetRepositoryElementsResponse
```

The SOAP body contains an [dsc:GetRepositoryElementsResponse](#) element.

### 3.2.4.1.2 Elements

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

Element	Description
<a href="#">dsc:ElementData</a>	An element that contains response data for a <a href="#">GetRepositoryElements</a> response.
<a href="#">dsc:GetRepositoryElementsRequest</a>	An element that contains the input data for a <a href="#">GetRepositoryElements</a> WSDL operation.
<a href="#">dsc:GetRepositoryElementsResponse</a>	An element that contains the output data for a <a href="#">GetRepositoryElements</a> WSDL operation.
<a href="#">dsc:Name</a>	An element that contains the qualified name for a Repository element. This name is used in the <a href="#">GetRepositoryElements</a> response to specify the data that the client is requesting.

Element	Description
<a href="#">dsc:RepositoryElements</a>	A collection of <a href="#">dsc:ElementData</a> elements in a <a href="#">GetRepositoryElements</a> response.
<a href="#">dsc:RequestedElements</a>	A collection of <a href="#">dsc:Name</a> elements that represent what information is to be returned in a <a href="#">GetRepositoryElements</a> .

### 3.2.4.1.2.1 dsc:ElementData

The [dsc:ElementData](#) element returns requested data in a [GetRepositoryElements](#) response.

```
<xs:element name="ElementData" type="dsc:RepositoryElementDataType"
maxOccurs="unbounded"/>
```

### 3.2.4.1.2.2 dsc:GetRepositoryElementsRequest

The [dsc:GetRepositoryElementsRequest](#) element contains input data for a [GetRepositoryElements](#) WSDL operation.

```
<xs:element name="GetRepositoryElementsRequest"
type="dsc:GetRepositoryElementsRequestType"/>
```

The [dsc:GetRepositoryElementsRequest](#) is an element of type [dsc:GetRepositoryElementsRequestType](#).

### 3.2.4.1.2.3 dsc:GetRepositoryElementsResponse

The [dsc:GetRepositoryElementsResponse](#) element contains output data for an [dsc:GetRepositoryElements](#) WSDL operation.

```
<xs:element name="GetRepositoryElementsResponse"
type="dsc:GetRepositoryElementsResponseType"/>
```

[dsc:GetRepositoryElementsResponse](#) is an element of type [dsc:GetRepositoryElementsResponseType](#).

### 3.2.4.1.2.4 dsc:Name

The [dsc:Name](#) element contains the qualified name for a Repository element. This name is used in the [dsc:GetRepositoryElements](#) request to specify the data the client is requesting.

```
<xs:element name="Name" type="xs:QName" maxOccurs="unbounded"/>
```

### 3.2.4.1.2.5 dsc:RepositoryElements

The [dsc:RepositoryElements](#) is a collection of [dsc:ElementData](#) elements in a [GetRepositoryElements](#) response.

```
<xs:element name="RepositoryElements"
```

```
type="dsc:RepositoryElementsType"/>
```

dsc:RepositoryElements is an element of type [dsc:RepositoryElementsType](#).

### 3.2.4.1.2.6 dsc:RequestedElements

The dsc:RequestedElements element contains a collection of [dsc:Name](#) elements that represent what information is to be returned for an [dsc:GetRepositoryElements](#) request.

```
<xs:element name="RequestedElements"  
type="dsc:RequestedRepositoryElementsType"/>
```

dsc:RequestedElements is an element of type [dsc:RequestedRepositoryElementsType](#).

### 3.2.4.1.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

Complex Type	Description
<a href="#">dsc:GetRepositoryElementsRequestType</a>	A complex type that contains the request data for an <a href="#">dsc:GetRepositoryElements</a> WSDL operation
<a href="#">dsc:GetRepositoryElementsResponseType</a>	A complex type that contains the response data for an <a href="#">dsc:GetRepositoryElements</a> WSDL operation
<a href="#">dsc:RepositoryElementDataType</a>	A type that contains all the scan repository elements that can be requested with an <a href="#">dsc:GetRepositoryElements</a> request and attributes that represent an element's qualified name and whether the element name requested is recognized
<a href="#">dsc:RequestedRepositoryElementsType</a>	A complex type that contains all the scan repositories and defines a collection of one or more elements that can be requested with an <a href="#">dsc:GetRepositoryElements</a> request as well as attributes that represent an element's qualified name and whether the element name requested is recognized
<a href="#">dsc:RepositoryElementsType</a>	A type that defines a collection of one or more <a href="#">dsc:RepositoryElementDataType</a> elements
<a href="#">dsc:RequestedRepositoryElementsType</a>	A complex type that defines a collection of one or more elements of type <a href="#">xs:QName</a> (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent what information is to be returned in an <a href="#">dsc:GetRepositoryElements</a> request

#### 3.2.4.1.3.1 dsc:GetRepositoryElementsRequestType

The [dsc:GetRepositoryElementsRequestType](#) type contains the request data for an [dsc:GetRepositoryElements](#) WSDL operation.

```
<xs:complexType name="GetRepositoryElementsRequestType">  
  <xs:sequence>  
    <xs:element name="RequestedElements" type="dsc:RequestedRepositoryElementsType"/>  
  </xs:sequence>  
</xs:complexType>
```

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

**dsc:RequestedElements:** The repository data element(s) the client requests to be returned.

### 3.2.4.1.3.2 dsc:GetRepositoryElementsResponseType

The `dsc:GetRepositoryElementsResponseType` type contains the response data for an [dsc:GetRepositoryElements](#) WSDL operation.

```
<xs:complexType name="GetRepositoryElementsResponseType">
  <xs:sequence>
    <xs:element name="RepositoryElements" type="dsc:RepositoryElementsType"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:RepositoryElements:** The repository data element(s) returned in response to the request. Each data element **MUST** contain:

- The name of the element.
- The **Valid** attribute indicating whether the element name was recognized by the scan repository:
  - If the element name is recognized by the scan repository, attribute equals "true".
  - If the element name is not recognized by the scan repository, attribute equals "false".
- The requested element's data.

### 3.2.4.1.3.3 dsc:RepositoryElementDataType

The `dsc:RepositoryElementDataType` type contains all the scan repository elements that can be requested with an [dsc:GetRepositoryElements](#) request and attributes that represent an element's qualified name and whether the element name requested is recognized.

```
<xs:complexType name="RepositoryElementDataType">
  <xs:choice>
    <xs:element name="RepositoryConfiguration"
      type="dsc:RepositoryConfigurationType"/>
    <xs:element name="RepositoryStatus"
      type="dsc:RepositoryStatusType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attribute ref="dsc:Name" use="required"/>
  <xs:attribute ref="dsc:Valid" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

### 3.2.4.1.3.4 dsc:RepositoryElementsType

The `dsc:RepositoryElementsType` type defines a collection of one or more [dsc:RepositoryElementDataType](#) elements.

```

<xs:complexType name="RepositoryElementsType">
  <xs:sequence>
    <xs:element name="ElementData"
      type="dsc:RepositoryElementDataType" maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

### 3.2.4.1.3.5 dsc:RequestedRepositoryElementsType

The `dsc:RequestedRepositoryElementsType` type defines a collection of one or more elements of type `xs:QName` (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent what information is to returned in an [dsc:GetRepositoryElements](#) request.

```

<xs:complexType name="RequestedRepositoryElementsType">
  <xs:sequence>
    <xs:element name="Name" type="xs:QName" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

#### Predefined Values:

[dsc:RepositoryConfiguration](#) - Get all the configuration information for the scan repository

[dsc:RepositoryStatus](#) - Get all the status information for the scan repository

### 3.2.4.2 CancelPostScanJob

The `CancelPostScanJob` operation is used to cancel a `PostScan` job that is being processed.

```

<wsdl:operation name="CancelPostScanJob">
  <wsdl:input message="dsc:CancelPostScanJobRequestMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
      configuration/CancelPostScanJob"/>
  <wsdl:output message="dsc:CancelPostScanJobResponseMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
      configuration/CancelPostScanJobResponse"/>
</wsdl:operation>

```

Upon receiving a [dsc:CancelPostScanJobRequestMsg](#), the server MUST respond with an [dsc:CancelPostScanJobResponseMsg](#) or a SOAP fault, as follows:

- If the server cannot validate the arguments of the `dsc:CancelPostScanJobRequestMsg`, the server MUST return a [dsc:InvalidArgs](#) SOAP fault. [<10>](#)
- If the server encounters an unexpected error from which it can recover while processing the `dsc:CancelPostScanJobRequestMsg`, the server MUST return a [dsc:ServerErrorTemporaryError](#) SOAP fault.

- If the server encounters an unexpected error from which it cannot recover while processing the `dsc:CancelPostScanJobRequestMsg`, the server MUST return a [dsc:ServerErrorInternalError](#) SOAP fault depending on the severity of the unexpected error.
- If the server does not have a currently processing PostScan job that matches the [dsc:JobToken](#) element of the [CancelPostScanJobRequest](#) element, the server MUST return an [dsc:ClientErrorJobTokenNotFound](#) fault.
- If the server does have a currently processing PostScan job that matches the `dsc:JobToken` element of the `dsc:CancelPostScanJobRequest` element, the server MUST take following steps:
  - Attempt to stop any filters currently processing scan documents.
  - Cancel any filters that have not begun processing.
  - Set the value of [dsc:JobState](#) to "Canceled" and the value of [dsc:JobStateReason](#) to "PostScanJobCanceled".
  - Respond with an [dsc:CancelPostScanJobResponse](#).

The `CancelPostScanJob` is a best-effort operation. The scan repository will accept or reject the operation, then the client can determine the outcome by checking the information returned in the [dsc:PostScanJobEndStateEvent](#) or by using [GetPostScanJobElements](#) after the job has completed processing.

### 3.2.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.2.4.2.1.1 CancelPostScanJobRequestMsg

The `CancelPostScanJobRequestMsg` is a request WSDL message for a [CancelPostScanJob](#) WSDL operation.

```
<wsdl:message name="CancelPostScanJobRequestMsg">
  <wsdl:part name="body" element="dsc:CancelPostScanJobRequest"/>
</wsdl:message>
```

The SOAP action value is:

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/configuration/
CancelPostScanJob
```

The SOAP body contains an [CancelPostScanJobRequest](#) element.

#### 3.2.4.2.1.2 CancelPostScanJobResponseMsg

The [CancelPostScanJobResponse](#) is a response WSDL message for a [CancelPostScanJob](#) WSDL operation.

```
<wsdl:message name="CancelPostScanJobResponseMsg">
  <wsdl:part name="body" element="dsc:CancelPostScanJobResponse"/>
</wsdl:message>
```

The SOAP action value is:

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/  
configuration/CancelPostScanJobResponse
```

The SOAP body contains an `CancelPostScanJobResponse` element.

### 3.2.4.2.2 Elements

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

#### 3.2.4.2.2.1 CancelPostScanJobRequest

The `CancelPostScanJobRequest` element contains input data for a [CancelPostScanJob](#) WSDL operation.

```
<xs:element name="CancelPostScanJobRequest"  
type="dsc:CancelPostScanJobRequestType"/>
```

`CancelPostScanJobRequest` is an element of type [CancelPostScanJobRequestType](#).

#### 3.2.4.2.2.2 CancelPostScanJobResponse

The `CancelPostScanJobResponse` element contains output data for a [CancelPostScanJob](#) WSDL operation.

```
<xs:element name="CancelPostScanJobResponse"  
type="dsc:CancelPostScanJobResponseType"/>
```

`CancelPostScanJobResponse` is an element of type [CancelPostScanJobResponseType](#).

### 3.2.4.2.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

#### 3.2.4.2.3.1 CancelPostScanJobRequestType

The `CancelPostScanJobRequestType` type contains the request data for a [CancelPostScanJob](#) WSDL operation.

```
<xs:complexType name="CancelPostScanJobRequestType">  
  <xs:sequence>  
    <xs:element name="JobToken" type="dsc:String255ExtType"/>  
  </xs:sequence>  
</xs:complexType>
```

**dsc:JobToken:** The unique identifier of the PostScan job to be canceled.

### 3.2.4.2.3.2 CancelPostScanJobResponseType

The CancelPostScanJobResponseType type contains the response data for a [CancelPostScanJob](#) WSDL operation.

```
<xs:complexType name="CancelPostScanJobResponseType">
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 3.2.4.3 GetPostScanJobElements

The GetPostScanJobElements operation is used to query the scan repository for information pertaining to a single PostScan job.

```
<wsdl:operation name="GetPostScanJobElements">
  <wsdl:input message="dsc:GetPostScanJobElementsRequestMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
distributedscan/configuration/GetPostScanJobElements"/>
  <wsdl:output message="dsc:GetPostScanJobElementsResponseMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
distributedscan/configuration/GetPostScanJobElementsResponse"/>
</wsdl:operation>
```

Upon receiving a [dsc:GetPostScanJobElementsRequestMsg](#) message, the server MUST respond with an [dsc:GetPostScanJobElementsResponseMsg](#) message or a SOAP fault, as follows:

- If the server cannot validate the arguments of the [dsc:GetPostScanJobElementsRequestMsg](#), the server MUST return a [dsc:InvalidArgs](#) SOAP fault. <11>
- If the server encounters an unexpected error from which it can recover while processing the [dsc:GetPostScanJobElementsRequestMsg](#), the server MUST return a [dsc:ServerErrorTemporaryError](#) SOAP fault.
- If the server encounters an unexpected error from which it cannot recover while processing the [dsc:GetPostScanJobElementsRequestMsg](#), the server MUST return a [dsc:ServerErrorInternalError](#) SOAP fault depending on the severity of the unexpected error.
- If the server does not have a PostScan job whose unique identifier matches the value specified in the [dsc:JobToken](#) element of the [dsc:GetPostScanJobElementsRequest](#) element, the server MUST return an [dsc:ClientErrorJobTokenNotFound](#) fault.
- If the server does have a PostScan job that matches the [dsc:JobToken](#) element of the [dsc:GetPostScanJobElementsRequest](#) element, the server MUST respond with an [dsc:GetPostScanJobElementsResponseMsg](#) message.
- The [dsc:GetPostScanJobElementsResponseMsg](#) element MUST have one [dsc:JobElements](#) child, which MUST contain as many [dsc:ElementData](#) children as there were [dsc:Name](#) children in the request message, and the [dsc:ElementData](#) children MUST be in the same order as the [dsc:Name](#) children in the request message.



- Each dsc:ElementData child corresponding to an dsc:Name child of the request's [dsc:RequestedElements](#) element MUST have the following characteristics:
  - The attributes of the dsc:ElementData child MUST be set as follows:
    - The **Name** attribute MUST be set to the same value as the value of the corresponding dsc:Name element of the request.
    - If the requested element is known by the server, the **Valid** attribute MUST be set to "true"; otherwise, if the requested element is not known by the server, the **Valid** attribute MUST be set to "false". [<12>](#)
    - Vendor-defined attributes MAY be added.
  - The contents of the element MUST be set as follows:
    - If the element is a status element, an [dsc:JobStatus](#) child element MUST be added.
    - If the element is a description element, an [dsc:JobDescription](#) element MUST be added.
    - If the element is a documents element, an [dsc:Documents](#) element MUST be added.
    - Vendor-defined child elements MAY be added.

### 3.2.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.2.4.3.1.1 GetPostScanJobElementsRequestMsg

The GetPostScanJobElementsRequestMsg message is the request WSDL message for a [GetPostScanJobElements](#) WSDL operation.

```
<wsdl:message name="GetPostScanJobElementsRequestMsg">
  <wsdl:part name="body" element="dsc:GetPostScanJobElementsRequest"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetPostScanJobElements
```

The SOAP body contains an [dsc:GetPostScanJobElementsRequest](#) element.

#### 3.2.4.3.1.2 GetPostScanJobElementsResponseMsg

The GetPostScanJobElementsResponseMsg message is the response WSDL message for a [GetPostScanJobElements](#) WSDL operation.

```
<wsdl:message name="GetPostScanJobElementsResponseMsg">
  <wsdl:part name="body" element="dsc:GetPostScanJobElementsResponse"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/  
configuration/GetPostScanJobElementsResponse
```

The SOAP body contains an [dsc:GetPostScanJobElementsResponse](#) element.

### 3.2.4.3.2 Elements

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

Element	Description
<a href="#">dsc:ElementData</a>	An element that contains response data for a <a href="#">GetPostScanJobElements</a> request.
<a href="#">dsc:GetPostScanJobElementsRequest</a>	An element that contains the input data for a <a href="#">GetPostScanJobElements</a> WSDL operation.
<a href="#">dsc:GetPostScanJobElementsResponse</a>	An element that contains the output data for a <a href="#">GetPostScanJobElements</a> WSDL operation.
<a href="#">dsc:Name</a>	An element that contains the qualified name for a PostScan job element. This name is used in the <a href="#">GetPostScanJobElements</a> request to specify the data the client is requesting.
<a href="#">dsc:JobElements</a>	A collection of <a href="#">dsc:ElementData</a> elements in an <a href="#">dsc:GetPostScanJobElements</a> response.
<a href="#">dsc:RequestedElements</a>	A collection of <a href="#">dsc:Name</a> elements that represent what information is to be returned in an <a href="#">dsc:GetPostScanJobElements</a> response.

#### 3.2.4.3.2.1 dsc:ElementData

The [dsc:ElementData](#) element is used for returning requested data in a [dsc:GetPostScanJobElements](#) response.

[dsc:ElementData](#) definition within [dsc:GetPostScanJobElements](#) response message is as follows.

```
<xs:element name="ElementData" type="dsc:JobElementDataType"  
maxOccurs="unbounded"/>
```

#### 3.2.4.3.2.2 dsc:GetPostScanJobElementsRequest

The [dsc:GetPostScanJobElementsRequest](#) element contains input data for an [dsc:GetPostScanJobElements](#) WSDL operation.

```
<xs:element name="GetPostScanJobElementsRequest"
```

```
type="dsc:GetPostScanJobElementsRequestType"/>
```

dsc:GetPostScanJobElementsRequest is an element of type [dsc:GetPostScanJobElementsRequestType](#).

### 3.2.4.3.2.3 dsc:GetPostScanJobElementsResponse

The dsc:GetPostScanJobElementsResponse element contains output data for an [dsc:GetPostScanJobElements](#) WSDL operation.

```
<xs:element name="GetPostScanJobElementsResponse" type="dsc:GetPostScanJobElementsResponseType"/>
```

dsc:GetPostScanJobElementsResponse is an element of type [dsc:GetPostScanJobElementsResponseType](#).

### 3.2.4.3.2.4 dsc:Name

The dsc:Name element contains the qualified name for a PostScan job element. This name is used in the [dsc:GetPostScanJobElements](#) request to specify the data the client is requesting.

```
<xs:element name="Name" type="xs:QName" maxOccurs="unbounded"/>
```

### 3.2.4.3.2.5 dsc:JobElements

The dsc:JobElements element contains job information elements requested by an [dsc:GetPostScanJobElements](#) request.

```
<xs:element name="JobElements" type="dsc:JobElementsType"/>
```

### 3.2.4.3.2.6 dsc:RequestedElements

The dsc:RequestedElements element contains a collection of [dsc:Name](#) elements that represent what information is to be returned for an [dsc:GetPostScanJobElements](#) request.

```
<xs:element name="RequestedElements" type="dsc:RequestedJobElementsType"/>
```

dsc:RequestedElements is an element of type [dsc:RequestedJobElementsType](#).

### 3.2.4.3.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

Complex Type	Description
<a href="#">dsc:GetPostScanJobElementsRequestType</a>	A complex type that contains the request data for an <a href="#">dsc:GetPostScanJobElements</a> WSDL operation

Complex Type	Description
<a href="#">dsc:GetPostScanJobElementsResponseType</a>	A complex type that contains the response data for an <a href="#">dsc:GetPostScanJobElements</a> WSDL operation
<a href="#">dsc:JobElementDataType</a>	A type that contains all the PostScan job elements that can be requested with a <a href="#">GetPostScanJobElements</a> request and attributes that represent an elements qualified name and if the element name requested is recognized
<a href="#">dsc:JobElementsType</a>	A type that defines a collection of one or more <a href="#">dsc:JobElementDataType</a> elements
<a href="#">dsc:RequestedJobElementsType</a>	A complex type that defines a collection of one or more elements of type <a href="#">xs:QName</a> (as specified in <a href="#">[XMLSCHEMA1]</a> and <a href="#">[XMLSCHEMA2]</a> ) that represent what information is to be returned in an <a href="#">dsc:GetPostScanJobElements</a> request

### 3.2.4.3.3.1 dsc:GetPostScanJobElementsRequestType

The [dsc:GetPostScanJobElementsRequestType](#) type contains the request data for an [dsc:GetPostScanJobElements](#) WSDL operation.

```
<xs:complexType name="GetPostScanJobElementsRequestType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
    <xs:element name="RequestedElements"
      type="dsc:RequestedJobElementsType"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:JobToken:** The unique identifier of the PostScan job for which the data is requested.

**dsc:RequestedElements:** The PostScan job data element(s) the client requests to be returned.

### 3.2.4.3.3.2 dsc:GetPostScanJobElementsResponseType

The [dsc:GetPostScanJobElementsResponseType](#) type contains the response data for an [dsc:GetPostScanJobElements](#) WSDL operation.

```
<xs:complexType name="GetPostScanJobElementsResponseType">
  <xs:sequence>
    <xs:element name="JobElements" type="dsc:JobElementsType"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:JobElements:** The PostScan job data element(s) returned in response to the request. Each data element MUST contain:

- The name of the element.
- The **Valid** attribute indicating whether the element name was recognized by the scan repository:

- If the element name is recognized by the scan repository, attribute equals "true".
- If the element name is not recognized by the scan repository, attribute equals "false".
- The requested element's data.

### 3.2.4.3.3.3 dsc:JobElementDataType

The dsc:JobElementDataType type contains all the PostScan job elements that can be requested with an [dsc:GetPostScanJobElements](#) request and attributes that represent an element's qualified name and whether the element name requested is recognized.

```
<xs:complexType name="JobElementDataType">
  <xs:choice>
    <xs:element name="JobStatus" type="dsc:JobStatusType"/>
    <xs:element name="JobDescription" type="dsc:JobDescriptionType"/>
    <xs:element name="Documents" type="dsc:DocumentsType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attribute ref="dsc:Name" use="required"/>
  <xs:attribute ref="dsc:Valid" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

### 3.2.4.3.3.4 dsc:JobElementsType

The dsc:JobElementsType type defines a collection of one or more [dsc:JobElementDataType](#) elements.

```
<xs:complexType name="JobElementsType">
  <xs:sequence>
    <xs:element name="ElementData" type="dsc:JobElementDataType"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

### 3.2.4.3.3.5 dsc:RequestedJobElementsType

The dsc:RequestedJobElementsType type defines a collection of one or more elements of type xs:QName (as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#)) that represent what information is to returned in an [dsc:GetPostScanJobElements](#) request.

```
<xs:complexType name="RequestedJobElementsType">
  <xs:sequence>
    <xs:element name="Name" type="xs:QName" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Predefined Values:

[dsc:JobStatus](#) - Get all the status information for the requested PostScan job.

[dsc:JobDescription](#) - Get all the descriptive information for the requested PostScan job.

[dsc:Documents](#) - Get the descriptive information for all scan documents in the requested PostScan job.

### 3.2.4.4 GetActiveJobs

The GetActiveJobs operation is used to query the scan repository for information about all currently processing PostScan jobs.

```
<wsdl:operation name="GetActiveJobs">
  <wsdl:input message="dsc:GetActiveJobsRequestMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
distributedscan/configuration/GetActiveJobs"/>
  <wsdl:output message="dsc:GetActiveJobsResponseMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
distributedscan/configuration/GetActiveJobsResponse"/>
</wsdl:operation>
```

Upon receiving an [dsc:GetActiveJobsRequestMsg](#) message, the server MUST respond with an [dsc:GetActiveJobsResponseMsg](#) message or a SOAP fault, as follows:

- If the server encounters an unexpected error from which it can recover while processing the [dsc:GetActiveJobsRequestMsg](#), the server MUST return a [dsc:ServerErrorTemporaryError](#) SOAP fault.
- If the server encounters an unexpected error from which it cannot recover while processing the [dsc:GetActiveJobsRequestMsg](#), the server MUST return a [dsc:ServerErrorInternalError](#) SOAP fault depending on the severity of the unexpected error.
- If the server can successfully process the [dsc:GetActiveJobsRequestMsg](#), the server MUST return a [dsc:GetActiveJobsResponseMsg](#) with the following requirements:
  - The [dsc:GetActiveJobsResponseMsg](#) MUST have one [dsc:ActiveJobs](#) child that MUST contain as many [dsc:JobSummary](#) children as there are PostScan jobs processing when the scan repository processes the [dsc:GetActiveJobsRequestMsg](#).
  - Each child corresponding to a processing PostScan job MUST have the following characteristics:
    - The contents of the element MUST be set as follows:
      - All the data elements of the [dsc:JobSummary](#) element MUST contain the corresponding data from the PostScan job.
      - Vendor-defined child elements MAY be added.

#### 3.2.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

### 3.2.4.4.1.1 GetActiveJobsRequestMsg

The GetActiveJobsRequestMsg message is a request WSDL message for a [GetActiveJobs](#) WSDL operation.

```
<wsdl:message name="GetActiveJobsRequestMsg">
  <wsdl:part name="body" element="dsc:GetActiveJobsRequest"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetActiveJobs
```

The SOAP body contains an [dsc:GetActiveJobsRequest](#) element.

### 3.2.4.4.1.2 GetActiveJobsResponseMsg

The GetActiveJobsResponseMsg message is the response WSDL message for a [GetActiveJobs](#) WSDL operation.

```
<wsdl:message name="GetActiveJobsResponseMsg">
  <wsdl:part name="body" element="dsc:GetActiveJobsResponse"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetActiveJobsResponse
```

The SOAP body contains an [dsc:GetActiveJobsResponse](#) element.

### 3.2.4.4.2 Elements

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

Element	Description
<a href="#">dsc:ActiveJobs</a>	A collection of zero or more <a href="#">dsc:JobSummary</a> elements that represent all currently active PostScan jobs in the server
<a href="#">dsc:GetActiveJobsRequest</a>	An element that contains the input data for a <a href="#">GetActiveJobs</a> WSDL operation
<a href="#">dsc:GetActiveJobsResponse</a>	An element that contains the output data for a <a href="#">GetActiveJobs</a> WSDL operation

#### 3.2.4.4.2.1 dsc:ActiveJobs

The dsc:ActiveJobs element is used to return the summary data for all active PostScan jobs.

```
<xs:element name="ActiveJobs" type="dsc:ListOfSummaryType"/>
```

#### 3.2.4.4.2.2 dsc:GetActiveJobsRequest

The dsc:GetActiveJobsRequest element contains input data for a [GetActiveJobs](#) WSDL operation.

```
<xs:element name="GetActiveJobsRequest"  
type="dsc: GetActiveJobsRequestType"/>
```

dsc:GetActiveJobsRequest is an element of type [dsc:GetActiveJobsRequestType](#).

#### 3.2.4.4.2.3 dsc:GetActiveJobsResponse

The dsc:GetActiveJobsResponse element contains output data for a [GetActiveJobs](#) WSDL operation.

```
<xs:element name="GetActiveJobsResponse"  
type="dsc:GetActiveJobsResponseType"/>
```

dsc:GetActiveJobsResponse is an element of type [dsc:GetActiveJobsResponseType](#).

#### 3.2.4.4.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

##### 3.2.4.4.3.1 dsc:GetActiveJobsRequestType

The dsc:GetActiveJobsRequestType type contains the request data for a [GetActiveJobs](#) WSDL operation.

```
<xs:complexType name="GetActiveJobsRequestType">  
  <xs:sequence>  
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

##### 3.2.4.4.3.2 dsc:GetActiveJobsResponseType

The dsc:GetActiveJobsResponseType type contains the response data for a [GetActiveJobs](#) WSDL operation.

```
<xs:complexType name="GetActiveJobsResponseType">  
  <xs:sequence>  
    <xs:element name="ActiveJobs" type="dsc:ListOfSummaryType"/>  
  </xs:sequence>  
</xs:complexType>
```



**dsc:ActiveJobs:** A collection of [dsc:JobSummaryType](#) elements that represent the PostScan jobs active in the scan repository when the GetActiveJobs operation was processed.

### 3.2.4.5 GetJobHistory

The GetJobHistory operation is used to query the scan repository for information about completed PostScan jobs.

```
<wsdl:operation name="GetJobHistory">
  <wsdl:input message="dsc:GetJobHistoryRequestMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/GetJobHistory"/>
  <wsdl:output message="dsc:GetJobHistoryResponseMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/GetJobHistoryResponse"/>
</wsdl:operation>
```

Upon receiving a [dsc:GetJobHistoryRequestMsg](#), the server MUST respond with an [dsc:GetJobHistoryResponseMsg](#) or a SOAP fault, as follows:

- If the server encounters an unexpected error from which it can recover while processing the [dsc:GetJobHistoryRequestMsg](#), the server MUST return a [dsc:ServerErrorTemporaryError](#) SOAP fault.
- If the server encounters an unexpected error from which it cannot recover while processing the [dsc:GetJobHistoryRequestMsg](#), the server MUST return a [dsc:ServerErrorInternalError](#) SOAP fault depending on the severity of the unexpected error.
- If the server can successfully process the [dsc:GetJobHistoryRequestMsg](#), the server MUST return a [dsc:GetJobHistoryResponseMsg](#) with the following requirements:
  - The [GetJobHistoryResponseMsg](#) MUST have one [dsc:JobHistory](#) child that MUST contain as many [dsc:JobSummary](#) children as there are completed PostScan jobs saved by the scan repository that received the [GetJobHistoryRequestMsg](#). The number of PostScan jobs saved is server dependent.
  - Each child corresponding to a completed PostScan job MUST have the following characteristics:
    - The contents of the element MUST be set as follows:
      - All the data elements of the [dsc:JobSummary](#) element MUST contain the corresponding data from the PostScan job.
      - Vendor-defined child elements MAY be added.

#### 3.2.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

##### 3.2.4.5.1.1 GetJobHistoryRequestMsg

The [GetJobHistoryRequestMsg](#) message is a request WSDL message for a [GetJobHistory](#) WSDL operation.

```
<wsdl:message name="GetJobHistoryRequestMsg">
  <wsdl:part name="body" element="dsc:GetJobHistoryRequest"/>
</wsdl:message>
```

The SOAP action value is:

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/configuration/GetJobHistory
```

The SOAP body contains an [dsc:GetJobHistoryRequest](#) element.

### 3.2.4.5.1.2 GetJobHistoryResponseMsg

The GetJobHistoryResponseMsg message is a response WSDL message for a [GetJobHistory](#) WSDL operation.

```
<wsdl:message name="GetJobHistoryResponseMsg">
  <wsdl:part name="body" element="dsc:GetJobHistoryResponse"/>
</wsdl:message>
```

The SOAP action value is:

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/GetJobHistoryResponse
```

The SOAP body contains an [dsc:GetJobHistoryResponse](#) element.

### 3.2.4.5.2 Elements

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

Element	Description
<a href="#">dsc:JobHistory</a>	A collection of <a href="#">dsc:JobSummary</a> elements that contain the summary information for completed PostScan jobs processed by the scan repository. This list does not necessarily contain all the jobs processed, just a subset of the most recent. This number kept is implementation dependent.
<a href="#">dsc:GetJobHistoryRequest</a>	An element that contains the input data for an <a href="#">dsc:GetJobHistory</a> WSDL operation.
<a href="#">dsc:GetJobHistoryResponse</a>	An element that contains the output data for an <a href="#">dsc:GetJobHistory</a> WSDL operation.

#### 3.2.4.5.2.1 dsc:JobHistory

The dsc:JobHistory element is a collection of [dsc:JobSummary](#) elements that contain the summary information for completed PostScan jobs processed by the scan repository. This list does not

necessarily contain all the jobs processed, just a subset of the most recent. This number kept is implementation dependent.

```
<xs:element name="JobHistory" type="dsc:ListOfSummaryType"/>
```

dsc:JobHistory is an element of type [dsc:ListOfSummaryType](#).

### 3.2.4.5.2.2 dsc:GetJobHistoryRequest

The dsc:GetJobHistoryRequest element contains input data for an [dsc:GetJobHistory](#) WSDL operation.

```
<xs:element name="GetJobHistoryRequest"
type="dsc:GetJobHistoryRequestType"/>
```

dsc:GetJobHistoryRequest is an element of type [dsc:GetJobHistoryRequestType](#).

### 3.2.4.5.2.3 dsc:GetJobHistoryResponse

The dsc:GetJobHistoryResponse element contains output data for an [dsc:GetJobHistory](#) WSDL operation.

```
<xs:element name="GetJobHistoryResponse"
type="dsc:GetJobHistoryResponseType"/>
```

dsc:GetJobHistoryResponse is an element of type [dsc:GetJobHistoryResponseType](#).

### 3.2.4.5.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

#### 3.2.4.5.3.1 dsc:GetJobHistoryRequestType

The dsc:GetJobHistoryRequestType type contains the request data for an [dsc:GetJobHistory](#) WSDL operation.

```
<xs:complexType name="GetJobHistoryRequestType">
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

#### 3.2.4.5.3.2 dsc:GetJobHistoryResponseType

The dsc:GetJobHistoryResponseType type contains the response data for an [dsc:GetJobHistory](#) WSDL operation.

```
<xs:complexType name="GetJobHistoryResponseType">
```

```

<xs:sequence>
  <xs:element name="JobHistory" type="dsc:ListOfSummaryType"/>
</xs:sequence>
</xs:complexType>

```

dsc:GetJobHistory: A collection of [dsc:JobSummaryType](#) elements that represent the PostScan jobs that have previously completed in the scan repository when the dsc:GetJobHistory operation was received. The depth of this list is server dependent.

### 3.2.4.6 RepositoryElementsChangeEvent

The RepositoryElementsChangeEvent event is sent by the scan repository any time an element or elements in the repository configuration changes. [<13>](#)

```

<wsdl:operation name="RepositoryElementsChangeEvent">
  <wsdl:output message="dsc:RepositoryElementsChangeEventMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/RepositoryElementsChangeEvent"/>
</wsdl:operation>

```

Upon detecting a change in the scan repository configuration, the server MUST send a [RepositoryElementsChangeEventMsg](#) message to all subscribed clients, as follows:

- RepositoryElementsChangeEventMsg MUST have one [dsc:ElementChanges](#) child that contains either an [dsc:RepositoryConfiguration](#) element or vendor-defined elements.
- The contents of the element(s) MUST be set as follows:
  - If the data change was detected in the scan repository configuration, an dsc:RepositoryConfiguration child element MUST be added.
  - If another data change was detected, vendor-defined child elements MAY be added.

#### 3.2.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

##### 3.2.4.6.1.1 RepositoryElementsChangeEventMsg

The RepositoryElementsChangeEventMsg message is an event WSDL message for a [RepositoryElementsChangeEvent](#) **WSDL Event**.

```

<wsdl:message name="RepositoryElementsChangeEventMsg">
  <wsdl:part name="body" element="dsc:RepositoryElementsChangeEvent"/>
</wsdl:message>

```

The SOAP action value is as follows.

```

http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/RepositoryElementsChangeEvent

```

The SOAP body contains a RepositoryElementsChangeEvent element.

### 3.2.4.6.2 Elements

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

Element	Description
<a href="#">dsc:ElementChanges</a>	An element in the <a href="#">RepositoryElementsChangeEvent</a> message that contains all of the current changes in the scan repository configuration.
<a href="#">RepositoryElementsChangeEvent</a>	An element that contains the event data for an RepositoryElementsChangeEvent WSDL operation.

#### 3.2.4.6.2.1 dsc:ElementChanges

The dsc:ElementChanges element is used to send changes in the scan repository configuration.

```
<xs:element name="ElementChanges" type="dsc:ElementChangesType"/>
```

#### 3.2.4.6.2.2 dsc:RepositoryElementsChangeEvent

The dsc:RepositoryElementsChangeEvent element contains event data for a [RepositoryElementsChangeEvent](#) WSDL operation.

```
<xs:element name="RepositoryElementsChangeEvent" type="dsc:RepositoryElementsChangeEvent"/>
```

RepositoryElementsChangeEvent is an element of type [dsc:RepositoryElementsChangeEvent](#).

### 3.2.4.6.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

Complex Type	Description
<a href="#">dsc:ElementChangesType</a>	A type that defines the elements of the scan repository that could change and produce a <a href="#">RepositoryElementsChangeEvent</a> .
<a href="#">dsc:RepositoryElementsChangeEvent</a>	A complex type that defines the event data for an <a href="#">dsc:RepositoryElementsChangeEvent</a> WSDL operation.

#### 3.2.4.6.3.1 dsc:ElementChangesType

The dsc:ElementChangesType type defines the elements of the scan repository that could change and produce an [RepositoryElementsChangeEvent](#) element.

```

<xs:complexType name="ElementChangesType">
  <xs:sequence>
    <xs:element name="RepositoryConfiguration"
      type="dsc:RepositoryConfigurationType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

### 3.2.4.6.3.2 dsc:RepositoryElementsChangeEvent

The `dsc:RepositoryElementsChangeEvent` type defines the event data for an [dsc:RepositoryElementsChangeEvent](#) WSDL operation.

```

<xs:complexType name="RepositoryElementsChangeEvent">
  <xs:sequence>
    <xs:element name="ElementChanges"
      type="dsc:ElementChangesType"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

**dsc:ElementChanges:** An element that contains any Repository elements in which a data change triggered the event.

### 3.2.4.7 RepositoryStatusSummaryEvent

The [dsc:RepositoryStatusSummaryEvent](#) event is sent by the scan repository any time the repository status changes.

```

<wsdl:operation name="RepositoryStatusSummaryEvent">
  <wsdl:output message="dsc:RepositoryStatusSummaryEventMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/RepositoryStatusSummaryEvent"/>
</wsdl:operation>

```

Upon detecting a change in the scan repository status, the server MUST send an [dsc:RepositoryStatusSummaryEventMsg](#) message to all subscribed clients, as follows:

- The `dsc:RepositoryStatusSummaryEventMsg` MUST have one [dsc:StatusSummary](#) child.
- The contents of the element MUST be set as follows:
  - An [dsc:RepositoryState](#) element MUST be added.
  - If one or more state reasons are available, an [dsc:RepositoryStateReasons](#) element MUST be added.
  - Vendor-defined child elements MAY be added.

### 3.2.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.2.4.7.1.1 dsc:RepositoryStatusSummaryEventMsg

The dsc:RepositoryStatusSummaryEventMsg message is an event WSDL message for an [dsc:RepositoryStatusSummaryEvent](#) WSDL Event.

```
<wsdl:message name="RepositoryStatusSummaryEventMsg">
  <wsdl:part name="body" element="dsc:RepositoryStatusSummaryEvent"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/RepositoryStatusSummaryEvent
```

The SOAP body contains an dsc:RepositoryStatusSummaryEvent element.

#### 3.2.4.7.2 Elements

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

Element	Description
<a href="#">dsc:RepositoryStatusSummaryEvent</a>	An element that contains the event data for an dsc:RepositoryStatusSummaryEvent WSDL operation.
<a href="#">dsc:StatusSummary</a>	An element of the dsc:RepositoryStatusSummaryEvent that contains the status of the scan repository at the time the event was triggered.

##### 3.2.4.7.2.1 dsc:RepositoryStatusSummaryEvent

The dsc:RepositoryStatusSummaryEvent element contains event data for an dsc:RepositoryStatusSummaryEvent WSDL operation.

```
<xs:element name="RepositoryStatusSummaryEvent"
  type="dsc:RepositoryStatusSummaryEventType"/>
```

[RepositoryStatusSummaryEvent](#) is an element of type dsc:RepositoryStatusSummaryEvent.

##### 3.2.4.7.2.2 dsc:StatusSummary

The dsc:StatusSummary element contains the status of the scan repository at the time the [dsc:RepositoryStatusSummaryEventType](#) was triggered.

```
<xs:element name="StatusSummary" type="dsc:StatusSummaryType"/>
```

### 3.2.4.7.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

Complex Type	Description
<a href="#">dsc:RepositoryStatusSummaryEventType</a>	A complex type that defines the event data for an <a href="#">dsc:RepositoryStatusSummaryEventType</a> WSDL operation
<a href="#">dsc:StatusSummaryType</a>	A type that defines the state elements of the scan repository that could change and produce an <a href="#">dsc:RepositoryStatusSummaryEventType</a>

#### 3.2.4.7.3.1 dsc:RepositoryStatusSummaryEventType

The [dsc:RepositoryStatusSummaryEventType](#) type defines the event data for an [dsc:RepositoryStatusSummaryEventType](#) WSDL operation.

```
<xs:complexType name="RepositoryStatusSummaryEventType">
  <xs:sequence>
    <xs:element name="StatusSummary" type="dsc:StatusSummaryType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:StatusSummary:** An element that contains the current scan repository state and state reasons.

#### 3.2.4.7.3.2 dsc:StatusSummaryType

The [dsc:StatusSummaryType](#) type defines the state elements of the scan repository that could change and produce an [dsc:RepositoryStatusSummaryEventType](#).

```
<xs:complexType name="StatusSummaryType">
  <xs:sequence>
    <xs:element name="RepositoryState" type="dsc:RepositoryStateType"/>
    <xs:element name="RepositoryStateReasons"
      type="dsc:RepositoryStateReasonsType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

### 3.2.4.8 PostScanJobStatusEvent

The [PostScanJobStatusEvent](#) event is sent by the scan repository any time the status of a [PostScanJob](#) changes.

```
<wsdl:operation name="PostScanJobStatusEvent">
  <wsdl:output message="dsc:PostScanJobStatusEventMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/PostScanJobStatusEvent"/>
</wsdl:operation>
```



Upon detecting a change in the status of a PostScan job, the server MUST send an [dsc:PostScanJobStatusEventMsg](#) message to all subscribed clients, as follows:

- The dsc:PostScanJobStatusEventMsg MUST have one [dsc:JobStatus](#) child.
- The contents of the element MUST be set as follows:
  - The current values of all PostScan job status related elements MUST be added.
  - Vendor-defined child elements MAY be added.

### 3.2.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.2.4.8.1.1 dsc:PostScanJobStatusEventMsg

The dsc:PostScanJobStatusEventMsg message is an event WSDL message for a PostScanJobStatusEventMsg WSDL Event.

```
<wsdl:message name="PostScanJobStatusEventMsg">
  <wsdl:part name="body" element="dsc:PostScanJobStatusEvent"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/PostScanJobStatusEvent
```

The SOAP body contains an [dsc:PostScanJobStatusEvent](#) element.

### 3.2.4.8.2 Elements

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

#### 3.2.4.8.2.1 dsc:PostScanJobStatusEvent

The dsc:PostScanJobStatusEvent element contains event data for an dsc:PostScanJobStatusEvent WSDL operation.

```
<xs:element name="PostScanJobStatusEvent"
  type="dsc:PostScanJobStatusEventType"/>
```

The dsc:PostScanJobStatusEvent element is of type [dsc:PostScanJobStatusEventType](#).

### 3.2.4.8.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

#### 3.2.4.8.3.1 dsc:PostScanJobStatusEventType

The dsc:PostScanJobStatusEventType type defines the event data for an [dsc:PostScanJobStatusEvent](#) WSDL operation.

```
<xs:complexType name="PostScanJobStatusEventType">
  <xs:sequence>
    <xs:element name="JobStatus" type="dsc:JobStatusType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:JobStatus:** An element that contains the current values of all PostScan job status related elements.

#### 3.2.4.9 PostScanJobEndStateEvent

The PostScanJobEndStateEvent event is sent by the scan repository every time a PostScan job completes.

```
<wsdl:operation name="PostScanJobEndStateEvent">
  <wsdl:output message="dsc:PostScanJobEndStateEventMsg"
    wsa:Action="http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/PostScanJobEndStateEvent"/>
</wsdl:operation>
```

Upon detecting the completion of a PostScan job, the server MUST send an [dsc:PostScanJobEndStateEventMsg](#) message to all subscribed clients, as follows:

- The dsc:PostScanJobEndStateEventMsg MUST have the following child elements:
  - [dsc:JobToken](#)
  - [dsc:PSP\\_Identifier](#)
  - [dsc:PSP\\_DisplayName](#)
  - [dsc:JobOriginatingUserName](#)
  - [dsc:JobCompletedState](#)
  - [dsc:JobCompletedStateReasons](#)
  - [dsc:FilterCompletedStatuses](#)
  - [dsc:ImagesReceived](#)
  - [dsc:JobCompletedTime](#)
- The contents of the elements MUST be set as follows:

- The final values of all the elements MUST be added.
- Vendor-defined child elements MAY be added.

### 3.2.4.9.1 Messages

The following WSDL message definitions are specific to this operation.

#### 3.2.4.9.1.1 dsc:PostScanJobEndStateEventMsg

The dsc:PostScanJobEndStateEventMsg message is an event WSDL message for a [PostScanJobEndStateEvent](#) WSDL Event.

```
<wsdl:message name="PostScanJobEndStateEventMsg">
  <wsdl:part name="body" element="dsc:PostScanJobEndStateEvent"/>
</wsdl:message>
```

The SOAP action value is as follows.

```
http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
configuration/PostScanJobEndStateEvent
```

The SOAP body contains an [dsc:PostScanJobEndStateEvent](#) element.

### 3.2.4.9.2 Elements

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

Element	Description
<a href="#">dsc:PostScanJobEndState</a>	An element that contains descriptive and final status information for a PostScan job
<a href="#">dsc:PostScanJobEndStateEvent</a>	An element that defines the event data for an dsc:PostScanJobEndStateEvent WSDL operation

#### 3.2.4.9.2.1 dsc:PostScanJobEndState

The dsc:PostScanJobEndState element contains descriptive and final status information for a PostScan job. This element is only sent within an [dsc:PostScanJobEndStateEvent](#) element.

```
<xs:element name="PostScanJobEndState"
  type="dsc:PostScanJobEndStateType"/>
```

#### 3.2.4.9.2.2 dsc:PostScanJobEndStateEvent

The dsc:PostScanJobEndStateEvent element contains event data for an dsc:PostScanJobEndStateEvent WSDL operation.

```
<xs:element name="PostScanJobEndStateEvent"
```

```
type="dsc:PostScanJobEndStateEventType"/>
```

dsc:PostScanJobEndStateEvent is an element of type [dsc:PostScanJobEndStateEventType](#).

### 3.2.4.9.3 Complex Types

The following XML schema complex type definitions are specific to this operation.

Complex Type	Description
<a href="#">dsc:PostScanJobEndStateType</a>	A complex type that defines descriptive and final status information for a PostScan job
<a href="#">dsc:PostScanJobEndStateEventType</a>	A complex type that defines the event data for an <a href="#">dsc:PostScanJobEndStateEvent</a> WSDL operation

#### 3.2.4.9.3.1 dsc:PostScanJobEndStateType

The dsc:PostScanJobEndStateType complex type defines descriptive and final status information for a PostScan job.

```
<xs:complexType name="PostScanJobEndStateType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
    <xs:element name="PSP_Identifier" type="dsc:String255ExtType"/>
    <xs:element name="PSP_DisplayName" type="dsc:String255ExtType"/>
    <xs:element name="JobOriginatingUserName" type="dsc:String255ExtType"/>
    <xs:element name="JobCompletedState" type="dsc:JobStateType"/>
    <xs:element name="JobCompletedStateReasons" type="dsc:JobStateReasonsType"/>
    <xs:element name="FilterCompletedStatuses" type="dsc:FilterStatusesType"/>
    <xs:element name="ImagesReceived" type="dsc:IntZeroExtType"/>
    <xs:element name="JobCompletedTime" type="dsc:DateTimeExtType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

#### 3.2.4.9.3.2 dsc:PostScanJobEndStateEventType

The dsc:PostScanJobEndStateEventType complex type defines the event data for an [dsc:PostScanJobEndStateEvent](#) WSDL operation.

```
<xs:complexType name="PostScanJobEndStateEventType">
  <xs:sequence>
    <xs:element name="PostScanJobEndState" type="dsc:PostScanJobEndStateType"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

**dsc:PostScanJobEndState:** An element that contains the final values of many PostScan job elements.

### 3.2.5 Timer Events

None.

### 3.2.6 Other Local Events

None.

## 3.3 RepositoryConfigurationServiceType Client 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.

## 4 Protocol Examples

### 4.1 WSDL Message Examples

A protocol client constructs the following WSDL message to query the scan repository server for its configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:WSA="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:DSC="http://schemas.microsoft.com/windows/2008/
  12/wdp/distributedscan/configuration">
  <S:Header>
    <WSA:MessageID>
      urn:uuid:5e00a5c1-29e7-4361-a42e-1f00013e973c
    </WSA:MessageID>
    <WSA:Action>
      http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/GetRepositoryElements
    </WSA:Action>
    <WSA:To>
      https://EDGUILLE64.enterprisescan1.nttest.microsoft.com:
      5362/ScanServer/553852db-6d5a-4eee-a34f-5b499c1a2944
    </WSA:To>
    <WSA:ReplyTo>
      <WSA:Address>
        http://schemas.xmlsoap.org/ws/2004/08/addressing/
        role/anonymous
      </WSA:Address>
    </WSA:ReplyTo>
  </S:Header>
  <S:Body>
    <DSC:GetRepositoryElementsRequest>
      <DSC:RequestedElements>
        <DSC:Name>
          DSC:RepositoryConfiguration
        </DSC:Name>
      </DSC:RequestedElements>
    </DSC:GetRepositoryElementsRequest>
  </S:Body>
</S:Envelope>
```

The protocol server would then respond with the following.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap=http://www.w3.org/2003/05/soap-envelope
  xmlns:wsa=http://schemas.xmlsoap.org/ws/2004/08/addressing
  xmlns:con=http://schemas.microsoft.com/windows/2008/
  12/wdp/distributedscan/configuration
  xmlns:dsc=http://schemas.microsoft.com/windows/2008/
  12/wdp/distributedscan/configuration
  xmlns:ftr="http://schemas.microsoft.com/windows/2007/
  10/imaging/postscan/filter">
  <soap:Header>
    <wsa:To>
```

```

        http://schemas.xmlsoap.org/ws/2004/
        08/addressing/role/anonymous
    </wsa:To>
    <wsa:Action>
        http://schemas.microsoft.com/windows/2008/12/wdp/
        distributedscan/configuration/GetRepositoryElementsResponse
    </wsa:Action>
    <wsa:MessageID>
        urn:uuid:4dbfd46d-e02a-4e03-8ad0-ab82c6bbbe77
    </wsa:MessageID>
    <wsa:RelatesTo>
        urn:uuid:5e00a5c1-29e7-4361-a42e-1f00013e973c
    </wsa:RelatesTo>
</soap:Header>
<soap:Body>
    <con:GetRepositoryElementsResponse>
        <con:RepositoryElements>
            <con:ElementData
                con:Name="dsc:RepositoryConfiguration"
                con:Valid="true">
                <con:RepositoryConfiguration>
                    <con:Filters>
                        <con:Filter>
                            <con:Dialect>
                                ftr:fileshare
                            </con:Dialect>
                            <con:FilterConfig/>
                        </con:Filter>
                        <con:Filter>
                            <con:Dialect>
                                ftr:sharepoint
                            </con:Dialect>
                            <con:FilterConfig/>
                        </con:Filter>
                        <con:Filter>
                            <con:Dialect>
                                ftr:email
                            </con:Dialect>
                            <con:FilterConfig/>
                        </con:Filter>
                    </con:Filters>
                </con:RepositoryConfiguration>
            </con:ElementData>
        </con:RepositoryElements>
    </con:GetRepositoryElementsResponse>
</soap:Body>
</soap:Envelope>

```

A protocol client constructs the following WSDL message to query the scan repository server for a list of the currently processing PostScan jobs.

```

<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
    xmlns:WSA=http://schemas.xmlsoap.org/ws/2004/08/addressing
    xmlns:DSC="http://schemas.microsoft.com/windows/2008/12/wdp/
    distributedscan/configuration">

```

```

<S:Header>
  <WSA:MessageID>
    urn:uuid:0eb870ee-f703-492a-8347-ba73a54e132d
  </WSA:MessageID>
  <WSA:Action>
    http://schemas.microsoft.com/windows/2008/12/wdp/distributedscan/
    configuration/GetActiveJobs
  </WSA:Action>
  <WSA:To>
    https://EDGUILLE64.enterprisescan1.nttest.microsoft.com:5362
    /ScanServer/553852db-6d5a-4eee-a34f-5b499c1a2944
  </WSA:To>
  <WSA:ReplyTo>
    <WSA:Address>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/
      role/anonymous
    </WSA:Address>
  </WSA:ReplyTo>
</S:Header>
<S:Body>
  <DSC:GetActiveJobsRequest />
</S:Body>
</S:Envelope>

```

If no PostScan jobs are processing, the protocol server would then respond with the following.

```

<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
  xmlns:WSA=http://schemas.xmlsoap.org/ws/2004/08/addressing
  xmlns:DSC="http://schemas.microsoft.com/windows/2008/12/wdp/
  distributedscan/configuration">
  <S:Header>
    <WSA:MessageID>
      urn:uuid:0eb870ee-f703-492a-8347-ba73a54e132d
    </WSA:MessageID>
    <WSA:Action>
      http://schemas.microsoft.com/windows/2008/12/wdp/
      distributedscan/configuration/GetActiveJobsResponse
    </WSA:Action>
    <WSA:To>
      https://EDGUILLE64.enterprisescan1.nttest.microsoft.com
      :5362/ScanServer/553852db-6d5a-4eee-a34f-5b499c1a2944
    </WSA:To>
    <WSA:ReplyTo>
      <WSA:Address>
        http://schemas.xmlsoap.org/ws/2004/08/addressing/
        role/anonymous
      </WSA:Address>
    </WSA:ReplyTo>
  </S:Header>
  <S:Body>
    <DSC:GetActiveJobsResponse />
  </S:Body>
</S:Envelope>

```



A protocol client constructs the following WSDL message to query the scan repository server for a list of the recently completed PostScan jobs.

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
  xmlns:WSA=http://schemas.xmlsoap.org/ws/2004/08/addressing
  xmlns:DSC="http://schemas.microsoft.com/windows/2008/12/wdp/
    distributedscan/configuration">
  <S:Header>
    <WSA:MessageID>
      urn:uuid:0eb870ee-f703-492a-8347-ba73a54e132d
    </WSA:MessageID>
    <WSA:Action>
      http://schemas.microsoft.com/windows/2008/12/wdp/
        distributedscan/configuration/GetJobHistory
    </WSA:Action>
    <WSA:To>
      https://EDGUILLE64.enterprisescan1.nttest.microsoft.com
        :5362/ScanServer/553852db-6d5a-4eee-a34f-5b499c1a2944
    </WSA:To>
    <WSA:ReplyTo>
      <WSA:Address>
        http://schemas.xmlsoap.org/ws/2004/08/addressing/
          role/anonymous
      </WSA:Address>
    </WSA:ReplyTo>
  </S:Header>
  <S:Body>
    <DSC:GetJobsHistoryRequest/>
  </S:Body>
</S:Envelope>
```

The protocol server would then respond with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap=http://www.w3.org/2003/05/soap-envelope
  xmlns:wsa=http://schemas.xmlsoap.org/ws/2004/08/addressing
  xmlns:con="http://schemas.microsoft.com/windows/2008/12/wdp/
    distributedscan/configuration">
  <soap:Header>
    <wsa:To>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/
        anonymous
    </wsa:To>
    <wsa:Action>
      http://schemas.microsoft.com/windows/2008/12/wdp/
        distributedscan/configuration/GetJobHistoryResponse
    </wsa:Action>
    <wsa:MessageID>
      urn:uuid:6ecae59e-93e1-42e4-affb-c63137b16992
    </wsa:MessageID>
    <wsa:RelatesTo>
      urn:uuid:3e26cab3-3759-45dc-a530-b6ea91e29e90
    </wsa:RelatesTo>
  </soap:Header>
  <soap:Body>
```

```

<con:GetJobHistoryResponse>
  <con:JobHistory>
    <con:JobSummary>
      <con:JobToken>
        6a8d7c92-0db3-4978-8aad-180530f8de53
      </con:JobToken>
      <con:PSP_Identifier>
        B6EB9213-C1BE-4BBF-880E-FF8F18A53469
      </con:PSP_Identifier>
      <con:PSP_DisplayName>
        EdTestAll
      </con:PSP_DisplayName>
      <con:JobOriginatingUserName>
        prtlab@ENTERPRISESCAN1
      </con:JobOriginatingUserName>
      <con:JobState>
        Aborted
      </con:JobState>
      <con:JobStateReasons>
        <con:JobStateReason>
          NoValidPostScanInstructionsFound
        </con:JobStateReason>
      </con:JobStateReasons>
      <con:FilterStatuses>
        <con:FilterStatus>
          <con:Dialect>
            http://schemas.microsoft.com/
            windows/2007/10/imaging/
            postscan/filter/email
          </con:Dialect>
          <con:FilterState>
            Canceled
          </con:FilterState>
        </con:FilterStatus>
      </con:FilterStatuses>
      <con:ImagesReceived>
        0
      </con:ImagesReceived>
    </con:JobSummary>
    <con:JobSummary>
      <con:JobToken>
        58e4590c-01e4-4933-a247-dc497b1d6c58
      </con:JobToken>
      <con:PSP_Identifier>
        B6EB9213-C1BE-4BBF-880E-FF8F18A53469
      </con:PSP_Identifier>
      <con:PSP_DisplayName>
        EdTestAll
      </con:PSP_DisplayName>
      <con:JobOriginatingUserName>
        prtlab@ENTERPRISESCAN1
      </con:JobOriginatingUserName>
      <con:JobState>
        Completed
      </con:JobState>
      <con:JobStateReasons>
        <con:JobStateReason>
          PostScanJobCompletedSuccessfully
        </con:JobStateReason>
    </con:JobSummary>
  </con:JobHistory>
</con:GetJobHistoryResponse>

```

```

        </con:JobStateReasons>
        <con:FilterStatuses>
            <con:FilterStatus>
                <con:Dialect>
                    http://schemas.microsoft.com/
                    windows/2007/10/imaging/
                    postscan/filter/sharepoint
                </con:Dialect>
                <con:FilterState>
                    CompletedSuccessfully
                </con:FilterState>
            </con:FilterStatus>
        </con:FilterStatuses>
        <con:ImagesReceived>
            0
        </con:ImagesReceived>
    </con:JobSummary>
</con:JobHistory>
</con:GetJobHistoryResponse>
</soap:Body>
</soap:Envelope>

```

## 4.2 SOAP Fault Example

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:dsc="http://schemas.microsoft.com/windows/2008/12/wdp/
  distributedscan/configuration">
  <soap:Header>
    <wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/addressing/fault</wsa:Action>
    <!-- Headers excluded for brevity -->
  </soap:Header>
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Receiver</soap:Value>
        <soap:Subcode>
          <soap:Value>dsc:ClientErrorJobTokenNotFound</soap:Value>
        </soap:Subcode>
      </soap:Code>
      <soap:Reason>
        <soap:Text xml:lang="en">A PostScan job identified by the
          specified dsc:JobToken argument could not be found.</soap:Text>
      </soap:Reason>
    </soap:Fault>
  </soap:Body></soap:Envelope>

```

## **5 Security**

### **5.1 Security Considerations for Implementers**

None.

### **5.2 Index of Security Parameters**

None.

## 6 Appendix A: Full WSDL

### 6.1 RepositoryConfigurationServiceType WSDL

For ease of implementation, the full WSDL is provided below.

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wSDP="http://schemas.xmlsoap.org/ws/2006/02/devprof"
  xmlns:WSP="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:WSD="http://schemas.xmlsoap.org/ws/2005/04/discovery"
  xmlns:WSE="http://schemas.xmlsoap.org/ws/2004/08/eventing"
  xmlns:WSA="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:dsc="http://schemas.microsoft.com/windows/2008/
    12/wdp/distributedscan/configuration"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/
    01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  targetNamespace="http://schemas.microsoft.com/
    windows/2008/12/wdp/distributedscan/configuration"
  name="WSDRepositoryConfigurationService">
  <xs:annotation>
    <xs:documentation>
      WSD Repository Configuration Service Web Service
      Description (WSDL) file
      Copyright 2007 Microsoft Corporation. All rights reserved
    </xs:documentation>
  </xs:annotation>
  <wsp:Policy wsu:Id="ServicePolicy">
    <wSDP:Profile/>
  </wsp:Policy>
  <types>
    <xs:schema targetNamespace="http://schemas.microsoft.com/
      windows/2008/12/wdp/distributedscan/configuration">
      <xs:include schemaLocation="WDPRepositoryConfig.xsd"/>
    </xs:schema>
  </types>
  <wSDL:message name="GetRepositoryElementsRequestMsg">
    <wSDL:part name="body"
      element="dsc:GetRepositoryElementsRequest"/>
  </wSDL:message>
  <wSDL:message name="GetRepositoryElementsResponseMsg">
    <wSDL:part name="body"
      element="dsc:GetRepositoryElementsResponse"/>
  </wSDL:message>
  <wSDL:message name="CancelPostScanJobRequestMsg">
    <wSDL:part name="body"
      element="dsc:CancelPostScanJobRequest"/>
  </wSDL:message>
  <wSDL:message name="CancelPostScanJobResponseMsg">
    <wSDL:part name="body"
      element="dsc:CancelPostScanJobResponse"/>
  </wSDL:message>
  <wSDL:message name="GetPostScanJobElementsRequestMsg">
    <wSDL:part name="body"
      element="dsc:GetPostScanJobElementsRequest"/>
  </wSDL:message>
```

```

</wsdl:message>
<wsdl:message name="GetPostScanJobElementsResponseMsg">
  <wsdl:part name="body"
    element="dsc:GetPostScanJobElementsResponse"/>
</wsdl:message>
<wsdl:message name="GetActiveJobsRequestMsg">
  <wsdl:part name="body"
    element="dsc:GetActiveJobsRequest"/>
</wsdl:message>
<wsdl:message name="GetActiveJobsResponseMsg">
  <wsdl:part name="body"
    element="dsc:GetActiveJobsResponse"/>
</wsdl:message>
<wsdl:message name="GetJobHistoryRequestMsg">
  <wsdl:part name="body"
    element="dsc:GetJobHistoryRequest"/>
</wsdl:message>
<wsdl:message name="GetJobHistoryResponseMsg">
  <wsdl:part name="body" element="dsc:GetJobHistoryResponse"/>
</wsdl:message>
<wsdl:message name="RepositoryElementsChangeEventMsg">
  <wsdl:part name="body"
    element="dsc:RepositoryElementsChangeEvent"/>
</wsdl:message>
<wsdl:message name="RepositoryStatusSummaryEventMsg">
  <wsdl:part name="body"
    element="dsc:RepositoryStatusSummaryEvent"/>
</wsdl:message>
<wsdl:message name="PostScanJobStatusEventMsg">
  <wsdl:part name="body"
    element="dsc:PostScanJobStatusEvent"/>
</wsdl:message>
<wsdl:message name="PostScanJobEndStateEventMsg">
  <wsdl:part name="body"
    element="dsc:PostScanJobEndStateEvent"/>
</wsdl:message>
<portType name="RepositoryConfigurationServiceType"
  wse:EventSource="true">
  <wsdl:operation name="GetRepositoryElements">
    <wsdl:input message="dsc:GetRepositoryElementsRequestMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
        12/wdp/distributedscan/configuration/
        GetRepositoryElements"/>
    <wsdl:output
      message="dsc:GetRepositoryElementsResponseMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
        12/wdp/distributedscan/configuration/
        GetRepositoryElementsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CancelPostScanJob">
    <wsdl:input message="dsc:CancelPostScanJobRequestMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
        12/wdp/distributedscan/configuration/CancelPostScanJob"/>
    <wsdl:output message="dsc:CancelPostScanJobResponseMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
        12/wdp/distributedscan/configuration/
        CancelPostScanJobResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetPostScanJobElements">

```

```

    <wsdl:input
      message="dsc:GetPostScanJobElementsRequestMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
GetPostScanJobElements"/>
    <wsdl:output
      message="dsc:GetPostScanJobElementsResponseMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
GetPostScanJobElementsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetActiveJobs">
    <wsdl:input message="dsc:GetActiveJobsRequestMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/GetActiveJobs"/>
    <wsdl:output message="dsc:GetActiveJobsResponseMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
GetActiveJobsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetJobHistory">
    <wsdl:input message="dsc:GetJobHistoryRequestMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/GetJobHistory"/>
    <wsdl:output message="dsc:GetJobHistoryResponseMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
GetJobHistoryResponse"/>
  </wsdl:operation>
  <wsdl:operation name="RepositoryElementsChangeEvent">
    <wsdl:output
      message="dsc:RepositoryElementsChangeEventMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
RepositoryElementsChangeEvent"/>
  </wsdl:operation>
  <wsdl:operation name="RepositoryStatusSummaryEvent">
    <wsdl:output
      message="dsc:RepositoryStatusSummaryEventMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
RepositoryStatusSummaryEvent"/>
  </wsdl:operation>
  <wsdl:operation name="PostScanJobStatusEvent">
    <wsdl:output message="dsc:PostScanJobStatusEventMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
PostScanJobStatusEvent"/>
  </wsdl:operation>
  <wsdl:operation name="PostScanJobEndStateEvent">
    <wsdl:output message="dsc:PostScanJobEndStateEventMsg"
      wsa:Action="http://schemas.microsoft.com/windows/2008/
12/wdp/distributedscan/configuration/
PostScanJobEndStateEvent"/>
  </wsdl:operation>
</portType>
<binding name="RepositoryConfigurationServiceBinding"
  type="dsc:RepositoryConfigurationServiceType">
  <wsoap12:binding style="document"

```

```

        transport="http://schemas.xmlsoap.org/soap/http"/>
<wsp:PolicyReference URI="#ServicePolicy"/>
<wsdl:operation name="GetRepositoryElements">
  <wsoap12:operation
    soapAction="http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/GetRepositoryElements"
    soapActionRequired="true"/>
  <wsdl:input>
    <wsoap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <wsoap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CancelPostScanJob">
  <wsoap12:operation
    soapAction="http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/CancelPostScanJob"
    soapActionRequired="true"/>
  <wsdl:input>
    <wsoap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <wsoap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetPostScanJobElements">
  <wsoap12:operation
    soapAction="http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/
      GetPostScanJobElements" soapActionRequired="true"/>
  <wsdl:input>
    <wsoap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <wsoap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetActiveJobs">
  <wsoap12:operation
    soapAction="http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/GetActiveJobs"
    soapActionRequired="true"/>
  <wsdl:input>
    <wsoap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <wsoap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetJobHistory">
  <wsoap12:operation
    soapAction="http://schemas.microsoft.com/windows/2008/
      12/wdp/distributedscan/configuration/GetJobHistory"
    soapActionRequired="true"/>
  <wsdl:input>
    <wsoap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>

```



```

        <wsoap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RepositoryElementsChangeEvent">
    <wsoap12:operation
        soapAction="http://schemas.microsoft.com/windows/2008/
            12/wdp/distributedscan/configuration/
            RepositoryElementsChangeEvent"
        soapActionRequired="true"/>
    <wsdl:output>
        <wsoap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RepositoryStatusSummaryEvent">
    <wsoap12:operation
        soapAction="http://schemas.microsoft.com/windows/2008/
            12/wdp/distributedscan/configuration/
            RepositoryStatusSummaryEvent"
        soapActionRequired="true"/>
    <wsdl:output>
        <wsoap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="PostScanJobStatusEvent">
    <wsoap12:operation
        soapAction="http://schemas.microsoft.com/windows/2008/
            12/wdp/distributedscan/configuration/
            PostScanJobStatusEvent" soapActionRequired="true"/>
    <wsdl:output>
        <wsoap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="PostScanJobEndStateEvent">
    <wsoap12:operation
        soapAction="http://schemas.microsoft.com/windows/2008/
            12/wdp/distributedscan/configuration/
            PostScanJobEndStateEvent" soapActionRequired="true"/>
    <wsdl:output>
        <wsoap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</binding>
<service name="RepositoryConfigurationService">
    <port name="RepositoryConfigurationPort"
        binding="dsc:RepositoryConfigurationServiceBinding">
        <wsoap12:address location="http://localhost/
            RepositoryConfigurationService/" />
    </port>
</service>
</definitions>

```

## 6.2 Full XSD

### 6.2.1 dsc

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<xs:schema xmlns:dsc="http://schemas.microsoft.com/windows/2008/12/
wdp/distributedscan/configuration"
xmlns:psp="http://schemas.microsoft.com/windows/2008/02/
imaging/postscan/psp"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/windows/
2008/12/wdp/distributedscan/configuration"
elementFormDefault="qualified">
  <xs:annotation>
    <xs:documentation>
      Distributed Scan Management -
      Scan Repository Configuration Web Service schema
      Copyright 2007 Microsoft Corp. All rights reserved
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://schemas.microsoft.com/windows/2008/
02/imaging/postscan/psp"
schemaLocation="PostScan.xsd"/>

  <xs:annotation>
    <xs:documentation>Extensions to basic element types to allow
IHV extensibility</xs:documentation>
  </xs:annotation>
  <xs:complexType name="String255ExtType">
    <xs:simpleContent>
      <xs:extension base="dsc:String255BaseType">
        <xs:anyAttribute namespace="##other"
processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="String255BaseType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="255"/>
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="IntOneExtType">
    <xs:simpleContent>
      <xs:extension base="dsc:IntOneBaseType">
        <xs:anyAttribute namespace="##other"
processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="IntOneBaseType">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="IntZeroExtType">
    <xs:simpleContent>
      <xs:extension base="dsc:IntZeroBaseType">
        <xs:anyAttribute namespace="##other"
processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>

```

```

</xs:complexType>
<xs:simpleType name="IntZeroBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="DateTimeExtType">
  <xs:simpleContent>
    <xs:extension base="xs:dateTime">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- Keyword extension federated by namespaces -->
<xs:simpleType name="KeywordNsExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+:[\w_\-\.]+"\>
  </xs:restriction>
</xs:simpleType>
<xs:annotation>
  <xs:documentation>
    Define the available SOAP Fault Subcodes
  </xs:documentation>
</xs:annotation>
<xs:simpleType name="FaultSubcodeValuesType">
  <xs:restriction base="xs:QName">
    <xs:enumeration value="dsc:InvalidArgs" />
    <xs:enumeration value="dsc:OperationFailed" />
    <xs:enumeration value="dsc:ClientErrorCurrentUserNotScanAdmin" />
    <xs:enumeration value="dsc:ClientErrorJobTokenNotFound" />
    <xs:enumeration value="dsc:ServerErrorInternalError" />
    <xs:enumeration value="dsc:ServerErrorTemporaryError" />
    <xs:enumeration value="dsc:ServerErrorPostScanJobCanceled" />
  </xs:restriction>
</xs:simpleType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that represent the Filters
    available on the Repository.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="FiltersType">
  <xs:sequence>
    <xs:element name="Filter" type="dsc:FilterType"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterType">
  <xs:sequence>
    <xs:element name="Dialect" type="xs:QName"/>
    <xs:element name="FilterConfig"
      type="dsc:FilterConfigType"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>

```

```

    </xs:sequence>
</xs:complexType>
<xs:complexType name="FilterConfigType">
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that represent the current
    Configuration of the Repository.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="RepositoryConfigurationType">
  <xs:sequence>
    <xs:element name="Filters" type="dsc:FiltersType"
      minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that represent the current basic
    State of the Repository.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="RepositoryStateType">
  <xs:simpleContent>
    <xs:extension base="dsc:RepositoryStateBaseType">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="RepositoryStateBaseType">
  <xs:union memberTypes="dsc:RepositoryStateWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="RepositoryStateWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="Idle"/>
    <xs:enumeration value="Processing"/>
    <xs:enumeration value="Stopped"/>
  </xs:restriction>
</xs:simpleType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that represent the detailed
    reasons for the State of the Repository.

```

```

    </xs:documentation>
</xs:annotation>
<xs:complexType name="RepositoryStateReasonsType">
  <xs:sequence>
    <xs:element name="RepositoryStateReason"
      type="dsc:RepositoryStateReasonType"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other"
    processContents="lax"/>
</xs:complexType>
<xs:complexType name="RepositoryStateReasonType">
  <xs:simpleContent>
    <xs:extension base="dsc:RepositoryStateReasonBaseType">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="RepositoryStateReasonBaseType">
  <xs:union memberTypes="dsc:RepositoryStateReasonWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="RepositoryStateReasonWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="None"/>
  </xs:restriction>
</xs:simpleType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that represent the current Status
    of the Repository.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="RepositoryStatusType">
  <xs:sequence>
    <xs:element name="RepositoryState"
      type="dsc:RepositoryStateType"/>
    <xs:element name="RepositoryStateReasons"
      type="dsc:RepositoryStateReasonsType"
      minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that will be used to request info
    from the Repository.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="RequestedRepositoryElementsType">
  <xs:sequence>
    <xs:element name="Name" type="xs:QName"

```

```

        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

```

<xs:annotation>
  <xs:documentation>
    Define the XML Elements that will contain the Repository
    information requested by a client.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="RepositoryElementDataType">
  <xs:choice>
    <xs:element name="RepositoryConfiguration"
      type="dsc:RepositoryConfigurationType"/>
    <xs:element name="RepositoryStatus"
      type="dsc:RepositoryStatusType"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attribute name="Name" type="xs:QName" use="required"
    form="qualified"/>
  <xs:attribute name="Valid" type="xs:boolean" use="required"
    form="qualified"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="RepositoryElementsType">
  <xs:sequence>
    <xs:element name="ElementData"
      type="dsc:RepositoryElementDataType"
      maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          Attribute: Name Attribute: Valid</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

```

```

<xs:annotation>
  <xs:documentation>Job Status Element
  definition</xs:documentation>
</xs:annotation>
<xs:complexType name="JobStatusType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
    <xs:element name="JobState" type="dsc:JobStateType"/>
    <xs:element name="JobStateReasons"
      type="dsc:JobStateReasonsType" minOccurs="0"/>
    <xs:element name="FilterStatuses"
      type="dsc:FilterStatusesType"/>
    <xs:element name="ImagesReceived"
      type="dsc:IntZeroExtType"/>
    <xs:element name="JobCreatedTime"

```

```

        type="dsc:DateTimeExtType" minOccurs="0"/>
      <xs:element name="JobCompletedTime"
        type="dsc:DateTimeExtType" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

  <xs:annotation>
    <xs:documentation>
      Job Status specific semantic elements
    </xs:documentation>
  </xs:annotation>
  <xs:complexType name="JobStateType">
    <xs:simpleContent>
      <xs:extension base="dsc:JobStateBaseType">
        <xs:anyAttribute namespace="##other"
          processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="JobStateBaseType">
    <xs:union memberTypes="dsc:JobStateWKVType
      dsc:KeywordNsExtensionPatternType"/>
  </xs:simpleType>
  <xs:complexType name="JobStateReasonsType">
    <xs:sequence>
      <xs:element name="JobStateReason"
        type="dsc:JobStateReasonType" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="JobStateReasonType">
    <xs:simpleContent>
      <xs:extension base="dsc:JobStateReasonBaseType">
        <xs:anyAttribute namespace="##other"
          processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="JobStateReasonBaseType">
    <xs:union memberTypes="dsc:JobStateReasonWKVType
      dsc:KeywordNsExtensionPatternType"/>
  </xs:simpleType>

  <xs:simpleType name="JobStateWKVType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:maxLength value="255"/>
      <xs:enumeration value="Aborted"/>
      <xs:enumeration value="Canceled"/>
      <xs:enumeration value="Completed"/>
      <xs:enumeration value="Creating"/>
      <xs:enumeration value="Pending"/>
      <xs:enumeration value="Pending-Held"/>
      <xs:enumeration value="Processing"/>
    </xs:restriction>
  </xs:simpleType>

```

```

        <xs:enumeration value="Started"/>
        <xs:enumeration value="Terminating"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="JobStateReasonWKVType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:maxLength value="255"/>
        <xs:enumeration value="None"/>
        <xs:enumeration value="PostScanJobTimedOut"/>
        <xs:enumeration value="NoValidPostScanInstructionsFound"/>
        <xs:enumeration value="CreatePostScanJobFailed"/>
        <xs:enumeration value="SendImageFailed"/>
        <xs:enumeration value="PostScanJobCanceled"/>
        <xs:enumeration value="PostScanJobProcessingFailed"/>
        <xs:enumeration value="PostScanJobCompletedSuccessfully"/>
        <xs:enumeration value="PostScanJobCompletedWithErrors"/>
        <xs:enumeration value="PostScanJobCompletedWithWarnings"/>
    </xs:restriction>
</xs:simpleType>

<xs:annotation>
    <xs:documentation>
        Job Description Element definition
    </xs:documentation>
</xs:annotation>
<xs:complexType name="JobDescriptionType">
    <xs:sequence>
        <xs:element name="PSP_Identifier"
            type="dsc:String255ExtType"/>
        <xs:element name="PSP_DisplayName"
            type="dsc:String255ExtType"/>
        <xs:element name="JobOriginatingUserName"
            type="dsc:String255ExtType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="FilterStatusesType">
    <xs:sequence>
        <xs:element name="FilterStatus"
            type="dsc:FilterStatusType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterStatusType">
    <xs:sequence>
        <xs:element name="Dialect"
            type="psp:anyURIExtType"/>
        <xs:element name="FilterState"
            type="dsc:FilterStateType"/>
        <xs:element name="FilterStateReasons"
            type="dsc:FilterStateReasonsType" minOccurs="0"/>
    </xs:sequence>

```



```
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

```
<xs:complexType name="FilterStateType">
  <xs:simpleContent>
    <xs:extension base="dsc:FilterStateBaseType">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="FilterStateBaseType">
  <xs:union memberTypes="dsc:FilterStateWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="FilterStateWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="Pending"/>
    <xs:enumeration value="Processing"/>
    <xs:enumeration value="Canceled"/>
    <xs:enumeration value="CompletedSuccessfully"/>
    <xs:enumeration value="CompletedWithErrors"/>
    <xs:enumeration value="CompletedWithWarnings"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:complexType name="FilterStateReasonsType">
  <xs:sequence>
    <xs:element name="FilterStateReason"
      type="dsc:FilterStateReasonType"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterStateReasonType">
  <xs:simpleContent>
    <xs:extension base="dsc:FilterStateReasonBaseType">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="FilterStateReasonBaseType">
  <xs:union memberTypes="dsc:FilterStateReasonWKVType
    dsc:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="FilterStateReasonWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="InvalidArgs"/>
    <xs:enumeration value="OutOfDiskSpace"/>
    <xs:enumeration value="ServerOutOfMemory"/>
    <xs:enumeration value="UnknownError"/>
  </xs:restriction>
</xs:simpleType>
```

```

        <xs:enumeration value="FileShareAccessDenied"/>
        <xs:enumeration value="FileShareOutOfDiskSpace"/>
        <xs:enumeration value="SharePointAccessDenied"/>
        <xs:enumeration value="SharePointOutOfDiskSpace"/>
        <xs:enumeration value="InvalidSMTPServer"/>
        <xs:enumeration value="UnableToSendEmail"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="JobSummaryType">
    <xs:sequence>
        <xs:element name="JobToken"
            type="dsc:String255ExtType"/>
        <xs:element name="PSP_Identifier"
            type="dsc:String255ExtType"/>
        <xs:element name="PSP_DisplayName"
            type="dsc:String255ExtType"/>
        <xs:element name="JobOriginatingUserName"
            type="dsc:String255ExtType"/>
        <xs:element name="JobState"
            type="dsc:JobStateType"/>
        <xs:element name="JobStateReasons"
            type="dsc:JobStateReasonsType" minOccurs="0"/>
        <xs:element name="FilterStatuses"
            type="dsc:FilterStatusesType"/>
        <xs:element name="ImagesReceived"
            type="dsc:IntZeroExtType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="ListOfSummarysType">
    <xs:sequence>
        <xs:element name="JobSummary" type="dsc:JobSummaryType"
            minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:annotation>
    <xs:documentation>
        Documents Element definition
    </xs:documentation>
</xs:annotation>
<xs:complexType name="DocumentsType">
    <xs:sequence>
        <xs:element name="Document" type="dsc:DocumentType"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

```

<xs:annotation>
  <xs:documentation>
    Document Element definition
  </xs:documentation>
</xs:annotation>
<xs:complexType name="DocumentType">
  <xs:sequence>
    <xs:element name="DocumentDescription"
      type="dsc:DocumentDescriptionType"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

```

<xs:annotation>
  <xs:documentation>
    Document Description Element Definition
  </xs:documentation>
</xs:annotation>
<xs:complexType name="DocumentDescriptionType">
  <xs:sequence>
    <xs:element name="DocumentId" type="dsc:IntOneExtType"/>
    <xs:element name="Format"
      type="dsc:DocumentFormatExtType"/>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="DocumentFormatExtType">
  <xs:simpleContent>
    <xs:extension base="dsc:DocumentFormatBaseType">
      <xs:anyAttribute namespace="##other"
        processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="DocumentFormatBaseType">
  <xs:union memberTypes="dsc:DocumentFormatWKVType
    dsc:MimeExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="DocumentFormatWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="dib"/>
    <xs:enumeration value="exif"/>
    <xs:enumeration value="jbig"/>
    <xs:enumeration value="jfif"/>
    <xs:enumeration value="jpeg2k"/>
    <xs:enumeration value="pdf-a"/>
    <xs:enumeration value="png"/>
    <xs:enumeration value="tiff-single-uncompressed"/>
    <xs:enumeration value="tiff-single-g4"/>
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="tiff-single-g3mh"/>
        <xs:enumeration value="tiff-single-jpeg-tn2"/>
        <xs:enumeration value="tiff-multi-uncompressed"/>
        <xs:enumeration value="tiff-multi-g4"/>
        <xs:enumeration value="tiff-multi-g3mh"/>
        <xs:enumeration value="tiff-multi-jpeg-tn2"/>
        <xs:enumeration value="xps"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MimeTypeExtensionPatternType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:pattern value="\w+[\w;/;_-\\|\.\]*/"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="RequestedJobElementsType">
    <xs:sequence>
        <xs:element name="Name" type="xs:QName"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="JobElementDataType">
    <xs:choice>
        <xs:element name="JobStatus" type="dsc:JobStatusType"/>
        <xs:element name="JobDescription"
            type="dsc:JobDescriptionType"/>
        <xs:element name="Documents" type="dsc:DocumentsType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attribute name="Name" type="xs:QName" use="required"
        form="qualified"/>
    <xs:attribute name="Valid" type="xs:boolean" use="required"
        form="qualified"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="JobElementsType">
    <xs:sequence>
        <xs:element name="ElementData"
            type="dsc:JobElementDataType" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>
                    Attribute: Name Attribute: Valid</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>

```

```
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

```
<xs:annotation>
  <xs:documentation>
    Define the XML Elements that will be the messages used in
    the GetRepositoryElements operation.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="GetRepositoryElementsRequestType">
  <xs:sequence>
    <xs:element name="RequestedElements"
      type="dsc:RequestedRepositoryElementsType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetRepositoryElementsRequest"
  type="dsc:GetRepositoryElementsRequestType"/>
<xs:complexType name="GetRepositoryElementsResponseType">
  <xs:sequence>
    <xs:element name="RepositoryElements"
      type="dsc:RepositoryElementsType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetRepositoryElementsResponse"
  type="dsc:GetRepositoryElementsResponseType"/>
```

```
<xs:annotation>
  <xs:documentation>
    Define the XML Elements that will be the messages used in
    the CancelPostScanJob operation.
  </xs:documentation>
</xs:annotation>
<xs:complexType name="CancelPostScanJobRequestType">
  <xs:sequence>
    <xs:element name="JobToken" type="dsc:String255ExtType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CancelPostScanJobRequest"
  type="dsc:CancelPostScanJobRequestType"/>
<xs:complexType name="CancelPostScanJobResponseType">
  <xs:sequence>
    <xs:any namespace="##other" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CancelPostScanJobResponse"
  type="dsc:CancelPostScanJobResponseType"/>
```

```
<xs:annotation>
  <xs:documentation>
```

```

        Define the XML Elements that will be the messages used in
        the GetJobElements operation.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="GetPostScanJobElementsRequestType">
    <xs:sequence>
        <xs:element name="JobToken" type="dsc:String255ExtType"/>
        <xs:element name="RequestedElements"
            type="dsc:RequestedJobElementsType"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetPostScanJobElementsRequest"
    type="dsc:GetPostScanJobElementsRequestType"/>
<xs:complexType name="GetPostScanJobElementsResponseType">
    <xs:sequence>
        <xs:element name="JobElements"
            type="dsc:JobElementsType"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetPostScanJobElementsResponse"
    type="dsc:GetPostScanJobElementsResponseType"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that will be the messages used in
        the GetActiveJobs operation.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="GetActiveJobsRequestType">
    <xs:sequence>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetActiveJobsRequest"
    type="dsc:GetActiveJobsRequestType"/>
<xs:complexType name="GetActiveJobsResponseType">
    <xs:sequence>
        <xs:element name="ActiveJobs"
            type="dsc:ListOfSummaryType"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetActiveJobsResponse"
    type="dsc:GetActiveJobsResponseType"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that will be the messages used in
        the GetJobHistory operation.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="GetJobHistoryRequestType">
    <xs:sequence>

```

```

        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetJobHistoryRequest"
    type="dsc:GetJobHistoryRequestType"/>
<xs:complexType name="GetJobHistoryResponseType">
    <xs:sequence>
        <xs:element name="JobHistory"
            type="dsc:ListOfSummariesType"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetJobHistoryResponse"
    type="dsc:GetJobHistoryResponseType"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that make up the
        RepositoryElementsChange Event.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="ElementChangesType">
    <xs:sequence>
        <xs:element name="RepositoryConfiguration"
            type="dsc:RepositoryConfigurationType"
            minOccurs="0"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="RepositoryElementsChangeEvent">
    <xs:sequence>
        <xs:element name="ElementChanges"
            type="dsc:ElementChangesType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="RepositoryElementsChangeEvent"
    type="dsc:RepositoryElementsChangeEvent"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that make up the
        RepositoryStatusSummary Event.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="StatusSummaryType">
    <xs:sequence>
        <xs:element name="RepositoryState"
            type="dsc:RepositoryStateType"/>
        <xs:element name="RepositoryStateReasons"
            type="dsc:RepositoryStateReasonsType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

```

```

        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="RepositoryStatusSummaryEventType">
    <xs:sequence>
        <xs:element name="StatusSummary"
            type="dsc:StatusSummaryType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="RepositoryStatusSummaryEvent"
    type="dsc:RepositoryStatusSummaryEventType"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that make up the
        PostScanJobStatus Event.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="PostScanJobStatusEventType">
    <xs:sequence>
        <xs:element name="JobStatus" type="dsc:JobStatusType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PostScanJobStatusEvent"
    type="dsc:PostScanJobStatusEventType"/>

<xs:annotation>
    <xs:documentation>
        Define the XML Elements that make up the
        PostScanJobEndState Event.
    </xs:documentation>
</xs:annotation>
<xs:complexType name="PostScanJobEndStateType">
    <xs:sequence>
        <xs:element name="JobToken"
            type="dsc:String255ExtType"/>
        <xs:element name="PSP_Identifier"
            type="dsc:String255ExtType"/>
        <xs:element name="PSP_DisplayName"
            type="dsc:String255ExtType"/>
        <xs:element name="JobOriginatingUserName"
            type="dsc:String255ExtType"/>
        <xs:element name="JobCompletedState"
            type="dsc:JobStateType"/>
        <xs:element name="JobCompletedStateReasons"
            type="dsc:JobStateReasonsType"/>
        <xs:element name="FilterCompletedStatuses"
            type="dsc:FilterStatusesType"/>
    </xs:sequence>
</xs:complexType>

```



```

        <xs:element name="ImagesReceived"
            type="dsc:IntZeroExtType"/>
        <xs:element name="JobCompletedTime"
            type="dsc:DateTimeExtType" minOccurs="0"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PostScanJobEndStateEventType">
    <xs:sequence>
        <xs:element name="PostScanJobEndState"
            type="dsc:PostScanJobEndStateType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PostScanJobEndStateEvent"
    type="dsc:PostScanJobEndStateEventType"/>
</xs:schema>

```

## 6.2.2 psp

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:psp="http://schemas.microsoft.com/windows/2008/
    02/imaging/postscan/psp"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:wscn="http://schemas.microsoft.com/windows/2006/
    08/wdp/scan"
    targetNamespace="http://schemas.microsoft.com/windows/2008/
    02/imaging/postscan/psp"
    elementFormDefault="qualified">
    <xs:annotation>
        <xs:documentation>
            WSD Enterprise Scanning -
            Post Scan Process schema
            Copyright 2007 Microsoft Corp. All rights reserved
        </xs:documentation>
    </xs:annotation>
    <xs:import namespace="http://schemas.microsoft.com/windows/2006/
    08/wdp/scan" schemaLocation="WDPScan.xsd"/>
    <xs:annotation>
        <xs:documentation>
            Define the XML Elements that represent the PostScan
            Process basic definition
        </xs:documentation>
    </xs:annotation>
    <xs:element name="PostScanProcess"
        type="psp:PostScanProcessType"/>
    <xs:complexType name="PostScanProcessType">
        <xs:sequence>
            <xs:element name="ScanRepository"
                type="psp:String255ExtType"/>
            <xs:element name="ScanTicket" type="wscn:ScanTicketType"/>
            <xs:element name="PostScanInstructions"
                type="psp:PostScanInstructionsType"/>

```

```

        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="PostScanInstructionsType">
    <xs:sequence>
        <xs:element name="ContinueOnError"
            type="psp:BoolExtType" minOccurs="0"/>
        <xs:element name="DocumentRootName"
            type="psp:String255ExtType"/>
        <xs:element name="FiltersToProcess"
            type="psp:FilterToProcessListType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterToProcessListType">
    <xs:sequence>
        <xs:element name="Filter"
            type="psp:FilterToProcessType" maxOccurs="unbounded"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterToProcessType">
    <xs:sequence>
        <xs:element name="Dialect" type="psp:anyURIExtType"/>
        <xs:element name="Instructions"
            type="psp:FilterInstructionType"/>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="FilterInstructionType">
    <xs:sequence>
        <xs:any namespace="##other" minOccurs="0"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:annotation>
    <xs:documentation>Extensions to basic element types to allow
        IHV extensibility</xs:documentation>
</xs:annotation>
<xs:complexType name="BoolExtType">
    <xs:simpleContent>
        <xs:extension base="xs:boolean">
            <xs:anyAttribute namespace="##other"
                processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="anyURIExtType">
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:anyAttribute namespace="##other"

```

```

        processContents="lax"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="String255ExtType">
    <xs:simpleContent>
        <xs:extension base="psp:String255BaseType">
            <xs:anyAttribute namespace="##other"
                processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="String255BaseType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="255"/>
        <xs:whiteSpace value="preserve"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

### 6.2.3 wscn

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/windows/2006/08/wdp/scan"
    xmlns:wscn="http://schemas.microsoft.com/windows/2006/08/wdp/scan"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <xs:annotation>
        <xs:documentation>Scan Ticket Element definition</xs:documentation>
    </xs:annotation>
    <xs:complexType name="ScanTicketType">
        <xs:sequence>
            <xs:element name="JobDescription" type="wscn:JobDescriptionType"/>
            <xs:element name="DocumentParameters" type="wscn:DocumentParametersType"
                minOccurs="0"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:annotation>
        <xs:documentation>Job Description Element definition</xs:documentation>
    </xs:annotation>
    <xs:complexType name="JobDescriptionType">
        <xs:sequence>
            <xs:element name="JobName" type="wscn:String255ExtType"/>
            <xs:element name="JobOriginatingUserName" type="wscn:String255ExtType"/>
            <xs:element name="JobInformation" type="wscn:String255ExtType" minOccurs="0"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:annotation>
        <xs:documentation>Document Parameters Element Definition</xs:documentation>

```

```

</xs:annotation>
<xs:complexType name="DocumentParametersType">
  <xs:sequence>
    <xs:element name="Format" type="wscn:DocumentFormatType" minOccurs="0"/>
    <xs:element name="CompressionQualityFactor"
      type="wscn:CompressionQualityFactorType" minOccurs="0"/>
    <xs:element name="ImagesToTransfer" type="wscn:ImagesToTransferType"
      minOccurs="0"/>
    <xs:element name="InputSource" type="wscn:DocumentInputSourceType"
      minOccurs="0"/>
    <xs:element name="FilmScanMode" type="wscn:FilmScanModeType" minOccurs="0"/>
    <xs:element name="ContentType" type="wscn:ContentTypeTicketType" minOccurs="0"/>
    <xs:element name="InputSize" type="wscn:DocumentInputSizeType" minOccurs="0"/>
    <xs:element name="Exposure" type="wscn:ScanExposureType" minOccurs="0"/>
    <xs:element name="Scaling" type="wscn:ScalingType" minOccurs="0"/>
    <xs:element name="Rotation" type="wscn:RotationType" minOccurs="0"/>
    <xs:element name="MediaSides" type="wscn:MediaSidesType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:annotation>
  <xs:documentation>Document Parameters specific semantic elements</xs:documentation>
</xs:annotation>
<xs:complexType name="DocumentFormatType">
  <xs:simpleContent>
    <xs:extension base="wscn:DocumentFormatExtType">
      <xs:attribute ref="wscn:MustHonor"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="DocumentFormatExtType">
  <xs:simpleContent>
    <xs:extension base="wscn:DocumentFormatBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="DocumentFormatBaseType">
  <xs:union memberTypes="wscn:DocumentFormatWKVType wscn:MimeExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="DocumentFormatWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:maxLength value="255"/>
    <xs:enumeration value="dib"/>
    <xs:enumeration value="exif"/>
    <xs:enumeration value="jbig"/>
    <xs:enumeration value="jfif"/>
    <xs:enumeration value="jpeg2k"/>
    <xs:enumeration value="pdf-a"/>
    <xs:enumeration value="png"/>
    <xs:enumeration value="tiff-single-uncompressed"/>
    <xs:enumeration value="tiff-single-g4"/>
    <xs:enumeration value="tiff-single-g3mh"/>
    <xs:enumeration value="tiff-single-jpeg-tn2"/>
    <xs:enumeration value="tiff-multi-uncompressed"/>
    <xs:enumeration value="tiff-multi-g4"/>
    <xs:enumeration value="tiff-multi-g3mh"/>
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="tiff-multi-jpeg-tn2"/>
        <xs:enumeration value="xps"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="CompressionQualityFactorType">
    <xs:simpleContent>
        <xs:extension base="wscn:CompressionQualityFactorExtType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="CompressionQualityFactorExtType">
    <xs:simpleContent>
        <xs:extension base="wscn:CompressionQualityFactorBaseType">
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="CompressionQualityFactorBaseType">
    <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="100"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="ImagesToTransferType">
    <xs:simpleContent>
        <xs:extension base="wscn:IntZeroExtType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="DocumentInputSourceType">
    <xs:simpleContent>
        <xs:extension base="wscn:DocumentInputSourceExtType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="DocumentInputSourceExtType">
    <xs:simpleContent>
        <xs:extension base="wscn:DocumentInputSourceBaseType">
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="DocumentInputSourceBaseType">
    <xs:union memberTypes="wscn:DocumentInputSourceWKVType
        wscn:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="DocumentInputSourceWKVType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:maxLength value="255"/>
        <xs:enumeration value="ADF"/>
        <xs:enumeration value="ADFDuplex"/>
        <xs:enumeration value="Film"/>
        <xs:enumeration value="Platen"/>
    </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="FilmScanModeType">
    <xs:simpleContent>
      <xs:extension base="wscn:FilmScanModeExtType">
        <xs:attribute ref="wscn:MustHonor"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="FilmScanModeExtType">
    <xs:simpleContent>
      <xs:extension base="wscn:FilmScanModeBaseType">
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="FilmScanModeBaseType">
    <xs:union memberTypes="wscn:FilmScanModeWKVType wscn:KeywordNsExtensionPatternType"/>
  </xs:simpleType>
  <xs:simpleType name="FilmScanModeWKVType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:maxLength value="255"/>
      <xs:enumeration value="NotApplicable"/>
      <xs:enumeration value="ColorSlideFilm"/>
      <xs:enumeration value="ColorNegativeFilm"/>
      <xs:enumeration value="BlackandWhiteNegativeFilm"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="ContentTypeTicketType">
    <xs:simpleContent>
      <xs:extension base="wscn:ContentTypeExtType">
        <xs:attribute ref="wscn:MustHonor"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="ContentTypeExtType">
    <xs:simpleContent>
      <xs:extension base="wscn:ContentTypeBaseType">
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="ContentTypeBaseType">
    <xs:union memberTypes="wscn:ContentTypeWKVType wscn:KeywordNsExtensionPatternType"/>
  </xs:simpleType>
  <xs:simpleType name="ContentTypeWKVType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:maxLength value="255"/>
      <xs:enumeration value="Auto"/>
      <xs:enumeration value="Text"/>
      <xs:enumeration value="Photo"/>
      <xs:enumeration value="Halftone"/>
      <xs:enumeration value="Mixed"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="DocumentInputSizeType">

```

```

<xs:choice>
  <xs:element name="DocumentSizeAutoDetect" type="wscn:BoolExtType"/>
  <xs:element name="InputMediaSize" type="wscn:DimensionsType"/>
</xs:choice>
<xs:attribute ref="wscn:MustHonor"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="DimensionsType">
  <xs:sequence>
    <xs:element name="Width" type="wscn:IntOneExtType"/>
    <xs:element name="Height" type="wscn:IntOneExtType"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="ScanExposureType">
  <xs:choice>
    <xs:element name="AutoExposure" type="wscn:BoolExtType"/>
    <xs:element name="ExposureSettings" type="wscn:ExposureSettingsType"/>
  </xs:choice>
  <xs:attribute ref="wscn:MustHonor"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="ExposureSettingsType">
  <xs:sequence>
    <xs:element name="Contrast" type="wscn:Adjustment1000Type" minOccurs="0"/>
    <xs:element name="Brightness" type="wscn:Adjustment1000Type" minOccurs="0"/>
    <xs:element name="Sharpness" type="wscn:Adjustment100Type" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="Adjustment1000Type">
  <xs:simpleContent>
    <xs:extension base="wscn:Adjustment1000BaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="Adjustment1000BaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-1000"/>
    <xs:maxInclusive value="1000"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Adjustment100Type">
  <xs:simpleContent>
    <xs:extension base="wscn:Adjustment100BaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="Adjustment100BaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="-100"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:complexType name="ScalingType">
  <xs:sequence>
    <xs:element name="ScalingWidth" type="wscn:ScalingRangeType"/>
    <xs:element name="ScalingHeight" type="wscn:ScalingRangeType"/>
  </xs:sequence>
  <xs:attribute ref="wscn:MustHonor"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="ScalingRangeType">
  <xs:simpleContent>
    <xs:extension base="wscn:ScalingRangeBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="ScalingRangeBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="1000"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="RotationType">
  <xs:simpleContent>
    <xs:extension base="wscn:RotationExtType">
      <xs:attribute ref="wscn:MustHonor"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="RotationExtType">
  <xs:simpleContent>
    <xs:extension base="wscn:RotationBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="RotationBaseType">
  <xs:union memberTypes="wscn:RotationWKVType wscn:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="RotationWKVType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="0"/>
    <xs:enumeration value="90"/>
    <xs:enumeration value="180"/>
    <xs:enumeration value="270"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="MediaSidesType">
  <xs:sequence>
    <xs:element name="MediaFront" type="wscn:MediaSideType"/>
    <xs:element name="MediaBack" type="wscn:MediaSideType" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="MediaSideType">
  <xs:sequence>
    <xs:element name="ScanRegion" type="wscn:ScanRegionType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```



```

        <xs:element name="ColorProcessing" type="wscn:ColorJobType" minOccurs="0"/>
        <xs:element name="Resolution" type="wscn:ResolutionPairExtType" minOccurs="0"/>
        <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="ScanRegionType">
    <xs:sequence>
        <xs:element name="ScanRegionXOffset" type="wscn:OffsetExtType" minOccurs="0"/>
        <xs:element name="ScanRegionYOffset" type="wscn:OffsetExtType" minOccurs="0"/>
        <xs:element name="ScanRegionWidth" type="wscn:DimensionExtType"/>
        <xs:element name="ScanRegionHeight" type="wscn:DimensionExtType"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="OffsetExtType">
    <xs:simpleContent>
        <xs:extension base="wscn:IntZeroExtType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="DimensionExtType">
    <xs:simpleContent>
        <xs:extension base="wscn:IntOneExtType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="ColorJobType">
    <xs:simpleContent>
        <xs:extension base="wscn:ColorConfigType">
            <xs:attribute ref="wscn:MustHonor"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="ColorConfigType">
    <xs:simpleContent>
        <xs:extension base="wscn:ColorConfigBaseType">
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="ColorConfigBaseType">
    <xs:union memberTypes="wscn:ColorWKVType wscn:KeywordNsExtensionPatternType"/>
</xs:simpleType>
<xs:simpleType name="ColorWKVType">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="BlackAndWhite1"/>
        <xs:enumeration value="Grayscale4"/>
        <xs:enumeration value="Grayscale8"/>
        <xs:enumeration value="Grayscale16"/>
        <xs:enumeration value="RGB24"/>
        <xs:enumeration value="RGB48"/>
        <xs:enumeration value="RGBa32"/>
        <xs:enumeration value="RGBa64"/>
    </xs:restriction>

```

```

</xs:simpleType>

<xs:complexType name="ResolutionPairExtType">
  <xs:sequence>
    <xs:element name="Width" type="wscn:IntOneExtType"/>
    <xs:element name="Height" type="wscn:IntOneExtType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="wscn:MustHonor"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- Keyword extension federated by namespaces -->
<xs:simpleType name="KeywordNsExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+:[\w_-\.\.]+"/>
  </xs:restriction>
</xs:simpleType>
<!-- Media type Keyword extension federated by MIME type names -->
<xs:simpleType name="MimeExtensionPatternType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="\w+[\w:/;_-\.\.\.]+\."/>
  </xs:restriction>
</xs:simpleType>

<xs:annotation>
  <xs:documentation>Common attributes and Types used by semantic
elements</xs:documentation>
</xs:annotation>
<xs:attribute name="MustHonor" type="xs:boolean" default="false"/>

<xs:annotation>
  <xs:documentation>Basic element types used in more than one sub-
schema</xs:documentation>
</xs:annotation>
<xs:complexType name="BoolExtType">
  <xs:simpleContent>
    <xs:extension base="xs:boolean">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="IntZeroExtType">
  <xs:simpleContent>
    <xs:extension base="wscn:IntZeroBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="IntZeroBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="IntOneExtType">
  <xs:simpleContent>
    <xs:extension base="wscn:IntOneBaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>

```

```
</xs:complexType>
<xs:simpleType name="IntOneBaseType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="String255ExtType">
  <xs:simpleContent>
    <xs:extension base="wscn:String255BaseType">
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="String255BaseType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="255"/>
    <xs:whiteSpace value="preserve"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

## 7 Appendix B: Product Behavior

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

- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

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

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

<1> [Section 1.8](#): Windows does not extend any of these types.

<2> [Section 2.2.2.1](#): The Windows Server 2008 R2 scan server repository only responds with the following standard errors:

- [wsa:ActionNotSupported](#)
- [dsc:InvalidArgs](#)

<3> [Section 2.2.3.28](#): Windows Server 2008 R2 scan server uses **GUIDs** for the JobToken.

<4> [Section 2.2.5.11](#): Windows Server 2008 R2 scan server does not return a value of "PostScanJobCompletedWithWarnings".

<5> [Section 2.2.5.12](#): Windows Server 2008 R2 scan server does not return a value of "Terminating".

<6> [Section 2.2.5.18](#): Windows Server 2008 R2 scan server does not return a value of "Stopped".

<7> [Section 3.1.4.1.4](#): The Windows Server 2008 R2 scan server does not return the standard error [dsc:ServerErrorTemporaryError](#).

<8> [Section 3.1.4.1.6](#): The Windows Server 2008 R2 scan server does not return the [dsc:JobToken](#) value in the detail element of the [dsc:ClientErrorJobTokenNotFound](#) SOAP fault.

<9> [Section 3.2.4.1](#): Windows Server 2008 R2 scan server will ignore the [dsc:GetRepositoryElementsRequestMsg](#) if it does not recognize the element requested in the [dsc:Name](#) element. Windows Server 2008 R2 scan server will issue a [dsc:InvalidArgs](#) SOAP fault to the [dsc:GetRepositoryElementsRequestMsg](#) if the element requested in the [dsc:Name](#) element is part of the current namespace but not [dsc:RepositoryConfiguration](#) or [dsc:RepositoryStatus](#).

<10> [Section 3.2.4.2](#): Windows Server 2008 R2 scan server will issue a [wsa:EndpointUnavailable](#) SOAP fault to the [dsc:CancelPostScanJobRequestMsg](#) if the [dsc:JobToken](#) element is not a correctly formatted GUID.

<11> [Section 3.2.4.3](#): Windows Server 2008 R2 scan server will issue a wsa:EndpointUnavailable SOAP fault to the [dsc:GetPostScanJobElementsRequestMsg](#) if the [dsc:JobToken](#) element is not a correctly formatted GUID.

<12> [Section 3.2.4.3](#): Windows Server 2008 R2 scan server will ignore the [dsc:GetPostScanJobElementsRequestMsg](#) if it does not recognize the element requested in the [dsc:Name](#) element.

<13> [Section 3.2.4.6](#): Windows Server 2008 R2 scan server does not generate the [RepositoryElementsChangeEvent](#) event.

## 8 Change Tracking

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

## 9 Index

### A

[Abstract data model - RepositoryConfigurationServiceType client](#) 43  
Abstract data model - RepositoryConfigurationServiceType server ([section 3.1.1](#) 43, [section 3.2.1](#) 45)  
[Applicability](#) 11  
[Attribute groups](#) 42  
Attributes  
    [Name](#) 42  
    [overview](#) 41  
    [Valid](#) 42

### C

[Capability negotiation](#) 11  
[Change tracking](#) 118  
Client - RepositoryConfigurationServiceType  
    [abstract data model](#) 43  
    initialization ([section 3.1.3](#) 43, [section 3.3](#) 77)  
    [overview](#) 77  
    SOAP faults ([section 3.1.4.1](#) 43, [section 3.3](#) 77)  
    timers ([section 3.1.2](#) 43, [section 3.3](#) 77)  
Complex types  
    [dsc:DateTimeExtType](#) 24  
    [dsc:DocumentDescriptionType](#) 25  
    [dsc:DocumentFormatExtType](#) 25  
    [dsc:DocumentsType](#) 25  
    [dsc:DocumentType](#) 25  
    [dsc:FilterConfigType](#) 26  
    [dsc:FilterStateReasonsType](#) 26  
    [dsc:FilterStateReasonType](#) 26  
    [dsc:FilterStateType](#) 26  
    [dsc:FilterStatusesType](#) 27  
    [dsc:FilterStatusType](#) 27  
    [dsc:FiltersType](#) 27  
    [dsc:FilterType](#) 27  
    [dsc:IntOneExtType](#) 28  
    [dsc:IntZeroExtType](#) 28  
    [dsc:JobDescriptionType](#) 28  
    [dsc:JobStateReasonsType](#) 29  
    [dsc:JobStateReasonType](#) 29  
    [dsc:JobStateType](#) 29  
    [dsc:JobStatusType](#) 29  
    [dsc:JobSummaryType](#) 30  
    [dsc:ListOfSummariesType](#) 28  
    [dsc:RepositoryConfigurationType](#) 30  
    [dsc:RepositoryStateReasonsType](#) 31  
    [dsc:RepositoryStateReasonType](#) 31  
    [dsc:RepositoryStateType](#) 30  
    [dsc:RepositoryStatusType](#) 31  
    [dsc:String255ExtType](#) 31  
    [overview](#) 23  
    [psp:anyURIExtType](#) 24

### D

[Data model - abstract - client](#) 43

Data model - abstract - server ([section 3.1.1](#) 43, [section 3.2.1](#) 45)  
[dsc:DateTimeExtType complex type](#) 24  
[dsc:Dialect element](#) 17  
[dsc:Document element](#) 17  
[dsc:DocumentDescription element](#) 17  
[dsc:DocumentDescriptionType complex type](#) 25  
[dsc:DocumentFormatBaseType simple type](#) 33  
[dsc:DocumentFormatExtType complex type](#) 25  
[dsc:DocumentFormatWKVType simple type](#) 33  
[dsc:DocumentId element](#) 18  
[dsc:Documents element](#) 17  
[dsc:DocumentsType complex type](#) 25  
[dsc:DocumentType complex type](#) 25  
[dsc:Filter element](#) 18  
[dsc:FilterCompletedStatuses element](#) 18  
[dsc:FilterConfig element](#) 18  
[dsc:FilterConfigType complex type](#) 26  
[dsc:Filters element](#) 18  
[dsc:FilterState element](#) 18  
[dsc:FilterStateBaseType simple type](#) 34  
[dsc:FilterStateReason element](#) 18  
[dsc:FilterStateReasonBaseType simple type](#) 35  
[dsc:FilterStateReasons element](#) 19  
[dsc:FilterStateReasonsType complex type](#) 26  
[dsc:FilterStateReasonType complex type](#) 26  
[dsc:FilterStateReasonWKVType simple type](#) 35  
[dsc:FilterStateType complex type](#) 26  
[dsc:FilterStateWKVType simple type](#) 36  
[dsc:FilterStatus element](#) 19  
[dsc:FilterStatuses element](#) 19  
[dsc:FilterStatusesType complex type](#) 27  
[dsc:FilterStatusType complex type](#) 27  
[dsc:FiltersType complex type](#) 27  
[dsc:FilterType complex type](#) 27  
[dsc:Format element](#) 19  
[dsc:ImagesReceived element](#) 19  
[dsc:IntOneBaseType simple type](#) 36  
[dsc:IntOneExtType complex type](#) 28  
[dsc:IntZeroBaseType simple type](#) 37  
[dsc:IntZeroExtType complex type](#) 28  
[dsc:JobCompletedState element](#) 19  
[dsc:JobCompletedStateReasons element](#) 20  
[dsc:JobCompletedTime element](#) 20  
[dsc:JobCreatedTime element](#) 20  
[dsc:JobDescription element](#) 20  
[dsc:JobDescriptionType complex type](#) 28  
[dsc:JobOriginatingUserName element](#) 20  
[dsc:JobState element](#) 20  
[dsc:JobStateBaseType simple type](#) 37  
[dsc:JobStateReason element](#) 21  
[dsc:JobStateReasonBaseType simple type](#) 37  
[dsc:JobStateReasons element](#) 21  
[dsc:JobStateReasonsType complex type](#) 29  
[dsc:JobStateReasonType complex type](#) 29  
[dsc:JobStateReasonWKVType simple type](#) 37  
[dsc:JobStateType complex type](#) 29  
[dsc:JobStateWKVType simple type](#) 39  
[dsc:JobStatus element](#) 21

[dsc:JobStatusType complex type](#) 29  
[dsc:JobSummary element](#) 21  
[dsc:JobSummaryType complex type](#) 30  
[dsc:JobToken element](#) 21  
[dsc:KeywordNsExtensionPatternType simple type](#)  
 39  
[dsc:ListOfSummarysType complex type](#) 28  
[dsc:MimeExtensionPatternType simple type](#) 40  
[dsc:PSP\\_DisplayName element](#) 21  
[dsc:PSP\\_Identifier element](#) 22  
[dsc:RepositoryConfiguration element](#) 22  
[dsc:RepositoryConfigurationType complex type](#) 30  
[dsc:RepositoryState element](#) 22  
[dsc:RepositoryStateBaseType simple type](#) 40  
[dsc:RepositoryStateReason element](#) 22  
[dsc:RepositoryStateReasonBaseType simple type](#) 40  
[dsc:RepositoryStateReasons element](#) 22  
[dsc:RepositoryStateReasonsType complex type](#) 31  
[dsc:RepositoryStateReasonType complex type](#) 31  
[dsc:RepositoryStateReasonWKVType simple type](#) 40  
[dsc:RepositoryStateType complex type](#) 30  
[dsc:RepositoryStateWKVType simple type](#) 41  
[dsc:RepositoryStatus element](#) 22  
[dsc:RepositoryStatusType complex type](#) 31  
[dsc:String255BaseType simple type](#) 41  
[dsc:String255ExtType complex type](#) 31

## E

### Elements

[dsc:Dialect](#) 17  
[dsc:Document](#) 17  
[dsc:DocumentDescription](#) 17  
[dsc:DocumentId](#) 18  
[dsc:Documents](#) 17  
[dsc:Filter](#) 18  
[dsc:FilterCompletedStatuses](#) 18  
[dsc:FilterConfig](#) 18  
[dsc:Filters](#) 18  
[dsc:FilterState](#) 18  
[dsc:FilterStateReason](#) 18  
[dsc:FilterStateReasons](#) 19  
[dsc:FilterStatus](#) 19  
[dsc:FilterStatuses](#) 19  
[dsc:Format](#) 19  
[dsc:ImagesReceived](#) 19  
[dsc:JobCompletedState](#) 19  
[dsc:JobCompletedStateReasons](#) 20  
[dsc:JobCompletedTime](#) 20  
[dsc:JobCreatedTime](#) 20  
[dsc:JobDescription](#) 20  
[dsc:JobOriginatingUserName](#) 20  
[dsc:JobState](#) 20  
[dsc:JobStateReason](#) 21  
[dsc:JobStateReasons](#) 21  
[dsc:JobStatus](#) 21  
[dsc:JobSummary](#) 21  
[dsc:JobToken](#) 21  
[dsc:PSP\\_DisplayName](#) 21  
[dsc:PSP\\_Identifier](#) 22  
[dsc:RepositoryConfiguration](#) 22  
[dsc:RepositoryState](#) 22

[dsc:RepositoryStateReason](#) 22  
[dsc:RepositoryStateReasons](#) 22  
[dsc:RepositoryStatus](#) 22

### Error codes

#### Events

[local - RepositoryConfigurationServiceType server](#)  
 77

[timer - RepositoryConfigurationServiceType server](#) 77

#### Examples

[overview](#) 78  
[SOAP fault examples](#) 83  
[WSDL message examples](#) 78

## F

[Fields - vendor-extensible](#) 11

#### Full WSDL

[overview](#) 85  
[RepositoryConfigurationServiceType](#) 85

#### Full XSD

[dsc](#) 89  
[psp](#) 105  
[wscn](#) 107

## G

[Glossary](#) 8

[Groups](#) 42

## I

[Implementer - security considerations](#) 84

[Index of security parameters](#) 84

[Informative references](#) 10

#### Initialization

[RepositoryConfigurationServiceType client](#)  
 ([section 3.1.3](#) 43, [section 3.3](#) 77)

[RepositoryConfigurationServiceType server](#)  
 ([section 3.1.3](#) 43, [section 3.2.3](#) 47)

[Introduction](#) 8

## L

[Local events - RepositoryConfigurationServiceType server](#) 77

## M

[Message processing -](#)

[RepositoryConfigurationServiceType server](#) 47

#### Messages

[attribute groups](#) 42  
[attributes](#) 41  
[complex types](#) 23  
[dsc:DateTimeExtType complex type](#) 24  
[dsc:Dialect element](#) 17  
[dsc:Document element](#) 17  
[dsc:DocumentDescription element](#) 17  
[dsc:DocumentDescriptionType complex type](#) 25  
[dsc:DocumentFormatBaseType simple type](#) 33  
[dsc:DocumentFormatExtType complex type](#) 25



[dsc:DocumentFormatWKVType simple type](#) 33  
[dsc:DocumentId element](#) 18  
[dsc:Documents element](#) 17  
[dsc:DocumentsType complex type](#) 25  
[dsc:DocumentType complex type](#) 25  
[dsc:Filter element](#) 18  
[dsc:FilterCompletedStatuses element](#) 18  
[dsc:FilterConfig element](#) 18  
[dsc:FilterConfigType complex type](#) 26  
[dsc:Filters element](#) 18  
[dsc:FilterState element](#) 18  
[dsc:FilterStateBaseType simple type](#) 34  
[dsc:FilterStateReason element](#) 18  
[dsc:FilterStateReasonBaseType simple type](#) 35  
[dsc:FilterStateReasons element](#) 19  
[dsc:FilterStateReasonsType complex type](#) 26  
[dsc:FilterStateReasonType complex type](#) 26  
[dsc:FilterStateReasonWKVType simple type](#) 35  
[dsc:FilterStateType complex type](#) 26  
[dsc:FilterStateWKVType simple type](#) 36  
[dsc:FilterStatus element](#) 19  
[dsc:FilterStatuses element](#) 19  
[dsc:FilterStatusesType complex type](#) 27  
[dsc:FilterStatusType complex type](#) 27  
[dsc:FiltersType complex type](#) 27  
[dsc:FilterType complex type](#) 27  
[dsc:Format element](#) 19  
[dsc:ImagesReceived element](#) 19  
[dsc:IntOneBaseType simple type](#) 36  
[dsc:IntOneExtType complex type](#) 28  
[dsc:IntZeroBaseType simple type](#) 37  
[dsc:IntZeroExtType complex type](#) 28  
[dsc:JobCompletedState element](#) 19  
[dsc:JobCompletedStateReasons element](#) 20  
[dsc:JobCompletedTime element](#) 20  
[dsc:JobCreatedTime element](#) 20  
[dsc:JobDescription element](#) 20  
[dsc:JobDescriptionType complex type](#) 28  
[dsc:JobOriginatingUserName element](#) 20  
[dsc:JobState element](#) 20  
[dsc:JobStateBaseType simple type](#) 37  
[dsc:JobStateReason element](#) 21  
[dsc:JobStateReasonBaseType simple type](#) 37  
[dsc:JobStateReasons element](#) 21  
[dsc:JobStateReasonsType complex type](#) 29  
[dsc:JobStateReasonType complex type](#) 29  
[dsc:JobStateReasonWKVType simple type](#) 37  
[dsc:JobStateType complex type](#) 29  
[dsc:JobStateWKVType simple type](#) 39  
[dsc:JobStatus element](#) 21  
[dsc:JobStatusType complex type](#) 29  
[dsc:JobSummary element](#) 21  
[dsc:JobSummaryType complex type](#) 30  
[dsc:JobToken element](#) 21  
[dsc:KeywordNsExtensionPatternType simple type](#) 39  
[dsc:ListOfSummariesType complex type](#) 28  
[dsc:MimeExtensionPatternType simple type](#) 40  
[dsc:PSP\\_DisplayName element](#) 21  
[dsc:PSP\\_Identifier element](#) 22  
[dsc:RepositoryConfiguration element](#) 22

[dsc:RepositoryConfigurationType complex type](#) 30  
[dsc:RepositoryState element](#) 22  
[dsc:RepositoryStateBaseType simple type](#) 40  
[dsc:RepositoryStateReason element](#) 22  
[dsc:RepositoryStateReasonBaseType simple type](#) 40  
[dsc:RepositoryStateReasons element](#) 22  
[dsc:RepositoryStateReasonsType complex type](#) 31  
[dsc:RepositoryStateReasonType complex type](#) 31  
[dsc:RepositoryStateReasonWKVType simple type](#) 40  
[dsc:RepositoryStateType complex type](#) 30  
[dsc:RepositoryStateWKVType simple type](#) 41  
[dsc:RepositoryStatus element](#) 22  
[dsc:RepositoryStatusType complex type](#) 31  
[dsc:String255BaseType simple type](#) 41  
[dsc:String255ExtType complex type](#) 31  
[elements](#) 15  
[enumerated](#) 14  
[error codes](#) 14  
[groups](#) 42  
[Name attribute](#) 42  
[namespaces](#) 13  
[psp:anyURIExtType complex type](#) 24  
[simple types](#) 32  
[syntax](#) 13  
[transport](#) 13  
[Validattribute](#) 42

## N

[Name attribute](#) 42  
[Namespaces](#) 13  
[Normative references](#) 9

## O

### Operations

[CancelPostScanJob](#) 53  
[GetActiveJobs](#) 62  
[GetJobHistory](#) 65  
[GetPostScanJobElements](#) 56  
[GetRepositoryElements](#) 47  
[PostScanJobEndStateEvent](#) 74  
[PostScanJobStatusEvent](#) 72  
[RepositoryElementsChangeEvent](#) 68  
[RepositoryStatusSummaryEvent](#) 70  
[Overview \(synopsis\)](#) 10

## P

[Parameters - security index](#) 84  
[Preconditions](#) 11  
[Prerequisites](#) 11  
[Product behavior](#) 116  
[psp:anyURIExtType complex type](#) 24

## R

References

- [informative](#) 10
- [normative](#) 9
- [Relationship to other protocols](#) 10
- RepositoryConfigurationServiceType client
  - [abstract data model](#) 43
  - initialization ([section 3.1.3](#) 43, [section 3.3](#) 77)
  - [overview](#) 77
  - SOAP faults ([section 3.1.4.1](#) 43, [section 3.3](#) 77)
  - timers ([section 3.1.2](#) 43, [section 3.3](#) 77)
- RepositoryConfigurationServiceType server
  - abstract data model ([section 3.1.1](#) 43, [section 3.2.1](#) 45)
  - [CancelPostScanJob operation](#) 53
  - [GetActiveJobs operation](#) 62
  - [GetJobHistory operation](#) 65
  - [GetPostScanJobElements operation](#) 56
  - [GetRepositoryElements operation](#) 47
  - initialization ([section 3.1.3](#) 43, [section 3.2.3](#) 47)
  - [local events](#) 77
  - [message processing](#) 47
  - [PostScanJobEndStateEvent operation](#) 74
  - [PostScanJobStatusEvent operation](#) 72
  - [RepositoryElementsChangeEvent operation](#) 68
  - [RepositoryStatusSummaryEvent operation](#) 70
  - [sequencing rules](#) 47
  - SOAP faults 43
  - [timer events](#) 77
  - timers ([section 3.1.2](#) 43, [section 3.2.2](#) 47)
- [RepositoryConfigurationServiceType WSDL](#) 85

## S

- Security
  - [implementer considerations](#) 84
  - [parameter index](#) 84
- [Sequencing rules - RepositoryConfigurationServiceType server](#) 47
- Server - RepositoryConfigurationServiceType
  - abstract data model ([section 3.1.1](#) 43, [section 3.2.1](#) 45)
  - [CancelPostScanJob operation](#) 53
  - [GetActiveJobs operation](#) 62
  - [GetJobHistory operation](#) 65
  - [GetPostScanJobElements operation](#) 56
  - [GetRepositoryElements operation](#) 47
  - initialization ([section 3.1.3](#) 43, [section 3.2.3](#) 47)
  - [local events](#) 77
  - [message processing](#) 47
  - [PostScanJobEndStateEvent operation](#) 74
  - [PostScanJobStatusEvent operation](#) 72
  - [RepositoryElementsChangeEvent operation](#) 68
  - [RepositoryStatusSummaryEvent operation](#) 70
  - [sequencing rules](#) 47
  - SOAP faults 43
  - [timer events](#) 77
  - timers ([section 3.1.2](#) 43, [section 3.2.2](#) 47)

## Simple types

- [dsc:DocumentFormatBaseType](#) 33
- [dsc:DocumentFormatWKVType](#) 33
- [dsc:FilterStateBaseType](#) 34
- [dsc:FilterStateReasonBaseType](#) 35
- [dsc:FilterStateReasonWKVType](#) 35

- [dsc:FilterStateWKVType](#) 36
- [dsc:IntOneBaseType](#) 36
- [dsc:IntZeroBaseType](#) 37
- [dsc:JobStateBaseType](#) 37
- [dsc:JobStateReasonBaseType](#) 37
- [dsc:JobStateReasonWKVType](#) 37
- [dsc:JobStateWKVType](#) 39
- [dsc:KeywordNsExtensionPatternType](#) 39
- [dsc:MimeExtensionPatternType](#) 40
- [dsc:RepositoryStateBaseType](#) 40
- [dsc:RepositoryStateReasonBaseType](#) 40
- [dsc:RepositoryStateReasonWKVType](#) 40
- [dsc:RepositoryStateWKVType](#) 41
- [dsc:String255BaseType](#) 41
- [overview](#) 32
- SOAP faults
  - [examples](#) 83
- RepositoryConfigurationServiceType client
  - ([section 3.1.4.1](#) 43, [section 3.3](#) 77)
- [RepositoryConfigurationServiceType server](#) 43
- [Standards assignments](#) 12
- [Syntax - messages - overview](#) 13

## T

- [Timer events - RepositoryConfigurationServiceType server](#) 77
- Timers - RepositoryConfigurationServiceType server
  - ([section 3.1.2](#) 43, [section 3.2.2](#) 47)
- [Tracking changes](#) 118
- [Transport](#) 13
- Types
  - [complex](#) 23
  - [simple](#) 32

## V

- [Valid attribute](#) 42
- [Vendor-extensible fields](#) 11
- [Versioning](#) 11

## W

- WSDL
  - [overview](#) 85
  - [RepositoryConfigurationServiceType](#) 85
  - [WSDL message examples](#) 78

## X

- XSD
  - [dsc](#) 89
  - [psp](#) 105
  - [wscn](#) 107