



# Content Management Interoperability Services Version 0.62b

## Part II – Web Services Binding

Draft

19 June 2009

**Specification URIs:**

**This Version:**

[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .html](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .html)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .doc](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .doc)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .pdf](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .pdf)

**Previous Version:**

[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .html](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .html)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .doc](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .doc)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .pdf](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .pdf)

**Latest Version:**

[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .html](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .html)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .doc](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .doc)  
[http://docs.oasis-open.org/\[tc-short-name\] / \[additional path/filename\] .pdf](http://docs.oasis-open.org/[tc-short-name] / [additional path/filename] .pdf)

**Technical Committee:**

OASIS Content Management Interoperability Services TC

**Chair(s):**

David Choy

**Editor(s):**

Al Brown  
Ethan Gur-Esh  
Ryan McVeigh

**Related work:**

This specification replaces or supercedes:

- CMIS Part II – ReSTful AtomPub Binding 0.62

This specification is related to:

- CMIS Part I Draft 0.62
- CMIS Part II – Web Services Draft 0.62

**Declared XML Namespace(s):**

<http://docs.oasis-open.org/ns/cmisis/core/200901>  
<http://docs.oasis-open.org/ns/cmisis/messaging/200901>  
<http://docs.oasis-open.org/ns/cmisis/ws/200901>

**Abstract:**

This specification defines the Web Services based binding.

**Status:**

This document was last revised or approved by the CMIS TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/CMIS/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/CMIS/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/CMIS/>.

---

## Notices

Copyright © OASIS® 2008. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

---

# Table of Contents

Part II – Web Services Binding .....	1
Draft.....	1
19 June 2009 .....	1
Specification URIs:.....	1
This Version: .....	1
Previous Version: .....	1
Latest Version: .....	1
Technical Committee: .....	1
OASIS Content Management Interoperability Services TC .....	1
Chair(s):.....	1
David Choy.....	1
Editor(s):.....	1
Al Brown .....	1
Related work: .....	1
Declared XML Namespace(s): .....	1
Abstract: .....	1
Status: .....	2
Notices .....	3
Table of Contents .....	4
1 Introduction.....	7
1.1 Terminology .....	7
1.2 Normative References .....	7
1.3 Non-Normative References .....	7
2 Simple Object Access Protocol (SOAP) Binding.....	8
2.1 Overview .....	8
2.1.1 Notational Conventions .....	8
2.1.2 Authentication.....	8
2.1.3 WS-I.....	8
2.1.4 Content Transfer .....	8
2.1.5 Reporting Errors .....	8
2.2 Object Representation Format .....	8
2.3 Common Properties.....	9
3 Services.....	10
3.1 RepositoryServicePort.....	10
3.1.1 getRepositories .....	10
3.1.2 getRepositoryInfo .....	10
3.1.3 getTypeChildren .....	11
3.1.4 getTypeDescendants .....	11
3.1.5 getTypeDefinition .....	12
3.2 NavigationServicePort .....	13
3.2.1 getDescendants .....	13
3.2.2 getChildren .....	14
3.2.3 getFolderParent.....	15

3.2.4	getFolderTree .....	15
3.2.5	getObjectParents .....	16
3.2.6	getCheckedOutDocs .....	17
3.3	ObjectServicePort .....	17
3.3.1	createDocument .....	18
3.3.2	createFolder .....	19
3.3.3	createRelationship .....	20
3.3.4	createPolicy .....	21
3.3.5	getAllowableActions .....	21
3.3.6	getProperties .....	22
3.3.7	getPropertiesOfLatestVersion .....	23
3.3.8	getFolderByPath .....	23
3.3.9	getContentStream .....	24
3.3.10	updateProperties .....	24
3.3.11	moveObject .....	25
3.3.12	deleteObject .....	25
3.3.13	deleteTree .....	26
3.3.14	setContentStream .....	26
3.3.15	deleteContentStream .....	27
3.3.16	getRenditions .....	27
3.4	MultiFilingServicePort .....	28
3.4.1	addObjectToFolder .....	28
3.4.2	removeObjectFromFolder .....	29
3.5	DiscoveryServicePort .....	29
3.5.1	query .....	30
3.5.2	getContentChanges .....	30
3.6	VersioningServicePort .....	31
3.6.1	checkOut .....	31
3.6.2	cancelCheckOut .....	32
3.6.3	checkIn .....	32
3.6.4	getPropertiesOfLatestVersion .....	33
3.6.5	getAllVersions .....	34
3.7	RelationshipServicePort .....	34
3.7.1	getRelationships .....	35
3.8	PolicyServicePort .....	35
3.8.1	applyPolicy .....	35
3.8.2	removePolicy .....	36
3.8.3	getAppliedPolicies .....	36
3.9	ACL Services .....	37
3.9.1	getACLCapabilities .....	37
3.9.2	getACL .....	37
3.9.3	applyACL .....	38
4	CMIS-Messaging Schema .....	39
5	CMISWS-Service WSDL .....	59
#	Conformance .....	76

A. Acknowledgements .....	77
B. Non-Normative Text .....	78
C. Revision History.....	79

---

# 1 Introduction

[All text is normative unless otherwise labeled]

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

## 1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [CMISDM] OASIS, Committee Draft 0.60, “Content Management Interoperability Services (CMIS) Domain Model”, March 2009
- [CMISSHEMA] OASIS, Committee Draft 0.60, “Content Management Interoperability Services (CMIS) Schema”, March 2009

## 1.3 Non-Normative References

**NOTE: The proper format for a citation to an OASIS Technical Committee’s work (whether Normative or Non-Normative) is:**

OASIS

Stage (Committee Draft 01, Committee Draft 02, Committee Specification 01, etc. or Standard)

Title (italicized or in quotation marks)

Approval Date (Month YYYY)

URI of the actual Authoritative Specification (namespace is not acceptable as the content changes over time)

For example:

**EDXL-HAVE** OASIS Standard, “Emergency Data Exchange Language (EDXL) Hospital Availability Exchange (HAVE) Version 1.0”, November 2008.  
[http://docs.oasis-open.org/emergency/edxl-have/os/emergency\\_edxl\\_have-1.0-spec-os.doc](http://docs.oasis-open.org/emergency/edxl-have/os/emergency_edxl_have-1.0-spec-os.doc)

---

## 2 Simple Object Access Protocol (SOAP) Binding

### 2.1 Overview

All services and operations defined in part I of the CMIS specification are presented in this SOAP binding. The WSDL for these services reference two XSD documents. One defines elements for the primary data types of documents, folders, relationships and policies as well as collections of these types of objects. Section 3.0 defines the mapping between all relevant pieces of the domain model as described in Part I of the CMIS specification. The second XSD defines the message formats for each of the CMIS services; the messages often refer to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services defined in the services section of Part I of the CMIS specification.

The normative CMIS SOAP binding is defined by the WSDL and XSD as well as the details given here in this part of the CMIS specification except the examples.

#### 2.1.1 Notational Conventions

The elements in both XSD schemas are defined in the CMIS namespace.

In each of the examples in this section the CMIS namespace is the default namespace, hence elements are shown without a namespace prefix.

Many of the services that are retrieving objects or sets of objects accept a filter that specifies the object properties that are returned. The examples below include only a limited set of properties for brevity. In most cases many more properties may be included in the XML that is returned. When examples are presented where only a limited set of properties are returned the accompanying text will clearly state that.

#### 2.1.2 Authentication

A CMIS SOAP binding **MUST** support WS-Security 1.1 and Username Token Profile 1.1.

#### 2.1.3 WS-I

A CMIS SOAP binding **SHALL** comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

#### 2.1.4 Content Transfer

A CMIS SOAP binding **SHOULD** support MTOM content transfers.

#### 2.1.5 Reporting Errors

Services **SHALL** report errors via SOAP faults. The `cmisMessageTypes.xsd` defines a basic fault structure that includes an error code and an error message and the WSDL for each service define specific messages that have the basic fault format.

## 2.2 Object Representation Format

The Domain Model in Part I of the CMIS specification defines Document, Folder, Relationship, and Policy object types. This section defines a mapping between the pieces of that domain model and the elements of the CMIS SOAP binding XSD.



## 2.3 Common Properties

Please see the Domain Model for base properties on objects.

---

## 3 Services

### 3.1 RepositoryServicePort

Please see the domain model for a description of the services contained.

```
<operation name="getRepositories">
  <input message="cmisw:getRepositoriesRequest" />
  <output message="cmisw:getRepositoriesResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getRepositoryInfo">
  <input message="cmisw:getRepositoryInfoRequest" />
  <output message="cmisw:getRepositoryInfoResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getTypes">
  <input message="cmisw:getTypesRequest" />
  <output message="cmisw:getTypesResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getTypeDefinition">
  <input message="cmisw:getTypeDefinitionRequest" />
  <output message="cmisw:getTypeDefinitionResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
```

#### 3.1.1 getRepositories

This returns the set of repositories available.

Input:

```
<xs:element name="getRepositories">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getRepositoriesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repository"
type="cmis:cmisRepositoryEntryType"
minOccurs="0" maxOccurs="unbounded" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

#### 3.1.2 getRepositoryInfo

Input:

```
<xs:element name="getRepositoryInfo">
  <xs:complexType>
```

```

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

```

Output:

```

<xs:element name="getRepositoryInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryInfo"
type="cmis:cmisRepositoryInfoType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.1.3 getTypeChildren

This service returns the list of all types in the repository.

Input:

```

<xs:element name="getTypeChildren">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="returnPropertyDefinitions"
type="xs:boolean"
minOccurs="0"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getTypeChildrenResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="type"
type="cmis:cmisTypeDefinitionType"
minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.1.4 getTypeDescendants

This service returns a tree of all types in the repository.

Input:

```
<xs:element name="getTypeDescendants">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includePropertyDefinitions"
        type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getTypeDescendantsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="type"
        type="cmis:cmisTypeDefinitionType"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.1.5 getTypeDefinition

Gets the definition for specified type

Input:

```
<xs:element name="getTypeDefinition">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getTypeDefinitionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="type"
        type="cmis:cmisTypeDefinitionType"
        minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 3.2 NavigationServicePort

The NavigationServicePort contains the following operations:

```
<operation name="getDescendants">
  <input message="cmisw:getDescendantsRequest" />
  <output message="cmisw:getDescendantsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getChildren">
  <input message="cmisw:getChildrenRequest" />
  <output message="cmisw:getChildrenResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getFolderParent">
  <input message="cmisw:getFolderParentRequest" />
  <output message="cmisw:getFolderParentResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getObjectParents">
  <input message="cmisw:getObjectParentsRequest" />
  <output message="cmisw:getObjectParentsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getCheckedOutDocs">
  <input message="cmisw:getCheckedOutDocsRequest" />
  <output message="cmisw:getCheckedOutDocsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
```

### 3.2.1 getDescendants

Gets the list of objects contained at one or more levels below the specified folder. Only the selected properties associated with each object are returned. The content stream is not returned.

Input:

```
<xs:element name="getDescendants">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="type"
        type="cmis:enumTypesOfFileableObjects"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="depth" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
        type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
        type="cmis:enumIncludeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="orderBy" type="xs:string"
        minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getDescendantsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.2.2 getChildren

Gets the list of objects contained at one level below the specified folder. Only the selected properties associated with each object are returned. The content stream is not returned.

Input:

```

<xs:element name="getChildren">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
      <xs:element name="type"
type="cmis:enumTypesOfFileableObjects"
minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
minOccurs="0"
maxOccurs="1" nillable="true" />
      <xs:element name="orderBy" type="xs:string"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getChildrenResponse">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="hasMoreItems" type="xs:boolean"
            minOccurs="1" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>

```

### 3.2.3 getFolderParent

Returns the parent folder object(s) above the specified folder.

Input:

```

<xs:element name="getFolderParent">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="0"
maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
maxOccurs="1" />
      <xs:element name="includeAllowableActions"
type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getFolderParentResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.2.4 getFolderTree

Returns the tree of folders under the specified folder

Input:

```

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

```

```

        <xs:element name="folderId" type="xs:string"
minOccurs="1"
            maxOccurs="1" />
        <xs:element name="filter" type="xs:string"
minOccurs="0"
            maxOccurs="1" nillable="true" />
        <xs:element name="includeAllowableActions"
type="xs:boolean"
            minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
</xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getFolderTreeResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

### 3.2.5 getObjectParents

Returns the parent folders for the specified non-folder fileable object.

Input:

```

<xs:element name="getObjectParents">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="filter" type="xs:string"
minOccurs="0"
                maxOccurs="1" />
            <xs:element name="includeAllowableActions"
type="xs:boolean"
                minOccurs="0" maxOccurs="1" nillable="true" />
            <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
                minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getObjectParentsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



## 3.2.6 getCheckedOutDocs

Gets the list of documents that are checked out that the user has access to. Most likely this will be the set of documents checked out by the user.

Input:

```
<xs:element name="getCheckedOutDocs" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
        minOccurs="0" \>
          maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0" \>
          maxOccurs="1" nillable="true" />
      <xs:element name="orderBy" type="xs:string"
        minOccurs="0" \>
          maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
        type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
        type="cmis:enumIncludeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="maxItems" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getCheckedOutDocsResponse" >
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 3.3 ObjectServicePort

```
<operation name="createDocument" >
  <input message="cmisw:createDocumentRequest" />
  <output message="cmisw:createDocumentResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="createFolder" >
  <input message="cmisw:createFolderRequest" />
  <output message="cmisw:createFolderResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="createRelationship" >
```

```

        <input message="cmisw:createRelationshipRequest" />
        <output message="cmisw:createRelationshipResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="createPolicy">
        <input message="cmisw:createPolicyRequest" />
        <output message="cmisw:createPolicyResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getAllowableActions">
        <input message="cmisw:getAllowableActionsRequest" />
        <output message="cmisw:getAllowableActionsResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getProperties">
        <input message="cmisw:getPropertiesRequest" />
        <output message="cmisw:getPropertiesResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getContentStream">
        <input message="cmisw:getContentStreamRequest" />
        <output message="cmisw:getContentStreamResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="updateProperties">
        <input message="cmisw:updatePropertiesRequest" />
        <output message="cmisw:updatePropertiesResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="moveObject">
        <input message="cmisw:moveObjectRequest" />
        <output message="cmisw:moveObjectResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteObject">
        <input message="cmisw:deleteObjectRequest" />
        <output message="cmisw:deleteObjectResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteTree">
        <input message="cmisw:deleteTreeRequest" />
        <output message="cmisw:deleteTreeResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="setContentStream">
        <input message="cmisw:setContentStreamRequest" />
        <output message="cmisw:setContentStreamResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteContentStream">
        <input message="cmisw:deleteContentStreamRequest" />
        <output message="cmisw:deleteContentStreamResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>

```

### 3.3.1 createDocument

Creates a document of the specified type

Input:

```
<xs:element name="createDocument" >
```

```

    <xs:complexType>
      <xs:sequence>
        <xs:element name="repositoryId" type="xs:string"
          minOccurs="1" maxOccurs="1" />
        <xs:element name="typeId" type="xs:string"
          minOccurs="1"
          maxOccurs="1" />
        <xs:element name="properties"
          type="cmis:cmisPropertiesType"
          minOccurs="1" maxOccurs="1" />
        <xs:element name="folderId" type="xs:string"
          minOccurs="0"
          maxOccurs="1" />
        <xs:element name="contentStream"
          type="cmis:cmisContentStreamType"
          minOccurs="0" maxOccurs="1" nillable="true" />
        <xs:element name="versioningState"
          type="cmis:enumVersioningState"
          minOccurs="0" maxOccurs="1" nillable="true" />
        <xs:element name="applyPolicies" type="xs:string"
          minOccurs="0" maxOccurs="unbounded"
          nillable="true" />
        <xs:element name="addACEs"
          type="cmis:cmisAccessControlEntryType"
          minOccurs="0" maxOccurs="unbounded"
          nillable="true" />
        <xs:element name="removeACEs"
          type="cmis:cmisAccessControlEntryType"
          minOccurs="0" maxOccurs="unbounded"
          nillable="true" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

Output:

```

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

```

### 3.3.2 createFolder

Creates a folder of the specified type

Input:

```

  <xs:element name="createFolder">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="repositoryId" type="xs:string"
          minOccurs="1" maxOccurs="1" />
        <xs:element name="typeId" type="xs:string"
          minOccurs="1"
          maxOccurs="1" />
        <xs:element name="properties"
          type="cmis:cmisPropertiesType"
          minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

properties object -->
minOccurs="1"
maxOccurs="1" />
<xs:element name="applyPolicies" type="xs:string"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
<xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
<xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
</xs:sequence>
</xs:complexType>
</xs:element>

```

Output:

```

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

```

### 3.3.3 createRelationship

Creates a relationship of the specified type

Input:

```

<xs:element name="createRelationship">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
minOccurs="1" maxOccurs="1" />
      <!-- must match the type attribute value of the
properties object -->
      <xs:element name="sourceObjectId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <!--
in then their
values MUST match
-->
      <xs:element name="targetObjectId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <!--
in then their
values MUST match
-->
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

Output:

```

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

```

### 3.3.4 createPolicy

Input:

```

<xs:element name="createPolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
minOccurs="1"
                                maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
minOccurs="1" maxOccurs="1" />
      <!--
in the object and
                                note that the object type may also be an element
                                if present the values must match
-->
      <xs:element name="folderId" type="xs:string"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

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

```

### 3.3.5 getAllowableActions

Returns the list of allowable actions for an object based on the current user's context.

Input:

```

<xs:element name="getAllowableActions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"

```

```

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

```

Output:

```

<xs:element name="getAllowableActionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="allowableActions"
type="cmis:cmisAllowableActionsType"
      minOccurs="1" maxOccurs="1" />
      <!--
actions are allowed
      -->
      returning no action elements indicates that no
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.3.6 getProperties

Returns the properties of an object

Input:

```

<xs:element name="getProperties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1"
      minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
      maxOccurs="1" />
      <xs:element name="returnVersion"
type="cmis:enumReturnVersion"
      minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
      maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
type="xs:boolean"
      minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
      minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getPropertiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.3.7 getPropertiesOfLatestVersion

Returns the properties of the latest version

Input:

```
<xs:element name="getPropertiesOfLatestVersion">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="versionSeriesId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="major" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getPropertiesOfLatestVersionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.3.8 getFolderByPath

Returns the properties of a folder specified by path

Input:

```
<xs:element name="getFolderByPath">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderPath" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
        type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
        type="cmis:enumIncludeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeACL" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getFolderByPathResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.3.9 getContentStream

Returns the content stream for the document

Input:

```
<xs:element name="getContentStream">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getContentStreamResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="contentStream"
type="cmis:cmisContentType"
minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.3.10 updateProperties

Updates properties of the specified object

Input:

```
<xs:element name="updateProperties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
      <!--
value must match this
      objectId
-->
      <xs:element name="changeToken" type="xs:string"
minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



```

        <xs:element name="properties"
type="cmis:cmisPropertiesType"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="updatePropertiesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="changeToken" type="xs:string"
                minOccurs="0" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

### 3.3.11 moveObject

Moves specified fileable object from a source folder to a target folder

Input:

```

<xs:element name="moveObject">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
minOccurs="1"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
                maxOccurs="1" />
            <xs:element name="targetFolderId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="sourceFolderId" type="xs:string"
                minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="moveObjectResponse">
    <xs:complexType>
        <xs:sequence />
    </xs:complexType>
</xs:element>

```

### 3.3.12 deleteObject

Deletes the specified object

Input:

```

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

```

```

        <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
        <xs:element name="allVersions" type="xs:boolean"
minOccurs="1"
                maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="deleteObjectResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>

```

### 3.3.13 deleteTree

Deletes specified folder and the entire sub-tree

Input:

```

<xs:element name="deleteTree">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
      <xs:element name="unfileNonfolderObjects"
type="cmis:enumUnfileNonfolderObjects"
                minOccurs="1" maxOccurs="1" />
      <xs:element name="continueOnFailure" type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="deleteTreeResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="failedToDelete">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="objectId"
type="xs:string"
                minOccurs="0"
maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.3.14 setContentStream

Sets (creates or replaces) the content stream for the specified document object.

Input:

```
<xs:element name="setContentStream">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
        minOccurs="1"
          maxOccurs="1" />
      <xs:element name="overwriteFlag" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="contentStream"
        type="cmis:cmisContentStreamType"
          minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="setContentStreamResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="documentId" type="xs:string"
        minOccurs="1"
          maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.3.15 deleteContentStream

Deletes the content stream of the specified document. This does not delete the document object or its properties.

Input:

```
<xs:element name="deleteContentStream">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
        minOccurs="1"
          maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="deleteContentStreamResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="documentId" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.3.16 getRenditions

Returns the renditions available on the specified object.

Input:

```
<xs:element name="getRenditions" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="renditionFilter" type="xs:string"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="maxItems" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getRenditionsResponse" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="rendition"
        type="cmis:cmisRenditionType" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 3.4 MultiFilingServicePort

```
<operation name="addObjectToFolder" >
  <input message="cmisw:addObjectToFolderRequest" />
  <output message="cmisw:addObjectToFolderResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="removeObjectFromFolder" >
  <input message="cmisw:removeObjectFromFolderRequest" />
  <output message="cmisw:removeObjectFromFolderResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
```

### 3.4.1 addObjectToFolder

Adds an existing, fileable object to a folder. This may fail based on repository rules such as multi-filing not being supported, documents only being allowed to be filed once in any folder, etc.

Input:

```
<xs:element name="addObjectToFolder" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

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

```

Output:

```

<xs:element name="addObjectToFolderResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>

```

### 3.4.2 removeObjectFromFolder

Removes a child object from a folder. This does not delete the object and does not change its ID.

Input:

```

<xs:element name="removeObjectFromFolder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="removeObjectFromFolderResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>

```

## 3.5 DiscoveryServicePort

```

<operation name="query">
  <input message="cmisw:queryRequest" />
  <output message="cmisw:queryResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getContentChanges">
  <input message="cmisw:getContentChangesRequest" />
  <output message="cmisw:getContentChangesResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>

```

### 3.5.1 query

Queries the repository for queryable objects based on properties and/or using full-text search. Query returns objects that matches the qualification. Content streams are not returned as part of query.

Input:

```
<xs:complexType name="cmisQueryType">
  <xs:sequence>
    <xs:element name="repositoryId" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="statement" type="xs:string" minOccurs="1"
      maxOccurs="1" />
    <xs:element name="searchAllVersions" type="xs:boolean"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="pageSize" type="xs:integer" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="skipCount" type="xs:integer" minOccurs="0"
      maxOccurs="1" />
    <xs:element name="returnAllowableActions" type="xs:boolean"
      minOccurs="0" maxOccurs="1" />

    <!-- added so jax-ws generates a nice WS interface -->
    <xs:element name="extension" type="cmis:cmisAnyXml"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
  <xs:attributeGroup ref="cmis:cmisUndefinedAttribute" />
</xs:complexType>
```

Output:

```
<xs:element name="queryResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3.5.2 getContentChanges

Provides support for getting content changed since a certain point.

Input:

```
<xs:element name="getContentChanges">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string" />
      <xs:element name="changeToken" type="xs:string"
        minOccurs="0" />
      <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeACL" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeProperties" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

        <xs:element name="filter" type="xs:string"
minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

Output:

```

  <xs:element name="getContentChangesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="changedObject"
type="cmis:cmisObjectType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
        <xs:element name="changeToken" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

## 3.6 VersioningServicePort

```

<operation name="checkOut">
  <input message="cmisw:checkOutRequest" />
  <output message="cmisw:checkOutResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="cancelCheckOut">
  <input message="cmisw:cancelCheckOutRequest" />
  <output message="cmisw:cancelCheckOutResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="checkIn">
  <input message="cmisw:checkInRequest" />
  <output message="cmisw:checkInResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getPropertiesOfLatestVersion">
  <input message="cmisw:getPropertiesOfLatestVersionRequest" />
  <output message="cmisw:getPropertiesOfLatestVersionResponse"
/>
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getAllVersions">
  <input message="cmisw:getAllVersionsRequest" />
  <output message="cmisw:getAllVersionsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="deleteAllVersions">
  <input message="cmisw:deleteAllVersionsRequest" />
  <output message="cmisw:deleteAllVersionsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>

```

### 3.6.1 checkOut

Create a private working copy of the document, copies the metadata and optionally content. It is up to the repository to determine if updates to the current version (not PWC) and prior versions are allowed if the document is checked-out.

Input:

```

<xs:element name="checkOut">
  <xs:complexType>

```

```

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

```

Output:

```

    <xs:element name="checkoutResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="documentId" type="xs:string"
            minOccurs="1"
            maxOccurs="1" />
          <xs:element name="contentCopied" type="xs:boolean"
            minOccurs="1" maxOccurs="1" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

### 3.6.2 cancelCheckout

Deletes the private working copy of the checked-out document object, allowing other documents in the version series to be checked out again.

Input:

```

    <xs:element name="cancelCheckout">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="repositoryId" type="xs:string"
            minOccurs="1" maxOccurs="1" />
          <xs:element name="documentId" type="xs:string"
            minOccurs="1"
            maxOccurs="1" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

Output:

```

    <xs:element name="cancelCheckoutResponse">
      <xs:complexType>
        <xs:sequence />
      </xs:complexType>
    </xs:element>

```

### 3.6.3 checkIn

Makes the private working copy the current version of the content.

Input:

```

    <xs:element name="checkIn">
      <xs:complexType>

```



```

        <xs:sequence>
          <xs:element name="repositoryId" type="xs:string"
            minOccurs="1" maxOccurs="1" />
          <xs:element name="documentId" type="xs:string"
            minOccurs="1"
            maxOccurs="1" />
          <xs:element name="major" type="xs:boolean"
            minOccurs="0"
            maxOccurs="1" nillable="true" />
          <xs:element name="properties"
            type="cmis:cmisPropertiesType"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="contentStream"
            type="cmis:cmisContentStreamType"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="checkinComment" type="xs:string"
            minOccurs="0" maxOccurs="1" nillable="true" />
          <xs:element name="applyPolicies" type="xs:string"
            minOccurs="0" maxOccurs="unbounded"
            nillable="true" />
          <xs:element name="addACEs"
            type="cmis:cmisAccessControlEntryType"
            minOccurs="0" maxOccurs="unbounded"
            nillable="true" />
          <xs:element name="removeACEs"
            type="cmis:cmisAccessControlEntryType"
            minOccurs="0" maxOccurs="unbounded"
            nillable="true" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

Output:

```

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

```

### 3.6.4 getPropertiesOfLatestVersion

Returns the properties of the latest version

Input:

```

    <xs:element name="getPropertiesOfLatestVersion" >
      <xs:complexType>
        <xs:sequence>
          <xs:element name="repositoryId" type="xs:string"
            minOccurs="1" maxOccurs="1" />
          <xs:element name="versionSeriesId" type="xs:string"
            minOccurs="1" maxOccurs="1" />
          <xs:element name="major" type="xs:boolean"
            minOccurs="1" maxOccurs="1" />
          <xs:element name="filter" type="xs:string"
            minOccurs="0"

```

```

                maxOccurs="1" nillable="true" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

Output:

```

<xs:element name="getPropertiesOfLatestVersionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.6.5 getAllVersions

Returns the list of all members of the version series for the specified document, sorted by CREATION\_DATE descending.

Input:

```

<xs:element name="getAllVersions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="versionSeriesId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getAllVersionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.7 RelationshipServicePort

```

<operation name="getRelationships">
  <input message="cmisw:getRelationshipsRequest" />
  <output message="cmisw:getRelationshipsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>

```

## 3.7.1 getRelationships

Returns a list of relationships associated with the object.

Input:

```
<xs:element name="getRelationships">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="direction"
        type="cmis:enumRelationshipDirection"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="typeId" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeSubRelationshipTypes"
        type="xs:boolean"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
        type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
        type="cmis:enumIncludeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="maxItems" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="getRelationshipsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 3.8 PolicyServicePort

### 3.8.1 applyPolicy

Applies a policy to a controllable object.

Input:

```
<xs:element name="applyPolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="policyId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="applyPolicyResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.8.2 removePolicy

Removes a previously applied policy from an object. The policy object is not deleted, and may still be applied to other objects.

Input:

```
<xs:element name="removePolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="policyId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Output:

```
<xs:element name="removePolicyResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.8.3 getAppliedPolicies

Returns the list of policy objects currently applied to the specified object.

Input:

```

<xs:element name="getAppliedPolicies">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
        minOccurs="1"
        maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
        minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getAppliedPoliciesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## 3.9 ACL Services

### 3.9.1 getACLCapabilities

Input:

```

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

```

Output:

```

<xs:element name="getACLCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="setType" type="cmis:enumACLsetType" />
      <xs:element name="permissions"
        type="cmis:cmisAccessControlPermissionType"
        minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="mapping"
        type="cmis:cmisPermissionMappingType"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.9.2 getACL

Input:

```

<xs:element name="getACL">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"

```

```

        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="getACLResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="acl"
type="cmis:cmisAccessControlListType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

### 3.9.3 applyACL

Input:

```

<xs:element name="applyACL">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Output:

```

<xs:element name="applyACLResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ACEs"
type="cmis:cmisAccessControlEntryType" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="exact" type="xs:boolean" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## 4 CMIS-Messaging Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://docs.oasis-open.org/ns/cmismessaging/200901"
  xmlns:cmis="http://docs.oasis-open.org/ns/cmismessaging/200901"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmismessaging/200901"
  xmlns:jaxb="http://java.sun.com/xml/ns/jaxb" version="0.62a"
  xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jaxb:extensionBindingPrefixes="xjc" jaxb:version="2.1"
  elementFormDefault="qualified">
  <xs:import schemaLocation="CMIS-Core.xsd"
    namespace="http://docs.oasis-open.org/ns/cmismessaging/200901" />

  <!--Exceptions-->
  <xs:complexType name="cmisFaultType">
    <xs:sequence>
      <xs:element name="type" type="cmism:enumServiceException" />
      <xs:element name="code" type="xs:integer" />
      <xs:element name="message" type="xs:string" />
      <xs:any minOccurs="0" maxOccurs="unbounded"
namespace="##other" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="cmisFault" type="cmism:cmisFaultType" />
  <xs:simpleType name="enumServiceException">
    <xs:restriction base="xs:string">
      <xs:enumeration value="constraint" />
      <xs:enumeration value="contentAlreadyExists" />
      <xs:enumeration value="filterNotValid" />
      <xs:enumeration value="invalidArgument" />
      <xs:enumeration value="objectNotFound" />
      <xs:enumeration value="offset" />
      <xs:enumeration value="notSupported" />
      <xs:enumeration value="permissionDenied" />
      <xs:enumeration value="runtime" />
      <xs:enumeration value="storage" />
      <xs:enumeration value="streamNotSupported" />
      <xs:enumeration value="type" />
      <xs:enumeration value="updateConflict" />
      <xs:enumeration value="versioning" />
    </xs:restriction>
  </xs:simpleType>

  <!--[Repository Services]-->
  <xs:element name="getRepositories">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="getRepositoriesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="repository"
type="cmis:cmisRepositoryEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```

<!-- getRepositoryInfo Operation -->
<xs:element name="getRepositoryInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getRepositoryInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryInfo"
type="cmis:cmisRepositoryInfoType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- getTypes Operation -->
<xs:element name="getTypeChildren">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includePropertyDefinitions"
type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="skipCount" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getTypeChildrenResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="type"
type="cmis:cmisTypeDefinitionType"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="hasMoreItems" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- getTypeDefinition Operation -->
<xs:element name="getTypeDefinition">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="typeId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

<xs:element name="getTypeDefinitionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="type"
type="cmis:cmisTypeDefinitionType"
        minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--[Navigation Services]-->
<!-- getDescendants Operation -->
<xs:element name="getDescendants">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="type"
type="cmis:enumTypesOfFileableObjects"
        minOccurs="0" maxOccurs="1" />
      <xs:element name="depth" type="xs:integer"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
      <xs:element name="includeAllowableActions"
type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="orderBy" type="xs:string"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getDescendantsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- getChildren Operation -->
<xs:element name="getChildren">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="type"
type="cmis:enumTypesOfFileableObjects"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

type="xs:boolean"
    <xs:element name="includeAllowableActions"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="includeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
type="cmis:enumIncludeRelationships"
    <xs:element name="maxItems" type="xs:integer"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="skipCount" type="xs:integer"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="orderBy" type="xs:string"
        minOccurs="0" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getChildrenResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
                maxOccurs="unbounded" />
            <xs:element name="hasMoreItems" type="xs:boolean"
                minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- getFolderParent Operation -->
<xs:element name="getFolderParent">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="folderId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="filter" type="xs:string"
                minOccurs="0" maxOccurs="1" />
            <xs:element name="includeAllowableActions"
                type="xs:boolean"
                minOccurs="0" maxOccurs="1" nillable="true" />
            <xs:element name="includeRelationships"
                type="cmis:enumIncludeRelationships"
                minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="getFolderParentResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
                maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- getObjectParents Operation-->
<xs:element name="getObjectParents">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />

```

```

                                <xs:element name="objectId" type="xs:string"
minOccurs="1"
                                maxOccurs="1" />
                                <xs:element name="filter" type="xs:string"
minOccurs="0"
                                maxOccurs="1" />
                                <xs:element name="includeAllowableActions"
type="xs:boolean"
                                minOccurs="0" maxOccurs="1" nillable="true" />
                                <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
                                minOccurs="0" maxOccurs="1" nillable="true" />
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                <xs:element name="getObjectParentsResponse">
                                <xs:complexType>
                                <xs:sequence>
                                <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>

                                <!--getCheckedOutDocs Operation -->
                                <xs:element name="getCheckedOutDocs">
                                <xs:complexType>
                                <xs:sequence>
                                <xs:element name="repositoryId" type="xs:string"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                <xs:element name="folderId" type="xs:string"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                <xs:element name="filter" type="xs:string"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                <xs:element name="orderBy" type="xs:string"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                <xs:element name="includeAllowableActions"
type="xs:boolean"
                                minOccurs="0" maxOccurs="1" nillable="true" />
                                <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
                                minOccurs="0" maxOccurs="1" nillable="true" />
                                <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                <xs:element name="skipCount" type="xs:integer"
minOccurs="0"
                                maxOccurs="1" nillable="true" />
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                <xs:element name="getCheckedOutDocsResponse">
                                <xs:complexType>
                                <xs:sequence>
                                <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
                                <xs:element name="hasMoreItems" type="xs:boolean"
minOccurs="1" maxOccurs="1" />
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>

```

```

<!--[Object Services]-->

<!-- createDocument Operation -->
<xs:element name="createDocument">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="0"
        maxOccurs="1" />
      <xs:element name="contentStream"
type="cmis:cmisContentStreamType"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="versioningState"
type="cmis:enumVersioningState"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="applyPolicies" type="xs:string"
        minOccurs="0" maxOccurs="unbounded"
nillable="true" />
      <xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType"
        minOccurs="0" maxOccurs="unbounded"
nillable="true" />
      <xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType"
        minOccurs="0" maxOccurs="unbounded"
nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="createDocumentResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- createFolder Operation -->
<xs:element name="createFolder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
        minOccurs="1" maxOccurs="1" />
      <!-- must match the type attribute value of the
properties object -->
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="applyPolicies" type="xs:string"
        minOccurs="0" maxOccurs="unbounded"
nillable="true" />
      <xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType"

```

```

minOccurs="0" maxOccurs="unbounded"
nillable="true" />
  <xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="createFolderResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- createRelationship Operation -->
<xs:element name="createRelationship">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
minOccurs="1" maxOccurs="1" />
      <!-- must match the type attribute value of the
properties object -->
      <xs:element name="sourceObjectId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <!--
      if sourceObjectId exist in the properties passed
in then their
      values MUST match
-->
      <xs:element name="targetObjectId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <!--
      if targetObjectId exist in the properties passed
in then their
      values MUST match
-->
      <xs:element name="applyPolicies" type="xs:string"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
  <xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="createRelationshipResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="objectId" type="xs:string"
minOccurs="1"

```

```

maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--createPolicy Operation-->
<xs:element name="createPolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="properties"
type="cmis:cmisPropertiesType"
        minOccurs="1" maxOccurs="1" />
      <!--
in the object and
        note that the object type may also be an element
        if present the values must match
-->
      <xs:element name="folderId" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="createPolicyResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
-----
-----
-->
<!-- getAllowableActions Operation -->
<xs:element name="getAllowableActions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getAllowableActionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="allowableActions"
type="cmis:cmisAllowableActionsType"
        minOccurs="1" maxOccurs="1" />
      <!--
actions are allowed
        returning no action elements indicates that no
-->
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!--
-----
-->
<!-- getProperties Operation -->
<xs:element name="getProperties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="returnVersion"
type="cmis:enumReturnVersion"
        minOccurs="0" maxOccurs="1" nillable="true" />
minOccurs="0"
      <xs:element name="filter" type="xs:string"
        maxOccurs="1" nillable="true" />
type="xs:boolean"
      <xs:element name="includeAllowableActions"
        minOccurs="0" maxOccurs="1" nillable="true" />
type="cmis:enumIncludeRelationships"
      <xs:element name="includeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeACL" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getPropertiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--
-----
-->
<!-- getFolderByPath Operation -->
<xs:element name="getFolderByPath">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
minOccurs="1"
      <xs:element name="folderPath" type="xs:string"
        maxOccurs="1" />
minOccurs="0"
      <xs:element name="filter" type="xs:string"
        maxOccurs="1" nillable="true" />
type="xs:boolean"
      <xs:element name="includeAllowableActions"
        minOccurs="0" maxOccurs="1" nillable="true" />
type="cmis:enumIncludeRelationships"
      <xs:element name="includeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
      <xs:element name="includeACL" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
    </xs:element>
    <xs:element name="getFolderByPathResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <!--
    -----
    -->
    <!-- getContentStream Operation -->
    <xs:element name="getContentStream">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
                <xs:element name="documentId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="getContentStreamResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="contentStream"
type="cmis:cmisContentType"
minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <!--
    -----
    -->
    <!-- updateProperties Operation -->
    <xs:element name="updateProperties">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
                <xs:element name="objectId" type="xs:string"
minOccurs="1"
maxOccurs="1" />
                <!--
                if objectId is present in the properties the
value must match this
                objectId
                -->
                <xs:element name="changeToken" type="xs:string"
minOccurs="0" maxOccurs="1" nillable="true" />
                <xs:element name="properties"
type="cmis:cmisPropertiesType"
minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="updatePropertiesResponse">
        <xs:complexType>

```



```

        <xs:sequence>
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="changeToken" type="xs:string"
                minOccurs="0" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!--
-----
-->
<!-- moveObject Operation -->
<xs:element name="moveObject">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="targetFolderId" type="xs:string"
minOccurs="1" maxOccurs="1" />
            <xs:element name="sourceFolderId" type="xs:string"
minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="moveObjectResponse">
    <xs:complexType>
        <xs:sequence />
    </xs:complexType>
</xs:element>
<!--
-----
-->
<!-- deleteObject Operation -->
<xs:element name="deleteObject">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="allVersions" type="xs:boolean"
minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="deleteObjectResponse">
    <xs:complexType>
        <xs:sequence />
    </xs:complexType>
</xs:element>
<!--
-----
-->
<!-- deleteTree Operation -->
<xs:element name="deleteTree">

```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="repositoryId" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
                <xs:element name="folderId" type="xs:string"
                    minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="unfileNonfolderObjects"
                    type="cmis:enumUnfileNonfolderObjects"
                    minOccurs="1" maxOccurs="1" />
                <xs:element name="continueOnFailure" type="xs:boolean"
                    minOccurs="0" maxOccurs="1" nillable="true" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="deleteTreeResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="failedToDelete">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="objectId"
                                type="xs:string"
                                minOccurs="0"
                                maxOccurs="unbounded" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <!--
    -----
    -->
    <!-- setContentStream Operation -->
    <xs:element name="setContentStream">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="repositoryId" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
                <xs:element name="documentId" type="xs:string"
                    minOccurs="1"
                    maxOccurs="1" />
                <xs:element name="overwriteFlag" type="xs:boolean"
                    minOccurs="0" maxOccurs="1" nillable="true" />
                <xs:element name="contentStream"
                    type="cmis:cmisContentStreamType"
                    minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="setContentStreamResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="documentId" type="xs:string"
                    minOccurs="1"
                    maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <!--
    -----

```

```

-->
<!-- deleteContentStream Operation -->
<xs:element name="deleteContentStream">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="deleteContentStreamResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="documentId" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
-----
-->
<!--[Multi-filing Services]-->
<!--
-----

-->
<!-- addObjectToFolder Operation-->
<xs:element name="addObjectToFolder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="addObjectToFolderResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<!--
-----

-->
<!-- removeObjectFromFolder Operation-->
<xs:element name="removeObjectFromFolder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="folderId" type="xs:string"
minOccurs="0"

```

```

                maxOccurs="1" nillable="true" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="removeObjectFromFolderResponse">
        <xs:complexType>
            <xs:sequence />
        </xs:complexType>
    </xs:element>
    <!--
        -----
        -----
    -->
    <!--[Discovery Services]-->
    <!--
        -----
        -----
    -->
    <!-- query Operation -->
    <!--
        This already provided in CMIS-Core <xs:element name="query"
        type="cmis:cmisQueryType" />
    -->
    <xs:element name="queryResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
                <xs:element name="hasMoreItems" type="xs:boolean"
minOccurs="1" maxOccurs="1" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="getContentChanges">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="repositoryId" type="xs:string" />
                <xs:element name="changeToken" type="xs:string"
minOccurs="0" />
                <xs:element name="maxItems" type="xs:integer"
minOccurs="0"
maxOccurs="1" nillable="true" />
                <xs:element name="includeACL" type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
                <xs:element name="includeProperties" type="xs:boolean"
minOccurs="0" maxOccurs="1" nillable="true" />
                <xs:element name="filter" type="xs:string"
minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="getContentChangesResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="changedObject"
type="cmis:cmisObjectType"
minOccurs="0" maxOccurs="unbounded"
nillable="true" />
                <xs:element name="changeToken" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

<!--
-----
-->
<!--[Versioning Services]-->
<!--
-----

-->
<!-- checkOut Operation -->
<xs:element name="checkOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checkOutResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="documentId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="contentCopied" type="xs:boolean"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
-----

-->
<!-- cancelCheckOut Operation -->
<xs:element name="cancelCheckOut">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="documentId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="cancelCheckOutResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<!--
-----

-->
<!-- checkIn Operation -->
<xs:element name="checkIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />

```

```

minOccurs="1"
    <xs:element name="documentId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
    <xs:element name="major" type="xs:boolean"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="properties"
        type="cmis:cmisPropertiesType"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="contentStream"
        type="cmis:cmisContentStreamType"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="checkinComment" type="xs:string"
        minOccurs="0" maxOccurs="1" nillable="true" />
    <xs:element name="applyPolicies" type="xs:string"
        minOccurs="0" maxOccurs="unbounded"
        nillable="true" />
    <xs:element name="addACEs"
        type="cmis:cmisAccessControlEntryType"
        minOccurs="0" maxOccurs="unbounded"
        nillable="true" />
    <xs:element name="removeACEs"
        type="cmis:cmisAccessControlEntryType"
        minOccurs="0" maxOccurs="unbounded"
        nillable="true" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="checkInResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="documentId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!--
-----
-->
<!--getPropertiesOfLatestVersion-->
<xs:element name="getPropertiesOfLatestVersion">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="versionSeriesId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="major" type="xs:boolean"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="filter" type="xs:string"
                minOccurs="0" maxOccurs="1" nillable="true" />
            <xs:element name="includeACL" type="xs:boolean"
                minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getPropertiesOfLatestVersionResponse">
    <xs:complexType>
        <xs:sequence>

```

```

        <xs:element ref="cmis:object" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<!--
-----
-->
<!-- getAllVersions Operation -->
<xs:element name="getAllVersions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="versionSeriesId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="filter" type="xs:string"
minOccurs="0"
                maxOccurs="1" nillable="true" />
            <xs:element name="includeAllowableActions"
type="xs:boolean"
                minOccurs="0" maxOccurs="1" nillable="true" />
            <xs:element name="includeRelationships"
type="cmis:enumIncludeRelationships"
                minOccurs="0" maxOccurs="1" nillable="true" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getAllVersionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!--
-----
-->
<!--
-----
-->
<!--[Relationship Services]-->
<!--
-----
-->
<!-- getRelationships Operation -->
<xs:element name="getRelationships">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
minOccurs="1"
                maxOccurs="1" />
            <xs:element name="direction"
type="cmis:enumRelationshipDirection"
                minOccurs="0" maxOccurs="1" nillable="true" />

```

```

minOccurs="0"
    <xs:element name="typeId" type="xs:string"
        minOccurs="1" maxOccurs="1" nillable="true" />
    <xs:element name="includeSubRelationshipTypes"
        minOccurs="1" maxOccurs="1" nillable="true" />
type="xs:boolean"
minOccurs="0"
    <xs:element name="filter" type="xs:string"
        minOccurs="1" maxOccurs="1" nillable="true" />
type="xs:boolean"
    <xs:element name="includeAllowableActions"
        minOccurs="0" maxOccurs="1" nillable="true" />
type="cmis:enumIncludeRelationships"
    <xs:element name="includeRelationships"
        minOccurs="0" maxOccurs="1" nillable="true" />
minOccurs="0"
    <xs:element name="maxItems" type="xs:integer"
        minOccurs="1" maxOccurs="1" nillable="true" />
minOccurs="0"
    <xs:element name="skipCount" type="xs:integer"
        minOccurs="1" maxOccurs="1" nillable="true" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getRelationshipsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
            <xs:element name="hasMoreItems" type="xs:boolean"
                minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!--
-----
-->
<!--[Policy Services]-->
<!--
-----
-->
<!--applyPolicy Operation-->
<xs:element name="applyPolicy">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="repositoryId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="policyId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
            <xs:element name="objectId" type="xs:string"
                minOccurs="1" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="applyPolicyResponse">
    <xs:complexType>
        <xs:sequence />
    </xs:complexType>
</xs:element>
<!--

```



```

-----
-->
<!--removePolicy Operation-->
<xs:element name="removePolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="policyId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="removePolicyResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<!--
-----

-->
<!--getAppliedPolicies Operation-->
<xs:element name="getAppliedPolicies">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
      <xs:element name="objectId" type="xs:string"
minOccurs="1"
        maxOccurs="1" />
      <xs:element name="filter" type="xs:string"
minOccurs="0"
        maxOccurs="1" nillable="true" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getAppliedPoliciesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cmis:object" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--
-----

-->
<!-- getACLCapabilities -->
<xs:element name="getACLCapabilities">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
        minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="getACLCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="setType" type="cmis:enumACLsetType"
/>
      <xs:element name="permissions"
type="cmis:cmisAccessControlPermissionType"
      minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="mapping"
type="cmis:cmisPermissionMappingType"
      minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="getACL">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getACLResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="acl"
type="cmis:cmisAccessControlListType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="applyACL">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="repositoryId" type="xs:string"
minOccurs="1" maxOccurs="1" />
      <xs:element name="addACEs"
type="cmis:cmisAccessControlEntryType"
      minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="removeACEs"
type="cmis:cmisAccessControlEntryType"
      minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="applyACLResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ACEs"
type="cmis:cmisAccessControlEntryType"
      minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="exact" type="xs:boolean" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>

```

## 5 CMISWS-Service WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200901"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200901"
  xmlns:cmisw="http://docs.oasis-open.org/ns/cmisis/ws/200901"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://docs.oasis-open.org/ns/cmisis/ws/200901"
  xmlns:jaxws="http://java.sun.com/xml/ns/jaxws" name="CMISWebServices">
  <types>
    <xsd:schema elementFormDefault="qualified"
      targetNamespace="http://docs.oasis-
open.org/ns/cmisis/ws/200901">
      <xsd:import schemaLocation="CMIS-Core.xsd"
        namespace="http://docs.oasis-
open.org/ns/cmisis/core/200901" />
      <xsd:import schemaLocation="CMIS-Messaging.xsd"
        namespace="http://docs.oasis-
open.org/ns/cmisis/messaging/200901" />
    </xsd:schema>
  </types>

  <message name="cmisException">
    <part name="fault" element="cmism:cmisFault" />
  </message>

  <message name="getACLCapabilitiesRequest">
    <part name="parameters" element="cmism:getACLCapabilities" />
  </message>
  <message name="getACLCapabilitiesResponse">
    <part name="parameters" element="cmism:getACLCapabilitiesResponse" />
  </message>

  <message name="getACLRequest">
    <part name="parameters" element="cmism:getACL" />
  </message>
  <message name="getACLResponse">
    <part name="parameters" element="cmism:getACLResponse" />
  </message>

  <message name="applyACLRequest">
    <part name="parameters" element="cmism:applyACL" />
  </message>
  <message name="applyACLResponse">
    <part name="parameters" element="cmism:applyACLResponse" />
  </message>

  <message name="queryRequest">
    <part name="parameters" element="cmis:query" />
  </message>
  <message name="queryResponse">
    <part name="parameters" element="cmism:queryResponse" />
  </message>
  <message name="getContentChangesRequest">
    <part name="parameters" element="cmism:getContentChanges" />
  </message>
  <message name="getContentChangesResponse">
    <part name="parameters" element="cmism:getContentChangesResponse" />
  </message>
```

```

</message>

<message name="addObjectToFolderRequest">
  <part name="parameters" element="cmism:addObjectToFolder" />
</message>
<message name="addObjectToFolderResponse">
  <part name="parameters" element="cmism:addObjectToFolderResponse" />
</message>
<message name="removeObjectFromFolderRequest">
  <part name="parameters" element="cmism:removeObjectFromFolder" />
</message>
<message name="removeObjectFromFolderResponse">
  <part name="parameters"
element="cmism:removeObjectFromFolderResponse" />
</message>

<message name="getDescendantsRequest">
  <part name="parameters" element="cmism:getDescendants" />
</message>
<message name="getDescendantsResponse">
  <part name="parameters" element="cmism:getDescendantsResponse" />
</message>
<message name="getChildrenRequest">
  <part name="parameters" element="cmism:getChildren" />
</message>
<message name="getChildrenResponse">
  <part name="parameters" element="cmism:getChildrenResponse" />
</message>
<message name="getFolderParentRequest">
  <part name="parameters" element="cmism:getFolderParent" />
</message>
<message name="getFolderParentResponse">
  <part name="parameters" element="cmism:getFolderParentResponse" />
</message>
<message name="getObjectParentsRequest">
  <part name="parameters" element="cmism:getObjectParents" />
</message>
<message name="getObjectParentsResponse">
  <part name="parameters" element="cmism:getObjectParentsResponse" />
</message>
<message name="getCheckedOutDocsRequest">
  <part name="parameters" element="cmism:getCheckedOutDocs" />
</message>
<message name="getCheckedOutDocsResponse">
  <part name="parameters" element="cmism:getCheckedOutDocsResponse" />
</message>

<message name="createDocumentRequest">
  <part name="parameters" element="cmism:createDocument" />
</message>
<message name="createDocumentResponse">
  <part name="parameters" element="cmism:createDocumentResponse" />
</message>
<message name="createFolderRequest">
  <part name="parameters" element="cmism:createFolder" />
</message>
<message name="createFolderResponse">
  <part name="parameters" element="cmism:createFolderResponse" />
</message>
<message name="createRelationshipRequest">
  <part name="parameters" element="cmism:createRelationship" />
</message>
<message name="createRelationshipResponse">
  <part name="parameters" element="cmism:createRelationshipResponse" />

```

```

</message>
<message name="createPolicyRequest">
  <part name="parameters" element="cmism:createPolicy" />
</message>
<message name="createPolicyResponse">
  <part name="parameters" element="cmism:createPolicyResponse" />
</message>
<message name="getAllowableActionsRequest">
  <part name="parameters" element="cmism:getAllowableActions" />
</message>
<message name="getAllowableActionsResponse">
  <part name="parameters" element="cmism:getAllowableActionsResponse"
/>
</message>
<message name="getPropertiesRequest">
  <part name="parameters" element="cmism:getProperties" />
</message>
<message name="getPropertiesResponse">
  <part name="parameters" element="cmism:getPropertiesResponse" />
</message>

<message name="getFolderByPathRequest">
  <part name="parameters" element="cmism:getFolderByPath" />
</message>
<message name="getFolderByPathResponse">
  <part name="parameters" element="cmism:getFolderByPathResponse" />
</message>

<message name="getContentStreamRequest">
  <part name="parameters" element="cmism:getContentStream" />
</message>
<message name="getContentStreamResponse">
  <part name="parameters" element="cmism:getContentStreamResponse" />
</message>
<message name="updatePropertiesRequest">
  <part name="parameters" element="cmism:updateProperties" />
</message>
<message name="updatePropertiesResponse">
  <part name="parameters" element="cmism:updatePropertiesResponse" />
</message>
<message name="moveObjectRequest">
  <part name="parameters" element="cmism:moveObject" />
</message>
<message name="moveObjectResponse">
  <part name="parameters" element="cmism:moveObjectResponse" />
</message>
<message name="deleteObjectRequest">
  <part name="parameters" element="cmism:deleteObject" />
</message>
<message name="deleteObjectResponse">
  <part name="parameters" element="cmism:deleteObjectResponse" />
</message>
<message name="deleteTreeRequest">
  <part name="parameters" element="cmism:deleteTree" />
</message>
<message name="deleteTreeResponse">
  <part name="parameters" element="cmism:deleteTreeResponse" />
</message>
<message name="setContentStreamRequest">
  <part name="parameters" element="cmism:setContentStream" />
</message>
<message name="setContentStreamResponse">
  <part name="parameters" element="cmism:setContentStreamResponse" />
</message>

```

```

<message name="deleteContentStreamRequest">
  <part name="parameters" element="cmism:deleteContentStream" />
</message>
<message name="deleteContentStreamResponse">
  <part name="parameters" element="cmism:deleteContentStreamResponse"
/>
</message>

<message name="applyPolicyRequest">
  <part name="parameters" element="cmism:applyPolicy" />
</message>
<message name="applyPolicyResponse">
  <part name="parameters" element="cmism:applyPolicyResponse" />
</message>
<message name="removePolicyRequest">
  <part name="parameters" element="cmism:removePolicy" />
</message>
<message name="removePolicyResponse">
  <part name="parameters" element="cmism:removePolicyResponse" />
</message>
<message name="getAppliedPoliciesRequest">
  <part name="parameters" element="cmism:getAppliedPolicies" />
</message>
<message name="getAppliedPoliciesResponse">
  <part name="parameters" element="cmism:getAppliedPoliciesResponse" />
</message>

<message name="getRelationshipsRequest">
  <part name="parameters" element="cmism:getRelationships" />
</message>
<message name="getRelationshipsResponse">
  <part name="parameters" element="cmism:getRelationshipsResponse" />
</message>

<message name="getRepositoriesRequest">
  <part name="parameters" element="cmism:getRepositories" />
</message>
<message name="getRepositoriesResponse">
  <part name="parameters" element="cmism:getRepositoriesResponse" />
</message>
<message name="getRepositoryInfoRequest">
  <part name="parameters" element="cmism:getRepositoryInfo" />
</message>
<message name="getRepositoryInfoResponse">
  <part name="parameters" element="cmism:getRepositoryInfoResponse" />
</message>
<message name="getTypeChildrenRequest">
  <part name="parameters" element="cmism:getTypeChildren" />
</message>
<message name="getTypeChildrenResponse">
  <part name="parameters" element="cmism:getTypeChildrenResponse" />
</message>
<message name="getTypeDefinitionRequest">
  <part name="parameters" element="cmism:getTypeDefinition" />
</message>
<message name="getTypeDefinitionResponse">
  <part name="parameters" element="cmism:getTypeDefinitionResponse" />
</message>

<message name="checkOutRequest">
  <part name="parameters" element="cmism:checkOut" />
</message>
<message name="checkOutResponse">
  <part name="parameters" element="cmism:checkOutResponse" />

```

```

</message>
<message name="cancelCheckOutRequest">
  <part name="parameters" element="cmism:cancelCheckOut" />
</message>
<message name="cancelCheckOutResponse">
  <part name="parameters" element="cmism:cancelCheckOutResponse" />
</message>
<message name="checkInRequest">
  <part name="parameters" element="cmism:checkIn" />
</message>
<message name="checkInResponse">
  <part name="parameters" element="cmism:checkInResponse" />
</message>
<message name="getPropertiesOfLatestVersionRequest">
  <part name="parameters" element="cmism:getPropertiesOfLatestVersion"
/>
</message>
<message name="getPropertiesOfLatestVersionResponse">
  <part name="parameters"
element="cmism:getPropertiesOfLatestVersionResponse" />
</message>
<message name="getAllVersionsRequest">
  <part name="parameters" element="cmism:getAllVersions" />
</message>
<message name="getAllVersionsResponse">
  <part name="parameters" element="cmism:getAllVersionsResponse" />
</message>

<portType name="DiscoveryServicePort">
  <operation name="query">
    <input message="cmisw:queryRequest" />
    <output message="cmisw:queryResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getContentChanges">
    <input message="cmisw:getContentChangesRequest" />
    <output message="cmisw:getContentChangesResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
</portType>

<portType name="MultiFilingServicePort">
  <operation name="addObjectToFolder">
    <input message="cmisw:addObjectToFolderRequest" />
    <output message="cmisw:addObjectToFolderResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="removeObjectFromFolder">
    <input message="cmisw:removeObjectFromFolderRequest" />
    <output message="cmisw:removeObjectFromFolderResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
</portType>

<portType name="NavigationServicePort">
  <operation name="getDescendants">
    <input message="cmisw:getDescendantsRequest" />
    <output message="cmisw:getDescendantsResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getChildren">
    <input message="cmisw:getChildrenRequest" />
    <output message="cmisw:getChildrenResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>

```

```

<operation name="getFolderParent">
  <input message="cmisw:getFolderParentRequest" />
  <output message="cmisw:getFolderParentResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getObjectParents">
  <input message="cmisw:getObjectParentsRequest" />
  <output message="cmisw:getObjectParentsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
<operation name="getCheckedOutDocs">
  <input message="cmisw:getCheckedOutDocsRequest" />
  <output message="cmisw:getCheckedOutDocsResponse" />
  <fault message="cmisw:cmisException" name="cmisException" />
</operation>
</portType>
<portType name="ObjectServicePort">
  <operation name="createDocument">
    <input message="cmisw:createDocumentRequest" />
    <output message="cmisw:createDocumentResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="createFolder">
    <input message="cmisw:createFolderRequest" />
    <output message="cmisw:createFolderResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="createRelationship">
    <input message="cmisw:createRelationshipRequest" />
    <output message="cmisw:createRelationshipResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="createPolicy">
    <input message="cmisw:createPolicyRequest" />
    <output message="cmisw:createPolicyResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getAllowableActions">
    <input message="cmisw:getAllowableActionsRequest" />
    <output message="cmisw:getAllowableActionsResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getProperties">
    <input message="cmisw:getPropertiesRequest" />
    <output message="cmisw:getPropertiesResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getFolderByPath">
    <input message="cmisw:getFolderByPathRequest" />
    <output message="cmisw:getFolderByPathResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="getContentStream">
    <input message="cmisw:getContentStreamRequest" />
    <output message="cmisw:getContentStreamResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="updateProperties">
    <input message="cmisw:updatePropertiesRequest" />
    <output message="cmisw:updatePropertiesResponse" />
    <fault message="cmisw:cmisException" name="cmisException" />
  </operation>
  <operation name="moveObject">
    <input message="cmisw:moveObjectRequest" />

```



```

        <output message="cmisw:moveObjectResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteObject">
        <input message="cmisw:deleteObjectRequest" />
        <output message="cmisw:deleteObjectResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteTree">
        <input message="cmisw:deleteTreeRequest" />
        <output message="cmisw:deleteTreeResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="setContentStream">
        <input message="cmisw:setContentStreamRequest" />
        <output message="cmisw:setContentStreamResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="deleteContentStream">
        <input message="cmisw:deleteContentStreamRequest" />
        <output message="cmisw:deleteContentStreamResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
</portType>
<portType name="PolicyServicePort">
    <operation name="applyPolicy">
        <input message="cmisw:applyPolicyRequest" />
        <output message="cmisw:applyPolicyResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="removePolicy">
        <input message="cmisw:removePolicyRequest" />
        <output message="cmisw:removePolicyResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getAppliedPolicies">
        <input message="cmisw:getAppliedPoliciesRequest" />
        <output message="cmisw:getAppliedPoliciesResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
</portType>

<portType name="RelationshipServicePort">
    <operation name="getRelationships">
        <input message="cmisw:getRelationshipsRequest" />
        <output message="cmisw:getRelationshipsResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
</portType>

<portType name="RepositoryServicePort">
    <operation name="getRepositories">
        <input message="cmisw:getRepositoriesRequest" />
        <output message="cmisw:getRepositoriesResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getRepositoryInfo">
        <input message="cmisw:getRepositoryInfoRequest" />
        <output message="cmisw:getRepositoryInfoResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>
    <operation name="getTypeChildren">
        <input message="cmisw:getTypeChildrenRequest" />
        <output message="cmisw:getTypeChildrenResponse" />
        <fault message="cmisw:cmisException" name="cmisException" />
    </operation>

```

```

        </operation>
        <operation name="getTypeDefinition">
            <input message="cmisw:getTypeDefinitionRequest" />
            <output message="cmisw:getTypeDefinitionResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
    </portType>
    <portType name="VersioningServicePort">
        <operation name="checkOut">
            <input message="cmisw:checkOutRequest" />
            <output message="cmisw:checkOutResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="cancelCheckOut">
            <input message="cmisw:cancelCheckOutRequest" />
            <output message="cmisw:cancelCheckOutResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="checkIn">
            <input message="cmisw:checkInRequest" />
            <output message="cmisw:checkInResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="getPropertiesOfLatestVersion">
            <input message="cmisw:getPropertiesOfLatestVersionRequest" />
            <output message="cmisw:getPropertiesOfLatestVersionResponse"
/>
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="getAllVersions">
            <input message="cmisw:getAllVersionsRequest" />
            <output message="cmisw:getAllVersionsResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
    </portType>
    <portType name="ACLServicePort">
        <operation name="getACLCapabilities">
            <input message="cmisw:getACLCapabilitiesRequest" />
            <output message="cmisw:getACLCapabilitiesResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="getACL">
            <input message="cmisw:getACLRequest" />
            <output message="cmisw:getACLResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
        <operation name="applyACL">
            <input message="cmisw:applyACLRequest" />
            <output message="cmisw:applyACLResponse" />
            <fault message="cmisw:cmisException" name="cmisException" />
        </operation>
    </portType>

    <binding name="DiscoveryServicePortBinding"
type="cmisw:DiscoveryServicePort">
        <soap:binding style="document"
            transport="http://schemas.xmlsoap.org/soap/http" />
        <operation name="query">
            <soap:operation soapAction="" />
            <input>
                <soap:body use="literal" />
            </input>
            <output>

```

```

        <soap:body use="literal" />
    </output>
    <fault name="cmisException">
        <soap:fault name="cmisException" use="literal" />
    </fault>
</operation>
<operation name="getContentChanges">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="cmisException">
        <soap:fault name="cmisException" use="literal" />
    </fault>
</operation>
</binding>
<binding name="MultiFilingServicePortBinding"
type="cmisw:MultiFilingServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="addObjectToFolder">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="removeObjectFromFolder">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>
<binding name="NavigationServicePortBinding"
type="cmisw:NavigationServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getDescendants">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</operation>

```

```

<operation name="getChildren">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getFolderParent">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getObjectParents">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getCheckedOutDocs">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
</binding>
<binding name="ObjectServicePortBinding" type="cmisw:ObjectServicePort">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="createDocument">
    <soap:operation soapAction="" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
    <fault name="cmisException">
      <soap:fault name="cmisException" use="literal" />
    </fault>
  </operation>

```

```

<operation name="createFolder">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="createRelationship">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="createPolicy">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getAllowableActions">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getProperties">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>
<operation name="getFolderByPath">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
  <fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
  </fault>
</operation>

```

```

        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getContentStream">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="updateProperties">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="moveObject">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="deleteObject">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="deleteTree">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

```

```

        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="setContentStream">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="deleteContentStream">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>
<binding name="PolicyServicePortBinding" type="cmisw:PolicyServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="applyPolicy">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="removePolicy">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getAppliedPolicies">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

```

```

        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>
<binding name="RelationshipServicePortBinding"
type="cmisw:RelationshipServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getRelationships">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>
<binding name="RepositoryServicePortBinding"
type="cmisw:RepositoryServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getRepositories">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getRepositoryInfo">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getTypeChildren">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getTypeDefinition">
        <soap:operation soapAction="" />

```



```

        <input>
            <soap:body use="literal" />
        </input>
    </output>
    <soap:body use="literal" />
</output>
<fault name="cmisException">
    <soap:fault name="cmisException" use="literal" />
</fault>
</operation>
</binding>
<binding name="VersioningServicePortBinding"
type="cmisw:VersioningServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="checkOut">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="cancelCheckOut">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="checkIn">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getPropertiesOfLatestVersion">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getAllVersions">

```

```

        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>
<binding name="ACLServicePortBinding" type="cmisw:ACLServicePort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getACLCapabilities">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="getACL">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
    <operation name="applyACL">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="cmisException">
            <soap:fault name="cmisException" use="literal" />
        </fault>
    </operation>
</binding>

<service name="DiscoveryService">
    <port name="DiscoveryServicePort"
binding="cmisw:DiscoveryServicePortBinding">
        <soap:address

location="http://cmis/services/DiscoveryService.DiscoveryServicePort" />
    </port>
</service>
<service name="MultiFilingService">

```

```

        <port name="MultiFilingServicePort"
binding="cmisw:MultiFilingServicePortBinding">
        <soap:address

        location="http://cmis/services/MultiFilingService.MultiFilingServicePort"
/>
        </port>
</service>
<service name="NavigationService">
        <port name="NavigationServicePort"
binding="cmisw:NavigationServicePortBinding">
        <soap:address

        location="http://cmis/services/NavigationService.NavigationServicePort" />
        </port>
</service>
<service name="ObjectService">
        <port name="ObjectServicePort"
binding="cmisw:ObjectServicePortBinding">
        <soap:address

        location="http://cmis/services/ObjectService.ObjectServicePort" />
        </port>
</service>
<service name="PolicyService">
        <port name="PolicyServicePort"
binding="cmisw:PolicyServicePortBinding">
        <soap:address

        location="http://cmis/services/PolicyService.PolicyServicePort" />
        </port>
</service>
<service name="RelationshipService">
        <port name="RelationshipServicePort"
binding="cmisw:RelationshipServicePortBinding">
        <soap:address

        location="http://cmis/services/RelationshipService.RelationshipServicePort"
/>
        </port>
</service>
<service name="RepositoryService">
        <port name="RepositoryServicePort"
binding="cmisw:RepositoryServicePortBinding">
        <soap:address

        location="http://cmis/services/RepositoryService.RepositoryServicePort" />
        </port>
</service>
<service name="VersioningService">
        <port name="VersioningServicePort"
binding="cmisw:VersioningServicePortBinding">
        <soap:address

        location="http://cmis/services/VersioningService.VersioningServicePort" />
        </port>
</service>
<service name="ACLService">
        <port name="ACLServicePort" binding="cmisw:ACLServicePortBinding">
        <soap:address

        location="http://cmis/services/ACLService.ACLServicePort" />
        </port>
</service>
</definitions>

```

---

## # Conformance

The last numbered section in the specification must be the Conformance section. Conformance Statements/Clauses go here.

---

## A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

**Participants:**

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

---

## B. Non-Normative Text

---

## C. Revision History

Revision	Date	Editor	Changes Made
.61	10 April 2009	Al Brown	
.62	8 July 2009	Ryan McVeigh	