



# Content Management Interoperability Services Version 0.62c

## Part II – ReSTful AtomPub Binding

### Committee Draft

29 June 2009

#### Specification URIs:

##### This Version:

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

##### Previous Version:

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

##### Latest Version:

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

- 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/restatom/200901>

#### Abstract:

This specification defines the ReSTful AtomPub 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

1	Introduction.....	7
1.1	Terminology.....	7
1.2	Normative References.....	7
1.3	Non-Normative References.....	7
2	Overview.....	9
2.1	Authentication.....	9
2.2	Response Formats.....	9
2.3	Optional Arguments.....	9
2.4	Errors and Exceptions.....	10
2.5	E-Tags.....	10
2.6	HTTP Options.....	10
3	CMIS.....	11
3.1	Mime Types.....	11
3.1.1	Atom Media Type with CMIS extensions.....	11
3.1.2	CMIS Query.....	11
	CMIS.....	12
3.1.3	Allowable Actions.....	12
3.1.4	CMIS Tree.....	12
3.1.5	CMIS ACL.....	12
3.2	CMIS Link Relations.....	12
3.2.1	Existing Link Relations.....	13
3.2.2	Hierarchy Navigation Internet Draft Link Relations.....	13
3.2.3	Versioning Internet Draft Link Relations.....	13
3.2.4	CMIS Specific Link Relations.....	13
3.3	Exceptions.....	14
3.3.1	Common Exceptions.....	14
3.3.2	Other Exceptions.....	14
3.3.3	Other notable HTTP Exceptions.....	14
3.4	Renditions.....	14
3.5	Content Streams.....	15
4	Atom Resources.....	16
4.1	Feeds.....	16
4.2	Entries.....	16
4.2.1	Hierarchical Atom Entries.....	17
4.3	Link relations.....	17
5	AtomPub Service Document (Repository).....	18
5.1	URI Templates.....	19
5.1.1	Entry By Id.....	19
5.1.2	Folder By Path.....	20
5.1.3	Query.....	20
5.2	HTTP Methods.....	21
5.2.1	GET.....	21
6	Service Collections.....	22

6.1	Root Folder Collections .....	22
6.2	Query Collection .....	22
6.2.1	POST .....	22
6.3	Checked Out Collection .....	23
6.3.1	GET .....	23
6.3.2	POST .....	23
6.4	Unfiled Collection .....	24
6.4.1	GET .....	24
6.4.2	POST .....	24
6.5	Types Children Collection .....	25
6.5.1	GET .....	25
7	Collections .....	26
7.1	Relationships Collection .....	26
7.1.1	GET .....	26
7.1.2	POST .....	26
7.2	Folder Children Collection .....	27
7.2.1	GET .....	27
7.2.2	POST .....	27
7.3	Policies Collection .....	29
7.3.1	GET .....	29
7.3.2	POST .....	30
8	Feeds .....	31
8.1	Object Parents Feed .....	31
8.1.1	GET .....	31
8.2	Changes .....	31
8.3	Folder Descendants .....	32
8.3.1	GET .....	32
8.3.2	DELETE .....	32
8.4	Folder Tree .....	32
8.4.1	GET .....	33
8.4.2	DELETE .....	33
8.5	AllVersions Feed .....	33
8.5.1	GET .....	34
8.5.2	DELETE .....	34
8.6	Type Descendants Feed .....	34
8.6.1	GET .....	34
9	Resources .....	35
9.1	Type Entry .....	35
9.1.1	GET .....	35
9.2	Document Entry .....	35
9.2.1	GET .....	36
9.2.2	PUT .....	36
9.2.3	DELETE .....	37
9.3	Document Private Working Copy (PWC) Entry .....	37
9.3.1	GET .....	37

9.3.2 PUT .....	38
9.3.3 DELETE.....	38
9.4 Folder Entry .....	38
9.4.1 GET .....	39
9.4.2 PUT .....	39
9.4.3 DELETE.....	39
9.5 Relationship Entry.....	39
9.5.1 GET .....	40
9.5.2 PUT .....	40
9.5.3 DELETE.....	40
9.6 Policy Entry.....	40
9.6.1 GET .....	41
9.6.2 PUT .....	41
9.6.3 DELETE.....	41
9.7 Content Stream.....	41
9.7.1 GET .....	41
9.7.2 PUT .....	42
9.7.3 DELETE.....	42
9.8 Folder Parent Entry.....	42
9.8.1 GET .....	42
9.9 ACL Resource .....	42
9.9.1 GET .....	42
10 CMIS-RestAtom Schema .....	43
11 IANA Considerations .....	44
11.1 Document Types.....	44
11.1.1 CMIS Query.....	44
11.1.2 CMIS AllowableActions .....	44
11.1.3 CMIS Tree .....	45
11.1.4 CMIS Atom .....	46
11.1.5 CMIS ACL.....	47
# Conformance.....	49
A. Acknowledgements .....	50
Participants: .....	50
B. Non-Normative Text .....	51
B.1 Atom Link Relations by Object Type .....	51
B.2 Atom and AtomPub Extensions .....	51
B.3 Examples.....	51
B.4 Expressing multiple content streams in REST .....	53
C. Revision History.....	54

---

# 1 Introduction

The Content Management Interoperability Services (CMIS) ReSTful AtomPub binding specification defines a specification based on AtomPub that can be used by applications to work with one or more Content Management Repositories.

It is expected that this binding will be leveraged to build applications such as:

- Quick UI components on the components (e.g., Widgets and Mashups)
- Consumed by feed-centric applications such as Yahoo pipes
- Content-centric applications close to the glass (Java/JSP, .NET, AJAX)
- Content-centric rich Internet Applications (e.g., flex, air)

## 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

- |           |  |
|-----------|--|
| [RFC4287] | M. Nottingham, R. Sayre, <i>Atom Syndication Format</i> ,<br><a href="http://www.ietf.org/rfc/rfc4287.txt">http://www.ietf.org/rfc/rfc4287.txt</a> , December 2005   |
| [RFC5023] | J. Gregorio, B. de hOra, <i>Atom Publishing Protocol</i> ,<br><a href="http://www.ietf.org/rfc/rfc5023.txt">http://www.ietf.org/rfc/rfc5023.txt</a> , October 2007   |
| [RFC2616] | R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, <i>Hypertext Transfer Protocol --HTTP/1.1</i> , <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> , June 1999 |
| [RFC2119] | S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> ,<br><a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a> , March 1997  |
| [CMISDM]  | OASIS, Committee Draft 0.52, “Content Management Interoperability Services (CMIS) Domain Model”, 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 Overview

This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023). Implementations of CMIS must be compliant with RFC4287 and RFC5023 and all RFCs that supersede them.

In the ReSTful AtomPub binding, the client starts with the service document. The client will request the service document by the URI provided by the vendor. The client will then choose a CMIS collection, and then start accessing the repository by following the references in the returned documents.

The CMIS binding consists of a service document specifying at least CMIS service collections, atom collections, feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to the requested resource.

When requesting a resource, optional parameters may be specified to change default behavior.

Special collections have been created that have semantic meaning beyond collection membership. These are:

- Unfiled – All documents added to this collection will be removed from all other collections
- CheckedOut – All documents added to this collection will be checkedout

### 2.1 Authentication

Authentication SHOULD be handled by the transport protocol. Please see AtomPub section 14.

### 2.2 Response Formats

The client can specify in HTTP the Accept header which specifies which formats are acceptable to the client. With this mechanism the client can chose which response format the CMIS implementation should respond with. The CMIS compliant implementation MUST support this one response format:

- application/atom+xml

The CMIS repository will chose the response format based on the Accept header if specified. If the Accept header is not specified, then the CMIS repository should respond with the format specified by the specification.

The CMIS repository may support other formats such as:

- application/json for JSON
- text/html for an HTML interface to the API

### 2.3 Optional Arguments

The binding supports adding optional parameters to CMIS resources to modify default behavior. These HTTP query string parameters would be appended to the URI specified.

CMIS implementations MUST support arguments being specified as HTTP query string parameters.

Names and valid values for HTTP query string parameters are as described in the appropriate CMIS Service descriptions [see Part I]. Valid values of enum types are also represented in the schema, as described in part 1 of CMIS

## 2.4 Errors and Exceptions

Exceptions shall be thrown as described in the appropriate CMIS Service description. Exceptions are to be mapped to an appropriate HTTP error code, and it is suggested that the body of an error reply be sufficient for a user to determine corrective action.

## 2.5 E-Tags

CMIS changeTokens are represented as E-Tags and follow HTTP's use of E-Tags. CMIS server implementations SHOULD support E-Tags.

## 2.6 HTTP Ranges

It is RECOMMENDED that HTTP Range requests be supported on Content Streams.

It is RECOMMENDED that HTTP compression is also supported.

## 2.7 HTTP Verb OPTIONS

The repository SHOULD support the HTTP Options verb on all the resources defined in this specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP verbs specified for that resource.

---

## 3 CMIS

### 3.1 Mime Types

CMIS introduces new media types for:

- a CMIS Query document (application/cmismquery+xml)
- a CMIS AllowableActions document (application/cmismallowableactions+xml)
- an Atom Document (Entry or Feed) with any CMIS Markup
- an Atom Feed Document with CMIS Hierarchy extensions (application/cmismtree+xml)

In addition to those media types specified by CMIS, CMIS also leverages these media types:

- AtomPub Service (application/atomsvc+xml)
- Atom Entry (application/atom+xml;type=entry)
- Atom Feed (application/atom+xml;type=feed)

#### 3.1.1 Atom Media Type with CMIS extensions

Media Type: application/cmismatom+xml

Starting tag: atom feed or atom entry

Type Parameters:

- type – the semantics of the type parameter MUST be the same as the media type parameter for atom documents.

This allows clients to differentiate between atom media type without CMIS extensions and atom media type with CMIS extensions.

This media type SHOULD NOT be used with CMIS extensions for nesting atom entries (CMIS Tree).

Example:

```
TODO: Add example here before Public Review
```

#### 3.1.2 CMIS Query

Media Type: application/cmismquery+xml

Starting tag: query

This document contains the representation of a query to be executed in a CMIS repository.

Example:

```
<query xmlns="http://www.cmism.org/CMIS/1.0">
  <statement>SELECT * FROM document</statement>
  <searchAllVersions>>false</searchAllVersions>
  <pageSize>0</pageSize>
</query>
```

```
</query> <skipCount>0</skipCount>
```

### 3.1.3 CMIS Allowable Actions

Media Type: application/cmisableactions+xml

Starting tag: allowableactions

This document contains the representation of the allowable actions the user may perform on the referenced object.

Example:

```
TODO: Add example here before Public Review
```

### 3.1.4 CMIS Tree

Media Type: application/cmistree+xml

Starting tag: atom feed

This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.

Example:

```
TODO: Add example here before Public Review
```

### 3.1.5 CMIS ACL

Media Type: application/cmisacl+xml

Starting tag: cmis:cmisAccessControlListType

This document specifies an Access Control List based on the schema in part 1.

Example:

```
TODO: Add example here before Public Review
```

## 3.2 CMIS Link Relations

The listing below outlines the different link types in CMIS. This is in addition to the link relations specified by Atom and Atom Publishing Protocol. The registry for link relations is located at <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

Links may have the following attributes in addition to the ones specified by Atom and Atom Publishing Protocol:

- (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. It is recommended to include this attribute for links that point to CMIS resources that have an id.

These are the link relation types specified by CMIS:

### 3.2.1 Existing Link Relations

Existing link relations should be used where appropriate by the implementation. In addition, the following link relations are leveraged for the CMIS specification:

- self
- service
- describedby
- via
- edit-media
-  alternate
- first
- previous
- next
- last

Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on these link relations.

### 3.2.2 Hierarchy Navigation Internet Draft Link Relations

Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-divilly-atom-hierarchy-03.txt>.

CMIS leverages the following link relations from the Internet Draft:

- up
- down

### 3.2.3 Versioning Internet Draft Link Relations

Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-brown-versioning-link-relations-01.txt>.

CMIS leverages the following link relations from the Internet Draft:

- all-versions
- current-version
- working-copy

### 3.2.4 CMIS Specific Link Relations

CMIS defines the following link relations:

- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>
  - Service: getAllowableActions
- <http://docs.oasis-open.org/ns/cmisis/link/200901/relationships>
  - Service: getRelationships
- <http://docs.oasis-open.org/ns/cmisis/link/200901/source>
  - Source Link on Relationship

- <http://docs.oasis-open.org/ns/cmisis/link/200901/target>
  - Target Link on Relationship
- <http://docs.oasis-open.org/ns/cmisis/link/200901/target>
  - Service: getACLPermissions
- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>
  - Service: getACL




## 3.3 Exceptions

### 3.3.1 Common Exceptions

The following listing defines the HTTP status codes that repositories will return for the various common exceptions defined in Part I of the CMIS specification.

CMIS Services Exception	HTTP Status Code
invalidArgument	400
objectNotFound	404
permissionDenied	403
operationNotSupported	405
updateConflict	409
runtime	500

### 3.3.2 Other Exceptions

 Services Exception	HTTP Status Code 
constraintViolation	409
filterNotValid	 403
streamNotSupported	403
storage	500
contentAlreadyExists	409
versioning	409
folderNotValid	400

### 3.3.3 Other notable HTTP Exceptions

- 415 Unsupported Media Type
  - When a media type is POST'ed to a collection that is not supported, this exception MUST be returned
- 422 Unprocessable Entity
  - When a request has been POST'ed but cannot be processed, this exception MUST be returned

## 3.4 Renditions

Each Rendition included in a CMIS AtomPub response is represented as an Atom link rel="alternate". The following attributes SHOULD be included on the link element:

- href: URI to the rendition content stream
- type: The Media Type of the Rendition

- cmis:renditionType: The Rendition Type

The following attributes MAY be included

- title: The Filename (or name property if object) of Rendition
- length: The length of the rendition

If a thumbnail is available, the repository SHOULD include the thumbnail as alternate link as part of the atom entry even if the optional includeRenditions argument is not specified.

### **3.5 Content Streams**

The content stream for a document SHOULD be referenced by the content src attribute as well as the edit-media link relation.

The following attributes SHOULD be included on the link element:

- href: URI to the content stream
- type: The Media Type of the content stream

---

## 4 Atom Resources

For all Atom Resources used in this specification, the following MUST be followed:

### 4.1 Feeds

Any feed MUST be a valid atom Feed document and conform to the guidelines below:

1. atom:updated MUST be the latest time the folder or its contents was updated. If unknown by the underlying repository, it should be the current time.
2. atom:author/atom:name MUST be the CMIS property cmis:CreatedBy
3. atom:title MUST be the CMIS property cmis:Name
4. The atom:link with relation self MUST be generated to return the URI of the feed
5. A feed SHOULD contain the element app:collection element, describing the appropriate media types supported for creation of new entries in the feed
6. atom:id must be the most stable ID available to the repository. This id MUST be compliant with atom's specification and be a valid URI.
7. Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than contained in the feed, then a link with the relation next MUST be included in the feed.

### 4.2 Entries

At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry document and conform to the guidelines below:

1. atom:Title MUST be the cmis:Name property
2. app:edited MUST be cmis:LastModifiedDate
3. atom:updated MUST be cmis:LastModifiedDate
4. atom:published MUST be cmis:CreatedDate
5. atom:author/atom:name MUST be cmis:CreatedBy
6. The repository SHOULD make a best effort at generating HTML or text that represents the object. For example, an HTML table containing the properties and their values for simple feed readers. That representation will be supplied as follows:
  - a. For Documents:
    - i. The atom:content/src attribute SHOULD be used to point to the content stream
    - ii. The repository SHOULD populate the atom:summary tag with text that at best efforts represents the document. For example, an HTML table containing the properties and their values for simple feed readers
  - b. Other Objects (Folders, Relationships, Types, etc):
    - i. The repository MUST comply with the atom specification and have a content element. How this is done, is repository specific. Any value in the content field MUST be ignored if the atom entry represents a non-document object.
    - ii. The repository SHOULD populate the atom:summary tag with text that at best efforts represents the object. For example, an HTML table containing the properties and their values for simple feed readers
7. Links will be used to provide URIs to CMIS functionality
8. Link relations MAY be omitted if the function is not allowed and that function would not show up in Allowable Actions
9. Links may be omitted if the repository does not support that capability



10. All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are duplicated in an atom element
11. atom:id MUST be the most stable ID available to the repository. This id MUST be compliant with atom's specification and be a valid URI.

When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding writable CMIS property. For example, atom:title will overwrite cmis:Name.

### 4.2.1 Hierarchical Atom Entries

For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate a cmisra:children element with the enclosing feed of its direct children. This pattern continues until the depth is satisfied.

The cmisra:children element that MUST be included in an atom entry:

- `<xs:element name="children" type="atom:feedType" />`

If an entry does not contain cmisra:children element, then the entry MAY have children even though it is not represented in the atom entry.

For Example,

TODO: Add in Example for F2F
------------------------------

### 4.3 Link relations

The link element with a specified relation MUST be included if client can perform the operation. The repository MAY omit the link relation if the operation is not available. The operation may not be available due to a variety of reasons such as access control, administrative policies, or other mechanisms.

---

## 5 AtomPub Service Document (Repository)

The AtomPub Service Document contains the set of repositories that are available. Each repository is mapped to a Workspace in the AtomPub Service document.

CMIS Services exposed:

GET: getRepositories, getRepositoryInfo

Media Type: application/atomsvc+xml

How the client will get the initial AtomPub (APP) service document or the URI for the service document is repository specific. Examples are via URI, or loading the service document from disk.

The service document will be available from Atom Entry and Atom Feed documents via a link relationship, service.

A workspace element for a CMIS repository MUST have a collection element for each of following collections:

- Root Folder Collection: Root folder of the Repository
  - 'root' for the children collection of the root folder
- Types Collection: Collection containing all the types in the repository
  - 'types' for the children collection

The workspace element SHOULD contain these collections if the repository supports this functionality:

- CheckedOut collection: collection containing all checked out documents user can see
  - 'checkedout'
- Query collection: Collection for posting queries to be executed
  - 'query'
- Changes collection: Collection for understanding changes in the repository
  - 'changes'
- Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
  - 'unfiled'

The workspace element will have two CMIS attributes:

- cmis:id: This is the id of the repository if available. This attribute is located on the workspace element.
- cmis:repositoryRelationship. RepositoryRelationship attribute is optional. RepositoryRelationship specifies the relationship of the repository to others listed in the service document. The repository name will be exposed in the workspace element via the atom:title element.

If the repository supports the URI templates, then the repository SHOULD include the uri templates in the workspace elements.

The workspace element SHOULD also contain the following link elements with the relation:

- `foldertree` (in CMIS namespace): This link relation points to the folder tree of the root folder. See Folder Tree resource for more information.
- `rootdescendants` (in CMIS namespace): This link relation points to the descendants collection of the root folder.
- `typesdescendants` (in CMIS namespace): This link relation points to the types descendants for the base types in the repository.

The workspace element may include `app:collection` elements for the collections that represent folders in the repository. However, an alternative approach, especially for a repository with many folders, is to not enumerate those collections here, but include the `app:collection` element per RFC5023 in the Atom Feed document.

## 5.1 URI Templates

Implementations SHOULD include URI templates in the workspace element of the service document as defined by CMIS. CMIS defines the following types:

- `entrybyid`
- `folderbypath`
- `query`

Repositories MAY extend that set of types. Those URI Template Types will be repository specific. Repositories MAY have more than one entry per uri type if the entries have different media types.

Structure of URI Template:

```
<xs:complexType name="cmisUriTemplateType" >
  <xs:sequence>
    <xs:element name="template" type="xs:string" />
    <xs:element name="type" type="xs:string" />
    <xs:element name="mediatype" type="xs:string" />
    <xs:any processContents="lax" namespace="##other"
minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

Repositories SHOULD include at least the URI templates with corresponding Media Type defined below in the AtomPub services document.

### 5.1.1 Entry By Id

Type: `entrybyid`

Media Type: `application/atom+xml;type=entry`

Service: `getProperties`

Variables that MUST be in the template:

- `{id}`: Id of object

Variables that MAY be in the template:

- {filter}: Property Filter
- {includeAllowableActions}: Property Filter
- {includeRelationships}: Include Relationships.
- {includeACL}: Include ACL

Example:

TODO: Add example here before Public Review

## 5.1.2 Folder By Path

Type: folderbypath

Media Type: application/atom+xml;type=entry

Service: getFolderByPath

Variables in Template:

- {path}: Path of folder.
  - No escaping is provided for the variables. It is suggested that this be an URI argument so no escaping is necessary, or the last part of the URI so it can be interpreted correctly.

Variables that MAY be in the template:

- {filter}: Property Filter
- {includeAllowableActions}: Property Filter
- {includeRelationships}: Include Relationships.
- {includeACL}: Include ACL

Example:

TODO: Add example here before Public Review

## 5.1.3 Query

Type: query

Media Type: application/atom+xml;type=feed

Service: query

Variables in Template:

- {q}: CMIS Query Statement
- {searchAllVersions}: Boolean, true if to search all versions

Variables that MAY be in the template:

- {maxItems}: Integer, Max items to return
- {skipCount}: Items to skip
- {includeAllowableActions}: Property Filter
- {includeRelationships}: Include Relationships.

Example:

TODO: Add example here before Public Review
---

## 5.2 HTTP Methods

### 5.2.1 GET

This retrieves the AtomPub Service document for a specified repository and potentially related repositories. This exposes the capabilities defined in `getRepositories` and `getRepositoryInfo` in the Domain Model.

---

## 6 Service Collections

These are the collections that are included on an AtomPub Service document in the workspace element.

### 6.1 Root Folder Collections

This is a collection described in the service document. Please see [Folder Children](#).

### 6.2 Query Collection

This is a collection for processing queries. If the implementation supports GET on this collection, then the implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom entries should represent persisted objects related to query such as persisted queries, long running queries or search templates.

CMIS Services exposed via HTTP verbs:

GET: Repository Specific.

POST: Query

Media Type: application/atom+xml;type=feed

Accept: Repository

- MUST support CMIS Query document,
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- paging link relations as appropriate: first, next, prev, last if the repository supports paging a query result set

#### 6.2.1 POST

This collection MUST accept CMIS Query documents.

Upon submission (creation) of a query document, a response must be returned with a Location header representing the feed for that query or an exception must be thrown. In addition, the server SHOULD return the feed directly. If the server does so, the server should also return the Content-Location header.

The feed returned MUST contain a set of atom entries representing the result set from the query.

The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The atom entries MUST contain the CMIS extension element (cmis:object) containing the properties specified by the query in the select clause of the query statement.

If all the selected properties can be mapped to the same type reference, then the repository MAY include addition information in the atom entry.

Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- HTTP Success: 201
- Location Header
- Content-Location Header

Example:

TODO: Add example here before Public Review
---

## 6.3 Checked Out Collection

This is a collection described in the service document that contains all the checkedout documents

CMIS Services:

GET: getCheckedoutDocs

POST: checkout

Media Type: application/atom+xml;type=feed

Accept: Repository

- MUST support Atom Entry Documents with CMIS extensions
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- paging link relations as appropriate: first, next, prev, last

### 6.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- folderId
- maxItems
- skipCount
- includeAllowableActions
- includeRelationships

### 6.3.2 POST

When an atom entry is POST'ed to this collection, the atom entry will be checked out. A Content-Location header MUST be returned containing the location of the private working copy.

Example:

TODO: Add example here before Public Review
---

## 6.4 Unfiled Collection

This is a collection described in the service document that contains all the unfiled documents in the repository.

CMIS Services:

GET: getUnfiled

POST: removeObjectFromFolder

Media Type: application/atom+xml;type=feed

Accept: Repository

- MUST support Atom Entry Documents with CMIS extensions
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml

### 6.4.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- folderId
- maxItems
- skipCount
- includeAllowableActions
- includeRelationships

### 6.4.2 POST

This removes the object from all folders in the repository by default. If the optional argument removeFrom is specified, the object will only be removed from that folder.

If the Atom Entry POST'ed, does not have the CMIS extensions with a valid cmis:ObjectId, the document does not exist, or the document is not in that folder, an exception is thrown.

This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- HTTP Success: 201
- Location Header

The following arguments may be supplied. Please see the domain model for more information:

- removeFrom: For repositories which support multi-filing, this parameter identifies which folder to remove this object from. If specified, it indicates the folder from which the object shall be moved. If not specified, the object will be removed from all folders.

Example:

```
TODO: Add example here before Public Review
```



## 6.5 Types Children Collection

This is a collection described in the service document that contains the types in the repository under the specified parent type. If no parent type is specified, then the base types are returned in the feed. This feed does not include any nesting and is a flat feed.

CMIS Services:

GET: getTypeChildren

Media Type: application/atom+xml;type=feed

Accept: Repository

- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the type definition whose children represent this feed
- down: points to the atom feed document representing the descendants collection for this same type with media type of application/cmistree+xml
- paging link relations as appropriate: first, next, prev, last

### 6.5.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- includePropertyDefinitions
  - maxItems
  - skipCount
-

---

## 7 Collections

### 7.1 Relationships Collection

This is the set of relationships available (either source or target or both) from a specific item such as a document, folder or policy.

CMIS Services:

GET: getRelationships

POST: createRelationship

Media Type: application/atom+xml;type=feed

Accept: Repository

- MUST support Atom Entry Documents with CMIS extensions
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: Points to the originating atom entry document if available that contained the relationship link the client followed to get to this collection.
- paging link relations as appropriate: first, next, prev, last

#### 7.1.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- typeId
- includeSubRelationshipTypes
- enumRelationshipDirection
- maxItems
- skipCount
- filter
- includeAllowableActions

#### 7.1.2 POST

When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new CMIS relationship, then that relationship will be created.

The server SHOULD throw an exception if the source is different than the sourceId or target different than the targeted for the source and targets specified in this collection.

The server SHOULD throw an exception if the ObjectTypeId is not specified.

Example:

TODO: Add example here before Public Review
---

## 7.2 Folder Children Collection

This is a pseudo-object comprised of all the direct children of a particular folder.

CMIS Services:

GET: getChildren

POST:

createDocument  
or createFolder  
or createPolicy  
or moveObject  
or addObjectToFolder

Media Type: application/atom+xml;type=feed

Accept: Repository

- MUST support Atom Entry Documents with CMIS extensions
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of the folder generating this collection
- up: points to the atom feed document for the folder children collection of this folder's parent
- down: points to the atom feed document representing the descendents collection for this same folder with media type of application/cmistree+xml
- paging link relations as appropriate: first, next, prev, last
- <http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree>: Points to the folder tree for this folder

### 7.2.1 GET

Request Format: N/A

Response Format: application/atom+xml;type=feed (flat listing) or application/cmistree+xml (tree listing)

HTTP Code:

- 200 OK (Success)

The following arguments may be supplied. Please see the domain model for more information:

- If flat listing, maxItems and skipCount
- if tree listing, depth
- type
- filter
- includeAllowableActions
- includeRelationships

### 7.2.2 POST

CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- HTTP Success: 201
- Location Header

The following arguments may be supplied.

- **removeFrom:** For repositories which support multi-filing, this parameter identifies which folder to remove this object from. If specified, it indicates the folder from which the object shall be moved. If not specified, **addObjectToFolder** will be performed.

POSTing an Atom Entry document with CMIS markup:

**Adding a document to a folder:**

If the atom entry has a **cmis:objectId** that is valid for the repository, the object will be added to the folder.

When an object is added to the folder, in repositories that do not support multi-filing it will be removed from the previous folder and the operation treated as move. If the repository supports multiple folders, it will be added to the new folder. If the optional argument **removeFrom** is specified, then the object will be removed from the folder specified.

The optional argument **sourceFolderId** MAY specify the folder from which to remove the object.

Example:

TODO: Add example here before Public Review


**Creating a CMIS Object (in that folder):**

If the **cmis:objectId** property is missing, it will be created and then added to the folder.

For Documents:

A content stream **MUST** be specified if required by the type definition. If not provided and it is required, an exception will be thrown. This is even true if the CMIS Document is not checked in on create.

Content Streams **MAY** be provided by any mechanism supported by Atom Publishing Protocol:

- As part of the atom entry via the **src** attribute on the content element
  - **src** attribute: Implementers **SHOULD** support external references to content
- As part of the atom entry inlining via the content element
  - 64: Implementers **MUST** support content base64 encoded
- At a later time
  - At a later time by replacing the **edit-media** link with a new content

The optional argument **versioningState** MAY specify additional versioning behavior such as **checkin**.

Example:

TODO: Add example here before Public Review

If the **cmis:objectId** property is present but not valid an exception will be thrown.

POSTing other document formats:

The behavior is repository specific when a non Atom entry or an atom document without the CMIS elements is posted to a folder collection.

For example, the repository MAY auto-create a document with a specific type (document) the client could edit. The repository MAY also throw an exception.

The repository MUST follow HTTP specification and return the following HTTP codes:

- 500 if internal resource is unavailable to complete the request such as database, network, storage is unavailable or when an condition not specified as part of HTTP is not met
- 415 Unsupported Media Type if the repository cannot support the supplied format.



Example:

TODO: Add example here before Public Review

Optional arguments:

- versioningState (for createDocument)
- sourceFolderId (for moveObject)
- removeFrom
- thisVersion

## 7.3 Policies Collection

is an atom feed of all the policy objects currently applied

OMIS Services:

GET: getAllVersions

POST: createPolicy

Media Type application/atom+xml;type=feed

Accept: repository

- MUST support Atom Entry Documents with CMIS extensions
- MAY support other media type

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of the resource generating this collection
- paging link relations as appropriate: first, next, prev, last

### 7.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter

### 7.3.2 POST

When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the object.

Example:

TODO: Add example here before Public Review

---

## 8 Feeds

### 8.1 Object Parents Feed

This is the set of parents for a specific object.

CMIS Services:

GET: getObjectParents

Media Type: application/atom+xml;type=feed

Accept: not-specified

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of object whose parents are represented by this collection
- paging link relations as appropriate: first, next, prev, last

Example:

TODO: Add example here before Public Review
---

#### 8.1.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter

### 8.2 Changes

This is a link relationship described in the service document that contains the changes in the repository in the workspace element. The link relation pointing to this feed is <http://docs.oasis-open.org/ns/cmris/link/200901/changes>.

CMIS Services:

GET: getChanges()

Media Type: application/atom+xml;type=feed

Accept: Atom Entry document, other repository-specific

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- paging link relations as appropriate: first, next, prev, last

If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is not available, the client should revisit the feed in the future and look for new items and the next link relation.

Example:

TODO: Add example here before Public Review
---

## 8.3 Folder Descendants

This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available via the link relation [http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree\\_](http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree_). Please see the Hierarchical Atom Entries for more information on format.

CMIS Services:

GET: getDescendants

DELETE: deleteTree

Media Type: application/atom+xml;type=feed

Accept: Atom Entry Document with CMIS extensions, other repository specific

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of the folder generating this collection
- up: points to the atom feed document for the folder children collection of this folder's parent
- down: points to the atom feed document representing the children feed for this same folder with media type of application/atom+xml
- paging link relations as appropriate: first, next, prev, last
- [http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree\\_](http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree_): Points to the folder tree for this folder

Example:

TODO: Add example here before Public Review
---

### 8.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions

### 8.3.2 DELETE

This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the domain model for more information:

- continueOnFailure
- unfileObjects

## 8.4 Folder Tree

This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link relation [http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree\\_](http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree_). Please see the Hierarchical Atom Entries for more information on format.

CMIS Services:

GET: getDescendants

DELETE: deleteTree

Media Type: application/atom+xml;type=feed



Accept: Atom Entry Document with CMIS extensions, other repository specific

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of the folder generating this collection
- up: points to the atom feed document for the folder children collection of this folder's parent
- down:
  - application/atom+xml : Points to the atom feed document representing the children feed for this same folder
  - application/cmistree+xml: Points to the descendants feed of the same folder
- paging link relations as appropriate: first, next, prev, last
- <http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree>: Points to the folder tree for this folder

Example:

TODO: Add example here before Public Review
---

## 8.4.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions

## 8.4.2 DELETE

This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the domain model for more information:

- continueOnFailure
- unfileObjects

## 8.5 AllVersions Feed

This is a feed comprised of all the versions of the given document.

CMIS Services:

GET: getAllVersions

DELETE: deleteAllVersions

Media Type: application/atom+xml;type=feed

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the atom entry of the resource generating this collection
- paging link relations as appropriate: first, next, prev, last

Example:

TODO: Add example here before Public Review
---

## 8.5.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships

## 8.5.2 DELETE

This removes the entire version history of the document.

Success HTTP code: 204

## 8.6 Type Descendants Feed

This is a feed described in the service document that contains all the types under a specific type in the repository to a specific depth. If no parent type is specified, then the base types are returned in the feed. The link relation is <http://docs.oasis-open.org/ns/cmisis/link/200901/typesdescendants>.

CMIS Services:

GET: getTypes

Media Type: application/atom+xml;type=feed

Accept: Atom Entry document, other repository-specific

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- via: points to the type definition whose descendents represent this feed
- down: points to the children feed for the same type

Example:

TODO: Add example here before Public Review
---

### 8.6.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- includePropertyDefinitions
  - depth
-

---

## 9 Resources

### 9.1 Type Entry

This represents a type definition in the repository.

This is enclosed as an atom entry

CMIS Services:

GET: getTypeDefinition

Media Type: application/atom+xml;type=entry

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- up: Points to the parent type as atom entry if applicable
- down: Points to the children of this type as atom feed if applicable
  - Media Type: application/cmistree+xml points to the atom feed document representing the descendents collection for this same type
- describedby: Points to the type definition atom entry of the base type

One of the following elements MUST be included inside the atom entry:

- `<xs:element name="documentType" type="cmis:cmisTypeDocumentDefinitionType" substitutionGroup="cmis:type" />`
- `<xs:element name="folderType" type="cmis:cmisTypeFolderDefinitionType" substitutionGroup="cmis:type" />`
- `<xs:element name="relationshipType" type="cmis:cmisTypeRelationshipDefinitionType" substitutionGroup="cmis:type" />`
- `<xs:element name="policyType" type="cmis:cmisTypePolicyDefinitionType" substitutionGroup="cmis:type" />`

Example:

TODO: Add example here before Public Review
---

#### 9.1.1 GET

### 9.2 Document Entry

This is a CMIS Document instance.

CMIS Services:

GET: getProperties, getPropertiesOfLatestVersion

PUT: updateProperties

DELETE: deleteObject

Media Type: application/atom+xml;type=entry

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- up: Points to the atom feed containing the set of parents. If there is only one parent, the repository MAY point this link relation directly to the atom entry of the parent.
- all-versions: Points to atom feed containing the versions of this document
- latest-version
- edit-media
- working-copy
- describedby
- alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/relationships>: Points to the relationships feed for this object
- <http://docs.oasis-open.org/ns/cmisis/link/200901/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>: Points to ACL document for this object

The following element MUST be included inside the atom entry:

- `<xs:element name="object" type="cmis:cmisObjectType" />`

Example:

TODO: Add example here before Public Review
---

## 9.2.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- major
- returnVersion
  - Used to differentiate between `getProperties` and `getPropertiesOfLatestVersion`. If TRUE, execute `getPropertiesOfLatestVersion` service.
- includeAllowableActions
- includeRelationships
- filter
- includeRenditions
  - If an object has a Thumbnail rendition, that rendition SHOULD be included in the atom entry representation if this parameter is not specified. If this parameter is specified, the behavior should follow the specified value.

## 9.2.2 PUT

This does a complete replacement of the atom entry with the atom entry document specified. If properties are not included, they will be unset.

### 9.2.3 DELETE

This removes the document.

Success HTTP code: 204

## 9.3 Document Private Working Copy (PWC) Entry

This is the private working copy of the document (checkedout version of document)

CMIS Services:

GET: `getProperties`

PUT: `updateProperties` or `checkin`

DELETE: `cancelCheckout`

Media Type: `application/atom+xml;type=entry`

Link relations:

- `service`: Points to service document containing CMIS repository
  - Media Type: `application/atomsvc+xml`
- `up`: Points to the atom feed containing the set of parents. If there is only one parent, the repository MAY point this link relation directly to the atom entry of the parent.
- `all-versions`
- `edit-media`
- `via`: atom entry that created this private working copy
- `describedby`
- `alternate`: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/relationships>: Points to the relationships feed for this object
- <http://docs.oasis-open.org/ns/cmisis/link/200901/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>: Points to ACL document for this object

The following element MUST be included inside the atom entry:

- `<xs:element name="object" type="cmis:cmisObjectType" />`

Example:

TODO: Add example here before Public Review
---

### 9.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- `filter`
- `includeAllowableActions`
- `includeRelationships`
- `includeRenditions`

- If an object has a Thumbnail rendition, that rendition SHOULD be included in the atom entry representation if this parameter is not specified. If this parameter is specified, the behavior should follow the specified value.

### 9.3.2 PUT

This does a complete replacement of the atom entry with the atom entry document specified. If properties are not included, they will be not set.

The following arguments may be supplied. Please see the domain model for more information:

- checkinComment
- major
- checkin
  - Used to differentiate between updateProperties or checkin services. If TRUE, execute checkin service.

### 9.3.3 DELETE

This removes the document entry, in this case, cancels the check out. The PWC will be removed.

Success HTTP code: 204

## 9.4 Folder Entry

This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

CMIS Services:

- GET: getProperties
- PUT: updateProperties
- DELETE: Delete [folder entry only – deleteObject]

Media Type: application/atom+xml;type=entry

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- describedby: Points to the type definition for this object
- down: Points to the children of this folder if they exist
  - Media Type: application/cmistree+xml points to the atom feed document representing the descendents collection for this same folder
- up: Points to the atom entry for the parent
- alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/relationships>: Points to the relationships feed for this object
- <http://docs.oasis-open.org/ns/cmisis/link/200901/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>: Points to ACL document for this object
- <http://docs.oasis-open.org/ns/cmisis/link/200901/foldertree>: Points to the folder tree for this folder

The following element MUST be included inside the atom entry:

- `<xs:element name="object" type="cmis:cmisObjectType" />`

Example:

TODO: Add example here before Public Review
---

## 9.4.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships
- includeRenditions
  - If an object has a Thumbnail rendition, that rendition SHOULD be included in the atom entry representation if this parameter is not specified. If this parameter is specified, the behavior should follow the specified value.

## 9.4.2 PUT

This does a complete replacement of the atom entry with the atom entry document specified. If properties are not included, they will be unset.

## 9.4.3 DELETE

This removes the object (folder) from the repository.

Success HTTP code: 204

## 9.5 Relationship Entry

This is a CMIS relationship instance. These objects are exposed via 'relationships' link type.

CMIS Services:

GET: `getProperties`

PUT: `updateProperties`

DELETE: `Delete`

Media Type: `application/atom+xml;type=entry`

Link Relations:

- `service`: Points to service document containing CMIS repository
  - Media Type: `application/atomsvc+xml`
- `describedby`
- `target`
- `source`
- `alternate`: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmis/link/200901/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200901/policies>: Points to the policy feed for this object.

- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>: Points to ACL document for this object

The following element MUST be included inside the atom entry:

- `<xs:element name="object" type="cmisis:cmisisObjectType" />`

Example:

TODO: Add example here before Public Review
---

## 9.5.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships
- includeRenditions
  - If an object has a Thumbnail rendition, that rendition SHOULD be included in the atom entry representation if this parameter is not specified. If this parameter is specified, the behavior should follow the specified value.

## 9.5.2 PUT

This does a complete replacement of the atom entry with the atom entry document specified. If properties are not included, they will be unset.

## 9.5.3 DELETE

This removes the relationship entry.

Successful HTTP code: 204

## 9.6 Policy Entry

This is a CMIS policy instance.

CMIS Services:

GET: getProperties

PUT: updateProperties

DELETE: Delete

Media Type: application/atom+xml;type=entry

Link Relations:

- service: Points to service document containing CMIS repository
  - Media Type: application/atomsvc+xml
- describedby
- target
- source
- alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.



- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmisis/link/200901/acl>: Points to ACL document for this object

The following element MUST be included inside the atom entry:

- `<xs:element name="object" type="cmis:cmisObjectType" />`

Example:

TODO: Add example here before Public Review
---

## 9.6.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships
- includeRenditions
  - If an object has a Thumbnail rendition, that rendition SHOULD be included in the atom entry representation if this parameter is not specified. If this parameter is specified, the behavior should follow the specified value.

## 9.6.2 PUT

This does a complete replacement of the atom entry with the atom entry document specified. If properties are not included, they will be unset.

## 9.6.3 DELETE

This removes the policy entry.

Success HTTP code: 204

## 9.7 Content Stream

This is the content stream portion of the document object.

CMIS Services:

GET: getContentStream

PUT: setContentStream

DELETE: deleteContentStream

Media Type: Mime/Type of resource (mime type of content stream on document)

### 9.7.1 GET

This returns the content stream.

It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED that HTTP compression is also supported.

Please see RFC2616 for more information on HTTP Range requests.

## 9.7.2 PUT

This does a replacement of the content stream.

The following optional arguments may be supplied. Please see the domain model for more information:

- overwriteFlag

Success HTTP code: 201

Returns headers:

- Content-Location: URI for content stream

## 9.7.3 DELETE

This removes the content stream.

## 9.8 Folder Parent Entry

This is the parent for a specific object. Please see the appropriate Entry Resource.

CMIS Services:

GET: getFolderParent

Media Type: application/atom+xml;type=entry

### 9.8.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter

## 9.9 ACL Resource

CMIS Services:

GET: getACL

Media Type: application/cmisacl+xml

Example:

TODO: Add example here before Public Review
---

### 9.9.1 GET

---

## 10CMIS-RestAtom Schema

Placeholder for schema to be included for Public Review Draft. See Schema zip file in the meantime.

---

---

# 11 IANA Considerations

## 11.1 Document Types

### 11.1.1 CMIS Query

A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmisquery+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmisquery

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

Al Brown <albertcbrown@us.ibm.com>

Intended usage: COMMON

Author/Change controller: IESG

### 11.1.2 CMIS AllowableActions

A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmisallowableactions+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmisallowableactions

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

Al Brown <albertcbrown@us.ibm.com>

Intended usage: COMMON

Author/Change controller: IESG

### 11.1.3 CMIS Tree

A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmistree+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmistree

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

Al Brown <albertcbrown@us.ibm.com>

Intended usage: COMMON

Author/Change controller: IESG

### 11.1.4 CMIS Atom

A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmisatom+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

"type": This parameter has semantics identical to the type parameter of the "application/atom+xml" as specified in [RFC4287]

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmisatom

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

Al Brown <albertcbrown@us.ibm.com>

Intended usage: COMMON

Author/Change controller: IESG

### 11.1.5 CMIS ACL

A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmisacl+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmisacl

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

Al Brown <albertcbrown@us.ibm.com>  
Intended usage: COMMON  
Author/Change controller: IESG

---



---

## # 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

### B.1 Atom Link Relations by Object Type

ATOM Link	Type	Document	Folder	Relationship	Policy
Self	X	X	X	X	X
alternate		X			
edit		X	X	X	X
edit-media		X			

### B.2 Atom and AtomPub Extensions

The following extensions to Atom and AtomPub (APP) have been leveraged in this specification:

1. Entries can contain Entries (for descendants) via the CmisAtomChildren type
2. Entries contain CMIS Type and CMIS Object information
3. Link relations have been extended
4. Links have been extended to add a cmis:id attribute
5. AtomPub Service document has been extended
  - a. CmisRepositoryInfo added to workspace
  - b. Collections have cmis collectionType

### B.3 Examples

Each of the following resources have an example as defined below:

Resource (Root Tag)	Description	Example
Repository (Service)	This is the repository logical object. It is represented by the atom service document. This is also exposed on entry as link 'service'	Please see Example-Service.xml
Root Folder Collection (Feed)	This is a collection described in the service document	Please see Example-FolderChildren.xml.
Checked Out Collection (Feed)	This is a collection described in the service document that contains all the checkedout documents	Please see Example-FolderChildren.xml. This will however, be a feed of private working copy document's only.
Unfiled Collection (Feed)	This is a collection of unfiled documents.	Please see Example-FolderChildren.xml. However there should be nothing returned.
Types Children Collection (Feed)	This is a collection described in the service document that contains all the types in the repository	

Type (Type)	This is the CMIS type of the object. This is exposed as link 'describedby',	Please see Example-Type.xml.
Document (Entry)	This is a CMIS Document instance. These are exposed in feeds or in an entry document. This can also be a private working copy (checkedout)	Please see Example-DocumentEntry.xml.
Document Private Working Copy (Entry)	This is a document in the private working copy of the document (checkedout version of document)	Please see Example-DocumentPWCEnter.xml.
Folder (Entry)	This is a CMIS Folder instance. These are exposed in feeds or in an entry document. The properties of a folder map onto the feed tag.	Please see Example-FolderEntry.xml.
Relationship (Entry)	This is a CMIS relationship instance. These objects are exposed via 'http://docs.oasis-open.org/ns/cmisis/link/200901/relationships'	Please see Example-Relationship.xml.
Policy (Entry)	This is a CMIS policy instance. This is exposed via 'cmisis-policies' as a feed	Please see Example-PolicyEntry.xml
Content Stream (Non-XML)	This is the content stream portion of the document object. This is exposed via 'edit-media' or 'alternate' on the entry	No example. This is the original document format.
Allowable Actions (Allowable-Actions)	This is the set of actions available to the current user on the current object. This is exposed as a link 'http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions'	Please see Example-AllowableActions.xml
Relationships Collection (for a specific item) (Feed)	This is the set of relationships available (either source or target or both) from a specific item such as a document, folder or policy. This is exposed as either 'http://docs.oasis-open.org/ns/cmisis/link/200901/source' or 'http://docs.oasis-open.org/ns/cmisis/link/200901/target'	Please see FolderChildren. However, the entries will be Relationships as in the RelationshipEntry example.
Parents Collection (for a specific document or policy object) (Feed)	This is the set of parents for a specific document or policy object. This is exposed via 'up' on the entry.	Please see FolderChildren. However, the entries will be Folder as in the FolderEntry Example
Children (Feed)	This is a pseudo-object comprised of all the direct children of a particular folder. This is exposed as 'down'	Please see FolderChildren.
Descendants (Feed)	This is a pseudo-object comprised of all the direct and indirect children of a particular folder. This is exposed as 'down' with a media	Please see FolderDescendants.

type 'application/cmistree+xml'

AllVersions  
(Feed)

This is a pseudo-object made up of all document versions related to the current document version. This collection is also known as the version history. This is exposed via 'all-versions' on the entry.

Please see FolderChildren. However, the entries will be Documents as in DocumentEntry and DocumentPWCEnter.

## B.4 Expressing multiple content streams in REST

CMIS does not currently support multiple content streams in the specification due to complexity and lack of support across a set of repositories. However, exposing the multiple content streams that already exist in a repository can be done quite readily with REST.

Inside the ATOM entry section:

```
<link rel="stream" foo:streamnumber=3  
      href="http://example.org/media/atom03">
```

The same can be done in the cmis:object tag as well. The foo:streamnumber attribute tag exposes the stream id. Non-aware applications will see many links of relationship cmis-stream. If they are aware, they can chose which stream they want to retrieve.

For setting multiple content streams, this must be done outside of CMIS today.

---

## C. Revision History

Revision	Date	Editor	Changes Made
.61	10 April 2009	Al Brown	Updated based on JIRA issues  <a href="http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&amp;pid=10021&amp;fixfor=">http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&amp;pid=10021&amp;fixfor=</a>
.62a	26 May 2009	Al Brown, Ryan McVeigh	<a href="http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&amp;mode=hide&amp;sorter/order=DESC&amp;s1&amp;pid=10021&amp;fixfor=10012">http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&amp;mode=hide&amp;sorter/order=DESC&amp;s1&amp;pid=10021&amp;fixfor=10012</a>