



Content Management Interoperability Services Version 0.62a

Part II – ReSTful AtomPub Binding

Committee Draft

16 April 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

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 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
3	CMIS.....	10
3.1	Mime Types	11
3.1.1	Atom Media Type with CMIS extensions	11
3.1.2	Query	11
3.1.3	AllowableActions	11
3.2	CMIS Link Relations	12
3.2.1	Existing Link Relations	12
3.2.2	Hierarchy Internet Draft Link Relations	12
3.2.3	Versioning Internet Draft Link Relations.....	12
3.2.4	CMIS Specific Link Relations	12
3.3	Exceptions	13
3.3.1	Common Exceptions	13
3.3.2	Other Exceptions.....	13
3.3.3	Other notable HTTP Exceptions.....	13
4	Atom Resources.....	14
4.1	Feeds	14
4.2	Entries.....	14
4.3	Categories.....	14
5	Repository	16
5.1	HTTP Methods.....	17
5.1.1	GET	17
6	Service Collections.....	18
6.1	Root Folder Collections	18
6.2	Query Collection	18
6.2.1	HTTP Methods	18
6.2.1.1	POST	18
6.3	Checked Out Collection	18
6.3.1	HTTP Methods	19
6.3.1.1	GET	19
6.3.1.2	POST	19
6.4	Unfiled Collection.....	19
6.4.1	HTTP Methods	19
6.4.1.1	GET	19
6.4.1.2	POST	20

6.5 Types Children Collection	20
6.5.1 HTTP Methods	20
6.5.1.1 GET	21
6.6 Types Descendants Collection	21
6.6.1 HTTP Methods	21
6.6.1.1 GET	21
6.7 Changes Collection.....	21
7 Collections.....	23
7.1 Relationships Collection	23
7.1.1 HTTP Methods	23
7.1.1.1 GET	23
7.1.1.2 POST	Error! Bookmark not defined.
7.2 Parents Collection.....	23
7.2.1 HTTP Methods	24
7.2.1.1 GET	24
7.3 Folder Children Collection	24
7.4 HTTP Methods.....	24
7.4.1.1 GET	24
7.4.1.2 POST	25
7.5 Folder Descendants Collection.....	26
7.5.1 HTTP Methods	26
7.5.1.1 GET	26
7.5.1.2 POST	26
7.5.1.3 DELETE	26
7.6 AllVersions Collection	27
7.6.1 HTTP Methods	27
7.6.1.1 GET	27
7.6.1.2 DELETE	27
7.7 Policies Collection.....	27
7.7.1 HTTP Methods	28
7.7.1.1 GET	28
7.7.1.2 POST	28
8 Entries	29
8.1 Type Entry.....	29
8.1.1 HTTP Methods	29
8.1.1.1 GET	29
8.2 Document Entry	29
8.2.1 HTTP Methods	30
8.2.1.1 GET	30
8.2.1.2 PUT.....	30
8.2.1.3 PATCH.....	30
8.2.1.4 DELETE.....	30
8.3 Document Private Working Copy (PWC) Entry	30
8.3.1 HTTP Methods	31
8.3.1.1 GET	31
8.3.1.2 PUT.....	31
8.3.1.3 PATCH.....	31

8.3.1.4 DELETE	31
8.4 Folder Entry	31
8.4.1 HTTP Methods	32
8.4.1.1 GET	32
8.4.1.2 PUT.....	32
8.4.1.3 PATCH.....	32
8.4.1.4 DELETE	32
8.5 Relationship Entry.....	32
8.5.1 HTTP Methods	33
8.5.1.1 GET	33
8.5.1.2 PUT.....	33
8.5.1.3 PATCH.....	33
8.5.1.4 DELETE.....	33
8.6 Policy Entry.....	33
8.6.1 GET	33
8.6.2 PUT	34
8.6.3 PATCH	34
8.6.4 DELETE.....	34
8.7 Content Stream.....	34
8.7.1 GET	34
8.7.2 PUT	34
8.7.3 DELETE.....	34
9 IANA Considerations	35
9.1 Document Types.....	37
9.1.1 Query.....	37
9.1.2 AllowableActions	37
# Conformance.....	39
A. Acknowledgements	40
B. Non-Normative Text	41
B.1 Atom Link Relations by Object Type	41
B.2 Atom and APP Extensions	41
B.3 Examples.....	41
B.4 Expressing multiple content streams in REST	43
C. Revision History.....	44

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, *Atom Syndication Format*, <http://www.ietf.org/rfc/rfc4287.txt>, December 2005
- [RFC5023] J. Gregorio, B. de hOra, *Atom Publishing Protocol*, <http://www.ietf.org/rfc/rfc5023.txt>, October 2007
- [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol --HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, June 1999
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, 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

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 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 APP documents utilizing the Atom and APP 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. Support for HTTP Basic authentication is recommended.

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 should respond with. CMIS compliant implementation MUST support this one response format:

- application/atom+xml;profile=cmis1.0

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.

3 CMIS

3.1 Mime Types

CMIS introduces two new media types for a CMIS Query document and a CMIS AllowableActions document.

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

- APP 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

This specification defines a new parameter to the Atom Media Type named 'profile'. The value of the parameter MUST be 'cmis1.0'. CMIS clients and servers should use this parameter when CMIS extensions are required.

Example:

- application/atom+xml;profile=cmis1.0

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

3.1.2 Query

Media Type: application/cmisquery+xml

Starting tag: query

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

Example:

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

3.1.3 AllowableActions

Media Type: application/cmisallowableactions+xml

Starting tag: allowableactions

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

3.2 CMIS Link Relations

The table 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 pointed at 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:

- service
- describedby
- via
- edit-media
- edit

3.2.2 Hierarchy Internet Draft Link Relations

CMIS leverages the following link relations from the Internet Draft:

- up
- uptree
- down (children in domain model)
- downtree (descendants in domain model)

3.2.3 Versioning Internet Draft Link Relations

CMIS leverages the following link relations from the Internet Draft:

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



3.2.4 CMIS Specific Link Relations


CMIS leverages the following link relations:

- <http://docs.oasis-open.org/ns/cmisis/link/200901/allowableactions>
- <http://docs.oasis-open.org/ns/cmisis/link/200901/relationships>
- <http://docs.oasis-open.org/ns/cmisis/link/200901/source>
- <http://docs.oasis-open.org/ns/cmisis/link/200901/target>
- <http://docs.oasis-open.org/ns/cmisis/link/200901/permissions>

3.3 Exceptions

3.3.1 Common Exceptions

The following table 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
InvalidArgumentException	400
ObjectNotFoundException	404
PermissionDeniedException	403 
OperationNotSupportedException	405
UpdateConflictException	409
RuntimeException	500

3.3.2 Other Exceptions

CMIS Services Exception	HTTP Status Code
ConstraintViolationException	409
FilterNotValidException	400
StreamNotSupportedException	403
StorageException	500
ContentAlreadyExistsException	409
VersioningException	409
FolderNotValidException	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

4 Atom Resources

4.1 Feeds

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

1. 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. Author/name **MUST** be the CMIS property createdBy
3. Title **MUST** be the CMIS property name
4. App:edited **MUST** be the CMIS property lastModifiedDate
5. Link with relation self **MUST** be generated to return the URI of the feed
6. A feed **SHOULD** contain the element app:collection element

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. Published **MUST** be cmis:createdDate
4. Atom:author **MUST** be cmis:creator
5. For content tags
 - a. Documents with content
 - i. The src attribute **SHOULD** be used to point to the content
 - ii. The repository **SHOULD** populate the summary tag with text that at best efforts represents the documents. For example, an HTML table containing the properties and their values for simple feed readers
 - b. Other (Content-less document, Folder, Relationship, Type, etc)
 - i. Best efforts at generating HTML text that represents the object. That text would normally go into the summary tag, but since there is no content, goes in the content tag.
6. Links will be used to provide URIs to CMIS functionality
7. Link relations be omitted if the function is not allowed and that function would not show up in Allowable Actions
8. Links may be omitted if the repository does not support that capability
9. All CMIS properties will be exposed in CMIS properties tag even if they are duplicated in an atom element

When POSTing an Atom Document, the atom fields take precedence over the CMIS property field for writeable properties. For example, atom:title will overwrite cmis:name

4.3 Categories

The categorization functionality of Atom will be repository specific. Specifically the repository may:

1. Internalize categories via some scheme and expose

2. Ignore the category tags

Category schema:

```
<?xml version="1.0" ?>
<app:categories
  xmlns:app="http://www.w3.org/2007/app"
  xmlns:atom="http://www.w3.org/2005/Atom"
  fixed="yes" scheme="http://example.com/cats/bag3">
  <atom:category term="animal" />
  <atom:category term="vegetable" />
  <atom:category term="mineral" />
</app:categories>
```

Categories included inside the Atom entry document:

```
<atom:category scheme="http://example.com/cats/b" g3" term="mineral"/>
```

5 Repository

This is the repository logical object. This is exposed on an Atom entry or Atom feed as link relation to an AtomPub service document. Each repository is mapped to a Workspace in the AtomPub Service document.

CMIS Services exposed:

GET: getRepositories, getRepositoryInfo

PUT: Not Supported

POST: Not Supported

DELETE: Not Supported

Media Type: APP Service

How the client will get the initial ATOM (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.

In the Atom (APP) Service Document, each workspace maps to a single repository. A workspace element for a CMIS repository MUST have a collection element for each of following collections:

- Root folder: Root folder of the Repository
 - 'rootchildren' for the children collection of the root folder
 - 'rootdescendants' for the descendants collection of the root folder
- Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
 - 'unfiled'
- Checkedout Folder: Folder containing all checked out documents user can see
 - 'checkedout'
- Types Folder: Folder containing all the types in the repository
 - 'typeschildren' for the children collection
 - 'typesdescendants' for the children collection
- Query collection: Collection for posting queries to be executed
 - 'query'
- Changes collection: Collection for understanding changes in the repository

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.

This service document represents both getRepositories() and getRepositoryInfo(repid). Each repository will be a workspace in the service document.

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 HTTP Methods

5.1.1 GET

This retrieves the APP 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 must be defined in a CMIS-compliant service document.

6.1 Root Folder Collections

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

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.

PUT: Not Supported

POST: Query

DELETE: Not Supported

PATCH: Not Supported

Media Type: Atom Feed

Accept: MUST CMIS Query document, MAY other media types

Link Relations:

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

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.

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

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

6.3 Checked Out Collection

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

CMIS Services:

GET: getCheckedOutDocuments

PUT: Not Supported

POST: checkout
DELETE: Not Supported
PATCH: Not Supported

Media 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

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.

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
PUT: Not Supported
POST: removeDocumentFromFolder, removeDocumentFromAllFolders
DELETE: Not Supported
PATCH: Not Supported

Media Type: Atom Feed Document

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

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 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 ObjectId, the document does not exist, or the document is not in that folder, an exception is thrown.

This adheres to APP 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

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: getTypes
 PUT: Not Supported
 POST: Not Supported
 DELETE: Not Supported
 PATCH: Not Supported

Media Type: Atom Feed Document

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 type definition whose children represent this feed
- downtree: points to the descendants feed for the same type

6.5.1 GET

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

- includePropertyDefinitions
- maxItems
- skipCount

6.6 Types Descendants Collection

This is a collection described in the service document that contains all the types under a specific type in the repository to a specific depth.

CMIS Services:

GET: getTypes

PUT: Not Supported

POST: Not Supported

DELETE: Not Supported

PATCH: Not Supported

Media Type: Atom Feed Document

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 children represent this feed
- dntree: points to the descendants feed for the same type

6.6.1 GET

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

- includePropertyDefinitions
- depth

6.7 Changes Collection

This is a collection described in the service document that contains the changes in the repository

CMIS Services:

GET: getChanges()

PUT: Not Supported

POST: Not Supported

DELETE: Not Supported

PATCH: Not Supported

Media Type: Atom Feed Document

Accept: Atom Entry document, other repository-specific

Link Relations:

- service: Points to service document containing CMIS repository
 - Media Type: application/atomsvc+xml
- next: This is the next page in the feed
- prev:

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
- PUT: Not Supported
- POST: createRelationship
- DELETE: Not Supported
- PATCH: Not Supported

Media Type: Atom 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 originating atom entry document if available. If source, then the source atom entry document. If target, then the target atom entry document.

7.1.1 GET

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

- relationshipType
- includeSubRelationshipTypes
- direction
- maxItems
- skipCount
- filter
- includeAllowableActions



7.2 Parents Collection

This is the set of parents for a specific object. This can also be the ancestor of folders if returnToRoot is selected.

CMIS Services:

- GET: getObjectParents or getFolderParent
- PUT: Not Supported
- POST: Not Supported
- DELETE: Not Supported

Media Type: Atom 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 the resource generating this collection

7.2.1 GET

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

- returnToRoot
- filter
- includeAllowableActions
- includeRelationships



7.3 Folder Children Collection

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

CMIS Services:

GET: getChildren

PUT: Not Supported

POST:

- createDocument (POST to folder)
- or createFolder (POST to folder)
- or createPolicy
- or moveObject
- or addObjectToFolder (Entry)

DELETE: Not Supported

Media Type: Atom Feed with CMIS extensions

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
- up: points to the atom feed document representing the collection containing this resource
- uptree: points to the atom feed document representing a tree of collections
- dntree: points to the atom feed document representing a tree of resources under the same parent

7.3.1 GET

Request Format: N/A

Response Format: Atom Feed (flat)

HTTP Code:

- 200 OK (Success)

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

- orderBy
- maxItems
- skipCount
- childTypes
- filter
- includeAllowableActions
- includeRelationships

7.3.2 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

If the atom entry has a cmis property objectId that is valid for the repository, the object will be added to the folder. If the objected property is missing, it will be created and then added to the folder. If the objected property is present but not valid an exception will be thrown.

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.

A content stream MUST be specified regardless of versioningState if required by the type definition. If not provided and it is required, ContentStreamNotProvided exception will be thrown.

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

- As part of the atom entry using the content element
- As a separate operation via the edit-media link

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.

If an OID for an existing document is specified, then the document will be filed into the specified folder.

Optional arguments:

- versioningState

- removeFrom
- thisVersion

7.4 Folder Descendants Collection

This is a pseudo-object comprised of all the direct and indirect children of a particular folder. This is exposed as 'descendants'

CMIS Services:

GET: getDescendants

PUT: Not Supported

POST:

createDocument (POST to folder)

createFolder (POST to folder)

createPolicy

or moveObject

or addDocument-ToFolder (Entry)

DELETE: deleteTree

Media Type: Atom Feed (Nested)

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
- up: points to the atom feed document representing the collection containing this resource
- uptree: points to the atom feed document representing a tree of collections
- down: points to the atom feed document representing a feed of children under the same parent



7.4.1 GET

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

- depth
- childTypes
- filter
- includeAllowableActions
- includeRelationships

7.4.2 POST

POST to the Folder Descendants Collection is identical to a POST to the Folder Children Collection.

7.4.3 DELETE

This removes the entire tree of collections

Success HTTP code: 204

If an error occurs, the client MAY do a GET on the collection to see the objects that are not deleted. The client may correct any errors and re-send the DELETE.

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

- continueOnFailure
- unfileMultiFiledDocuments

7.5 AllVersions Collection

This is a pseudo-object comprised of all the direct and indirect children of a particular folder. This is exposed as 'descendants'

CMIS Services:

GET: getAllVersions
PUT: Not Supported
POST: Not Supported
DELETE: deleteAllVersions

Document Type: Atom 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

7.5.1 GET

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

- filter
- includeAllowableActions
- includeRelationships

7.5.2 DELETE

This removes the entire version history of the document.

Success HTTP code: 204

7.6 Policies Collection

This is an atom feed of all the policy objects currently applied

CMIS Services:

GET: getAllVersions
PUT: Not Supported
POST: Not Supported
DELETE: deleteAllVersions

Media Type: Atom Feed Document

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

7.6.1 GET

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

- filter

7.6.2 POST

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

8 Entries

8.1 Type Entry

This represents a type definition in the repository.

This is enclosed as an atom entry

CMIS Services:

GET: getTypeDefinition

PUT: Not Supported

POST: Not Supported

DELETE: Not Supported

PATCH: Not Supported

Media Type: Atom Entry with CMIS Extensions

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
- up: Points to the parent type as atom entry if applicable
- down: Points to the children of this type as atom feed if applicable
- downtree: Points to the descendants of this type as atom feed if applicable
- describedby: Points to the type definition atom entry of the base type

8.1.1 GET

8.2 Document Entry

This is a CMIS Document instance.

CMIS Services:

GET: getProperties, getPropertiesOfLatestVersion

PUT: updateProperties

POST: Not Supported

DELETE: Delete [This specific version]

Media Type: Atom Entry with CMIS extensions

Link Relations:

- service: Points to service document containing CMIS repository
 - Media Type: application/atomsvc+xml
- up: Points to the atom entry of the parent
- allowableactions: Points to the allowable actions entry for this resource
- all-versions: Points to atom feed containing the versions of this resource
- latest-version

- edit-media
- working-copy
- describedby
- policies
- relationships
- acl

8.2.1 GET

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

- returnVersion
- filter

8.2.2 PUT

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

8.2.3 PATCH

Request Media Type: Atom Entry with CMIS extensions

The client will send the Atom Entry with CMIS extensions to the server. The server will inspect the document. The server will update the resource with any CMIS properties specified in the request. Any properties not specified in the request will not be changed.

8.2.4 DELETE

This removes the private working copy.

Success HTTP code: 204

8.3 Document Private Working Copy (PWC) Entry

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

CMIS Services:

- GET: getProperties
- PUT: updateProperties or checkin
- POST: Not Supported
- DELETE: cancelCheckout

Media Type: Atom Entry with CMIS extensions

Link relations:

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

- all-versions
- relationships
- edit-media
- via: atom entry that created this private working copy
- describedby
- policies
- acl

8.3.1 GET

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

- filter

8.3.2 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

8.3.3 PATCH

Please see PATCH for Document Entry .

8.3.4 DELETE

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

Success HTTP code: 204

8.4 Folder Entry

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

CMIS Services:

- GET: getProperties
- PUT: updateProperties
- POST: Not Supported
- DELETE: Delete [folder entry only – deleteObject]

Media Type: Atom Entry with CMIS extensions

Link Relations:

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

- describedby
- down
- downtree
- up
- allowableactions
- relationships
- policies
- acl

8.4.1 GET

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

- filter

8.4.2 PUT

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

8.4.3 PATCH

Please see PATCH for Document Entry.

8.4.4 DELETE

This removes the object (folder) from the repository.

Success HTTP code: 204

8.5 Relationship Entry

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

CMIS Services:

- GET: getProperties
- PUT: updateProperties
- POST: Not Supported
- DELETE: Delete

Media Type: Atom Entry with CMIS extensions

Link Relations:

- service: Points to service document containing CMIS repository
 - Media Type: application/atomsvc+xml
- describedby
- target
- source
- allowableactions
- policies
- acl

8.5.1 GET

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

- filter

8.5.2 PUT

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

8.5.3 PATCH

This does a partial replacement of the atom entry with the atom entry document specified. This is the same as a PUT, except CMIS properties can be omitted. If properties are omitted, then they will not be updated rather than changed to not-set as in the case of PUT.

8.5.4 DELETE

This removes the relationship entry.

Successful HTTP code: 204

8.6 Policy Entry

This is a CMIS policy instance. This is exposed via 'policies' as a feed.

CMIS Services:

- GET: getProperties
- PUT: updateProperties
- POST: Not Supported
- DELETE: Delete

Media Type: Atom Entry with CMIS extensions

Link Relations:

- service: Points to service document containing CMIS repository
 - Media Type: application/atomsvc+xml
- describedby
- target
- source
- allowableactions
- policies
- acl

8.6.1 GET

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

* filter

8.6.2 PUT

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

8.6.3 PATCH

Please see PATCH for Document Entry.

8.6.4 DELETE

This removes the policy entry.

Success HTTP code: 204

8.7 Content Stream

This is the content stream portion of the document object. This is exposed via 'stream' and edit-media on the entry

CMIS Services:

GET: getContentStream

PUT: setContentStream

POST: Not Supported

DELETE: deleteContentStream

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

8.7.1 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.

8.7.2 PUT

This does a replacement of the content stream. If the document is not checked out, an exception will be thrown. This should follow AtomPub.

Success HTTP code: 201

Returns headers:

- Content-Location: URI for content stream

8.7.3 DELETE

This removes the content stream. If the document is not checked out, an exception will be thrown.

9 RestAtom Schema

```
<?xml version="1.0" encoding="UTF-8"?>
  <!--
    Common CMIS XSD
  -->
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" targetNamespace="http://docs.oasis-
open.org/ns/cmisis/restatom/200901"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:xhtml="http://www.w3.org/1999/xhtml"
    xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
    xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
    jaxb:extensionBindingPrefixes="xjc" jaxb:version="2.1"
    xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200901"
    xmlns:cmisra="http://docs.oasis-open.org/ns/cmisis/restatom/200901"
    version="0.62a">
    <xs:import namespace="http://docs.oasis-open.org/ns/cmisis/core/200901"
      schemaLocation="CMIS-Core.xsd" />
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="xml.xsd" />

    <xs:simpleType name="enumRestOutputHeaders">
      <xs:restriction base="xs:string">
        <xs:enumeration value="contentCopied" />
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="enumLinkRelations">
      <xs:restriction base="xs:string">
        <xs:enumeration value="self" />
        <xs:enumeration value="edit" />
        <xs:enumeration value="edit-media" />
        <xs:enumeration value="via" />
        <xs:enumeration value="up" />
        <xs:enumeration value="up-tree" />
        <xs:enumeration value="down" />
        <xs:enumeration value="down-tree" />
        <xs:enumeration value="all-versions" />
        <xs:enumeration value="current-version" />
        <xs:enumeration value="working-copy" />
        <xs:enumeration value="service" />
        <xs:enumeration value="describedby" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/allowableactions" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/relationships" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/source" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/target" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/policies" />
        <xs:enumeration value="http://docs.oasis-
open.org/ns/cmisis/link/200901/acl" />
      </xs:restriction>
    </xs:simpleType>

    <!-- CMIS Rest Arguments -->
    <xs:simpleType name="enumArguments">
      <xs:restriction base="xs:string">
```

```

<xs:enumeration value="childTypes" />
<xs:enumeration value="continueOnFailure" />
<xs:enumeration value="checkin" />
<xs:enumeration value="checkinComment" />
<xs:enumeration value="depth" />
<xs:enumeration value="direction" />
<xs:enumeration value="filter" />
<xs:enumeration value="folderByPath" />
<xs:enumeration value="folderId" />
<xs:enumeration value="includeAllowableActions" />
<xs:enumeration value="includePropertyDefinitions" />
<xs:enumeration value="includeRelationships" />
<xs:enumeration value="includeSubrelationshipTypes" />
<xs:enumeration value="length" />
<xs:enumeration value="major" />
<xs:enumeration value="majorVersion" />
<xs:enumeration value="maxItems" />
<xs:enumeration value="offset" />
<xs:enumeration value="removeFrom" />
<xs:enumeration value="relationshipType" />
<xs:enumeration value="repositoryId" />
<xs:enumeration value="returnToRoot" />
<xs:enumeration value="returnVersion" />
<xs:enumeration value="skipCount" />
<xs:enumeration value="thisVersion" />
<xs:enumeration value="typeId" />
<xs:enumeration value="types" />
<xs:enumeration value="unfileMultiFiledDocuments" />
<xs:enumeration value="versioningState" />
</xs:restriction>
</xs:simpleType>

<xs:attribute name="id" type="xs:string" />
<xs:element name="hasMoreItems" type="xs:boolean" />
</xs:schema>
<!-- EOF -->

```

10IANA Considerations

10.1 Document Types

10.1.1 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

10.1.2 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

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 APP Extensions

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

1. Entries can contain Entries (for descendants)
2. Entries contain CMIS Type and CMIS Object information
3. Link relations have been extended
4. Links have been extended to add an cmis:id attribute (see atom.xsd line 296)
5. APP 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 'cmis-repository'	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 'cmis-type',	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. This is exposed as a feed. 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 'cmis-relationships'	Please see Example-Relationship.xml.
Policy (Entry)	This is a CMIS policy instance. This is exposed via 'cmis-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 'cmis-stream' 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 'cmis-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	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 can also be the ancestor of folders if returnToRoot is selected	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 'cmis-children'	Please see FolderChildren.
Descendants	This is a pseudo-object comprised of all the direct and indirect children of	Please see FolderDescendants.

(Feed)	a particular folder. This is exposed as 'cmis-descendants'	
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	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 http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&pid=10021&fixfor=1
.62	26 May 2009	Al Brown	http://tools.oasis-open.org/issues/secure/IssueNavigator.jspa?reset=true&mode=hide&sorter/order=DESC&sort=1&pid=10021&fixfor=10012